SIEMENS

SIMATIC

WinCC
WinCC Basic V14 SP1 - Communication

System Manual

Online help printout

03/2017
Online help printout

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.

⚠ DANGER

indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ WARNING

indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ CAUTION

indicates that minor personal injury can result if proper precautions are not taken.

NOTICE

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:

⚠ WARNING

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.

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG
GERMANY

Document order number: Online help printout
℗ 03/2017 Subject to change

Copyright © Siemens AG 2017.
All rights reserved

Table of contents

1 Communicating with PLCs (Basic Panels)....9

1.1 Basics of communication (Basic Panels)....9
1.1.1 Communication between devices (Basic Panels)....9
1.1.2 Devices and networks in the automation system (Basic Panels)....10
1.1.3 Data exchange using tags (Basic Panels)....14
1.1.4 Data exchange using area pointers (Basic Panels)....15
1.1.5 Communication drivers (Basic Panels)....16

1.2 Editors for communication (Basic Panels)....16
1.2.1 "Devices & networks" editor (Basic Panels)....16
1.2.2 Network view (Basic Panels)....18
1.2.3 Network data (Basic Panels)....21
1.2.4 Diagnostics of online connections (Basic Panels)....24
1.2.5 Device view (Basic Panels)....25
1.2.6 Topology view (Basic Panels)....28
1.2.7 Inspector window (Basic Panels)....31
1.2.8 Hardware catalog (Basic Panels)....32
1.2.9 Information on hardware components (Basic Panels)....35

1.3 Networks and connections (Basic Panels)....36
1.3.1 SIMATIC communication networks (Basic Panels)....36
1.3.1.1 Communication networks (Basic Panels)....36
1.3.1.2 PROFINET and Ethernet (Basic Panels)....37
1.3.1.3 PROFIBUS (Basic Panels)....38
1.3.1.4 MPI (Basic Panels)....39
1.3.1.5 PPI (Basic Panels)....40
1.3.2 Configuring networks and connections (Basic Panels)....41
1.3.2.1 Networking devices (Basic Panels)....41
1.3.2.2 Configuring an integrated connection in the "Devices & Networks" editor (Basic Panels)....42
1.3.2.3 Special considerations of the "Devices & Networks" editor (Basic Panels)....44
1.3.2.4 Configuring a non-integrated connection in the "Connections" editor (Basic Panels)....46
1.3.2.5 Configuring a routed connection in the "Devices & Networks" editor (Basic Panels)....50

1.4 Data exchange (Basic Panels)....53
1.4.1 Data exchange using tags (Basic Panels)....53
1.4.1.1 Basics of tags (Basic Panels)....53
1.4.1.2 Overview of HMI tag tables (Basic Panels)....54
1.4.1.3 External tags (Basic Panels)....56
1.4.1.4 Addressing external tags (Basic Panels)....58
1.4.1.5 Internal Tags (Basic Panels)....60
1.4.2 Data exchange using area pointers (Basic Panels)....61
1.4.2.1 Basic information on area pointers (Basic Panels)....61
1.4.2.2 Area pointers for connections (Basic Panels)....62
1.4.2.3 Use of area pointers (Basic Panels)....64
1.4.2.4 Accessing data areas (Basic Panels)....66
1.4.2.5 Configuring area pointers (Basic Panels)....66

1.5 Device dependency (Basic Panels)....71

1.5.1 Basic Panel (Basic Panels)....71
1.5.1.1 Communication drivers (Basic Panels)....71
1.5.1.2 Interfaces of the Basic Panels (Basic Panels)....75
1.5.1.3 Area pointers for Basic Panels (Basic Panels)....76
1.6 Communicating with SIMATIC S7 1500 (Basic Panels)....77
1.6.1 Communication with SIMATIC S7-1500 (Basic Panels)....77
1.6.2 Communication via PROFINET (Basic Panels)....78
1.6.2.1 Configuring an HMI connection (Basic Panels)....78
1.6.2.2 PROFINET parameters (Basic Panels)....81
1.6.2.3 Setting port options (Basic Panels)....93
1.6.3 Communication via PROFIBUS (Basic Panels)....97
1.6.3.1 Configuring an HMI connection (Basic Panels)....97
1.6.3.2 PROFIBUS parameters (Basic Panels)....100
1.6.4 Data exchange (Basic Panels)....110
1.6.4.1 Data exchange using area pointers (Basic Panels)....110
1.6.4.2 Trends (Basic Panels)....125
1.6.4.3 Alarms (Basic Panels)....126
1.6.5 Performance features of communication (Basic Panels)....128
1.6.5.1 SIMATIC S7-1500 device dependency (Basic Panels)....128
1.6.5.2 Valid data types for SIMATIC S7 1500 (Basic Panels)....130
1.6.6 Configuring connections in the "Connections" editor (Basic Panels)....131
1.6.6.1 Parameters for the connection (SIMATIC S7 1500) (Basic Panels)....131
1.6.6.2 Ethernet parameters (Basic Panels)....133
1.6.6.3 PROFIBUS parameters (Basic Panels)....134
1.6.7 Configuring time synchronization (Basic Panels)....136
1.6.7.1 Time synchronization restrictions (Basic Panels)....136
1.6.7.2 Configuring time synchronization for integrated connections....137
1.6.7.3 Configuring time synchronization for non-integrated connections (Basic Panels)....138
1.6.7.4 Time synchronization (Basic Panels)....140
1.7 Communicating with SIMATIC S7 1200 (Basic Panels)....140
1.7.1 Communication with SIMATIC S7-1200 (Basic Panels)....140
1.7.2 Communication via PROFINET (Basic Panels)....141
1.7.2.1 Communication via PROFINET (Basic Panels)....141
1.7.2.2 Configuring an HMI connection via PROFINET (Basic Panels)....142
1.7.2.3 PROFINET parameters (Basic Panels)....145
1.7.2.4 Setting port options (Basic Panels)....157
1.7.3 Communication via PROFIBUS (Basic Panels)....161
1.7.3.1 Communication via PROFIBUS (Basic Panels)....161
1.7.3.2 Configuring an HMI connection via PROFIBUS (Basic Panels)....162
1.7.3.3 PROFIBUS parameters (Basic Panels)....164
1.7.4 Data exchange (Basic Panels)....173
1.7.4.1 Data exchange using area pointers (Basic Panels)....173
1.7.4.2 Trends (Basic Panels)....189
1.7.4.3 Alarms (Basic Panels)....190
1.7.5 Performance features of communication (Basic Panels)....193
1.7.5.1 S7-1200 device dependency (Basic Panels)....193
1.7.5.2 Valid data types for SIMATIC S7-1200 (Basic Panels)....195
1.7.6 Creating connections in the "Connections" editor (Basic Panels)....196
1.7.6.1 Creating a PROFINET connection (Basic Panels)....196
1.7.6.2 Creating a PROFIBUS DP connection (Basic Panels)....198
1.7.6.3 Parameters for the connection (Basic Panels)....200

1.7.7 Configuring time synchronization (Basic Panels)....204
1.7.7.1 Time synchronization (Basic Panels)....204
1.7.7.2 Time synchronization restrictions (Basic Panels)....205
1.7.7.3 Configuring time synchronization for integrated connections....206
1.7.7.4 Configuring time synchronization for non-integrated connections (Basic Panels)....207
1.8 Communicating with SIMATIC S7 300/400 (Basic Panels)....209
1.8.1 Communication with SIMATIC S7 300/400 (Basic Panels)....209
1.8.2 Communication via PROFINET (Basic Panels)....209
1.8.2.1 Communication via PROFINET (Basic Panels)....209
1.8.2.2 Configuring an HMI connection via PROFINET (Basic Panels)....210
1.8.2.3 PROFINET parameters (Basic Panels)....212
1.8.2.4 Setting port options (Basic Panels)....220
1.8.3 Communication via PROFIBUS (Basic Panels)....224
1.8.3.1 Communication via PROFIBUS (Basic Panels)....224
1.8.3.2 Configuring an HMI connection via PROFIBUS (Basic Panels)....224
1.8.3.3 PROFIBUS parameters (Basic Panels)....227
1.8.4 Communication via MPI (Basic Panels)....232
1.8.4.1 Communication via MPI (Basic Panels)....232
1.8.4.2 Configuring an HMI connection via MPI (Basic Panels)....233
1.8.4.3 MPI parameters (Basic Panels)....235
1.8.5 Data exchange (Basic Panels)....241
1.8.5.1 Data exchange using area pointers (Basic Panels)....241
1.8.5.2 Trends (Basic Panels)....257
1.8.5.3 Alarms (Basic Panels)....257
1.8.6 Performance features of communication (Basic Panels)....260
1.8.6.1 S7-300/400 device dependency (Basic Panels)....260
1.8.6.2 Permitted data types for SIMATIC S7 300/400 (Basic Panels)....262
1.8.7 Creating connections in the "Connections" editor (Basic Panels)....262
1.8.7.1 Creating a PROFINET connection (Basic Panels)....262
1.8.7.2 Creating a PROFIBUS connection (Basic Panels)....265
1.8.7.3 Creating an MPI connection (Basic Panels)....267
1.8.7.4 Parameters for the connection (Basic Panels)....268
1.9 Communicating with the SIMATIC S7-1500 Software Controller (Basic Panels)....275
1.9.1 Communication with SIMATIC S7-1500 Software Controller (Basic Panels)....275
1.9.2 Communication via PROFINET (Basic Panels)....276
1.9.2.1 Configuring an HMI connection (Basic Panels)....276
1.9.2.2 PROFINET parameters (Basic Panels)....278
1.9.2.3 Setting port options (Basic Panels)....285
1.9.3 Communication via PROFIBUS (Basic Panels)....290
1.9.3.1 Configuring an HMI connection (Basic Panels)....290
1.9.3.2 PROFIBUS parameters (Basic Panels)....293
1.9.4 Data exchange (Basic Panels)....303
1.9.4.1 Data exchange using area pointers (Basic Panels)....303
1.9.4.2 Trends (Basic Panels)....320
1.9.4.3 Alarms (Basic Panels)....321
1.9.4.4 LED mapping (Basic Panels)....324
1.9.5 Performance features of communication (Basic Panels)....324
1.9.5.1 Device dependencies of the SIMATIC S7-1500 Software Controller (Basic Panels)....324
1.9.6 Configuring time synchronization (Basic Panels)....326
1.9.6.1 Time synchronization (Basic Panels)....326
1.9.6.2 Time synchronization restrictions (Basic Panels)....327

1.9.6.3 Configuring time synchronization for integrated connections....328
1.9.6.4 Configuring time synchronization for non-integrated connections (Basic Panels)....329
1.10 Communicating with SIMATIC ET 200 CPU (Basic Panels)....331
1.10.1 Communication with SIMATIC ET 200 CPU (Basic Panels)....331
1.10.2 Communication via PROFINET (Basic Panels)....331
1.10.2.1 Communication via PROFINET (Basic Panels)....331
1.10.2.2 Configuring an HMI connection via PROFINET (Basic Panels)....332
1.10.2.3 PROFINET parameters (Basic Panels)....335
1.10.2.4 Setting port options (Basic Panels)....342
1.10.3 Communication via PROFIBUS (Basic Panels)....346
1.10.3.1 Communication via PROFIBUS (Basic Panels)....346
1.10.3.2 Configuring an HMI connection via PROFIBUS (Basic Panels)....347
1.10.3.3 PROFIBUS parameters (Basic Panels)....348
1.10.4 Data exchange (Basic Panels)....353
1.10.4.1 Data exchange using area pointers (Basic Panels)....353
1.10.4.2 Trends (Basic Panels)....371
1.10.4.3 Alarms (Basic Panels)....372
1.10.4.4 LED mapping (Basic Panels)....374
1.10.5 Performance features of communication (Basic Panels)....375
1.10.5.1 Device dependencies of SIMATIC ET 200 CPU (Basic Panels)....375
1.10.5.2 Valid data types for SIMATIC ET 200 CPU (Basic Panels)....382
1.10.6 Creating connections in the "Connections" editor (Basic Panels)....383
1.10.6.1 Creating a PROFINET connection (Basic Panels)....383
1.10.6.2 Creating a PROFIBUS DP connection (Basic Panels)....385
1.10.6.3 Connection parameters (Basic Panels)....388
1.10.7 Configuring time synchronization (Basic Panels)....391
1.10.7.1 Time synchronization (Basic Panels)....391
1.10.7.2 Time synchronization restrictions (Basic Panels)....392
1.10.7.3 Configuring time synchronization for integrated connections....393
1.10.7.4 Configuring time synchronization for non-integrated connections (Basic Panels)....394
1.11 Communicating with SIMATIC S7 200 (Basic Panels)....396
1.11.1 Communication with SIMATIC S7 200 (Basic Panels)....396
1.11.2 Creating a connection to SIMATIC S7 200 (Basic Panels)....396
1.11.3 Parameters for the connection (Basic Panels)....399
1.11.3.1 Cyclic operation (Basic Panels)....399
1.11.3.2 Parameters for the connection (SIMATIC S7 200) (Basic Panels)....399
1.11.3.3 Ethernet parameters (Basic Panels)....401
1.11.3.4 PROFIBUS parameters (Basic Panels)....402
1.11.3.5 MPI parameters (Basic Panels)....404
1.11.3.6 PPI parameters (Basic Panels)....405
1.11.4 Data exchange (Basic Panels)....407
1.11.4.1 Data exchange using area pointers (Basic Panels)....407
1.11.4.2 Trends (Basic Panels)....422
1.11.4.3 Alarms (Basic Panels)....423
1.11.5 Performance features of communication (Basic Panels)....425
1.11.5.1 Device dependency S7 200 (Basic Panels)....425
1.11.5.2 Permitted data types for SIMATIC S7 200 (Basic Panels)....427
1.12 Communicating with SIMATIC LOGO! (Basic Panels)....428
1.12.1 Communication with SIMATIC LOGO! (Basic Panels)....428
1.12.2 Creating a connection to SIMATIC LOGO! (Basic Panels)....429
1.12.3 Connection parameters (Basic Panels)....431

1.12.3.1 Connection parameters (Basic Panels)....431
1.12.3.2 Ethernet parameters (Basic Panels)....432
1.12.3.3 Cyclic operation (Basic Panels)....433
1.12.4 Data exchange (Basic Panels)....434
1.12.4.1 Trends (Basic Panels)....434
1.12.4.2 Alarms (Basic Panels)....434
1.12.5 Performance features of communication (Basic Panels)....437
1.12.5.1 SIMATIC LOGO! device dependency (Basic Panels)....437
1.12.5.2 Valid data types for SIMATIC LOGO! (Basic Panels)....438
1.13 Communication with other PLCs (Basic Panels)....439
1.13.1 Communication with other PLCs (Basic Panels)....439
1.13.2 Distinctive features when configuring (Basic Panels)....440
1.13.3 Communication drivers (Basic Panels)....441
1.13.3.1 Allen-Bradley (Basic Panels)....441
1.13.3.2 Mitsubishi (Basic Panels)....486
1.13.3.3 Modicon Modbus (Basic Panels)....515
1.13.3.4 Omron (Basic Panels)....545
1.13.3.5 Channel support packages (Basic Panels)....563
1.13.4 Data exchange using area pointers (Basic Panels)....649
1.13.4.1 General information on area pointers (Basic Panels)....649
1.13.4.2 Accessing data areas (Basic Panels)....651
1.13.4.3 "Screen number" area pointer (Basic Panels)....651
1.13.4.4 "Date/time" area pointer (Basic Panels)....652
1.13.4.5 "Date/time PLC" area pointer (Basic Panels)....653
1.13.4.6 "Coordination" area pointer (Basic Panels)....654
1.13.4.7 "Project ID" area pointer (Basic Panels)....655
1.13.4.8 "PLC job" area pointer (Basic Panels)....656
1.13.4.9 "Data record" area pointer (Basic Panels)....659
Index....667

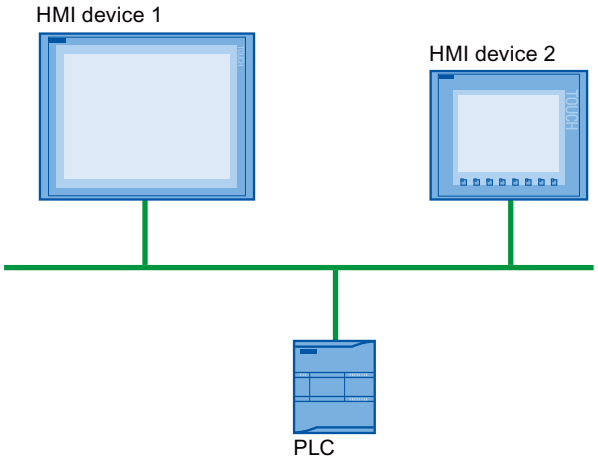
Communicating with PLCs (Basic Panels) 1

1.1 Basics of communication (Basic Panels)

1.1.1 Communication between devices (Basic Panels)

Communication

Data exchange between devices is known 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
- Administer process parameters and machine parameters

Communication partners

Communication between the following devices is described in more detail in this section:

- PLC
  The PLC controls a process by means of a user program.
- HMI device
  You use the HMI device to operate and monitor the process.

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.

Automation system

The following characteristics describe an automation system:

- The PLC and HMI device are interconnected
- The network between the PLC and HMI device is configured

Communication between HMI devices

The HTTP protocol is available for communication between HMI devices.

For more detailed information, refer to the documentation on the SIMATIC HMI HTTP protocol.

Communication via a uniform and vendor-neutral interface

With OPC (Openess Productivity Collaboration), WinCC has a uniform and manufacturer-neutral software interface. This interface enables standardized data exchange between industrial, office, and manufacturing applications.

For more detailed information, refer to the documentation for OPC.

See also

Devices and networks in the automation system (Page 10)

Data exchange using tags (Page 14)

Data exchange using area pointers (Page 15)

Communication drivers (Page 16)

1.1.2 Devices and networks in the automation system (Basic Panels)

Introduction

To set up an automation system, you must configure, parameterize, and interconnect the individual devices.

You insert PLCs and HMI devices into the project in the same way. Likewise, you configure the two devices in the same way.

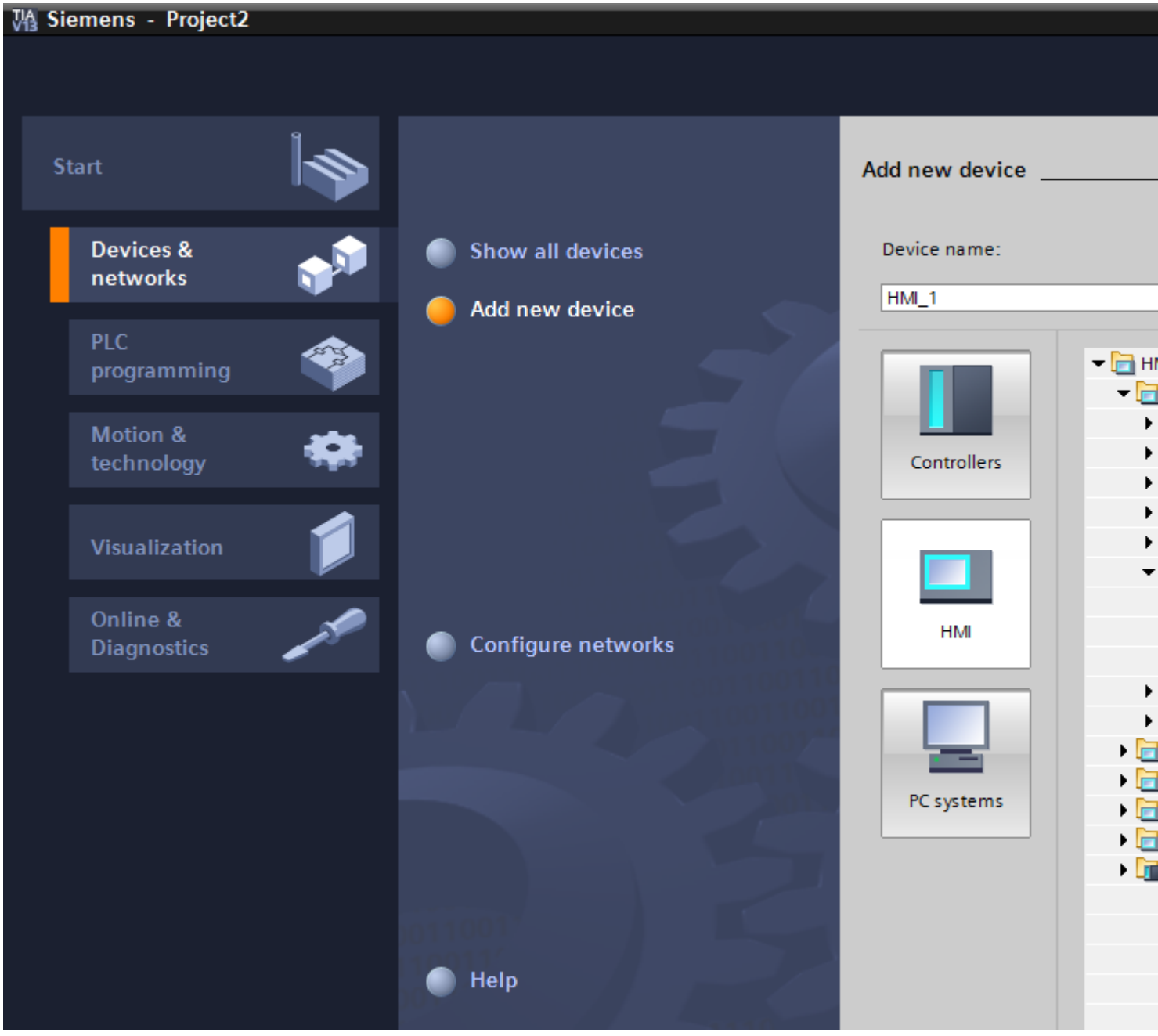
Automation system setup:

1. Insert PLC into the project.
2. Insert HMI device into the project.
3. Network the devices together.
4. Interconnect the devices.

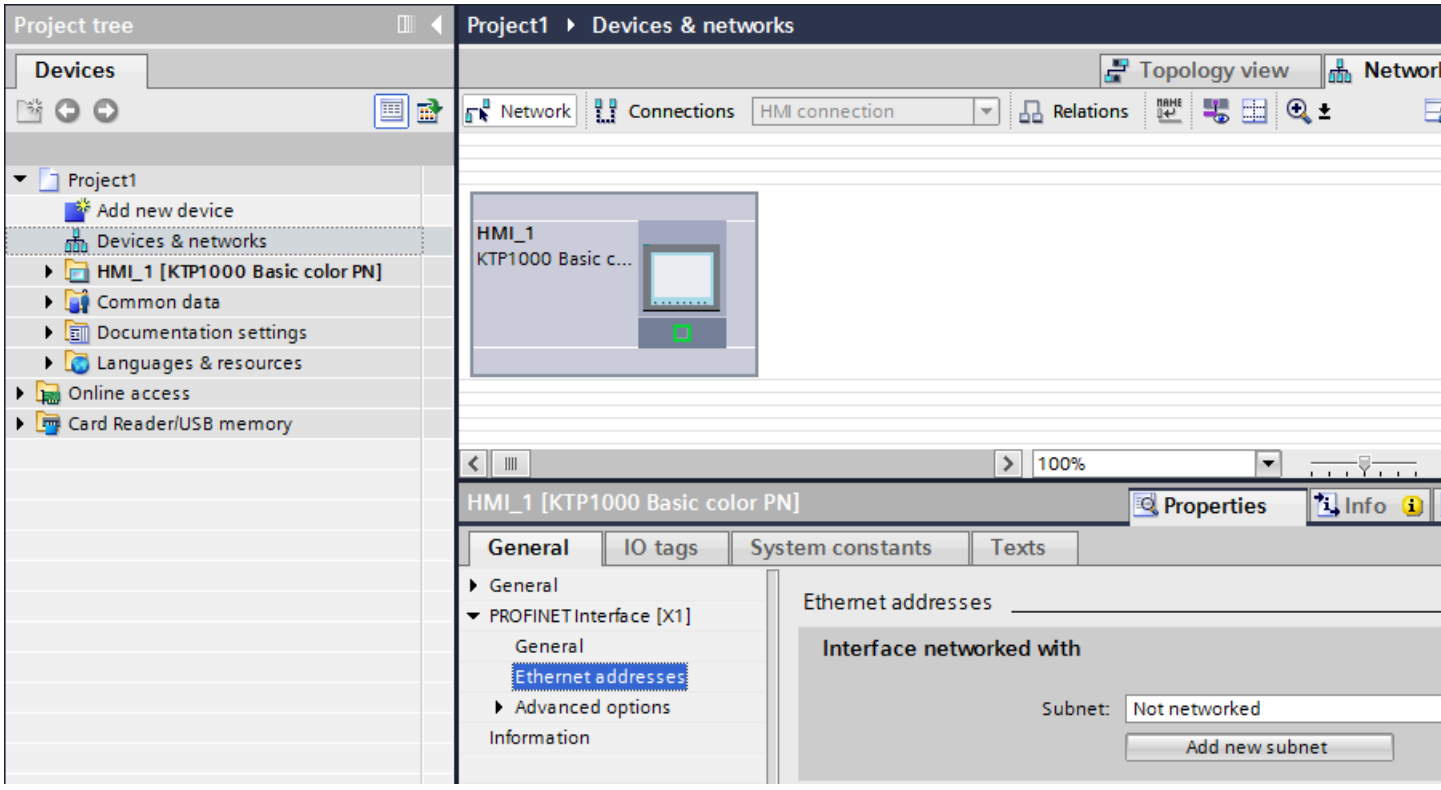
Inserting devices

If you have created a project, you can add a device in the portal view or project view.

- Portal view

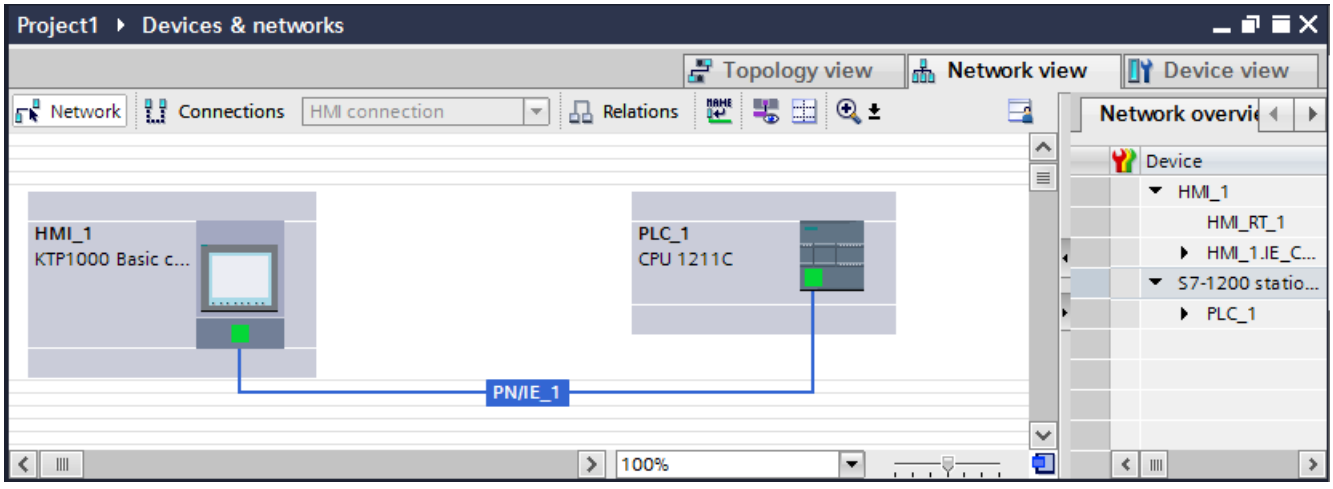


- Project view



Networking devices

You can network the interfaces of the communication-capable devices conveniently in the "Devices & Networks" editor. In the networking step, you configure the physical device connections.

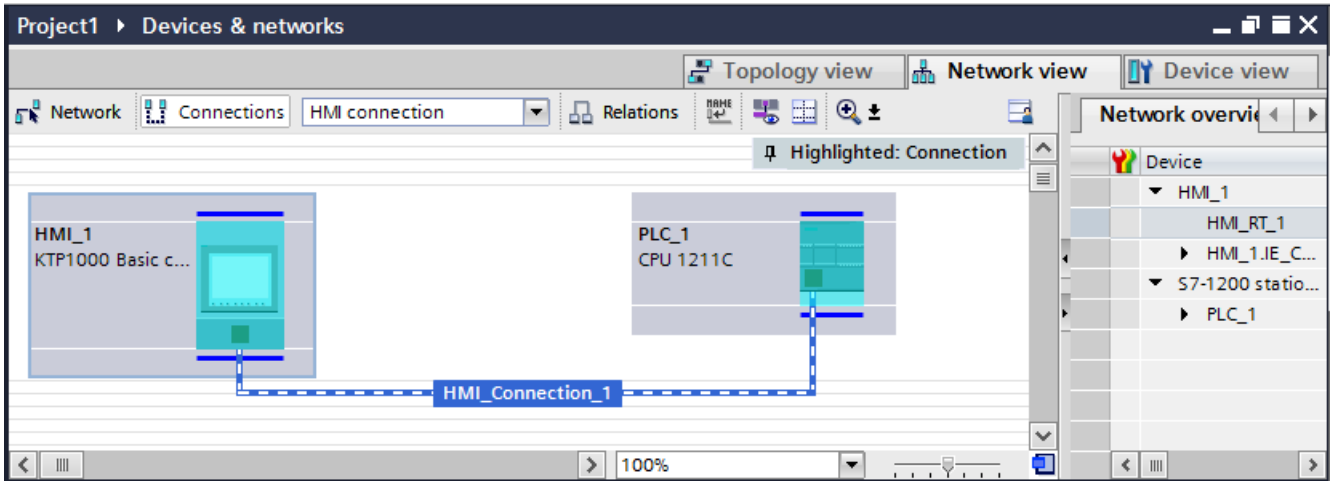


The tabular network overview supplements the graphical network view with the following additional functions:

- You obtain detailed information on the structure and parameter settings of the devices.
- Using the "Subnet" column, you can connect communication-capable components to subnets that have been created.

Connecting devices

After you network the devices together, you configure the connection. You configure the "HMI connection" connection type for communication with the HMI device.



See also

Communication between devices (Page 9)

Data exchange using tags (Page 14)

Data exchange using area pointers (Page 15)

Communication drivers (Page 16)

1.1.3 Data exchange using tags (Basic Panels)

Communication using tags

Process values are forwarded in runtime using tags. Process values are data which is stored in the memory of one of the connected automation systems. They represent the status of a plant in the form of temperatures, fill levels or switching states, for example. Define external tags for processing the process values in WinCC.

WinCC works with two types of tag:

- External tags
- Internal tags

Working with tags

For additional information on configuring tags see section "Working with tags".

See also

Communication between devices (Page 9)

Devices and networks in the automation system (Page 10)

Data exchange using area pointers (Page 15)

Communication drivers (Page 16)

1.1.4 Data exchange using area pointers (Basic Panels)

Communication using area pointers

Area pointers are parameter fields. WinCC receives information from these parameter fields in Runtime during the course of the project. This information contains data on the location and size of data areas in the PLC.

During communication, the PLC and the HMI device alternately access these data areas for read and write operations.

Based on the evaluation of data stored in the data areas, the PLC and HMI device initiate mutually defined actions.

The area pointers are managed centrally in the "Connections" editor. Area pointers are used to exchange data from specific user data areas.

You use the following area pointers in WinCC:

- Data record
- Date/time
- Coordination
- Job mailbox
- Date/time PLC
- Project ID
- Screen number

The availability of the various area pointers is determined by the HMI device used.

See also

Communication between devices (Page 9)
Devices and networks in the automation system (Page 10)
Data exchange using tags (Page 14)
Communication drivers (Page 16)

1.1.5 Communication drivers (Basic Panels)

Communication drivers

A communication driver is a software component that establishes a connection between a PLC and an HMI device. The communication driver thus enables the assignment of process values to HMI tags.

The interface as well as the profile and transmission speed can be chosen, depending on the HMI device used and the connected communication partner.

Note

Applies only to RT Professional.

If the correct communication driver is not installed, no communication is possible between the controller and the HMI device.

Install the correct drivers or the SIMATIC NET Station Manager through which drivers for most of the controllers are automatically installed.

See also

Communication between devices (Page 9)
Devices and networks in the automation system (Page 10)
Data exchange using tags (Page 14)
Data exchange using area pointers (Page 15)

1.2 Editors for communication (Basic Panels)

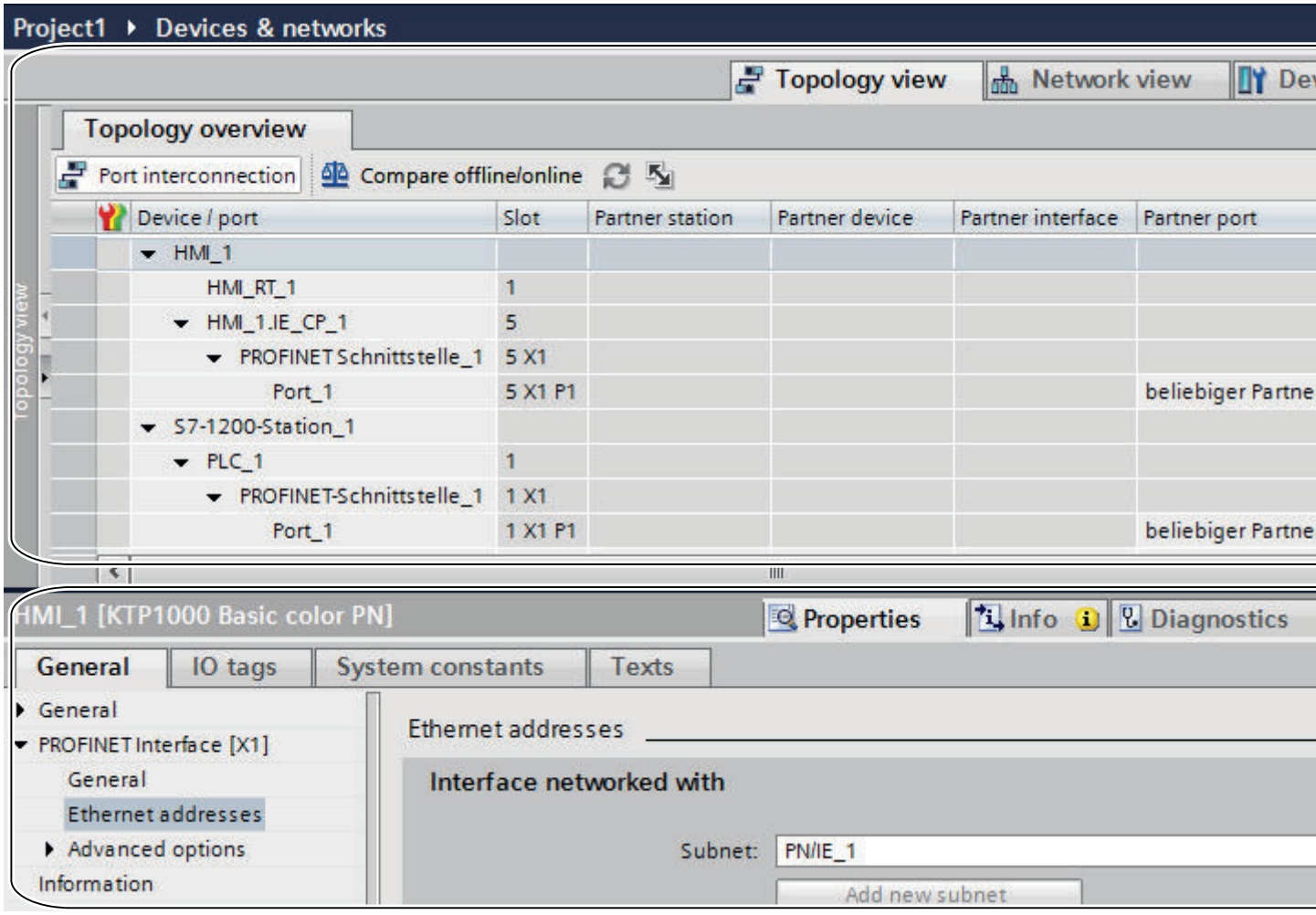
1.2.1 "Devices & networks" editor (Basic Panels)

Function of the hardware and network editor

The "Devices & networks" editor is the development environment for networking, configuring and assigning parameters to devices and modules.

Configuration

The "Devices & networks" editor consists of the following components:



1 Device view, network view and topology view
2 Hardware catalog
3 Inspector window

The "Devices & networks" editor provides you with three different views of your project. You can switch between these three views at any time depending on whether you want to produce and edit individual devices and modules, entire networks and device configurations or the topological structure of your project.

The inspector window contains information on the object currently marked. Here you can change the settings for the object marked.

Drag the devices and modules you need for your automation system from the hardware catalog to the network, device or topology view.

See also

Network view (Page 18)
Network data (Page 21)
Diagnostics of online connections (Page 24)
Device view (Page 25)
Topology view (Page 28)
Inspector window (Page 31)
Hardware catalog (Page 32)
Information on hardware components (Page 35)

1.2.2 Network view (Basic Panels)

Introduction

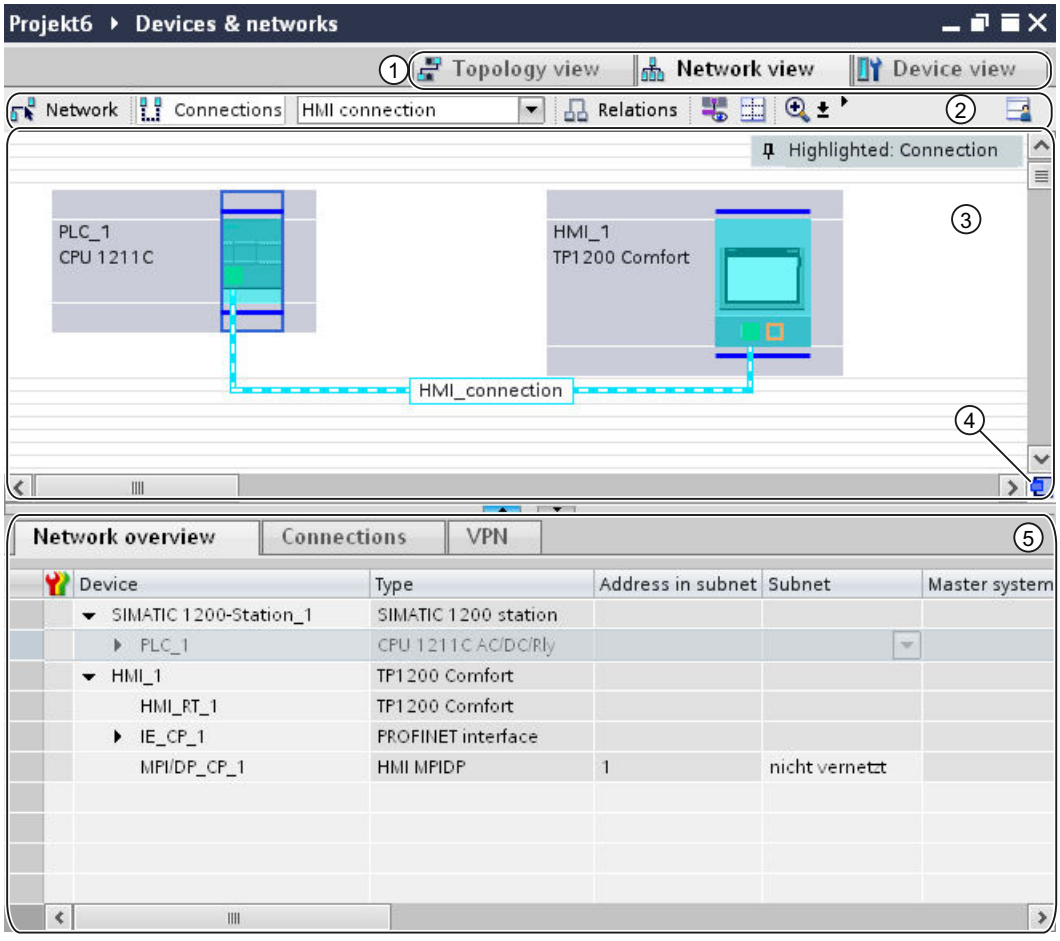
The network view is a working area of the hardware and network editor.

You undertake the following tasks in network view:

- Configuring and assign device parameters
- Networking devices with one another

Structure

The following diagram shows the components of the network view:



1 Changeover switch: network view/device view/topology view

2 Toolbar of network view

3 Graphic area of network view

4 Overview navigation

5 Table area of network view

You can use your mouse to change the spacing between the graphic and table areas of the network view.

To do this, click on the upper edge of the table view and expand or contract this by moving the mouse with the mouse button held down.

You can use the two small arrow keys to minimize, maximize or select the latest table structure of the table view with just one click.

Toolbar

The toolbar provides the following functions:

Icon	Meaning
	Mode to network devices.
	Mode to create connections. You can select the connection type using the adjacent drop-down list.
	Mode to create relations.
	Show interface addresses.
	Adjust the zoom setting. You can select the zoom setting or enter it directly in the adjacent drop-down list. You can also zoom in or zoom out the view in steps using the zoom symbol or draw a frame around an area to be zoomed in.
	Show page breaks Enables the page break preview. Dashed lines will be displayed at the positions where the pages will break when printed.
	Remember layout Saves the current table view. The layout, width and visibility of columns in the table view is stored.

Graphic area

The graphic area of the network view displays any network-related devices, networks, connections and relations. In this area, you add devices from the hardware catalog, connect them with each other via their interfaces and configure the communication settings.

Overview navigation

Click in the overview navigation to obtain an overview of the created objects in the graphic area. By holding down the mouse button, you can quickly navigate to the desired objects and display them in the graphic area.

Table area

The table area of the network view includes various tables for the devices, connections and communication settings present:

- Network overview
- Connections
- I/O communication

You can use the shortcut menu of the title bar of the table to adjust the tabular display.

See also

"Devices & networks" editor (Page 16)

Network data (Page 21)

Diagnostics of online connections (Page 24)
Device view (Page 25)
Topology view (Page 28)
Inspector window (Page 31)
Hardware catalog (Page 32)
Information on hardware components (Page 35)

1.2.3 Network data (Basic Panels)

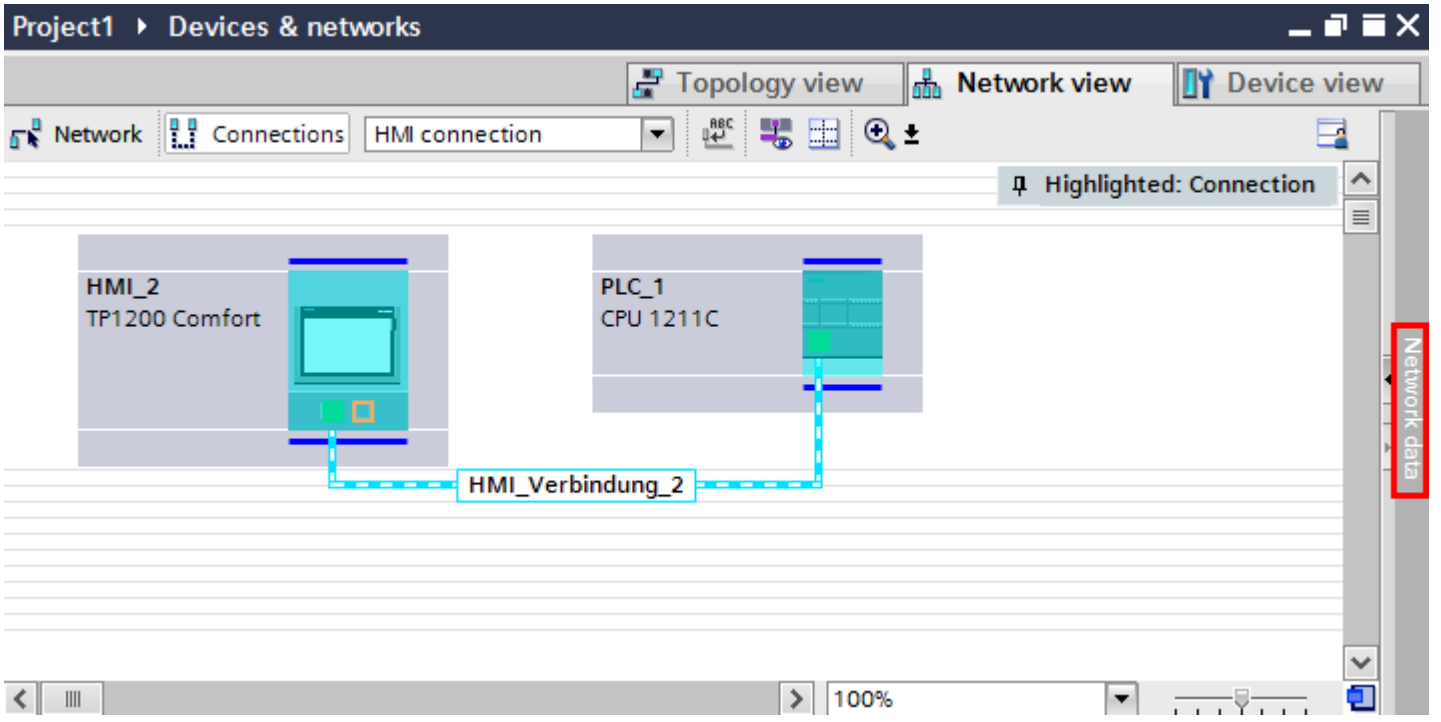
Introduction

The "Network view" editor also gives you a tabular view of the "Network data" in addition to the graphic network view.

You have the following selections in the "Network data" editor:

- Network overview
- Connections
- VPN
- I/O communication

You open the "Network data" below the graphic network overview.



Basic functions

The network data are displayed in tabular form and support the following basic functions for editing a table:

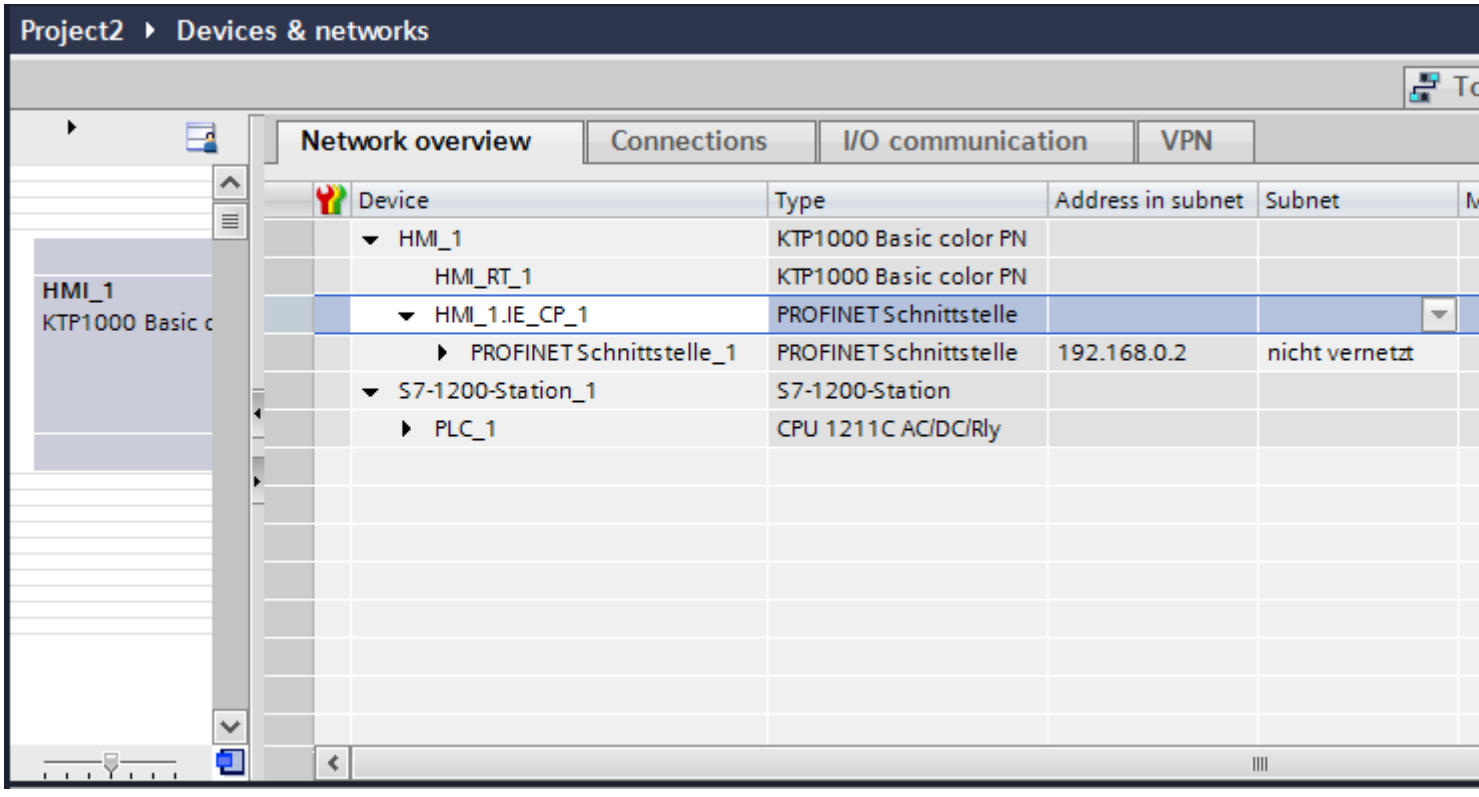
- Displaying and hiding table columns
  Note: The columns of relevance to configuration cannot be hidden.
- Optimizing column width
- Sorting table
- Displaying the meaning of a column, a row or cell using tooltips.

Network overview

The tabular network overview adds the following functions to the graphic network view:

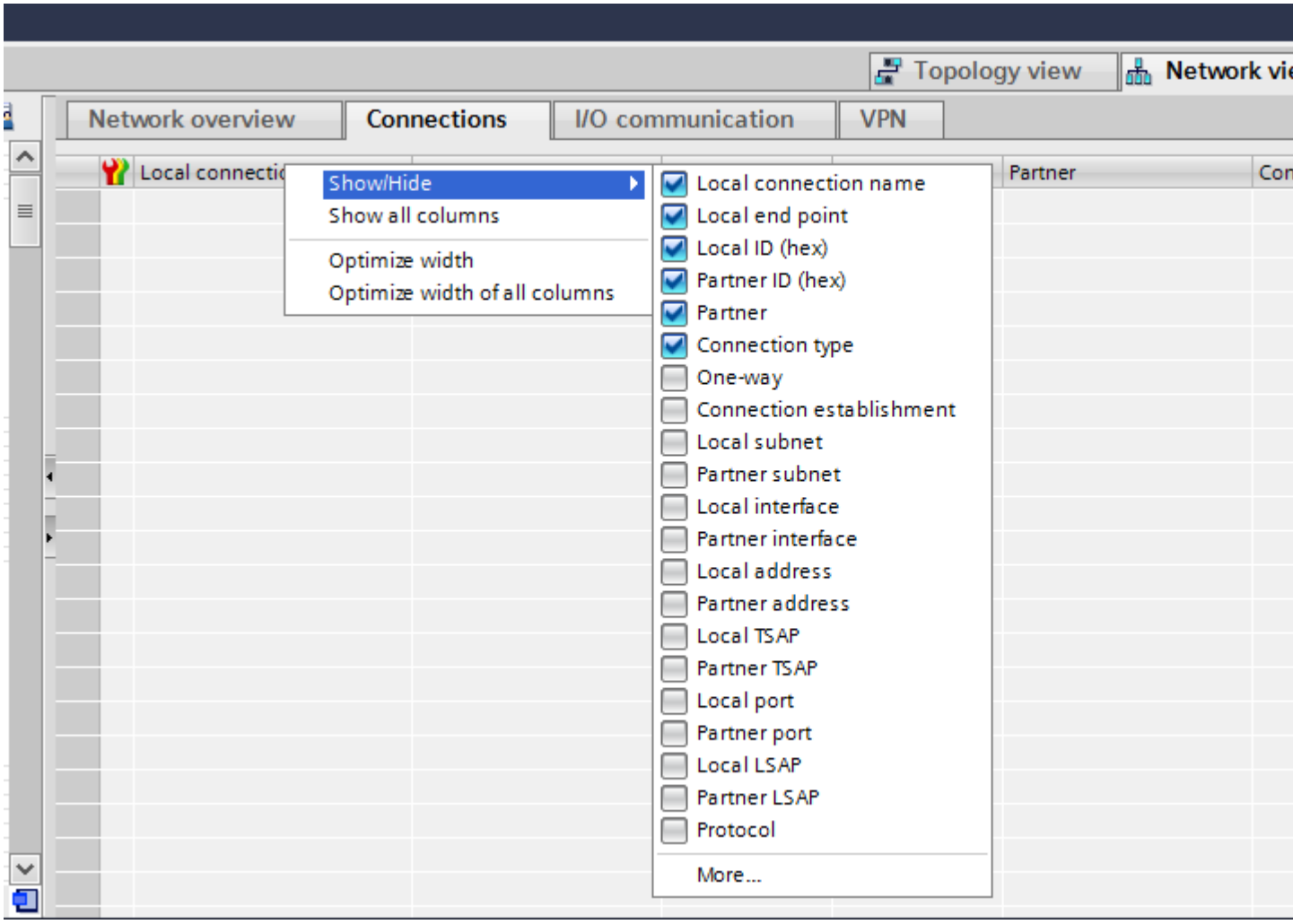
- You obtain detailed information on the structure and parameter settings of the devices.
- Using the "Subnet" column, you can connect components capable of communication with created subnets.

The



Connections

You will find additional network data under "Connections".



See also

"Devices & networks" editor (Page 16)

Network view (Page 18)

Diagnostics of online connections (Page 24)

Device view (Page 25)

Topology view (Page 28)

Inspector window (Page 31)

Hardware catalog (Page 32)

Information on hardware components (Page 35)

1.2.4 Diagnostics of online connections (Basic Panels)

Diagnostics of online connections

You can read out the diagnostics data of existing connections in the TIA Portal.

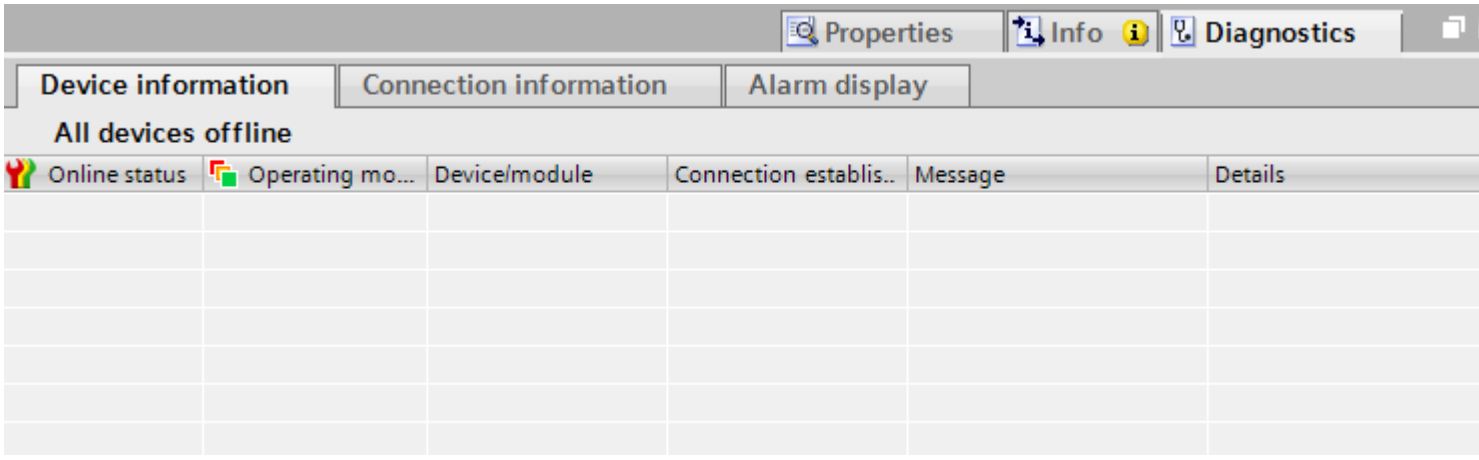
The "Diagnostics" function shows the connection data in tabular form in the Inspector window.

Requirements

- Devices must be in "Online" mode.

Device information

The diagnostics data of all devices in "Online" mode is displayed in the "Diagnostics > Device information" Inspector window.



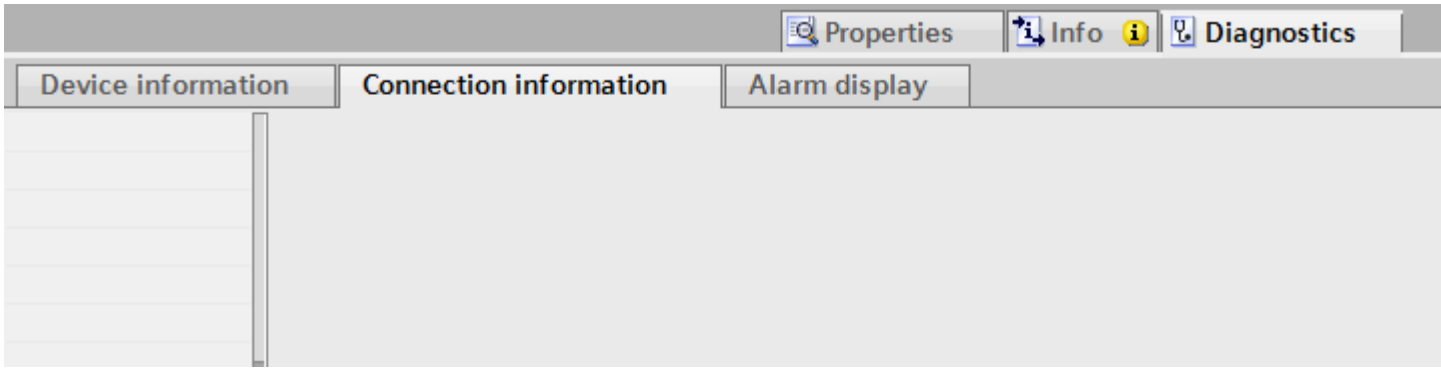
The following data is displayed:

- Online status
- Operating mode
- Device / module
- Alarm
- Details
- Help

Connection information

Use the "Connection information" function to display the diagnostics data of the connection selected in the "Devices&Networks" editor.

A graphic displays the communication partners of the connection and by which communication driver they are connected with each other.



The following data is displayed:

- End point
- Interface
- Subnet
- Address
- TSAP
- Number of HMI resources

See also

"Devices & networks" editor (Page 16)
Network view (Page 18)
Network data (Page 21)
Device view (Page 25)
Topology view (Page 28)
Inspector window (Page 31)
Hardware catalog (Page 32)
Information on hardware components (Page 35)

1.2.5 Device view (Basic Panels)

Introduction

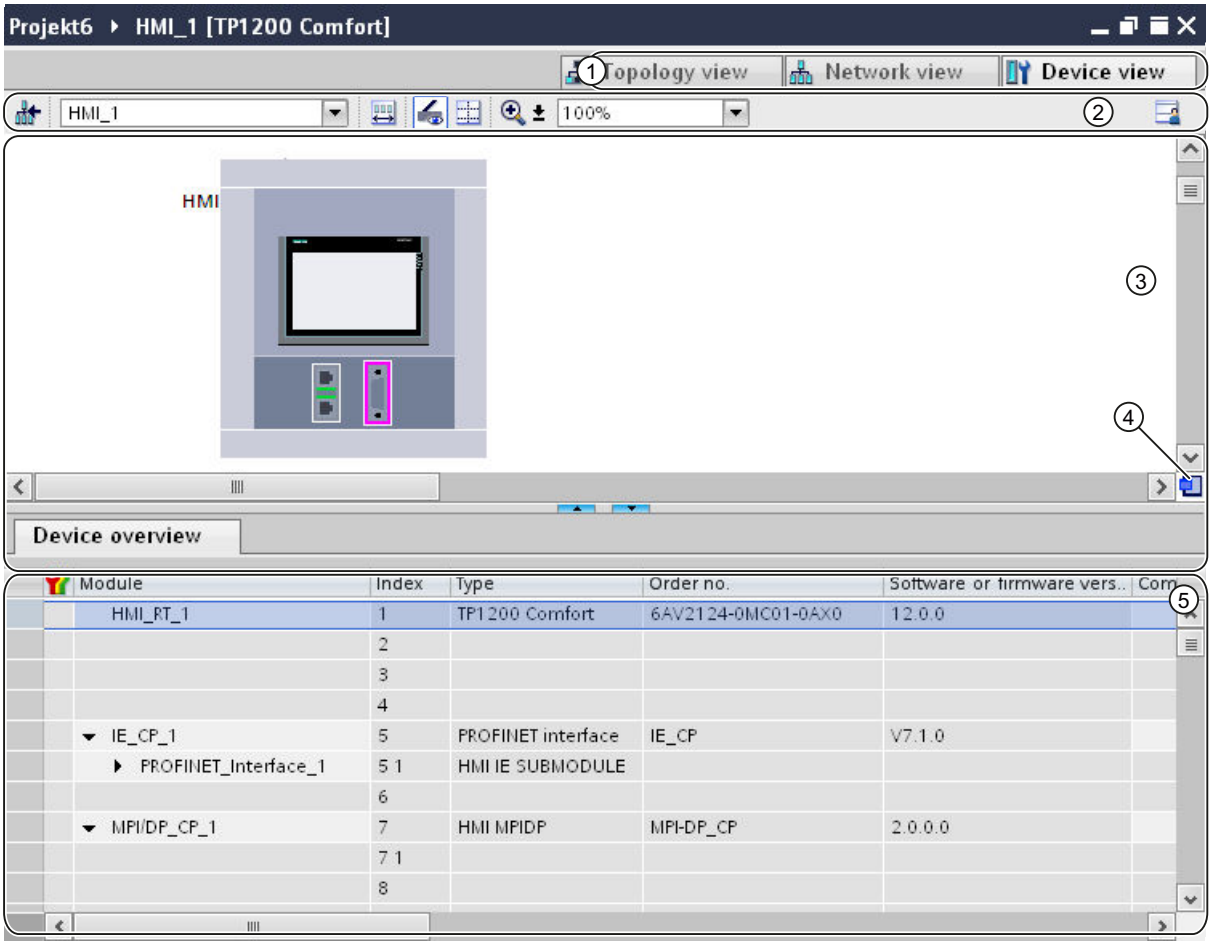
The device view is a working area of the hardware and network editor.

You undertake the following tasks in the device view:

- Configuring and assign device parameters
- Configuring and assign module parameters

Structure

The following diagram shows the components of the device view:



1 Changeover switch: network view/device view/topology view

2 Toolbar of device view

3 Graphic area of the device view

4 Overview navigation

5 Table area of device view

You can use your mouse to change the spacing between the graphic and table areas of the device view.

To do this, click on the upper edge of the table view and expand or contract this by moving the mouse with the mouse button held down. You can use the two small arrow keys to minimize, maximize or select the latest table structure of the table view with just one click.

Toolbar

The toolbar provides the following functions:

Icon	Meaning
	Switches to the network view. Note: The device view can switch between the existing devices using the drop-down list.
	Show the area of unplugged modules.
	Show module labels.
	Adjust the zoom setting. Select the zoom setting or enter it directly in the adjacent drop-down list. You can use the Zoom button to zoom in or out incrementally or to drag a frame around an area to be enlarged. You can read the address descriptions of the I/O channels of signal modules at a zoom level setting of 200% or higher.
	Show page breaks Enables the page break preview. Dashed lines will be displayed at the positions where the pages will break when printed.
	Remember layout Saves the current table view. The layout, width and visibility of columns in the table view is stored.

Graphic area

The graphic area of the device view displays hardware components and if necessary the associated modules that are assigned to each other via one or more racks. In the case of devices with racks, you have the option of installing additional hardware objects from the hardware catalog into the slots on the racks.

Overview navigation

Click in the overview navigation to obtain an overview of the created objects in the graphic area. By holding down the mouse button, you can quickly navigate to the desired objects and display them in the graphic area.

Table area

The table area of the device view gives you an overview of the modules used and the most important technical and organizational data.

You can use the shortcut menu of the title bar of the table to adjust the tabular display.

See also

"Devices & networks" editor (Page 16)

Network view (Page 18)

Network data (Page 21)

Diagnostics of online connections (Page 24)

Topology view (Page 28)
Inspector window (Page 31)
Hardware catalog (Page 32)
Information on hardware components (Page 35)

1.2.6 Topology view (Basic Panels)

Introduction

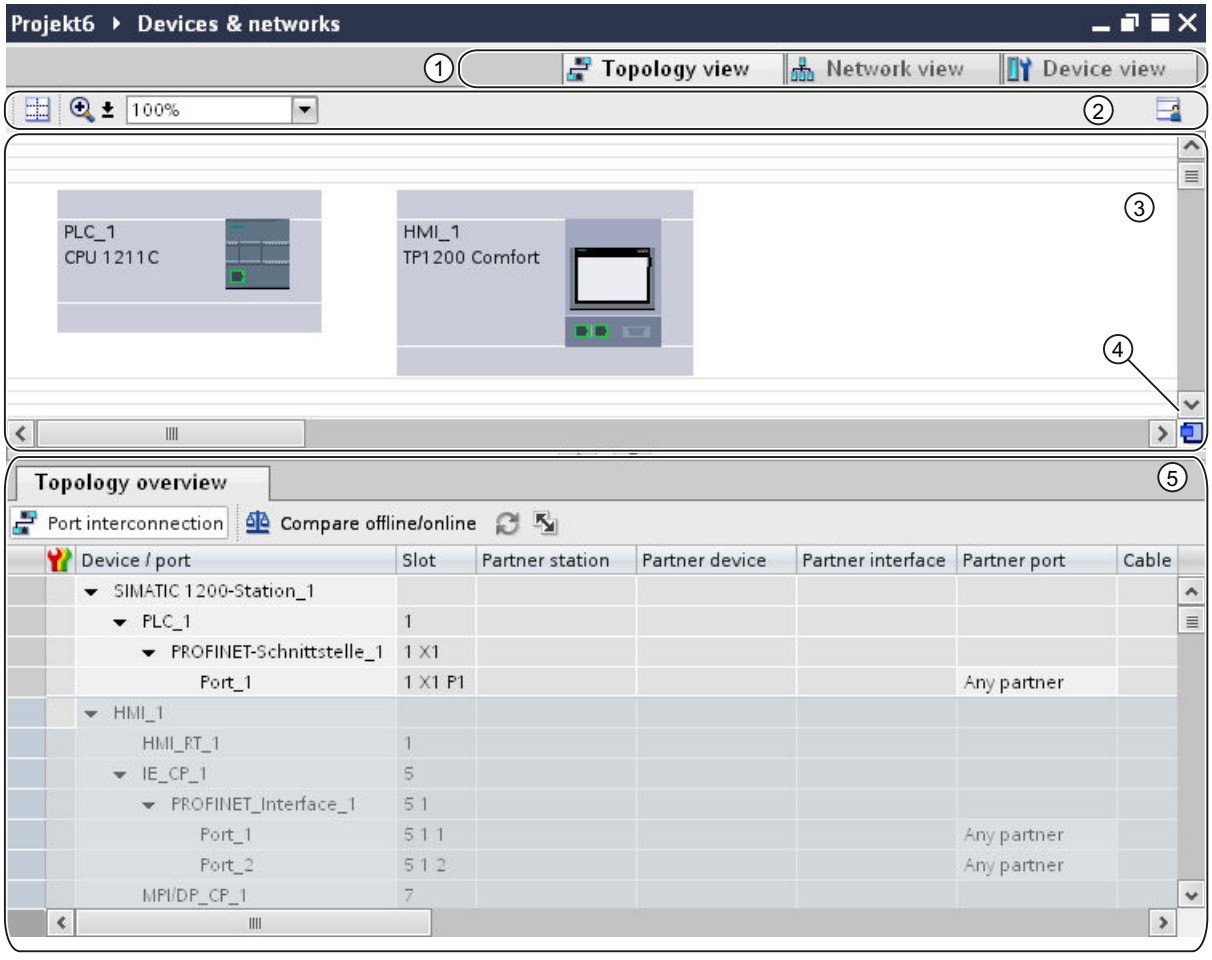
The topology view is a working area of the hardware and network editor.

You undertake the following tasks in topology view:

- Displaying the Ethernet topology
- Configuring the Ethernet topology
- Identify and minimize differences between the desired and actual topology

Structure

The following figure provides an overview of the topology view.



1 Changeover switch: device view / network view / topology view

2 Topology view toolbar

3 Graphic area of the topology view

4 Overview navigation

5 Table area of the topology view

You can use your mouse to change the spacing between the graphic and table areas of the topology view.

To do this, click on the upper edge of the table view and expand or contract this by moving the mouse with the mouse button held down. You can use the two small arrow keys to minimize, maximize or select the latest table structure of the table view with just one click.

Toolbar

The toolbar provides the following functions:

Icon	Meaning
	Adjust the zoom setting. You can select the zoom setting via the adjacent drop-down list or enter it directly. You can also zoom in or zoom out the view in steps using the zoom symbol or draw a frame around an area to be zoomed in.
	Show page breaks Enables the page break preview. Dashed lines will be displayed at the positions where the pages will break when printed.
	Remember layout Saves the current table view. The layout, width and visibility of columns in the table view is stored.

Graphic area

The graphic area of the topology view displays Ethernet modules with their appropriate ports and port connections. Here you can add additional hardware objects with Ethernet interfaces.

Overview navigation

Click in the overview navigation to obtain an overview of the created objects in the graphic area. By holding down the mouse button, you can quickly navigate to the desired objects and display them in the graphic area.

Table area

This displays the Ethernet or PROFINET modules with their appropriate ports and port connections in a table. This table corresponds to the network overview table in the network view.

You can use the shortcut menu of the title bar of the table to adjust the tabular display.

See also

"Devices & networks" editor (Page 16)

Network view (Page 18)

Network data (Page 21)

Diagnostics of online connections (Page 24)

Device view (Page 25)

Inspector window (Page 31)

Hardware catalog (Page 32)

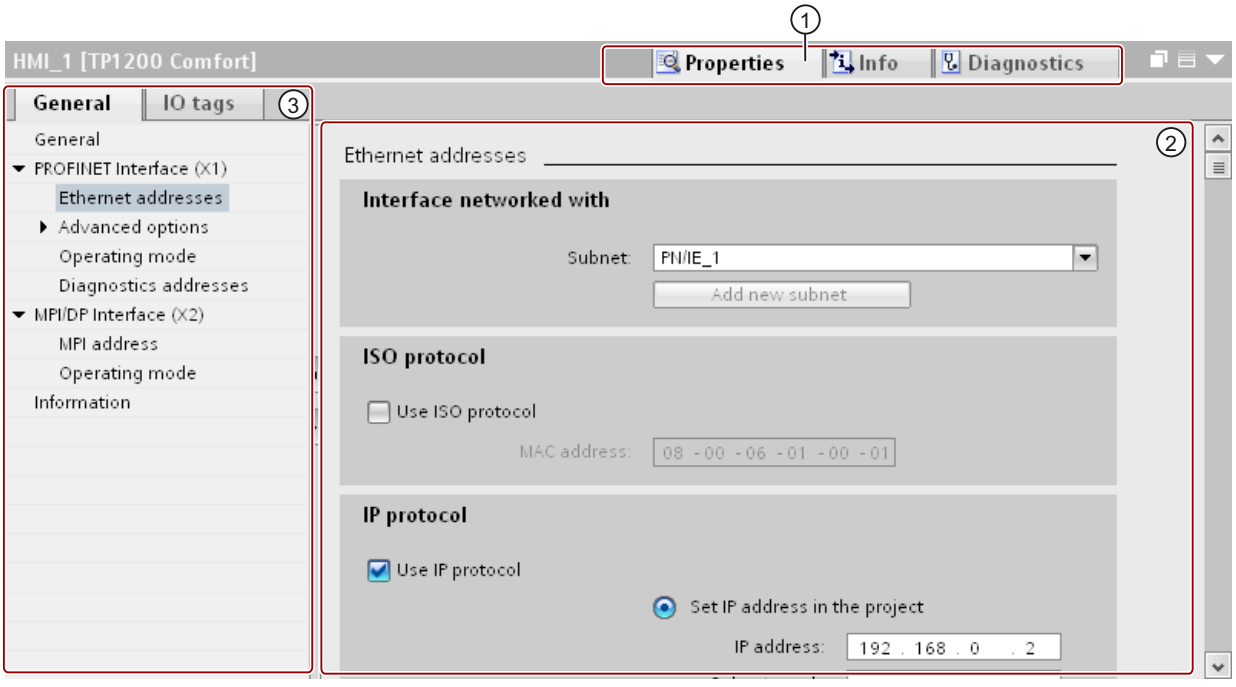
Information on hardware components (Page 35)

1.2.7 Inspector window (Basic Panels)

The properties and parameters shown for the object selected can be edited in the inspector window.

Structure

The inspector window consists of the following components:



1	Switch between various information and work areas
2	Navigation between various pieces of information and parameters
3	Display showing the selected information and parameters

Function

The information and parameters in the inspector window are split into different types of information:

- Properties
- Info
- Diagnostics

To display the corresponding information and parameters, click in the area you want. The "Properties" area is the most important one for configuring an automation system. This area is displayed by default.

The left pane of the inspector window is used for area navigation. Information and parameters are arranged there in groups. If you click on the arrow symbol to the left of the group name, you can expand the group if sub-groups are available. If you select a group or sub-group, the

corresponding information and parameters are displayed in the right pane of the inspector window and can be edited there too.

See also

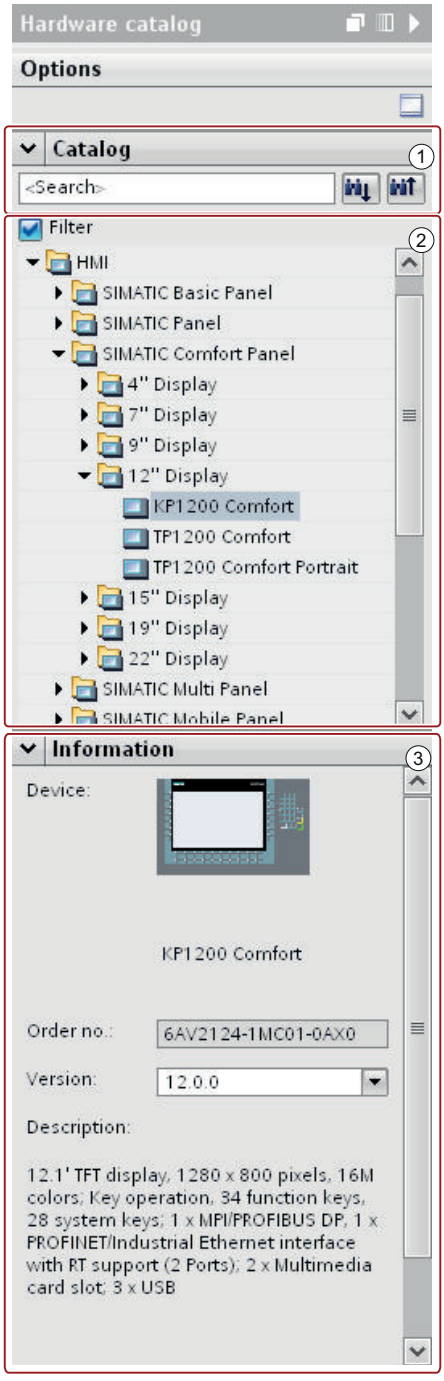
"Devices & networks" editor (Page 16)
Network view (Page 18)
Network data (Page 21)
Diagnostics of online connections (Page 24)
Device view (Page 25)
Topology view (Page 28)
Hardware catalog (Page 32)
Information on hardware components (Page 35)

1.2.8 Hardware catalog (Basic Panels)

The "Hardware catalog" task card gives you easy access to a wide range of hardware components.

Structure

The "Hardware catalog" task card consists of the following panes:



1 "Catalog" pane, search and filter function

2 "Catalog" pane, component selection

3 "Information" pane

Search and filter function

The search and filter functions of the "Catalog" pane make it easy to search for particular hardware components. You can limit the display of the hardware components to certain criteria using the filter function. For example, you can limit the display to objects that you can also place within the current context or which contain certain functions.

Objects that can be used in the current context include, for example, interconnectable objects in the network view or only modules compatible with the device in the device view.

Component selection

The component selection in the "Catalog" pane contains the installed hardware components in a tree structure. You can move the devices or modules you want from the catalog to the graphic work area of the device or network view.

Installed hardware components without a license are grayed out. You cannot use non-licensed hardware components.

Hardware components belonging to various components groups thematically are partially implemented as linked objects. When you click on such linked hardware components, a catalog tree opens in which you can find the appropriate hardware components.

Information

The "Information" pane contains detailed information on the object selected from the catalog:

- Schematic representation
- Name
- Order number
- Version number
- Description

See also

"Devices & networks" editor (Page 16)

Network view (Page 18)

Network data (Page 21)

Diagnostics of online connections (Page 24)

Device view (Page 25)

Topology view (Page 28)

Inspector window (Page 31)

Information on hardware components (Page 35)

1.2.9 Information on hardware components (Basic Panels)

In the hardware catalog, you can display information on selected hardware components in the "Information" pane. You can also display further information on the selected hardware components using the shortcut menu.

Access to further information

If you select a hardware object in the hardware catalog and open the shortcut menu, you not only have the "Copy" function available but also three options for accessing information on Service & Support:

- Information regarding product support
- FAQs
- Manuals

The required information is displayed in the work area of the hardware and network editor.

Note

You can only access Service & Support when you are connected to the Internet and the function is enabled. By default, the function is disabled.

To enable the function, refer to the instructions in the section "Enabling product support".

Information regarding product support

Here, you have access to general information on hardware and software components. The order number of the selected hardware object is entered automatically in the search mask. You can, however, also search for other hardware and software components.

FAQs

Here, you have access to "Frequently Asked Questions" (FAQs). You can view various entries on hardware and software questions. Using a detailed search mask, you can filter the required topics.

Manuals

Here, you have access to the manuals of the various hardware components. This is particularly useful if the configuration, addressing or parameter assignment you are planning requires more detailed knowledge of the hardware you are using.

See also

"Devices & networks" editor (Page 16)

Network view (Page 18)

Network data (Page 21)

Diagnostics of online connections (Page 24)
Device view (Page 25)
Topology view (Page 28)
Inspector window (Page 31)
Hardware catalog (Page 32)

1.3 Networks and connections (Basic Panels)

1.3.1 SIMATIC communication networks (Basic Panels)

1.3.1.1 Communication networks (Basic Panels)

Overview

Communication networks are a central component of modern automation solutions. Industrial networks have to fulfill special requirements, for example:

- Coupling of automation systems as well as simple sensors, actuators, and computers.
- The information has to be correct and has to be transferred at the right moment.
- Robust against electromagnetic disturbances, mechanical stresses and soiling
- Flexible adaptation to the production requirements

Industrial networks belong to the LANs (Local Area Networks) and allow communication within a limited area.

Industrial networks fulfill the following communication functions:

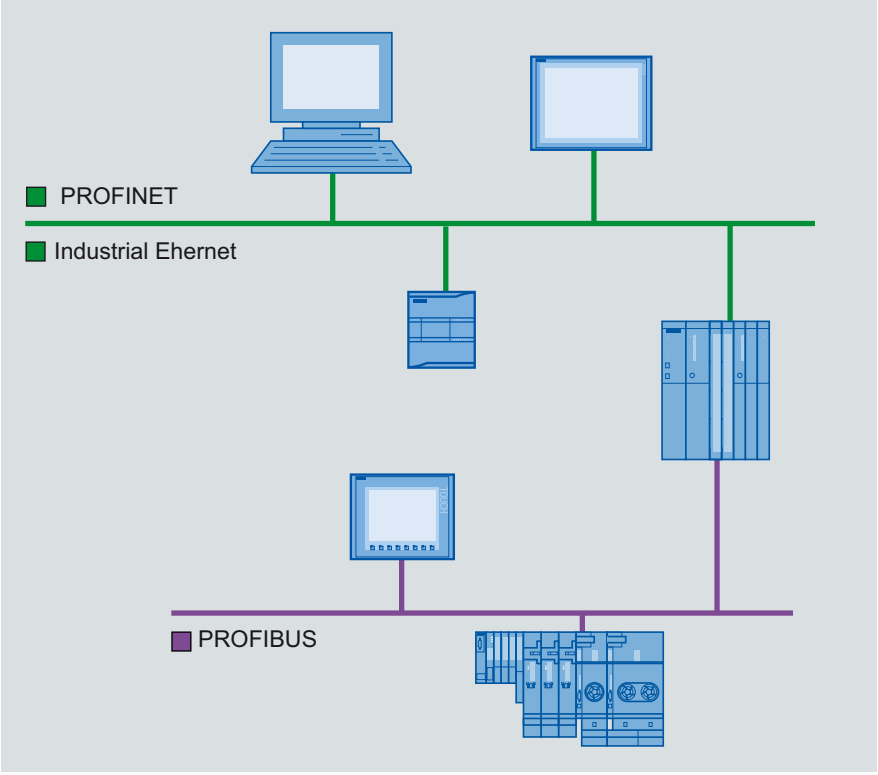
- Process and field communication of the automation systems including sensors and actuators
- Data communication between automation systems
- IT communication for integrating the modern information technology

Overview of the networks

This section examines the following networks:

- **Industrial Ethernet**
  The industrial network standard for all levels
- **PROFINET**
  The open Industrial Ethernet standard for automation
- **PROFIBUS**
  The international standard for the field area and market leader at the field busses

- **MPI**
  The integrated interface of the SIMATIC products
- **PPI**
  The integrated interface specially for the S7-200



See also

PROFINET and Ethernet (Page 37)

PROFIBUS (Page 38)

MPI (Page 39)

PPI (Page 40)

1.3.1.2 PROFINET and Ethernet (Basic Panels)

Industrial Ethernet

Industrial Ethernet, which is based on IEEE 802.3, enables you to connect your automation system to your office networks. Industrial Ethernet provides IT services that you can use to access production data from the office environment.

Ethernet network

An Ethernet network allows you to interconnect all devices that are connected to the network via an integrated Ethernet interface or a communication module. This enables connection of multiple HMI devices to one SIMATIC S7 PLC and multiple SIMATIC S7 PLCs to one HMI device. The maximum number of communication partners that you can connect to an HMI device is dependent on the HMI device used. Additional information is available in the documentation for the respective HMI device.

PROFINET

PROFINET is an open standard for industrial automation defined by IEEE 61158 and based on Industrial Ethernet. PROFINET makes use of IT standards all the way to the field level and enables plant-wide engineering.

With PROFINET, you can realize high-performance automation solutions for applications with stringent real-time requirements.

See also

Communication networks (Page 36)

1.3.1.3 PROFIBUS (Basic Panels)

PROFIBUS DP

PROFIBUS DP (distributed I/O) is used to connect the following devices:

- PLCs, PCs, HMI devices
- Distributed I/O devices, e.g., SIMATIC ET 200
- Valves
- Drives

PROFIBUS DP's fast response times make it ideally suited for the manufacturing industry.

Its basic functionality includes cyclic exchange of process data between the master and PROFIBUS DP slaves, as well as diagnostics.

PROFIBUS network

You can connect an HMI device within the PROFIBUS network to any SIMATIC S7 module that has an integrated PROFIBUS or PROFIBUS DP interface. You can thereby connect multiple HMI devices to one SIMATIC S7 PLC and multiple SIMATIC S7 PLCs to one HMI device.

The maximum number of communication partners that you can connect to an HMI device is dependent on the HMI device used. Additional information is available in the documentation for the respective HMI device.

You configure the SIMATIC S7-200 PLC as a passive device in the network. You connect the SIMATIC S7-200 using the DP connector or a PROFIBUS communication module.

See also

Communication networks (Page 36)

1.3.1.4 MPI (Basic Panels)

MPI

MPI (Multi-Point Interface) is the integrated interface for SIMATIC products:

- PLCs
- HMI devices
- Programming device/PC

Small subnets with the following characteristics are set up with MPI:

- Short distances
- Few devices
- Small data quantities

MPI network

You connect the HMI device to the MPI interface of the SIMATIC S7 PLC. This enables connection of multiple HMI devices to one SIMATIC S7 PLC and multiple SIMATIC S7 PLCs to one HMI device. The maximum number of communication partners that you can connect to an HMI device is dependent on the HMI device used. Additional information is available in the documentation for the respective HMI device.

Network architectures

MPI is based on the PROFIBUS standard (IEC 61158 and EN 50170) and supports the following bus topologies:

- Line
- Star
- Tree

An MPI subnet contains a maximum of 127 devices and consists of multiple segments. Each segment contains a maximum of 32 devices and is limited by terminating resistors. Repeaters are used to connect segments. The maximum cable length without a repeater is 50 m.

See also

Communication networks (Page 36)

1.3.1.5 PPI (Basic Panels)

Introduction

PPI (point-to-point interface) is an integrated interface that was developed specially for the SIMATIC S7-200. A PPI network typically connects S7-200 PLCs. However, other SIMATIC PLCs (e.g., S7-300 and S7-400) or HMI devices can communicate with a SIMATIC S7-200 in the PPI network.

PPI network

A PPI connection is a point-to-point connection. The HMI device is the master. The SIMATIC S7-200 is the slave.

You can connect a maximum of one SIMATIC S7-200 to an HMI device. You use the serial connector of the CPU to connect the HMI device. You can connect multiple HMI devices to one SIMATIC S7-200. From the perspective of the SIMATIC S7-200, only one connection at a time is possible.

Note

The PPI network can contain a maximum of four masters in addition to the HMI device. For performance reasons, do not configure more than four devices at a time as a master in the PPI network.

Network architectures

PPI is based on the PROFIBUS standard (IEC 61158 and EN 50170) and supports the following bus topologies:

- Line
- Star

Multi-master networks with a maximum of 32 masters are set up with PPI:

- An unlimited number of masters can communicate with each slave.
- A slave can be assigned to multiple masters.

The RS 485 repeater can be used to extend the PPI network. Modems can also be connected to the PPI network.

See also

Communication networks (Page 36)

1.3.2 Configuring networks and connections (Basic Panels)

1.3.2.1 Networking devices (Basic Panels)

Introduction

The "Devices & Networks" editor is provided for configuring connections. You can network devices in the editor. You can also configure and assign parameters to devices and interfaces. You then configure the required connections between the networked devices.

In the "Devices & Networks" editor you configure HMI connections with the PLCs:

- SIMATIC S7 1500
- SIMATIC S7 1200
- SIMATIC S7 300
- SIMATIC S7 400
- SIMATIC ET 200 CPU
- SIMATIC S7-1500 software controller

You configure the HMI connections to other PLCs in the "Connections" editor of the respective HMI device.

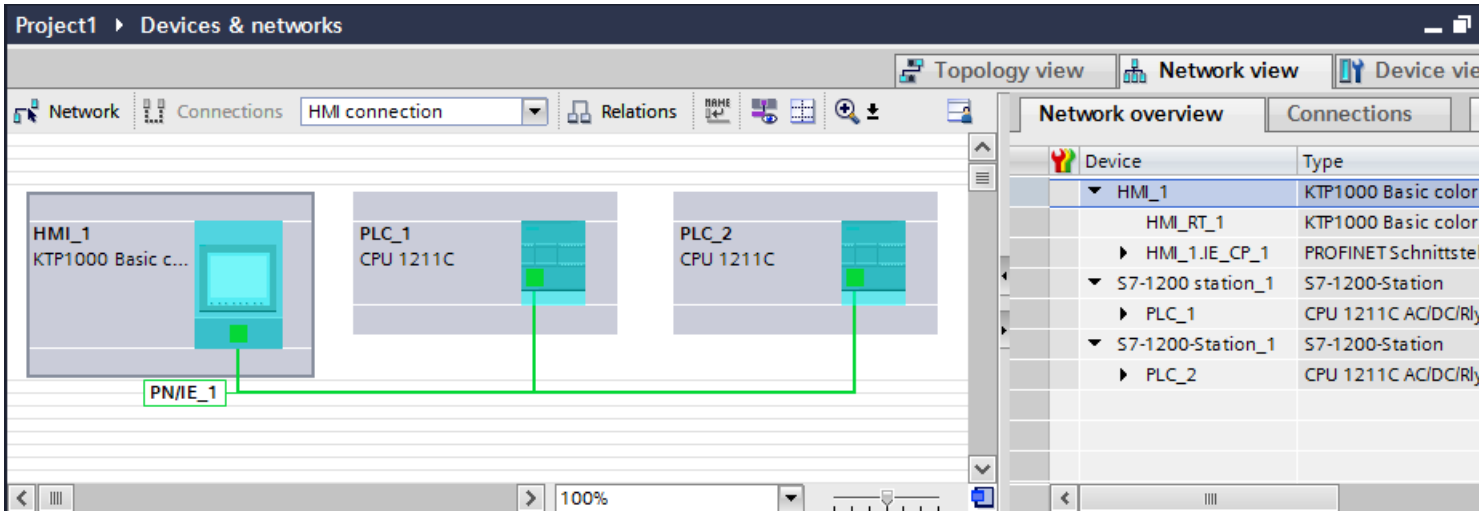
Networking devices

The network view of the "Devices & Networks" editor includes a graphical area and a tabular area. You can use the graphical area to network the devices in the project with drag-and-drop. The tabular area provides an overview of the devices and their components.

You can network the following PLCs together with HMI devices in the "Devices & Networks" area.

- SIMATIC S7 1500
- SIMATIC S7 1200
- SIMATIC S7 300
- SIMATIC S7 400
- SIMATIC ET 200 CPU
- SIMATIC S7-1500 software controller

All other PLCs are available in the TIA Portal and are configured "not integrated". You configure "not integrated" connections in the "Connections" editor of the HMI device.



With the networking step, you configure the physical connection of the communication partners. The networking of devices is depicted by lines that are colored according to the interface.

See also

Configuring an integrated connection in the "Devices & Networks" editor (Page 42)

Special considerations of the "Devices & Networks" editor (Page 44)

Configuring a non-integrated connection in the "Connections" editor (Page 46)

Configuring a routed connection in the "Devices & Networks" editor (Page 50)

1.3.2.2 Configuring an integrated connection in the "Devices & Networks" editor (Basic Panels)

Introduction

You configure an HMI connection between the HMI device and a SIMATIC S7 PLC in the "Devices & Networks" editor. This HMI connection is the direct connection between the communication partners that you have created in a project.

Integrated connections

Connections of devices within a project are referred to as integrated connections. In the case of integrated connections, you can directly configure addresses of PLC tags.

Note

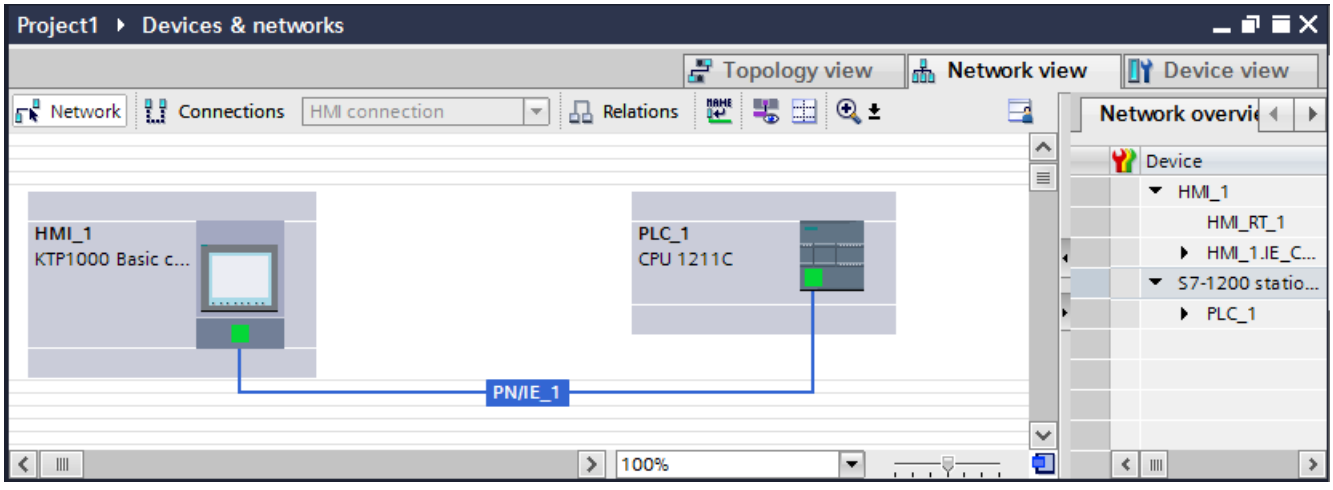
An HMI connection can be configured in the "Devices & Networks" editor for the following PLCs only:

- SIMATIC S7 1500
- SIMATIC S7 1200
- SIMATIC S7 300
- SIMATIC S7 400
- SIMATIC ET 200 CPU
- SIMATIC S7-1500 software controller

You configure the HMI connections to all other PLCs in the "Connections" editor of the HMI device.

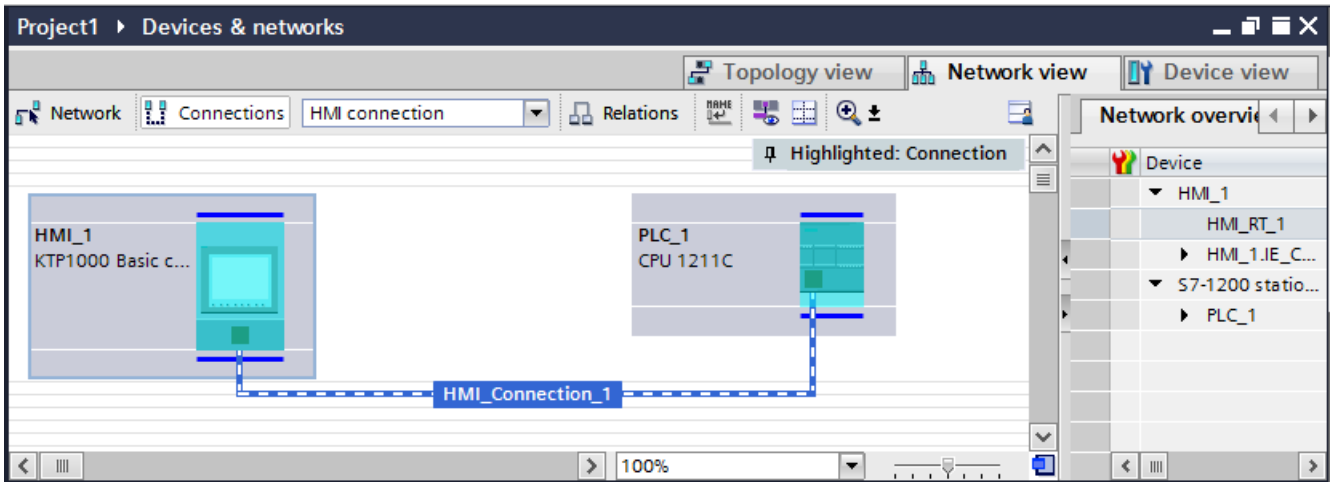
Configuring an HMI connection in the "Devices & Networks" editor

1. Insert an HMI device and a SIMATIC S7 PLC in your project.



2. Switch to "Connections" mode.
3. Select the "HMI connection" connection type.

4. Use a drag-and-drop operation to interconnect the two PROFINET interfaces.



5. Change the IP address and subnet mask address parameters according to the requirements of your project.

See also

Networking devices (Page 41)

Special considerations of the "Devices & Networks" editor (Page 44)

Configuring a non-integrated connection in the "Connections" editor (Page 46)

Configuring a routed connection in the "Devices & Networks" editor (Page 50)

1.3.2.3 Special considerations of the "Devices & Networks" editor (Basic Panels)

Introduction

If you are configuring or have already configured networks or HMI connections, the "Devices & Networks" editor supports you with the following functions:

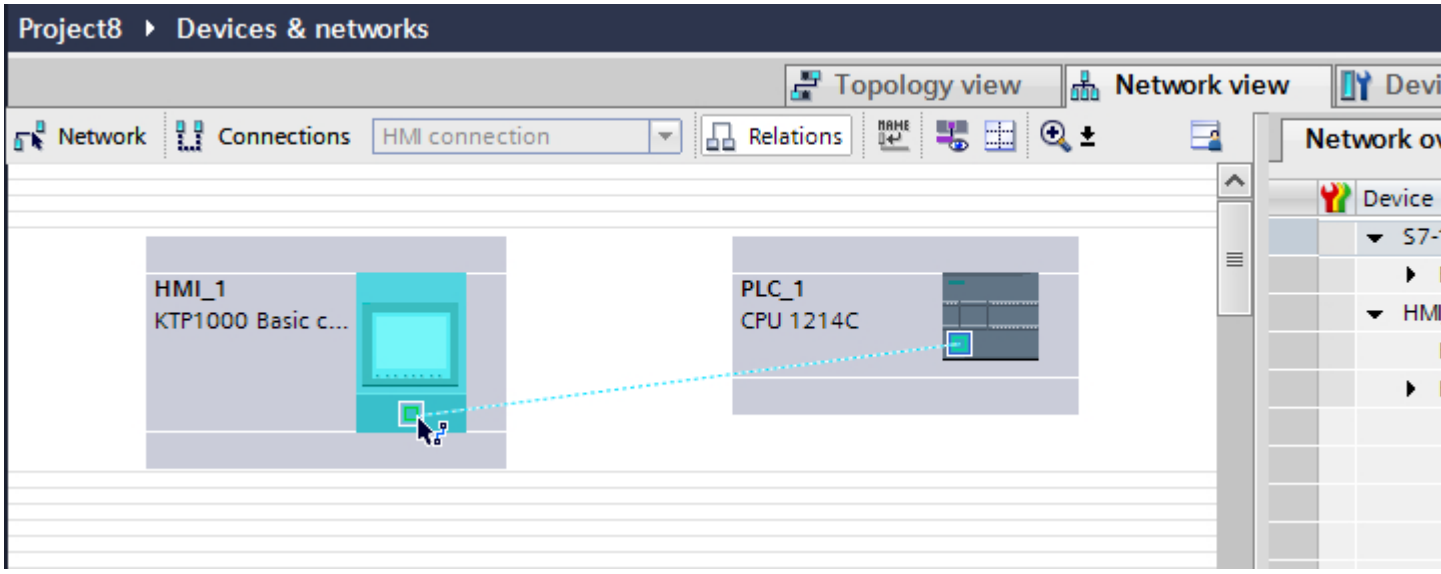
- Highlighting of communication partners
- Highlighting of HMI connections
- Automatic creation of subnets

Highlighting of communication partners

All communication partners for which an HMI connection is possible are highlighted in turquoise if you have selected the "HMI connection" type.

Starting from the interface of a device create an HMI connection to the device of another device using a drag-and-drop operation. During the drag-and-drop operation all potential communication partners are highlighted in turquoise.

Use the ESC key to stop connecting interfaces using a drag-and-drop operation.



When the mouse pointer is moved over the interface of a device, the following icons indicate whether a connection is possible:



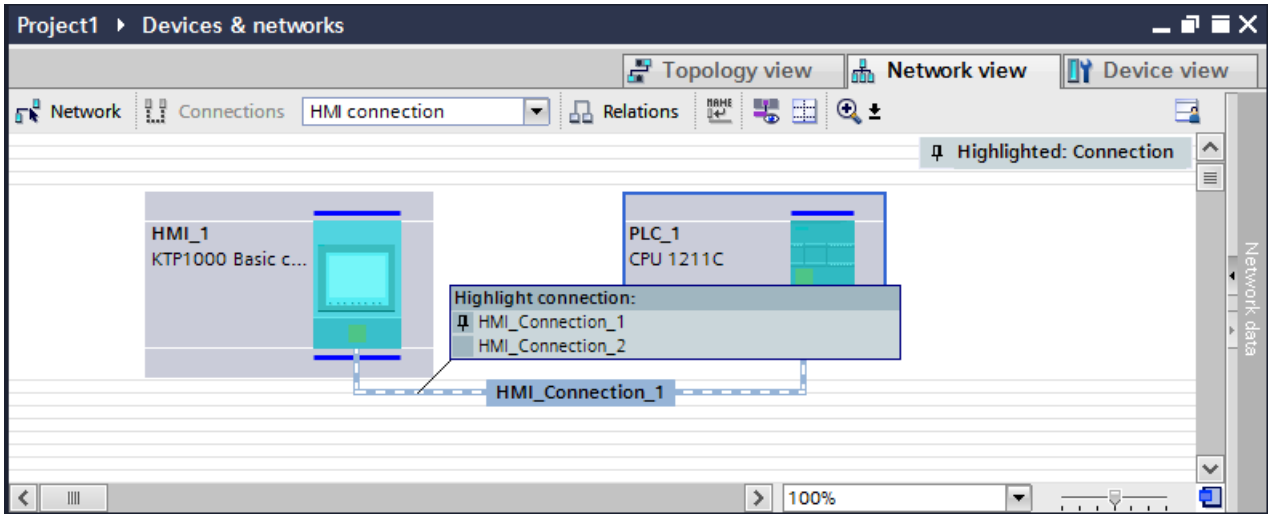
A connection is possible.



A connection is not possible.

Highlighting of HMI connections

A turquoise highlighting of the connection indicates that a HMI connection was created. If several HMI connections are created, you can select one of the already created HMI connections in a dialog.



Then you can configure the parameters of the selected HMI connection and the communication partners in the inspector window.

Subnets

Subnetworks are automatically created and used under the following conditions:

- If both communication partners are not already interconnected in different networks
- If a free interface is available to both communication partners.
- If a subnetwork already exists then that subnetwork is automatically used for the HMI connection.

See also

Networking devices (Page 41)

Configuring an integrated connection in the "Devices & Networks" editor (Page 42)

Configuring a non-integrated connection in the "Connections" editor (Page 46)

Configuring a routed connection in the "Devices & Networks" editor (Page 50)

1.3.2.4 Configuring a non-integrated connection in the "Connections" editor (Basic Panels)

Introduction

You use the "Connections" editor of the HMI device to configure a connection between an HMI device and a PLC that cannot be configured in the "Devices & Networks" editor.

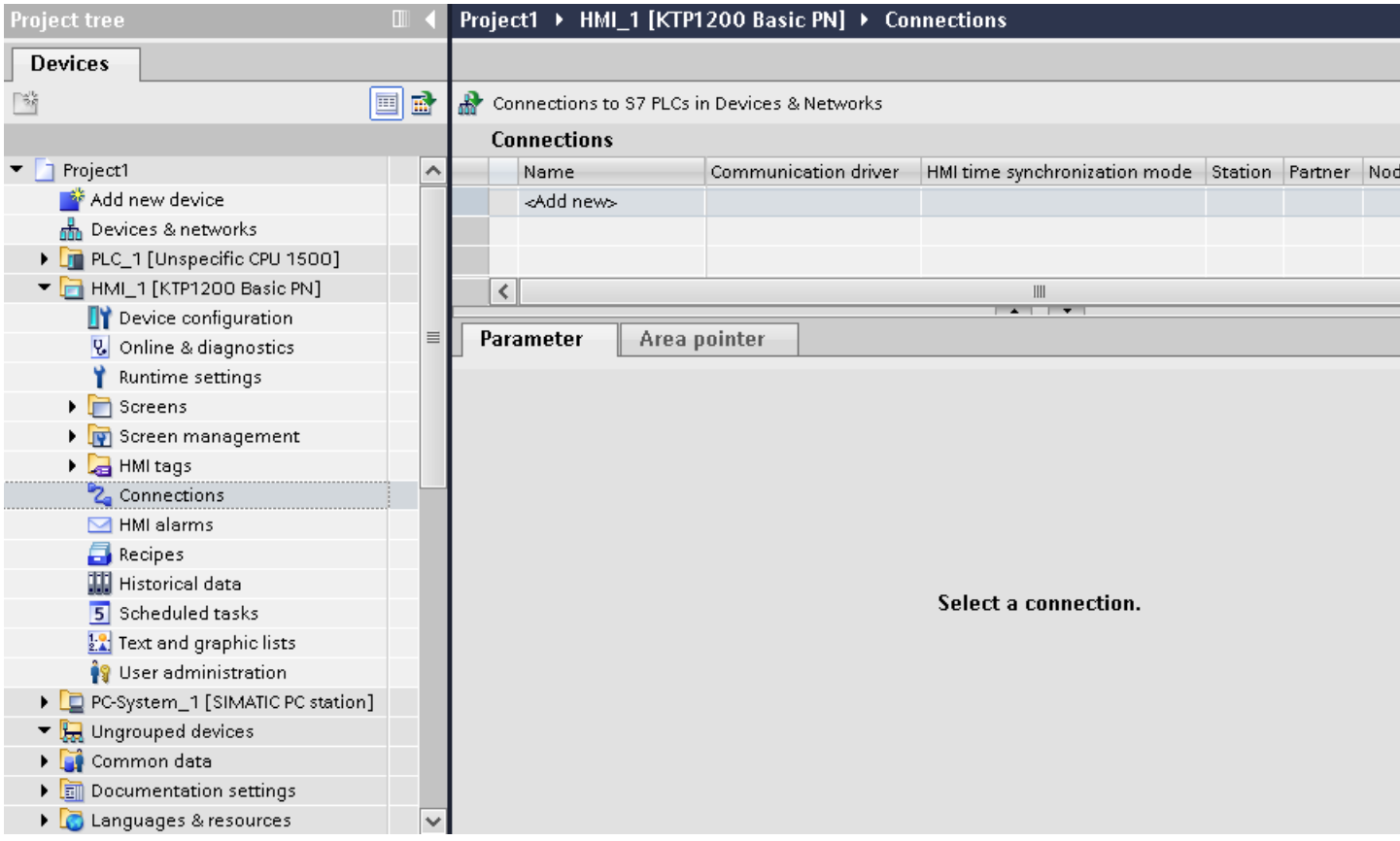
These connections are referred to as non-integrated connections.

Requirements

- A project is open.
- An HMI device has been created.

Configuring a connection in the "Connections" editor

1. Open the "Connections" editor of the HMI device.
2. Create a new connection.

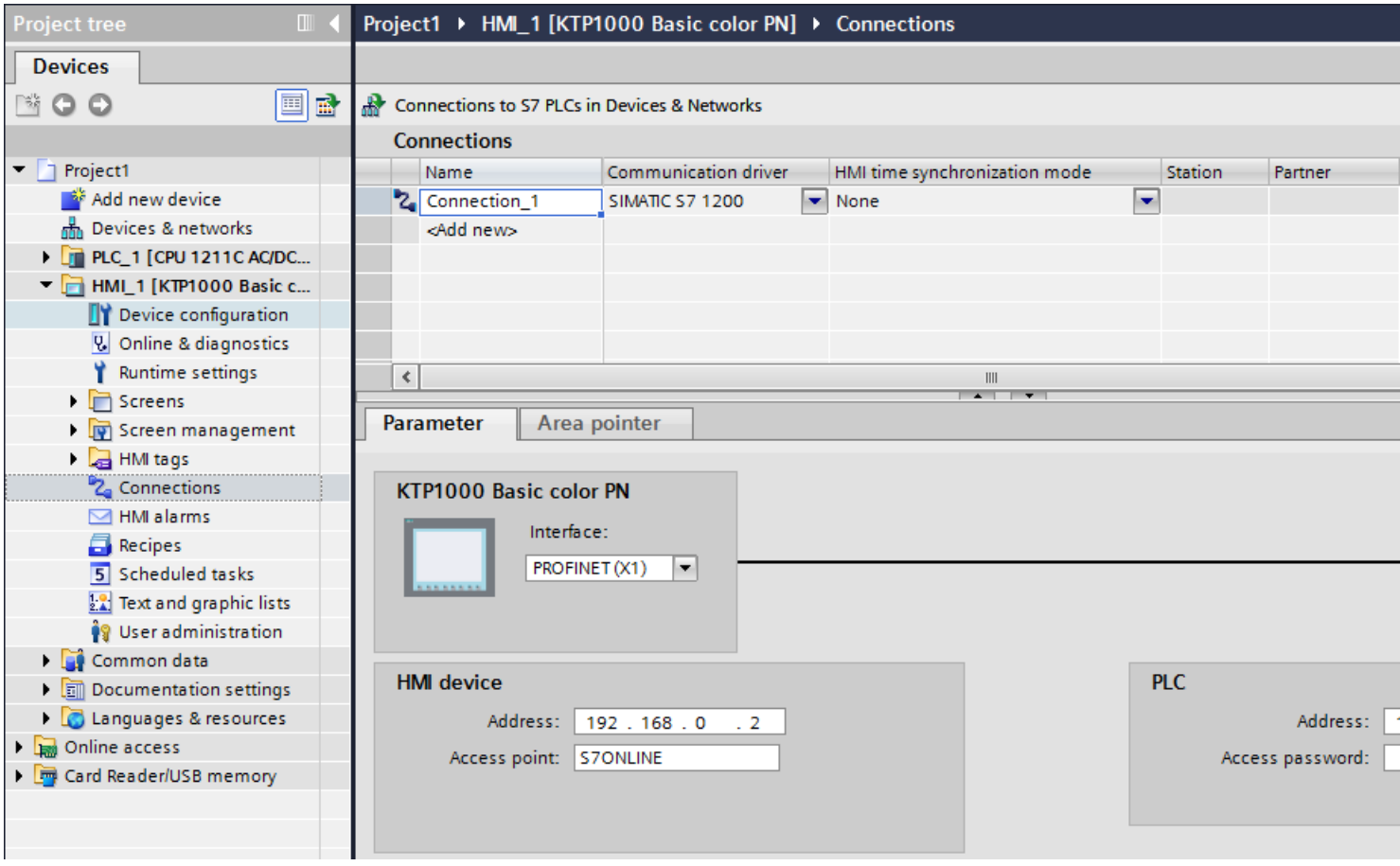


3. Select the communication driver.

Note

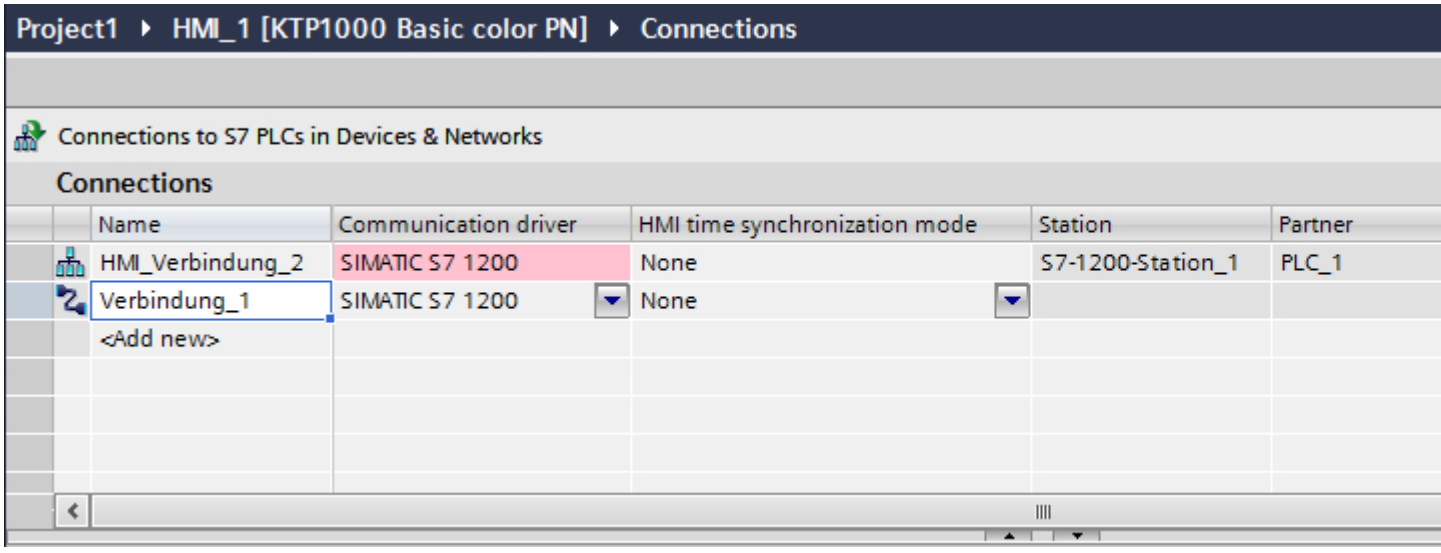
A non-integrated connection between an HMI device and a SIMATIC S7-1500 software controller is not supported in WinCC.

4. Set the connection parameters.



Integrated connections in the "Connections" editor

If you have already configured the integrated connections of the HMI device in the "Devices & Networks" editor, they are also displayed in the "Connections" editor.



Meaning of the icons used:

- Integrated connection
- Non-integrated connection

See also

Networking devices (Page 41)

Configuring an integrated connection in the "Devices & Networks" editor (Page 42)

Special considerations of the "Devices & Networks" editor (Page 44)

Configuring a routed connection in the "Devices & Networks" editor (Page 50)

1.3.2.5 Configuring a routed connection in the "Devices & Networks" editor (Basic Panels)

Introduction

You can configure a routed HMI connection to a PLC in another subnet in the "Devices & Networks" editor.

Note

A routed HMI connection can only be configured for the following PLCs:

- SIMATIC S7 1500
- SIMATIC S7 1200
- SIMATIC ET 200 CPU
- SIMATIC S7-1500 Software Controller
- SIMATIC S7 300/400

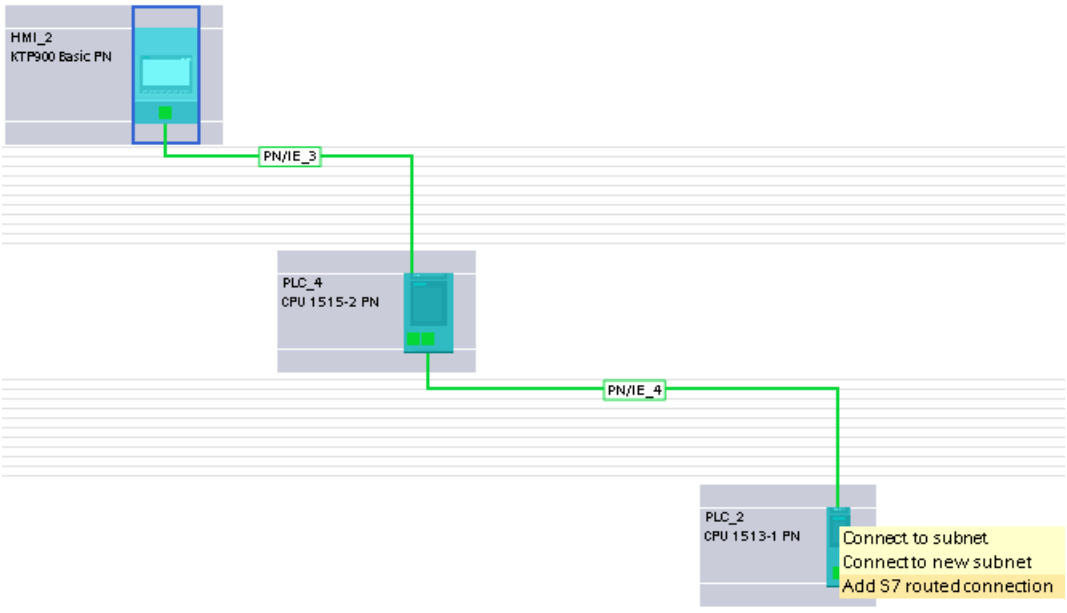
Requirement

- An HMI device has been created.
- PLCs have been set up in different networks
- The network view in the "Devices & Networks" editor is open.

Configuring a routed HMI connection

1. Switch to "Connections" mode.
2. Select the "HMI connection" connection type.

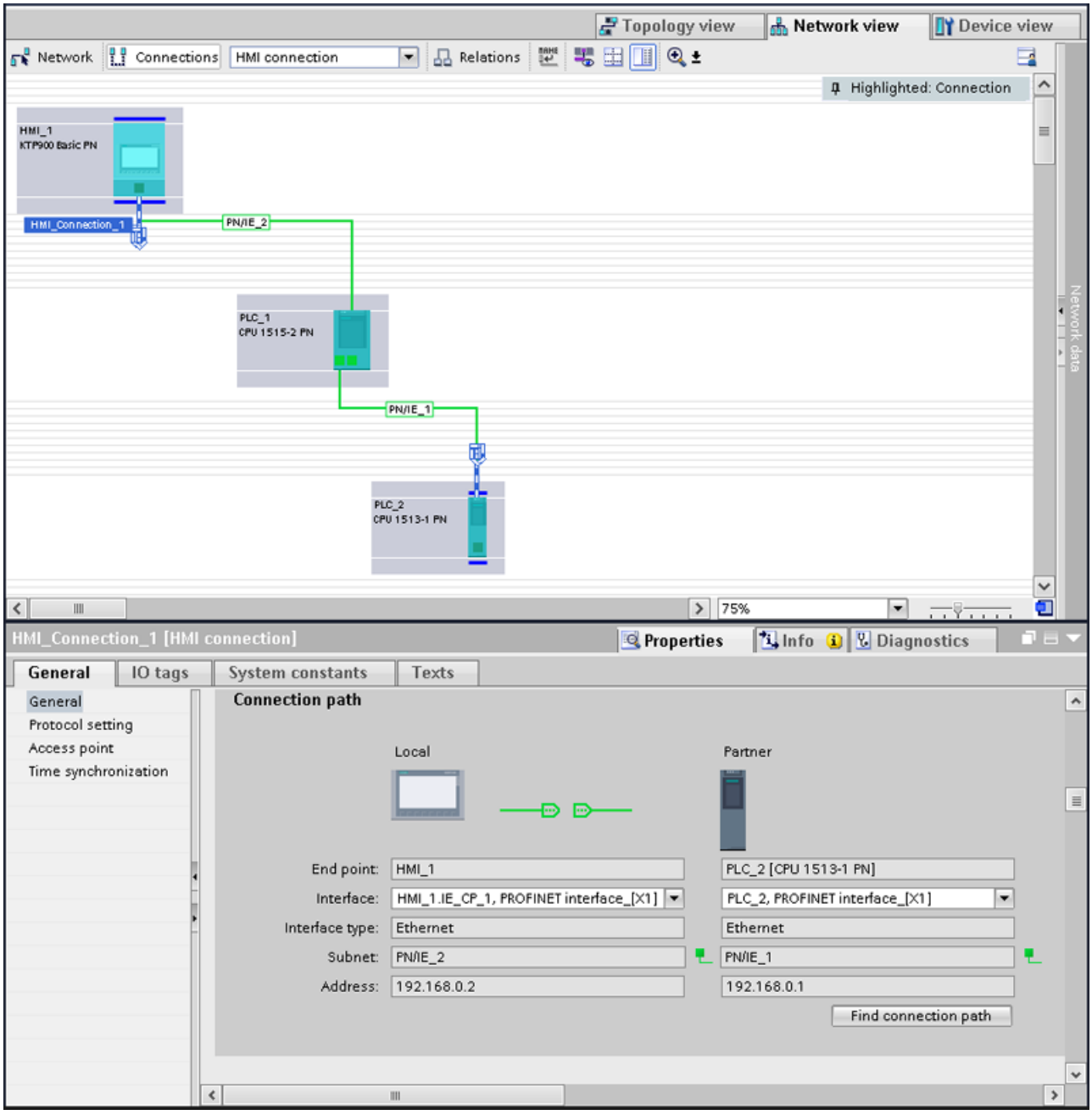
3. Drag-and-drop the HMI device to the PLC in the other subnet.
   A dialog opens.



4. Select "Add routed connection".

Result

The routed HMI connection has now been created. If you change the HMI device type or the configuration of the PLC, you will need to adapt the routed HMI connection again.



See also

Networking devices (Page 41)

Configuring an integrated connection in the "Devices & Networks" editor (Page 42)

Special considerations of the "Devices & Networks" editor (Page 44)

Configuring a non-integrated connection in the "Connections" editor (Page 46)

1.4 Data exchange (Basic Panels)

1.4.1 Data exchange using tags (Basic Panels)

1.4.1.1 Basics of tags (Basic Panels)

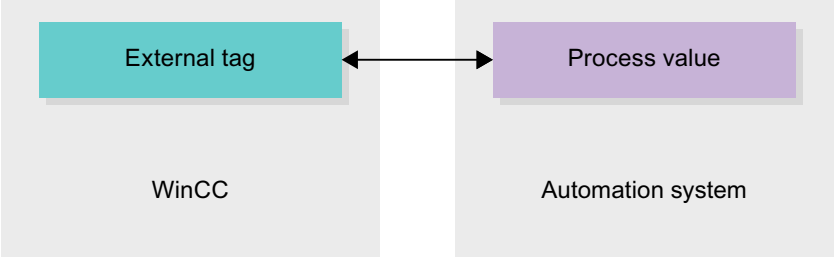
Introduction

Process values are forwarded in runtime using tags. Process values are data which is stored in the memory of one of the connected automation systems. They represent the status of a plant in the form of temperatures, fill levels or switching states, for example. Define external tags for processing the process values in WinCC.

WinCC works with two types of tag:

- External tags
- Internal tags

The external tags form the link between WinCC and the automation systems. The values of external tags correspond to the process values from the memory of an automation system. The value of an external tag is determined by reading the process value from the memory of the automation system. It is also possible to rewrite a process value in the memory of the automation system.



Internal tags do not have a process link and only convey values within the WinCC. The tag values are only available as long as runtime is running.

Tags in WinCC

For external tags, the properties of the tag are used to define the connection that the WinCC uses to communicate with the automation system and form of data exchange.

Tags that are not supplied with values by the process - the internal tags - are not connected to the automation system. In the tag's "Connection" property, this is identified by the "Internal tag" entry.

You can create tags in different tag tables for greater clarity. You then directly access the individual tag tables in the "HMI tags" node in the project tree. The tags from all tag tables can be displayed with the help of the table "Show all tags".

With structures you bundle a number of different tags that form one logical unit. Structures are project-associated data and are available for all HMI devices of the project. You use the "Types" editor in the project library to create and edit a structure.

See also

Overview of HMI tag tables (Page 54)

External tags (Page 56)

Addressing external tags (Page 58)

Internal Tags (Page 60)

1.4.1.2 Overview of HMI tag tables (Basic Panels)

Introduction

HMI tag tables contain the definitions of the HMI tags that apply across all devices. A tag table is created automatically for each HMI device created in the project.

In the project tree there is an "HMI tags" folder for each HMI device. The following tables can be contained in this folder:

- Default tag table
- User-defined tag tables
- Table of all tags

In the project tree you can create additional tag tables in the HMI tags folder and use these to sort and group tags and constants. You can move tags to a different tag table using a drag-and-drop operation or with the help of the "Tag table" field. Activate the "Tags table" field using the shortcut menu of the column headings.

Default tag table

There is one default tag table for each HMI device of the project. It cannot be deleted or moved. The default tag table contains HMI tags and, depending on the HMI device, also system tags. You can declare all HMI tags in the standard tags table or, as necessary, additional user-defined tables of tags.

User-defined tag tables

You can create multiple user-defined tag tables for each HMI device in order to group tags according to your requirements. You can rename, gather into groups, or delete user-defined tag tables. To group tag tables, create additional subfolders in the HMI tags folder.

All tags

The "All tags" table shows an overview of all HMI tags and system tags of the HMI device in question. This table cannot be deleted, renamed or moved. This table also contains the "Tags table" column, which indicates the tag table of where a tag is included. Using the "Tags table" field, the assignment of a tag to a tags table can be changed.

With devices for Runtime Professional, the table "All tags" contain an additional tab "System tags". The system tags are created by the system and used for internal management of the project. The names of the system tags begin with the "@" character. System tags cannot be deleted or renamed. You can evaluate the value of a system tag, but cannot modify it.

Additional tables

The following tables are also available in an HMI tag table:

- Discrete alarms
- Analog alarms
- Logging tags

With the help of these tables you configure alarms and logging tags for the currently selected HMI tag.

Discrete alarms table

In the "Discrete alarms" table, you configure discrete alarms to the HMI tag selected in the HMI tag table. When you configure a discrete alarm, multiple selection in the HMI tag table is not possible. You configure the discrete alarms for each HMI tag separately.

Analog alarms table

In the "Analog alarms" table, you configure analog alarms to the HMI tag selected in the HMI tag table. When you configure an analog alarm, multiple selection in the HMI tag table is not possible. You configure the analog alarms for each HMI tag separately.

Logging tags table

In the "Logging tags" table, you configure logging tags to the HMI tag selected in the HMI tag table. When you configure a logging tag, multiple selection in the HMI tag table is not possible. You configure the logging tags for each HMI tag separately. The "Logging tags" table is only available if the HMI device used supports logging.
If WinCC Runtime Professional is used, you can also assign several log tags to a tag. With the other HMI devices, you can only assign one log tag to a tag.

See also

Basics of tags (Page 53)

External tags (Page 56)

Addressing external tags (Page 58)

Internal Tags (Page 60)

1.4.1.3 External tags (Basic Panels)

Introduction

External tags allow communication (data exchange) between the components of an automation system, for example, between an HMI device and a PLC.

Principle

An external tag is the image of a defined memory location in the PLC. You have read and write access to this storage location from both the HMI device and from the PLC.

Since external tags are the image of a storage location in the PLC, the applicable data types depend on the PLC which is connected to the HMI device.

If you write a PLC control program in STEP 7, the PLC tags created in the control program will be added to the PLC tag table. If you want to connect an external tag to a PLC tag, access the PLC tags directly via the PLC tag table and connect them to the external tag.

Data types

All the data types which are available at the connected PLC are available at an external tag in WinCC. Information about data types which are available for connection to other PLCs can be found in the documentation about the respective communication drivers.

See "Communication between devices (Page 9)" for additional information.

Note

As well as external tags, area pointers are also available for communication between the HMI device and PLC. You can set up and enable the area indicators in the "Connections" editor.

Central tag management in STEP 7

You can connect also connect DB instances of user-defined PLC data types (UDT) to the HMI tags.

The PLC data type and the corresponding DB instances are created and updated centrally in STEP 7. In WinCC, you can use the following sources as PLC tag (DB instances):

- Data block elements that use a UDT as data type
- Data block instances of a UDT

The data type is taken from STEP 7 and is not converted to an HMI data type. The access type is always "Symbolic access". Depending on the release for WinCC in STEP 7, elements and structured elements of the PLC data type are applied to WinCC. Elements of a structured

UDT are applied and displayed in the PLC tag table if the instance-specific properties "Visible in HMI" and "Accessible from HMI" have been set.

Note

Accessing PLC data types

Access to PLC data types is only available in conjunction with SIMATIC S7-1200 or S7-1500.

Synchronization with PLC tags

A variety of options for synchronizing external tags with the PLC tags are available in the Runtime settings under "Settings for tags".

When you perform synchronization, you have the option of automatically applying the tag names of the PLC to external tags and reconnecting the existing tags.

The generated tag name is derived from the position of the data value in the hierarchical structure of the data block.

Update of tag values

For external tags, the current tag values are transmitted in Runtime via the communication connection between WinCC and the connected automation systems and then saved in the runtime memory. Next, the tag value will be updated to the set cycle time. For use in the runtime project, WinCC accesses tag values in the runtime memory that were read from the PLC at the previous cycle time. As a result, the value in the PLC can already change while the value from the runtime memory is being processed.

Note

PLC array elements in conjunction with S7-1200 or S7-1500

The index of the PLC array elements can begin with any number. In WinCC, indexing always starts with 0.

A PLC tag "Array [1..3] of Int", for example, is mapped to "Array [0..2] of Int" in WinCC.

When you access an array in a script, pay attention to the correct indexing.

See also

Basics of tags (Page 53)

Overview of HMI tag tables (Page 54)

Addressing external tags (Page 58)

Internal Tags (Page 60)

Communication between devices (Page 9)

1.4.1.4 Addressing external tags (Basic Panels)

Introduction

The options for addressing external tags depend on the type of connection between WinCC and the PLC in question. A distinction must be made between the following connection types:

- Integrated connection
  Connections of devices which are within a project and were created with the "Devices & Networks" editor are referred to as integrated connections.
- Non-integrated connection
  Connections of devices which were created with the "Connections" editor are referred to as non-integrated connections. It is not necessary that all of the devices be within a single project.

The connection type can also be recognized by its icon.

	Integrated connection
	Non-integrated connection

You can find additional information in the section "Basics of communication (Page 9)".

Addressing with integrated connections

An integrated connection offers the advantage that you can address a tag both symbolically and absolutely.

For symbolic addressing, you select the PLC tag via its name and connect it to the HMI tag. The valid data type for the HMI tag is automatically selected by the system. You have to distinguish between the following cases when you address elements in data blocks:

Symbolic addressing of data blocks with optimized access and standard access:

During the symbolic addressing of a data block with optimized access and standard access, the address of an element in the data block is dynamically assigned and is automatically adopted in the HMI tag in the event of a change. You do not need to compile the connected data block or the WinCC project for this step.
For data blocks with optimized access, only symbolic addressing is available.

For symbolic addressing of elements in a data block, you only need to recompile and reload the WinCC project in case of the following changes:

- If the name or the data type of the linked data block element or global PLC tag has changed.
- If the name or the data type of the higher level structure node of a linked element in the data block element or global PLC tag has changed.
- If the name of the connected data block has changed.

Symbolic addressing is currently available with the following PLCs:

- SIMATIC S7-1200
- SIMATIC S7-1500

- SIMATIC ET 200 CPU
- SIMATIC S7-1500 software controller

Symbolic addressing is also available if you have an integrated link.

You can also use absolute addressing with an integrated connection. You have to use absolute addressing for PLC tags from a SIMATIC S7-300/400 PLC. If you have connected an HMI tag with a PLC tag and the address of the PLC tag changes, you only have to recompile the control program to update the new address in WinCC. Then you recompile the WinCC project and load it onto the HMI device.

In WinCC, symbolic addressing is the default method. To change the default setting, select the menu command "Options > Settings". Select "Visualization > Tags" in the "Settings" dialog. If required, disable the "Symbolic access" option.

The availability of an integrated connection depends on the PLC used. The following table shows the availability:

PLC	Integrated connection	Comments
S7-300/400	Yes	The linking of tags is not checked in Runtime. If the tag address changes in the PLC and the HMI device is not compiled again and loaded, the change is not registered in runtime.
S7-1200	Yes	A validity check of the tag connection is performed in runtime during symbolic addressing. If an address is changed in the PLC, the change is registered and an error message is issued. In the case of absolute addressing, the behavior described for the S7-300/400 applies.
S7-1500	Yes	A validity check of the tag connection is performed in runtime during symbolic addressing. If an address is changed in the PLC, the change is registered and an error message is issued. In the case of absolute addressing, the behavior described for the S7-300/400 applies.
SIMATIC ET 200 CPU	Yes	A validity check of the tag connection is performed in runtime during symbolic addressing. If an address is changed in the PLC, the change is registered and an error message is issued. In the case of absolute addressing, the behavior described for the S7-300/400 applies.
SIMATIC S7-1500 software controller	Yes	A validity check of the tag connection is performed in runtime during symbolic addressing. If an address is changed in the PLC, the change is registered and an error message is issued. In the case of absolute addressing, the behavior described for the S7-300/400 applies.

Create an integrated connection in the "Devices & Networks" editor. If the PLC is contained in the project and integrated connections are supported, you can then also have the connection created automatically. To do this, when configuring the HMI tag, simply select an existing PLC tag to which you want to connect the HMI tag. The integrated connection is then automatically created by the system.

Addressing with non-integrated connections

In the case of a project with a non-integrated connection, you always configure a tag connection with absolute addressing. Select the valid data type yourself. If the address of a PLC tag changes in a project with a non-integrated connection during the course of the project, you also have to make the change in WinCC. The tag connection cannot be checked for validity in Runtime, an error message is not issued.

A non-integrated connection is available for all supported PLCs.

Symbolic addressing is not available in a non-integrated connection.

With a non-integrated connection, the control program does not need to be part of the WinCC project. You can perform the configuration of the PLC and the WinCC project independently of each other. For configuration in WinCC, only the addresses used in the PLC and their function have to be known.

See also

Basics of tags (Page 53)

Overview of HMI tag tables (Page 54)

External tags (Page 56)

Internal Tags (Page 60)

Basics of communication (Page 9)

1.4.1.5 Internal Tags (Basic Panels)

Introduction

Internal tags do not have any connection to the PLC. Internal tags transport values within the HMI device. The tag values are only available as long as runtime is running.

Principle

Internal tags are stored in the memory of the HMI device. Therefore, only this HMI device has read and write access to the internal tags. You can create internal tags to perform local calculations, for example.

You can use the HMI data types for internal tags. Availability depends on the HMI device being used.

The following HMI data types are available:

HMI data type	Data format
Array	One-dimensional array
Bool	Binary tag
DateTime	Date/time format
DInt	Signed 32-bit value
Int	Signed 16-bit value

HMI data type	Data format
LReal	Floating-point number 64-bit IEEE 754
Real	Floating-point number 32-bit IEEE 754
SInt	Signed 8-bit value
UDnt	Unsigned 32-bit value
UInt	Unsigned 16-bit value
USInt	Unsigned 8-bit value
WString	Text tag, 16-bit character set

See also

Basics of tags (Page 53)

Overview of HMI tag tables (Page 54)

External tags (Page 56)

Addressing external tags (Page 58)

1.4.2 Data exchange using area pointers (Basic Panels)

1.4.2.1 Basic information on area pointers (Basic Panels)

Introduction

You use an area pointer to access a data area in the PLC. During communication, the PLC and the HMI device alternately access these data areas for read and write operations. The PLC and the HMI device trigger defined interactions based on the evaluation of stored data.

For example, area pointers are required for the following data:

- Recipes
- Job mailboxes
- Sign-of-life monitoring

Area pointer

The following area pointers are supported:

Area pointer

Area pointers can be configured for connections.

- Data record
- Date/time

- Coordination
- Job mailbox

Global area pointers of the HMI device

Global area pointers can be configured to only one connection for each project.

- Screen number
- Date/time PLC
- Project ID

See also

Area pointers for connections (Page 62)

Use of area pointers (Page 64)

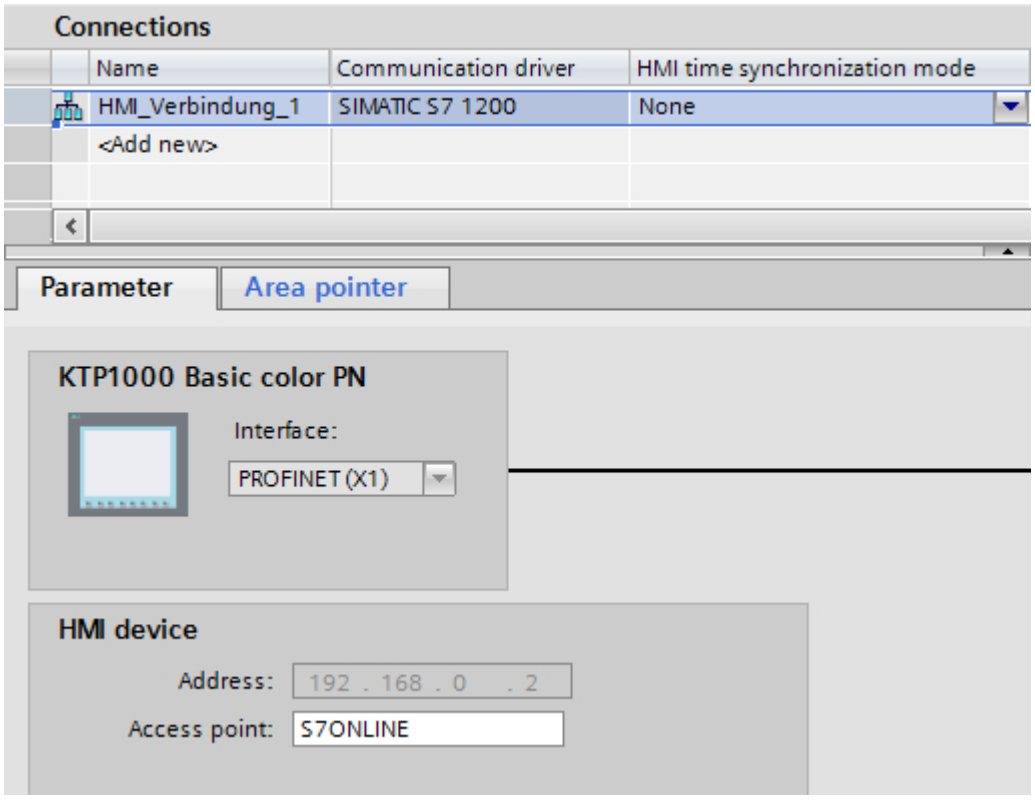
Accessing data areas (Page 66)

1.4.2.2 Area pointers for connections (Basic Panels)

Introduction

Using the "Area pointer" tab of the "Connections" editor, you can configure the usage of the available area pointers.

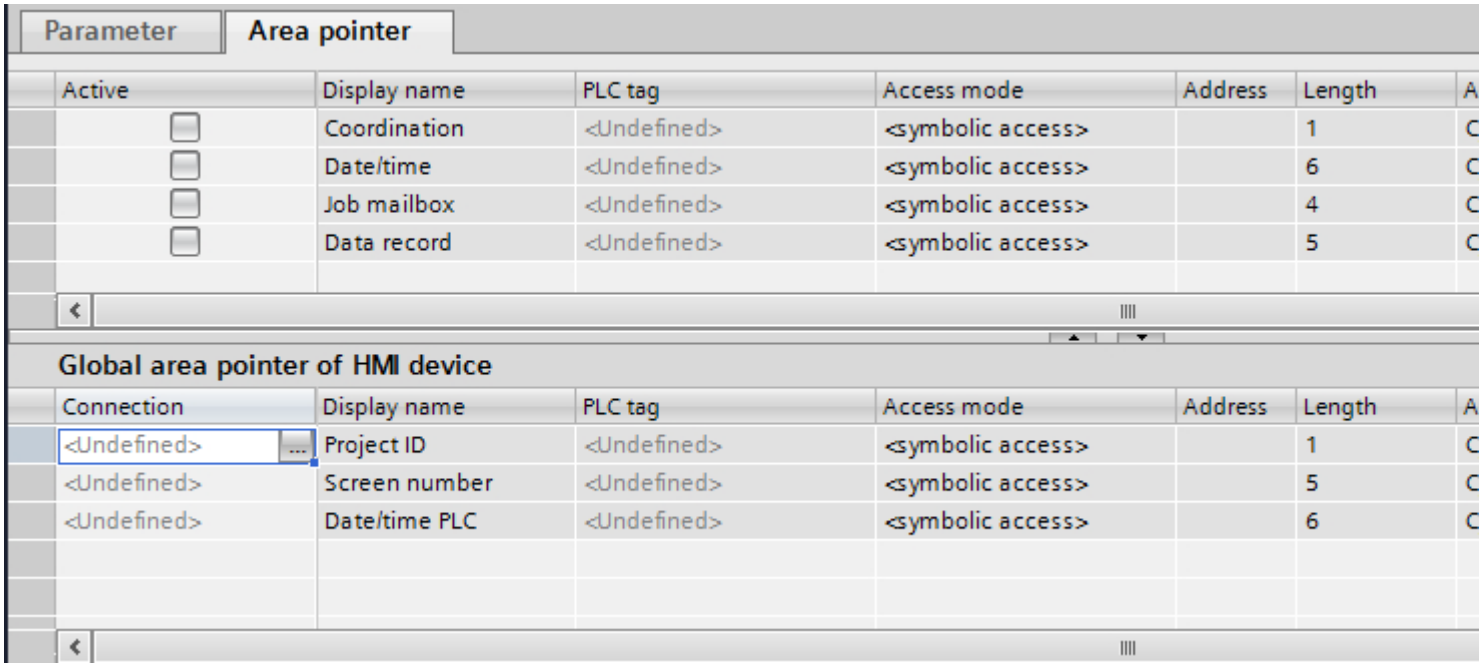
To configure the area pointers, open the "Connections" editor and open the "Area pointer" tab.



Structure

The "Area pointer" tab contains two tables of area pointers. The top part of the table contains the area pointers you can create and enable separately for each available connection.

The "Global area pointers of HMI device" table contains the area pointers which are created only once in the project and can be used for only one connection.



See also

Basic information on area pointers (Page 61)

Use of area pointers (Page 64)

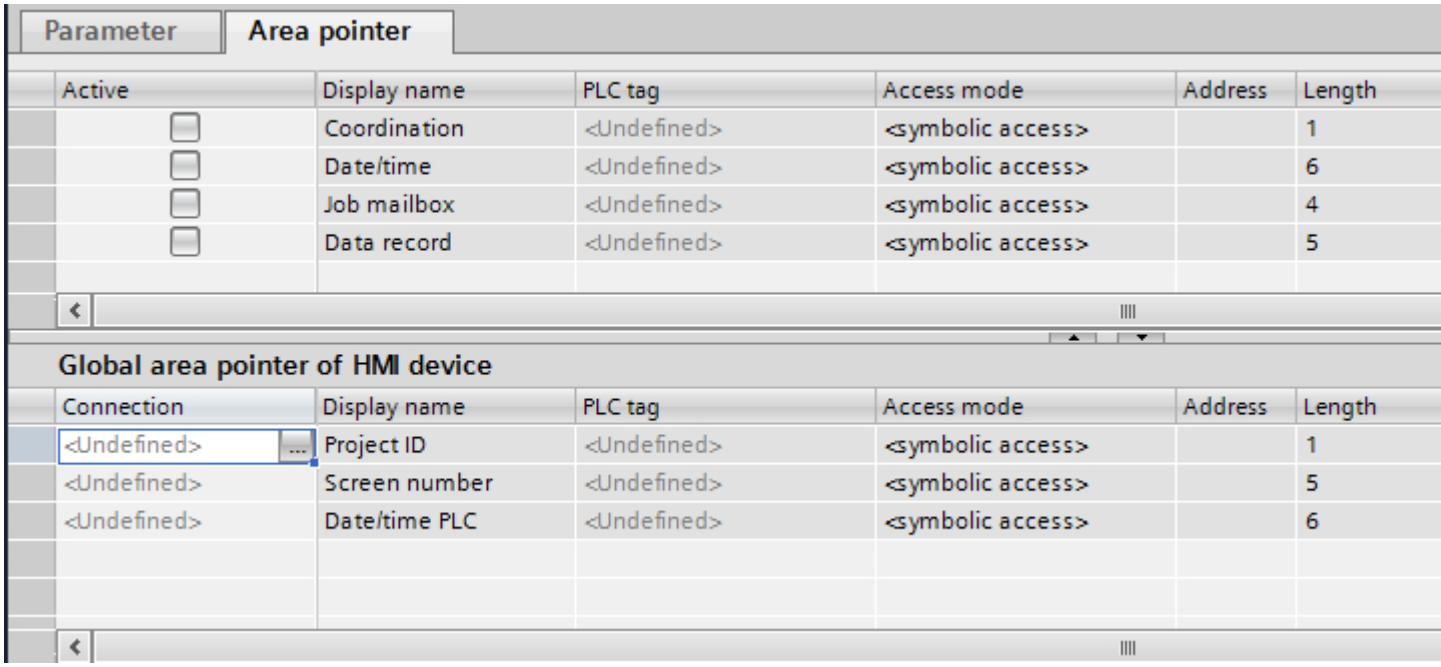
Accessing data areas (Page 66)

1.4.2.3 Use of area pointers (Basic Panels)

"Area pointer" tab

Before you use an area pointer, you enable it under "Connections > Area pointer". You then assign the area pointer parameters.

You assign the following parameters in the "Area pointer" tab:



- Active
  Enables the area pointer.
- Pointer name
  Name of the area pointer specified by WinCC.
- PLC tag
  Here you select the PLC tag or the tag array that you have configured as the data area for the area pointer.
- Access mode
  Here you can select from the following access modes:
  - Symbolic access
  - Absolute access
- Address
  If you selected "Symbolic access", no address is entered in this field.
  If you selected "Absolute access", enter the address of a tag in the "Address" field.
- Length
  WinCC specifies the length of the area pointer.
- Acquisition cycle
  You specify the acquisition cycle in this field for area pointers that are read by the HMI device. Note that a very short acquisition time may have a negative impact on HMI device performance.
- Comment
  Enter a comment, for example, to describe the purpose of the area pointer.

See also

Basic information on area pointers (Page 61)

Area pointers for connections (Page 62)

Accessing data areas (Page 66)

1.4.2.4 Accessing data areas (Basic Panels)

Accessing data areas

The following table shows how HMI devices and PLCs access individual data areas for read (R) or write (W) operations.

Data area	Required for	HMI device	PLC
Screen number	Evaluation by the PLC in order to determine the active screen.	W	R
Data record	Transfer of data records with synchronization	R/W	R/W
Date/time	Transfer of the date and time from the HMI device to the PLC	W	R
Date/time PLC	Transfer of the date and time from the PLC to the HMI device	R	W
Coordination	Requesting the HMI device status in the PLC program	W	R
Project ID	Runtime checks for consistency between the WinCC project ID and the project in the PLC	R	W
Job mailbox	Triggering of HMI device functions by the PLC program	R/W	R/W

See also

Basic information on area pointers (Page 61)

Area pointers for connections (Page 62)

Use of area pointers (Page 64)

1.4.2.5 Configuring area pointers (Basic Panels)

Configuration of area pointers (Basic Panels)

Introduction

You use an area pointer to access a data area in the PLC. The data area is stored in the PLC.

Prior to configuring area pointers

Before you use the area pointer, you must enable and parameterize it under "Connections > Area Pointer".

Global data block

To access the data area in the PLC, you have to create a global data block in the PLC program. The following example shows the use of a data block.

Length of area pointers

For area pointers with a length >= 1, you set up the data area as a tag array in a global data block or instance data block.

You also have the option to use a PLC tag for area pointers with a length = 1.

The configuration of the tags in a data block is dependent on the length of the area pointer you want to use. The unit for the length of an area pointer is a 16-bit word.

If, for example, you want to use an area pointer with a length of "5", you must create an array with 5 array elements of the data type UINT in the data block.

Alternative procedure

Alternatively, you can also use the absolute access mode to access area pointers. Absolute access mode only works on standard PLC data blocks.

See also

Basic information on area pointers (Page 61)

Parameterizing a global data block (Page 67)

Configuring an area pointer for a connection (Page 69)

General information on area pointers (Page 303)

Parameterizing a global data block (Basic Panels)

Introduction

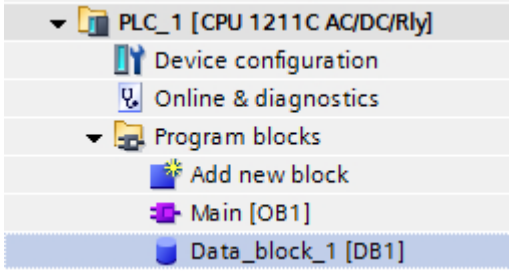
To access the data area in the PLC, a global data block for the area pointer must be parameterized in the PLC program.

Requirements

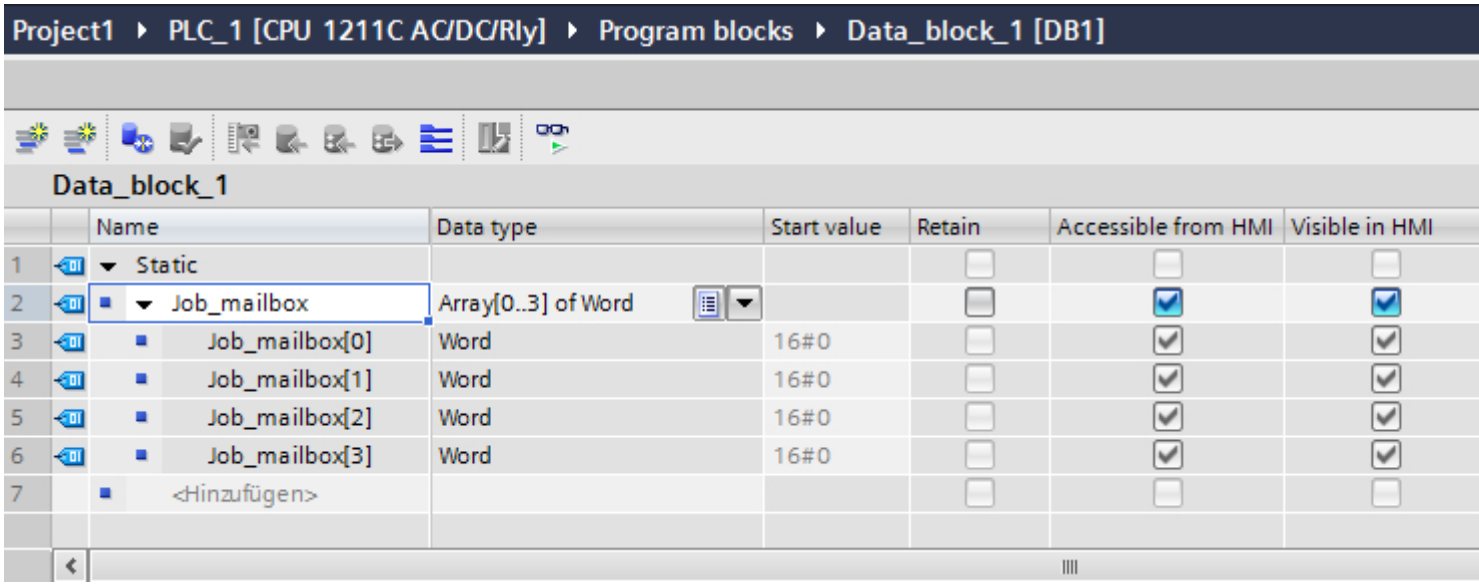
- A PLC is created in the project.
- A connection is configured between the PLC and the HMI device.
- The PLC program contains a global data block.

Procedure

1. Open "PLC > Program blocks" in the project tree.
2. Double-click the global data block you created previously.
   The data block opens.



3. Enter a tag name in the "Name" column.
4. Select "Array[lo .. hi] of type" as the data type in the "Data type" column.
5. Replace the "lo" entry by the low value for the dimension of the array.
6. Replace the "hi" entry by the high value for the dimension of the array.
   Example: If you configure an area pointer with the length "4", enter the value "0" for "lo" and the value "3" for "hi" inside the brackets.
7. Replace the "type" designation with the "word" data type.
   The full data type for an array of 4 tags appears as follows: "Array[0 .. 3] of word".
   The tag array is created after the entry is confirmed.
8. Click "Compile".
   The project is compiled.



See also

Basic information on area pointers (Page 61)

Configuration of area pointers (Page 66)

Configuring an area pointer for a connection (Page 69)

Configuring an area pointer for a connection (Basic Panels)

Introduction

After you have parameterized the global data block, you now create the area pointer for the connection.

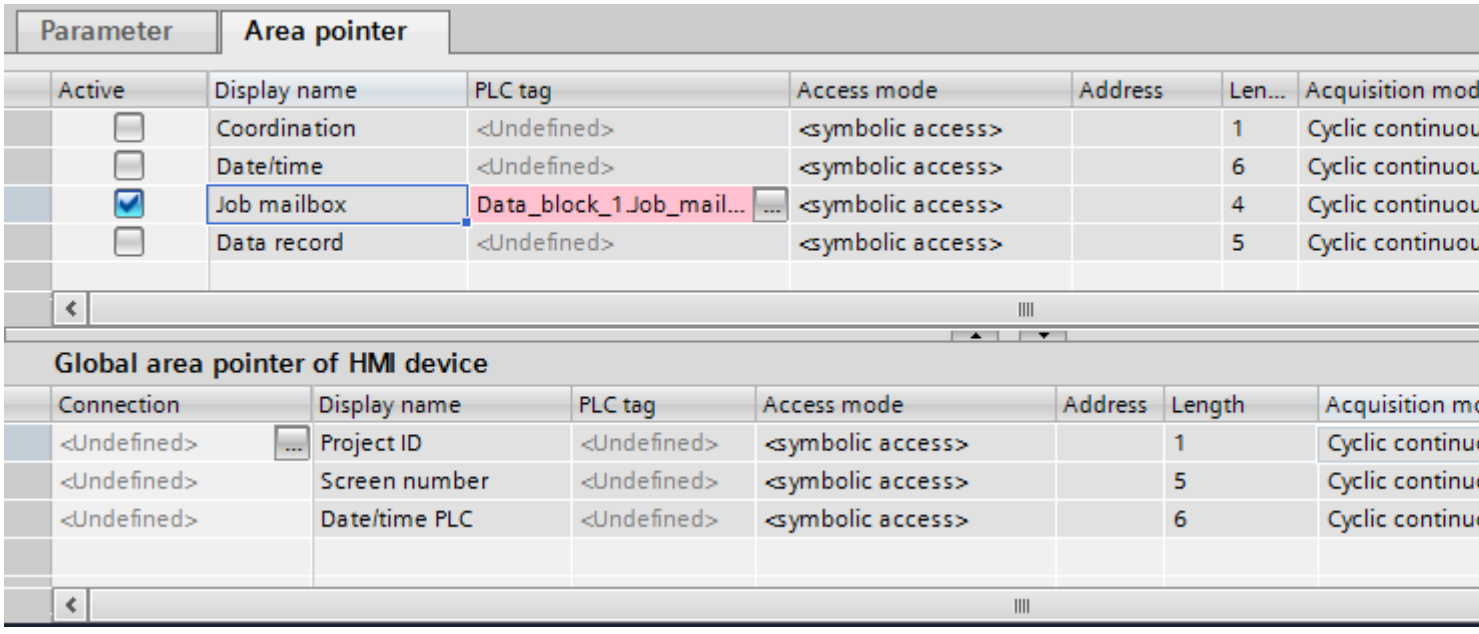
Requirements

- The global data block has been parameterized in the PLC program.

Procedure

1. Open "HMI >Connections" in the project tree.
2. Click the "Area pointer" tab.
3. Enable the required area pointer.
   You enable a global area pointer by selecting the connection in the "Connection" field.
4. Click the navigation button in the "PLC tag" field.
   The object list opens.

5. Navigate to the data block in the object list, and select the tag in the right window.
   You do not need an array tag to configure an area pointer with the length of "1".



6. Select the "Word" data type when creating the tag in the data block.

If required, set additional parameters, such as the acquisition cycle, during configuration.

Result

The area pointer is enabled and connected to the PLC tag in the global data block.

See also

Basic information on area pointers (Page 61)
Configuration of area pointers (Page 66)
Parameterizing a global data block (Page 67)

1.5 Device dependency (Basic Panels)

1.5.1 Basic Panel (Basic Panels)

1.5.1.1 Communication drivers (Basic Panels)

SIMATIC communication drivers (Basic Panels)

Device dependency of the Basic Panels

You configure different communications drivers depending on the communication. The following tables show the communication drivers released for integrated and non-integrated communication.

There are different versions of the HMI devices within the integrated communication.

Communication drivers for integrated communication (V11.0)

HMI devices	SIMATIC S7-1200 (V1)	SIMATIC S7-1200 (V2)	SIMATIC S7-1200 (V2.2)	SIMATIC S7-1200 (V3)	SIMATIC S7-1200 (V4)	SIMATIC S7-1500	SIMATIC S7-300/400	SIMATIC ET 200 CPU	SIMATIC S7-1500 software controller
KP300 Basic	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No
KP400 Basic	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No
KTP400 Basic PN	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No
KTP600 Basic DP	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No
KTP600 Basic PN	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No
KTP1000 Basic DP	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No
KTP1000 Basic PN	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No
TP1500 Basic PN	Yes	Yes	Yes	Yes	Yes	No	Yes	No	No

Communication drivers for integrated communication (V12.0)

HMI devices	SIMATIC S7-1200 (V1)	SIMATIC S7-1200 (V2)	SIMATIC S7-1200 (V2.2)	SIMATIC S7-1200 (V3)	SIMATIC S7-1200 (V4)	SIMATIC S7-1500	SIMATIC S7-300/400	SIMATIC ET 200 CPU	SIMATIC S7-1500 software controller
KP300 Basic	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
KP400 Basic	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
KTP400 Basic PN	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
KTP600 Basic DP	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
KTP600 Basic PN	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
KTP1000 Basic DP	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
KTP1000 Basic PN	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
TP1500 Basic PN	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No

Communication drivers for integrated communication (V13.0)

HMI devices	SIMATIC S7-1200 (V1)	SIMATIC S7-1200 (V2)	SIMATIC S7-1200 (V2.2)	SIMATIC S7-1200 (V3)	SIMATIC S7-1200 (V4)	SIMATIC S7-1500	SIMATIC S7-300/400	SIMATIC ET 200 CPU	SIMATIC S7-1500 software controller
KTP400 Basic	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
KTP700 Basic	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
KTP900 Basic	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
KTP1200 Basic	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No

Communication drivers for integrated communication (V13.0.1)

HMI devices	SIMATIC S7-1200 (V1)	SIMATIC S7-1200 (V2)	SIMATIC S7-1200 (V2.2)	SIMATIC S7-1200 (V3)	SIMATIC S7-1200 (V4)	SIMATIC S7-1500	SIMATIC S7-300/400	SIMATIC ET 200 CPU	SIMATIC S7-1500 software controller
KTP400 Basic	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KTP700 Basic	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

HMI devices	SIMATIC S7-1200 (V1)	SIMATIC S7-1200 (V2)	SIMATIC S7-1200 (V2.2)	SIMATIC S7-1200 (V3)	SIMATIC S7-1200 (V4)	SIMATIC S7-1500	SIMATIC S7-300/400	SIMATIC ET 200 CPU	SIMATIC S7-1500 software controller
KTP900 Basic	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KTP1200 Basic	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Communication drivers for integrated communication (V14.0)

HMI devices	SIMATIC S7-1200 (V1)	SIMATIC S7-1200 (V2)	SIMATIC S7-1200 (V2.2)	SIMATIC S7-1200 (V3)	SIMATIC S7-1200 (V4)	SIMATIC S7-1500	SIMATIC S7-300/400	SIMATIC ET 200 CPU	SIMATIC S7-1500 software controller
KTP400 Basic	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KTP700 Basic	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KTP900 Basic	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KTP1200 Basic	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Communication drivers for non-integrated communication

HMI devices	SIMATIC S7-1200	SIMATIC S7-1500	SIMATIC S7-300/400	SIMATIC S7-200	SIMATIC LOGO!	SIMATIC HTTP protocol	SIMATIC ET 200 CPU	SIMATIC S7-1500 software controller
KP300 Basic	Yes	Yes	Yes	Yes	Yes	No	Yes	No
KP400 Basic	Yes	Yes	Yes	Yes	Yes	No	Yes	No
KTP400 Basic PN	Yes	Yes	Yes	Yes	Yes	No	Yes	No
KTP600 Basic DP	Yes	Yes	Yes	Yes	Yes	No	Yes	No
KTP600 Basic PN	Yes	Yes	Yes	Yes	Yes	No	Yes	No
KTP700 Basic PN	Yes	Yes	Yes	Yes	Yes	No	Yes	No
KTP900 Basic PN	Yes	Yes	Yes	Yes	Yes	No	Yes	No
KTP1000 Basic DP	Yes	Yes	Yes	Yes	Yes	No	Yes	No
KTP1000 Basic PN	Yes	Yes	Yes	Yes	Yes	No	Yes	No
TP1500 Basic PN	Yes	Yes	Yes	Yes	Yes	No	Yes	No

See also

Other communication drivers (Page 74)

Other communication drivers (Basic Panels)

Device dependency of the Basic Panels

The following table shows which communication drivers you can configure with the various Basic Panels.

Communication drivers

HMI devices	OPC	SIMATIC HTTP protocol	Allen-Bradley EtherNet/IP	Allen-Bradley DF1	Mitsubishi MC TCP/IP	Mitsubishi FX	Modicon Modbus TCP/IP	Modicon Modbus RTU	Omron Host Link
KP300 Basic	No	No	Yes	No	Yes	No	Yes	No	No
KP400 Basic	No	No	Yes	No	Yes	No	Yes	No	No
KTP400 Basic PN	No	No	Yes	No	Yes	No	Yes	No	No
KTP600 Basic DP	No	No	No	Yes [2)]	No	Yes	No	Yes [1)]	Yes
KTP600 Basic PN	No	No	Yes	No	Yes	No	Yes	No	No
KTP700 Basic PN	No	No	Yes	No	Yes	No	Yes	No	No
KTP700 Basic DP	No	No	No	Yes [2)]	No	Yes	No	Yes [1)]	Yes
KTP900 Basic PN	No	No	Yes	No	Yes	No	Yes	No	No
KTP1000 Basic DP	No	No	No	Yes [2)]	No	Yes	No	Yes [1)]	Yes
KTP1000 Basic PN	No	No	Yes	No	Yes	No	Yes	No	No
KTP1200 Basic PN	No	No	Yes	No	Yes	No	Yes	No	No
KTP1200 Basic DP	No	No	No	Yes [2)]	No	Yes	No	Yes [1)]	Yes
TP1500 Basic PN	No	No	Yes	No	Yes	No	Yes	No	No

[1] only with RS 422-RS232 converter
Order number of the converter: 6AV6 671-8XE00-0AX0

[2] Direct communication with PLC 5 or KF2 module, otherwise only approved with RS422-RS232 converter (option).
Order number of the converter: 6AV6 671-8XE00-0AX0

See also

SIMATIC communication drivers (Page 71)

1.5.1.2 Interfaces of the Basic Panels (Basic Panels)

Device dependency of the Basic Panels

The following table shows which HMI device interfaces are available for the communication driver protocols.

Table 1-1 Basic Panels

	KP300 Basic **KP400 Basic** **KTP400 Basic PN** **KTP600 Basic PN** **KTP700 Basic PN** **KTP900 Basic PN** **KTP1200 Basic PN** **KTP1000 Basic PN** **TP1500 Basic PN**	**KTP600 Basic DP** **KTP700 Basic DP** **KTP1000 Basic DP** **KTP1200 Basic DP**
SIMATIC S7 - PPI [1]	—	MPI/DP (X2)
SIMATIC S7 - MPI	—	MPI/DP (X2)
SIMATIC S7 - PROFIBUS	—	MPI/DP (X2)
SIMATIC S7 - PROFINET	PROFINET (X1)	—
SIMATIC HMI HTTP protocol	—	—
OPC	—	—
Allen-Bradley EtherNet/IP	PROFINET (X1)	—
Allen-Bradley DF1	—	MPI/DP (X2) [2]
Mitsubishi TCP/IP	PROFINET (X1)	—
Mitsubishi FX	—	MPI/DP (X2) (RS422)
Modicon Modbus TCP	PROFINET (X1)	—
Modicon Modbus RTU	—	MPI/DP (X2) [3]
Omron Host Link	—	MPI/DP (X2) (RS422)

[1] For SIMATIC S7-200 only

[2] Direct communication with PLC5 or KF2 module, otherwise only approved with RS422-RS232 converter (option).
Order number: 6AV6 671-8XE00-0AX0

[3] only approved with RS 422-RS232 converter
Order number: 6AV6 671-8XE00-0AX0

See also

Area pointers for Basic Panels (Page 76)

1.5.1.3 Area pointers for Basic Panels (Basic Panels)

Introduction

Area pointers are parameter fields from which the HMI device obtains information about the location and size of data areas in the PLC. During communication, the PLC and the HMI device alternately access these data areas for read and write operations. Based on the evaluation of data stored in the data areas, the PLC and HMI device initiate mutually defined actions.

WinCC uses the following area pointers:

- Job mailbox
- Project ID
- Screen number
- Data record
- Date/time
- Date/time PLC
- Coordination

Availability of the area pointers

The following table shows the availability of the area pointers on the HMI devices. Note that the area pointers can be used only for available communication drivers.

Area pointer

	KP300 Basic	KTP400 Basic PN	KTP600 Basic PN	KTP600 Basic DP	KTP700 Basic PN / DP	KTP900 Basic PN	KTP1000 Basic PN / DP	KTP1200 Basic PN / DP	TP1500 Basic PN
Screen number	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Data record	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

	KP300 Basic	KTP400 Basic PN	KTP600 Basic PN	KTP600 Basic DP	KTP700 Basic PN / DP	KTP900 Basic PN	KTP1000 Basic PN / DP	KTP1200 Basic PN / DP	TP1500 Basic PN
Date/time	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Date/time PLC	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Coordination	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Project ID	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Job mailbox	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

See also

Interfaces of the Basic Panels (Page 75)

1.6 Communicating with SIMATIC S7 1500 (Basic Panels)

1.6.1 Communication with SIMATIC S7-1500 (Basic Panels)

Introduction

This section describes communication between the HMI device and the SIMATIC S7-1500 controller.

You can configure the following communication channels for the SIMATIC S7-1500 controller:

- PROFINET
- PROFIBUS

HMI connection for communication

Connections between the HMI device and SIMATIC S7-1500 are configured in the "Devices & Networks" editor.

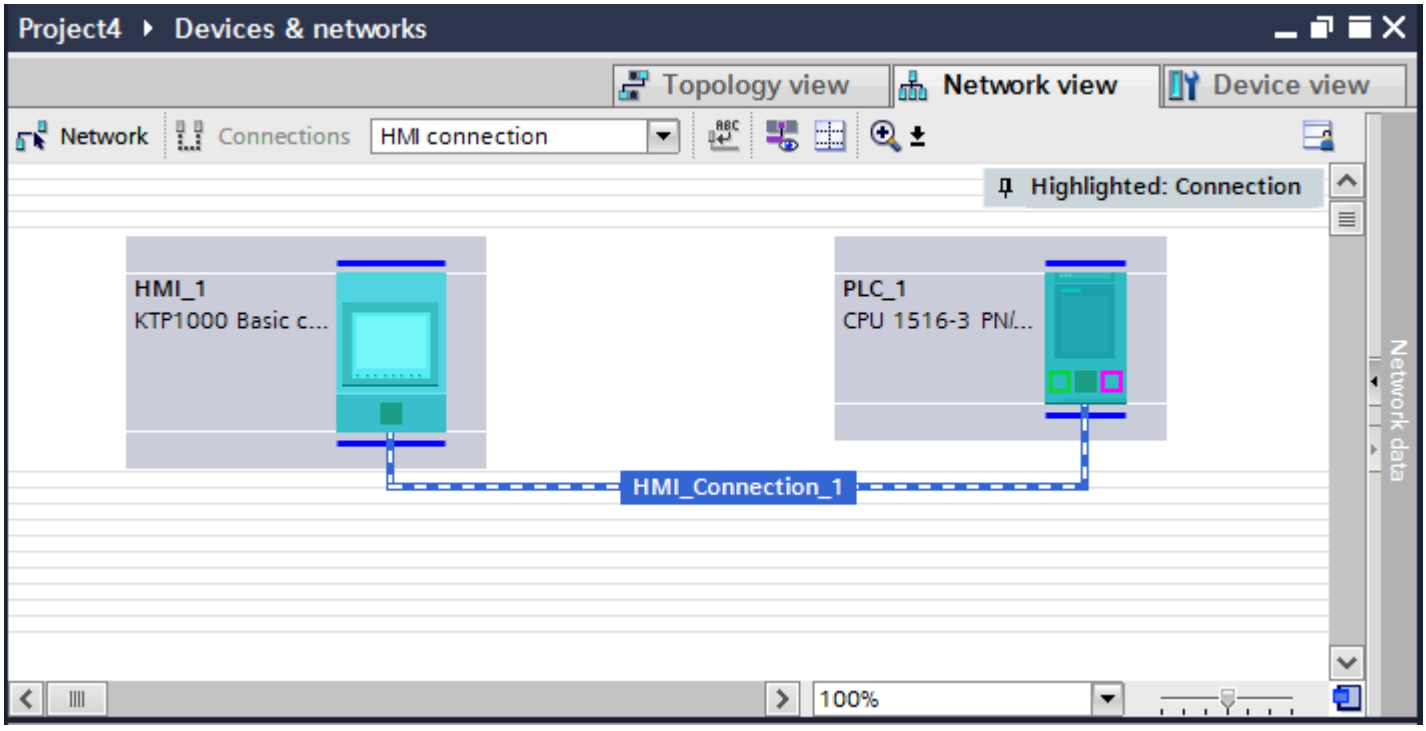
1.6.2 Communication via PROFINET (Basic Panels)

1.6.2.1 Configuring an HMI connection (Basic Panels)

Communication via PROFINET (Basic Panels)

HMI connections via PROFINET

If you have inserted an HMI device and a SIMATIC S7 1500 into the project, interconnect the two PROFINET interfaces in the "Devices & Networks" editor.



You can also connect multiple HMI devices to a single SIMATIC S7 1500 and multiple SIMATIC S7 1500 to a single HMI device.

The maximum number of communication partners that you can connect to an HMI device is dependent on the HMI device used.

Additional information is available in the documentation for the respective HMI device.

HMI connection in the "Devices & Networks" editor

You configure the HMI connection between the PLC and the HMI device via PROFINET in the "Devices & Networks" editor.

Connection in the "Connections" editor

Alternatively, you configure the connection between the PLC and HMI device via PROFINET in the "Connections" editor of the HMI device.

See also

Communication with SIMATIC S7-1500 (Page 77)

Configuring an HMI connection via PROFINET (Page 79)

Configuring an HMI connection via PROFINET (Basic Panels)

Introduction

You configure an HMI connection between HMI devices and a SIMATIC S7 1500 via PROFINET or Ethernet in the "Devices & Networks" editor.

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

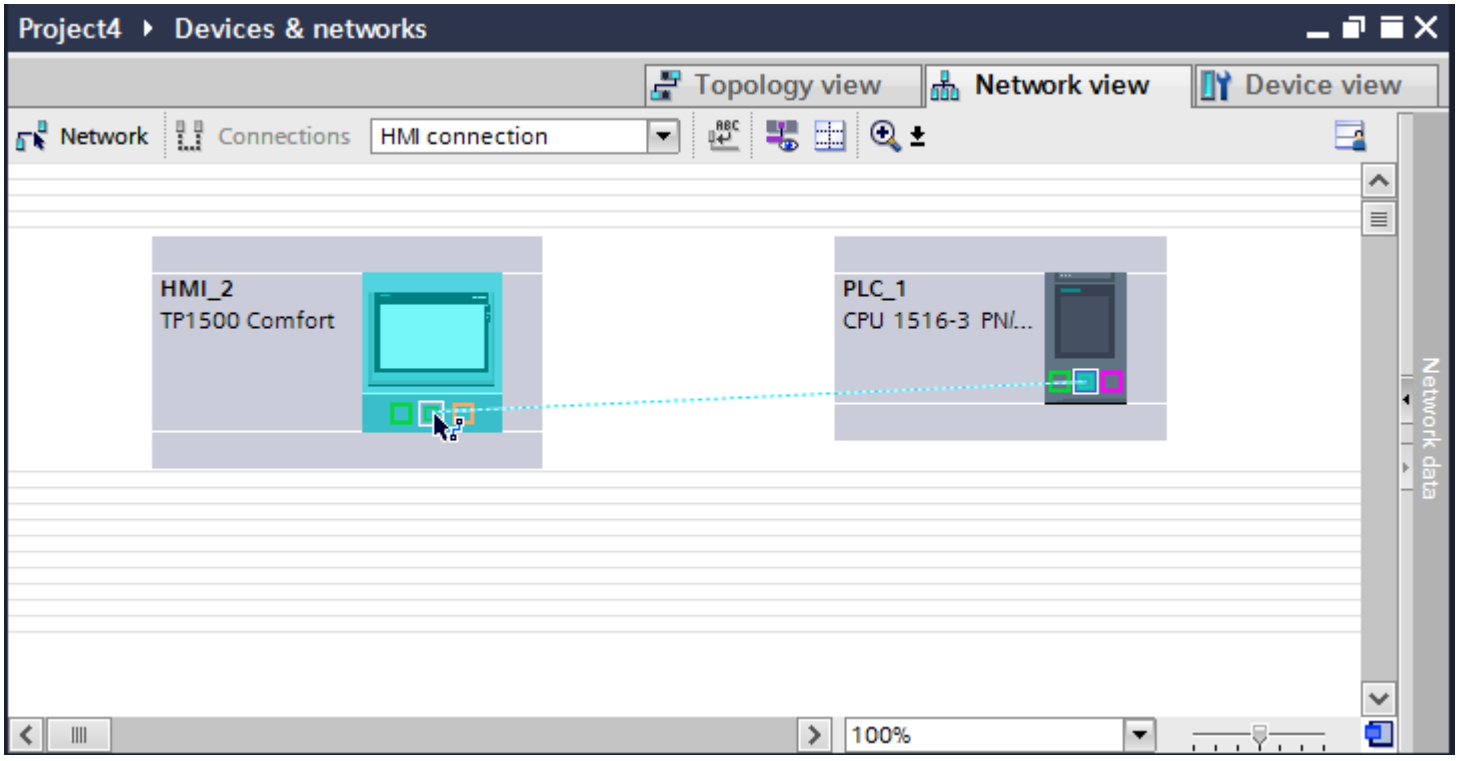
The following communication partners are created in the "Devices & Networks" editor:

- HMI device with PROFINET or Ethernet interface
- SIMATIC S7 1500 with PROFINET interface

Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed graphically in the network view.
2. Click the "Connections" button and select "HMI connection" for the connection type.
   The devices available for connection are highlighted in color.

3. Click the PROFINET interface of the PLC and use a drag-and-drop operation to draw a connection to the PROFINET or Ethernet interface of the HMI device.



4. Click the connecting line.
5. Click "Highlight HMI connection" and select the HMI connection.
   The connection is displayed graphically in the Inspector window.
6. Click the communication partners in the "Network view" and change the PROFINET parameters in the Inspector window according to the requirements of your project.
   For additional information on this, see section "PROFINET parameters (Page 81)".

Note

The created HMI connection is also shown in the tabular area of the editor on 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 a SIMATIC S7 1500. The IP address and subnet mask connection parameters are configured.

See also

Communication with SIMATIC S7-1500 (Page 77)

Communication via PROFINET (Page 78)

PROFINET parameters (Page 81)

1.6.2.2 PROFINET parameters (Basic Panels)

PROFINET parameters for the HMI connection (Basic Panels)

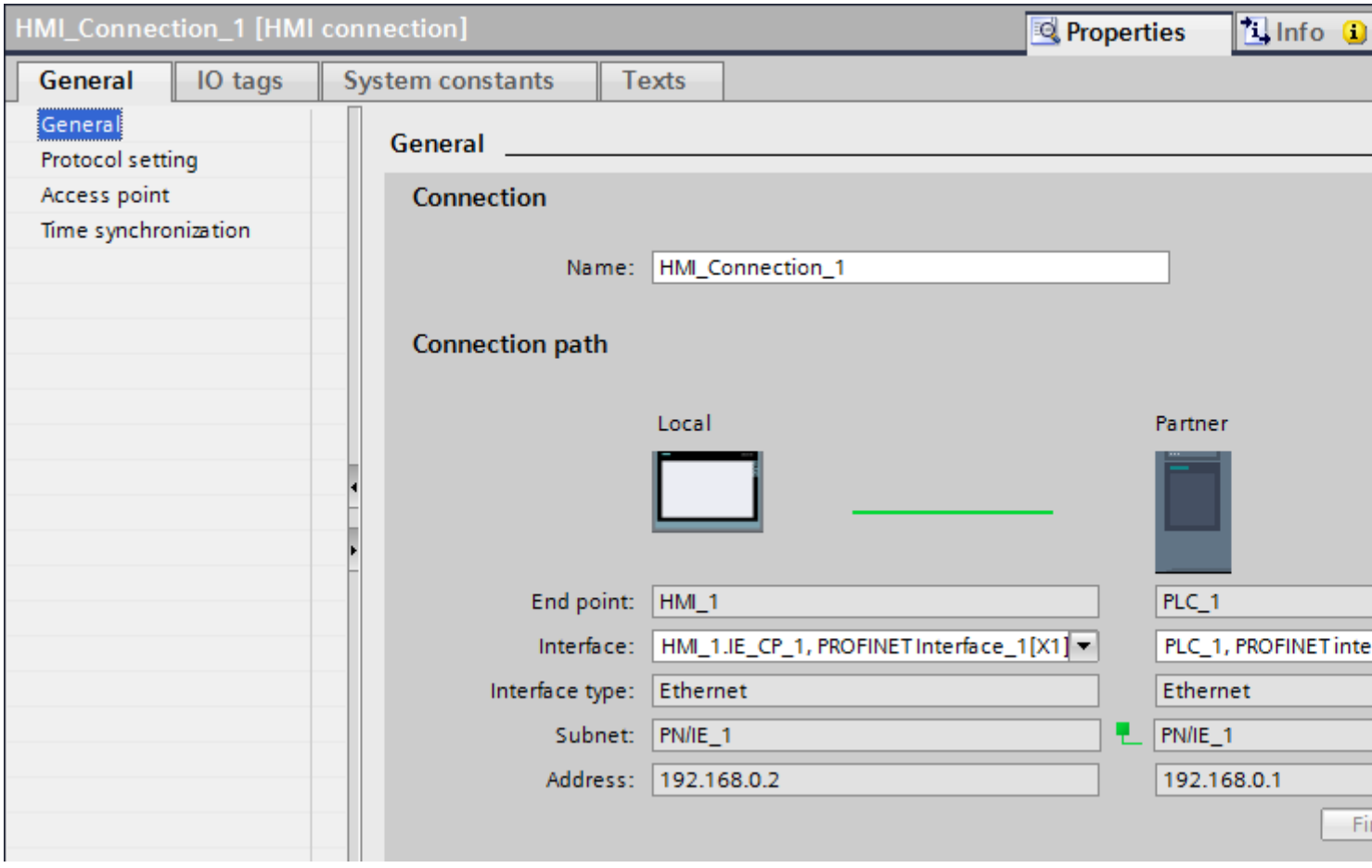
PROFINET parameters for the HMI connection

An overview of the configured HMI connection parameters can be found in the properties for the HMI connection.

Only limited changes are possible in this Inspector window.

Displaying and changing the HMI connection parameters

1. Click the HMI connection in the "Devices & Networks" editor.
2. Change the parameters of the HMI connection in the Inspector window under "Properties > General > General".



"Connection"

Shows the name of the HMI connection.

"Connection path"

The communication partners of the selected HMI connection and the associated PROFINET parameters are displayed in the "Connection path" area. Some of the areas displayed cannot be edited in this dialog.

- "End point"
  Displays the device name. This area cannot be edited.
- "Interface"
  Displays the selected interface of the device. You can choose between several interfaces, depending on the device.

- "Interface type"
  Displays the selected interface type. This area cannot be edited.
- "Subnet"
  Displays the selected subnet. This area cannot be edited.
- "Address"
  Displays the selected IP address of the device. This area cannot be edited.
- "Find connection path" button
  Enables the subsequent specification of connections.

See also

Communication with SIMATIC S7-1500 (Page 77)

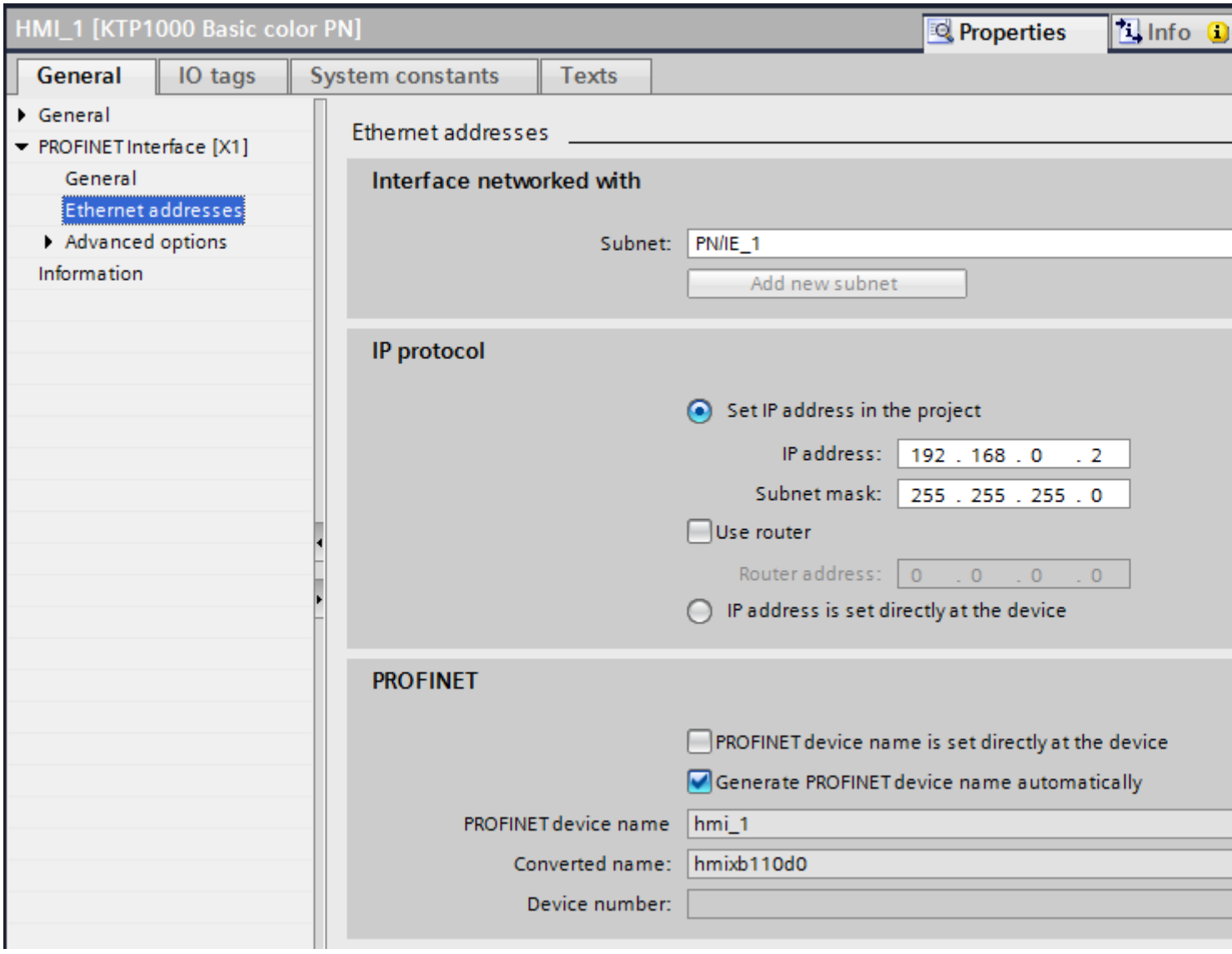
PROFINET parameters for the HMI device (Basic Panels)

PROFINET parameters for the HMI device

An overview of the configured HMI device parameters can be found in the properties for the HMI device.

Displaying and changing PROFINET parameters of the HMI device

1. Click the HMI device in the "Devices & Networks" editor.
2. Change the parameters of the HMI device in the Inspector window under "Properties > General".



"Interface networked with"

In the "Interface networked with" area, select the subnet of the HMI connection via which the HMI device is connected to the network. You use the "Add new subnet" button to create a new subnet.

"IP protocol"

- "Set IP address in the project"
  When you transfer the WinCC project to the HMI device, this IP address is set up directly in the HMI device.

  Note

  The device is automatically restarted in the case of HMI devices with the Windows CE 3.0 operating system.

  HMI devices with Windows CE 3.0:
  - OP 77B
  - TP 177B color PN/DP
  - TP 177B mono DP
  - OP 177B color PN/DP
  - OP 177B mono DP
  - Mobile Panel 177 PN
  - Mobile Panel 177 DP
  - TP 277 6"
  - OP 277 6"

- "Subnet mask"
  You assign data of the subnet mask in the "Subnet mask" area.
- "Use IP router"
  If you are using an IP router, select "Use IP router" and enter the router address in the "Router address" field.
- "Set IP address using a different method"
  If the function "Set IP address using a different method" is activated, the IP address is not taken from the project. You have to enter the IP address directly in the Control Panel of the HMI device.

See also

Communication with SIMATIC S7-1500 (Page 77)

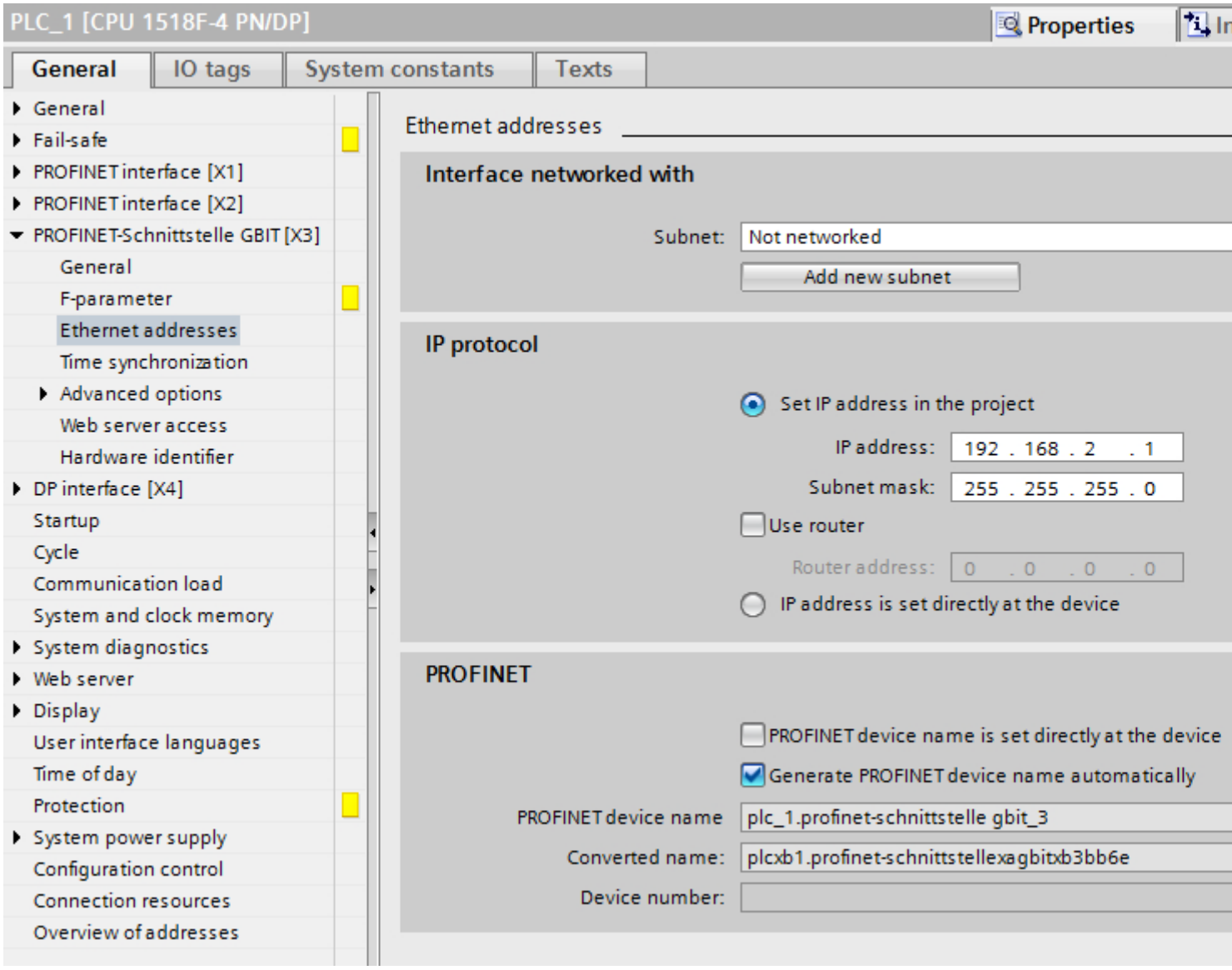
PROFINET parameters for the PLC (Basic Panels)

PROFINET parameters for the PLC

An overview of the configured parameters can be found in the properties for the PLC.

Displaying and changing PROFINET parameters of the PLC

1. Click the PLC in the "Devices & Networks" editor.
2. Change the parameters of the PLC in the Inspector window under "Properties > General > General".



"Interface networked with"

In the "Subnet" area, select the subnet of the HMI connection via which the PLC is connected to the network. You use the "Add new subnet" button to create a new subnet.

"IP protocol"

- "Interface type"
  Depending on the HMI device type, you have various interfaces to choose from.
- "IP address"
  You assign the IP address of the HMI device in the "IP address" area.
- "Subnet mask"
  You assign data of the subnet mask in the "Subnet mask" area.
  If you are using an IP router, select "Use IP router" and enter the router address in the field.

See also

Communication with SIMATIC S7-1500 (Page 77)

Protection of communication (Basic Panels)

Security levels (Basic Panels)

If you want to protect the controller and HMI device communication, you can assign protection levels for the communication.

For a SIMATIC S7-1500 CPU, you can enter multiple passwords and thereby set up different access rights for various user groups.

The passwords are entered in a table, so that exactly one protection level is assigned to each password.

The effect of the password is given in the "Protection" column.

For the SIMATIC S7-1500 controller, several aspects need to be considered when setting protection levels.

For additional information on this, see section Setting options for the protection (Page 88)

Example

When configuring the controller, you select the "Complete protection" protection level for a standard CPU (i.e., not an F-CPU).

Afterwards, you enter a separate password for every protection level above it in the table.

For users who do not know any of the passwords, the CPU is completely protected. Not even HMI access is possible.

For users who know one of the assigned passwords, the effect depends on the table row in which the password occurs:

- The password in row 1 (no protection) allows access as if the CPU were completely unprotected. Users who know this password have unrestricted access to the CPU.
- The password in row 2 (write protection) allows access as if the CPU were write-protected. Despite knowing the password, users who know this password only have read access to the CPU.
- The password in row 3 (read and write protection) allows access as if the CPU were read-protected and write-protected, so that only HMI access is possible for users who know this password.

See also

Setting options for the protection (Page 88)

Communication with SIMATIC S7-1500 (Page 77)

Access password for the HMI connection (Page 92)

Setting options for the protection (Basic Panels)

Access levels

The following section describes how to use the various access levels of the S7-1500 CPUs.

S7-1500 CPUs provide various access levels to limit the access to specific functions.

The individual access levels as well as their associated passwords are specified in the object properties of the CPU. You assign parameters for the access level in a table.

Access level	Access			Access permission
	HMI	Read	Write	Password
Full access (no protection)	✓	✓	✓	
Read access	✓	✓		
HMI access	✓			
No access (complete protection)				

The green checkmarks in the columns to the right of the respective access level specify which operations are possible without knowing the password of this access level.

If you want to use the functions of fields that are not selected in the "Access" column, a password has to be entered:

Example:

You set the access level "Read access". You can see from the table that write access is not possible during operation without entering a password.

The table also shows that full access is necessary for the "Write" function.

To use a function requiring write access during operation, the password for full access must therefore be entered.

NOTICE
Configuring an access level does not replace know-how protection Configuring access levels offers a high degree of protection against unauthorized changes to the CPU by restricting download privileges to the CPU. However, blocks on the memory card are not write- or read-protected. Use know-how protection to protect the code of blocks on the memory card.

Default characteristics

The default access level is "Full access (no protection)". Every user can read and change the hardware configuration and the blocks. A password is not set and is also not required for online access.

The access levels in detail

Below you can find an explanation of the existing access levels and which functions are possible at the respective access level.

- Full access (no protection):
  The hardware configuration and the blocks can be read and changed by all users.
- Read access for F-blocks (F-CPUs only):
  F-blocks of the safety program cannot be changed without authorization by the password of this access level or a higher level.
  Further information is available in the programming and operating manual *SIMATIC Safety - Configuring and Programming.*
- 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, as is changing the operating state (RUN/STOP).
  You cannot load blocks or a hardware configuration into the CPU. Moreover, no writing test functions and firmware updates are possible.

- HMI access:
  Only HMI access and access to diagnostics data is possible. Tags can be read and written via a HMI device.
  At this access level, you can neither load blocks and the hardware configuration into the CPU nor load blocks and the hardware configuration from the CPU into the programming device.
  In addition, the following is **not** possible: Writing test functions, changing the operating state (RUN/STOP) and firmware updates.
- No access (complete protection):
  Only identification data can be read, via "Accessible devices", for example.
  Neither read nor 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).
  Legitimation with a configured password provides you with access in accordance with the associated protection level.

Behavior of functions at different access levels

The table below describes which online functions are possible in the various protection levels.

Function	Full access	Read access	HMI access	No access
Identification of the device, via "Accessible devices", for example	Yes	Yes	Yes	Yes
HMI diagnostics view	Yes	Yes	Yes	No
Monitoring tags (M, I, Q, DB content) via HMI device	Yes	Yes	Yes	No
Modifying tags (M, I, Q, DB content) via HMI device	Yes	Yes	Yes	No
Diagnostics display (for example, device information, connection display, alarm display, diagnostic buffer)	Yes	Yes	Yes	No
Reading cycle time statistics (Online & Diagnostics)	Yes	Yes	Yes	No
Reading information from the hardware configuration (Online & Diagnostics)	Yes	Yes	Yes	No
Reading time-of-day	Yes	Yes	Yes	No
Executing online functions within the hardware configuration (Online & Diagnostics)	Yes	Yes	Yes	No
Acknowledging alarms	Yes	Yes	Yes	No
Receiving alarms	Yes	Yes	Yes	No
Enabling/disabling alarms	Yes	Yes	No	No
Reading tags via test function (STEP 7, tag table or watch table)	Yes	Yes	No	No

Requesting operating state change online (RUN/STOP/ warm restart)	Yes	Yes	No	No
Downloading data blocks, code blocks, hardware configuration to PG/PC	Yes	Yes	No	No
Set time-of-day	Yes	Yes	No	No
Deleting data blocks, code blocks, hardware configuration in the CPU	Yes	No	No	No
Downloading individual data blocks, code blocks, hardware configurations to the CPU	Yes	No	No	No
Loading PLC program to the device and resetting	Yes	No	No	No
Firmware update of CPUs or I/O modules	Yes	No	No	No
Modifying tags via test function (STEP 7, watch table)	Yes	No	No	No
Reading tags in the program status	Yes	No	No	No
Online editing of blocks	Yes	No	No	No
Modifying outputs in STOP mode	Yes	No	No	No

Behavior of a password-protected module during operation

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

Before an online function is executed, the necessary permission is checked and, if necessary, the user is prompted to enter a password.

Example: The module was configured with read access and you want to execute the "Modify tags" function. This requires write access to a test function; therefore, the assigned password must be entered to execute the function.

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. If the online connection is restored after an interruption, you do not have to enter the access data again. To manually cancel access authorization, click "Online > Delete access rights".

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

Access to a password-protected S7-1500 CPU can be restricted locally on the display. The restriction is only in effect when the operating mode switch is set to RUN.

See also

Communication with SIMATIC S7-1500 (Page 77)

Security levels (Page 87)

Access password for the HMI connection (Page 92)

Access password for the HMI connection (Basic Panels)

Introduction

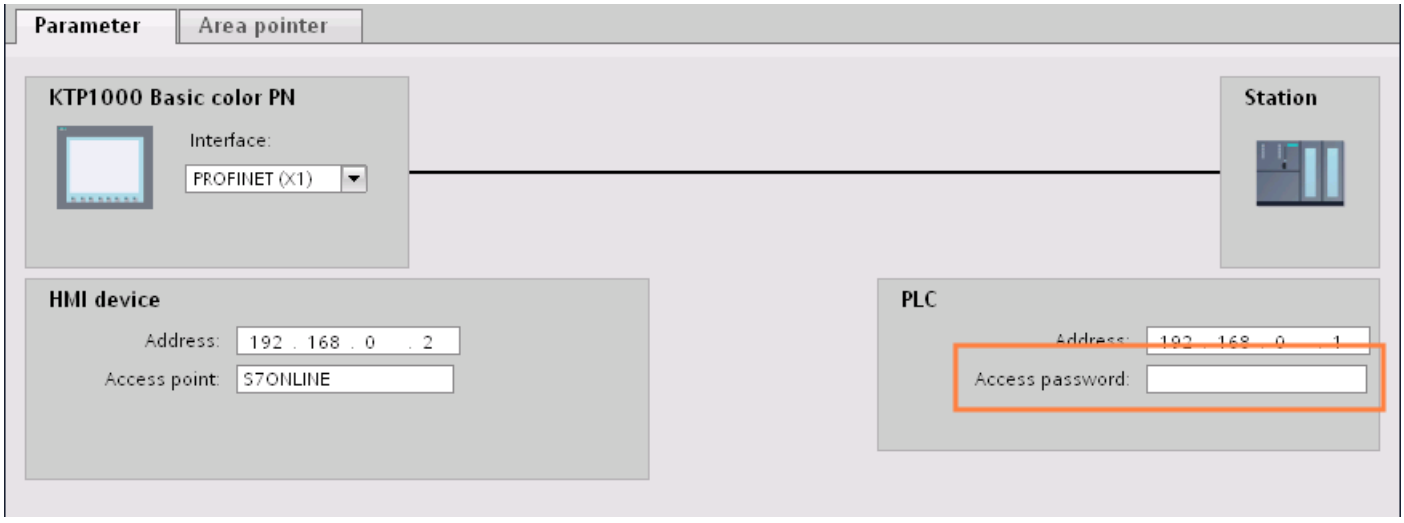
You secure access to a PLC by assigning a password.

You assign the password when you configure the password.

A password is mandatory as of protection level "Complete protection". Communication to the PLC is denied if an incorrect password or no password is entered.

Entering access password

You enter the password for the PLC in the "Connections" editor.



See also

Communication with SIMATIC S7-1500 (Page 77)

Security levels (Page 87)

Setting options for the protection (Page 88)

1.6.2.3 Setting port options (Basic Panels)

Setting the port options (Basic Panels)

Changing connection settings for the PROFINET IO port

You can change the network settings for the PROFINET IO port as required. By default, the settings are made automatically. In normal situations, this guarantees problem-free communication.

Possible settings for transmission rate / duplex

Depending on the selected device, you can make the following settings for "Transmission rate / duplex":

- Automatic setting
  Recommended default setting of the port. The transmission settings are automatically "negotiated" with the peer port. The "Enable autonegotiation" option is also enabled as a default, in other words, you can use cross cables or patch cables for the connection.
- TP/ITP at x Mbps full duplex (half duplex)
  Setting of the transmission rate and the full duplex/half duplex mode. The effectiveness depends on the "Enable autonegotiation" setting:
  - Autonegotiation enabled
    You can use both cross cable and patch cable.
  - Autonegotiation disabled
    Make sure that you use the correct cable (patch cable or cross cable)! The port is also monitored with this setting.
- Deactivated
  Depending on the module type, the drop down list box can contain the "- Disabled -" option. This option, for example, allows you to prevent access to an unused port for security reasons. With this setting, diagnostic events are not generated.

"Monitor" option

This option enables or disables port diagnostics. Examples of port diagnostics: The link status is monitored, in other words, the diagnostics are generated during link-down and the system reserve is monitored in the case of fiber optic ports.

Option "Enable autonegotiation "

The autonegotiation setting can only be changed if a concrete medium (for example, TP 100 Mbps full duplex) is selected. Whether or not a concrete medium can be set depends on the properties of the module.

If autonegotiation is disabled, this causes the port to be permanently specified, as for example, is necessary for a prioritized startup of the IO device.

You must make sure the partner port has the same settings because with this option the operating parameters of the connected network are not detected and the data transmission rate and transmission mode can accordingly not be optimally set.

Note

When a local port is connected, STEP 7 makes the setting for the partner port if the partner port supports the setting. If the partner port does not accept the setting, an error message is generated.

See also

Wiring rules for disabled autonegotiation (Page 94)

Boundaries at the port (Page 95)

Wiring rules for disabled autonegotiation (Basic Panels)

Requirements

You have made the following settings for the port in question, for example, to accelerate the startup time of the IO device:

- Fixed transmission rate
- Autonegotiation incl. autocrossing disabled

The time for negotiating the transmission rate during startup has been saved.

If you have disabled autonegotiation, you must observe the wiring rules.

Wiring rules for disabled autonegotiation

PROFINET devices have the following two types of ports:

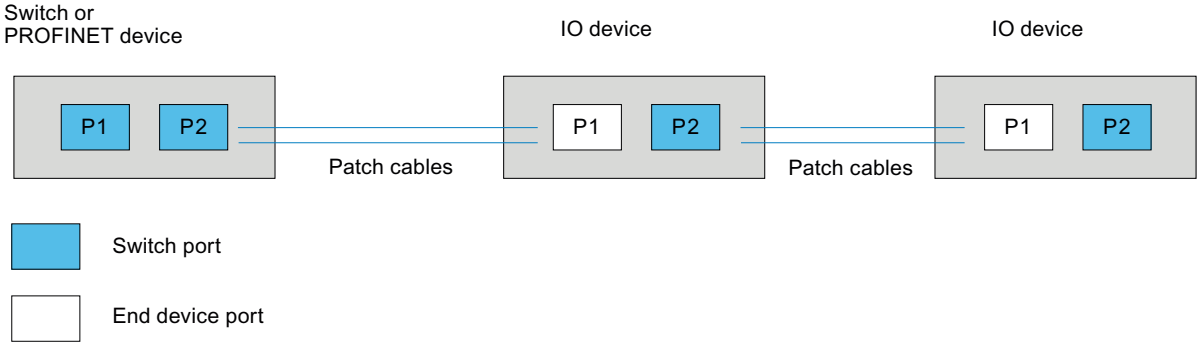
Type of port	PROFINET devices	Note
Switch port with crossed pin assignment	For IO devices: Port 2 For S7 CPUs with 2 ports: Ports 1 and 2	Crossed pin assignment means that the pin assignment for the ports for sending and receiving between the respective PROFINET devices is exchanged internally.
End device port with uncrossed pin assignment	For IO devices: Port 1 For S7 CPUs with one port: Port 1	-

Validity of the wiring rules

The cabling rules described in the following paragraph apply exclusively for the situation in which you have specified a fixed port setting.

Rules for cabling

You can connect several IO devices in line using a single cable type (patch cable). To do this, you connect port 2 of the IO device (distributed I/O) with port 1 of the next IO device. The following graphic gives an example with two IO devices.



See also

Setting the port options (Page 93)

Boundaries at the port (Page 95)

Boundaries at the port (Basic Panels)

Requirements

The respective device must support boundaries settings in order to work with boundaries. If the device for PROFINET does not support boundaries settings, the corresponding parameters are disabled.

Enable boundaries

"Boundaries" are limits for transmission of certain Ethernet frames. The following boundaries can be set at a port:

- "End of discovery of accessible devices"
  No forwarding of DCP frames to identify accessible devices. Devices downstream from this port cannot be reached by the project tree under "Accessible devices". Devices downstream from this port cannot be reached by the CPU.
- "End of topology discovery"
  LLDP frames (Link Layer Discovery Protocol) are not forwarded for topology detection.
- "End of sync domain"
  No forwarding of sync frames transmitted to synchronize nodes within a sync domain.
  If you operate, for example, a PROFINET device with more than two ports in a ring, you should prevent the sync frame from being fed into the ring by setting a sync boundary (at the ports not inside the ring).
  Additional example: If you want to use several sync domains, configure a sync domain boundary for the port connected to a PROFINET device from the other sync domain.

Restrictions

The following restrictions must be observed:

- The individual check boxes can only be used if the port supports the function in question.
- If a partner port has been determined for the port, the following check boxes cannot be used:
  - "End of discovery of accessible devices"
  - "End of topology discovery"

See also

Setting the port options (Page 93)

Wiring rules for disabled autonegotiation (Page 94)

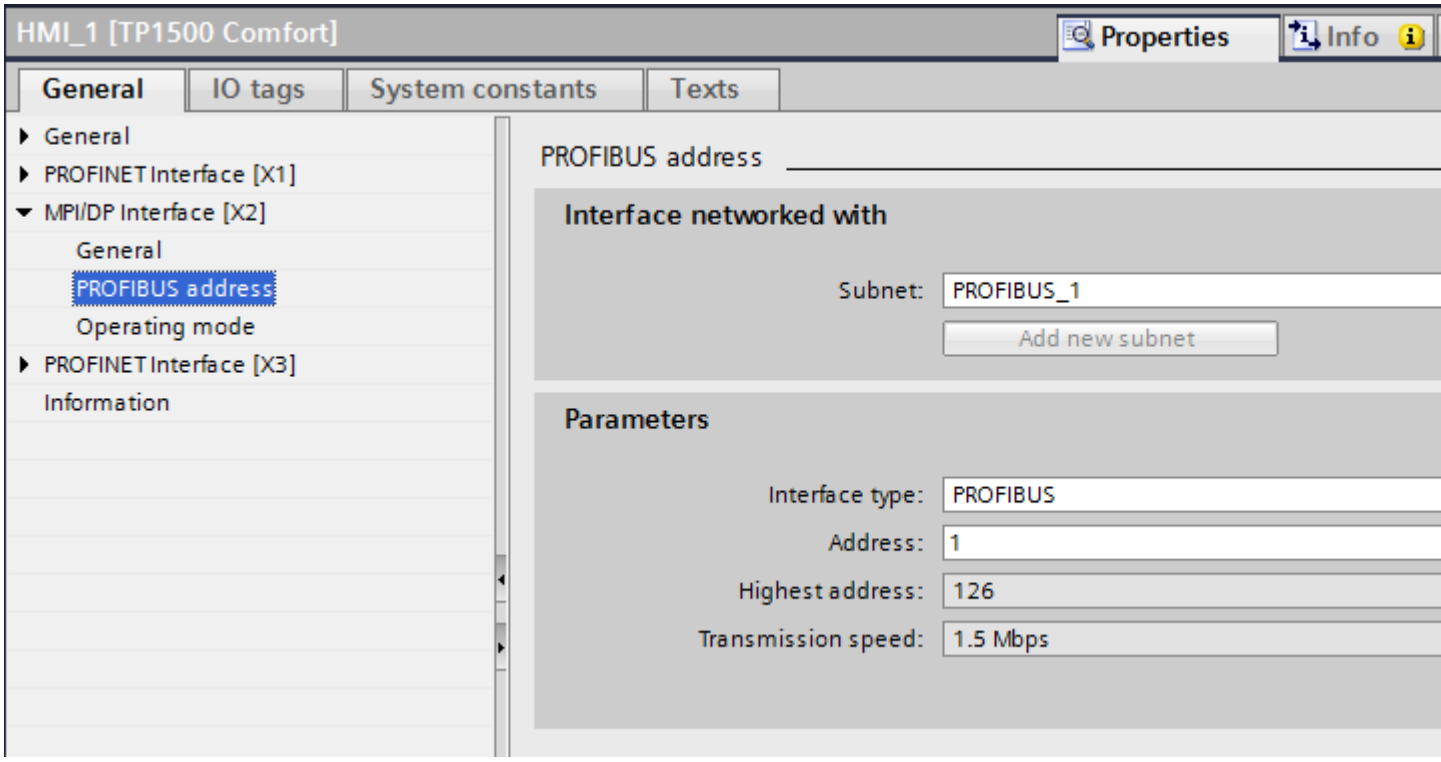
1.6.3 Communication via PROFIBUS (Basic Panels)

1.6.3.1 Configuring an HMI connection (Basic Panels)

Communication via PROFIBUS (Basic Panels)

HMI connections via PROFIBUS

If you have inserted an HMI device and a SIMATIC S7 1500 into the project, interconnect the two PROFIBUS interfaces in the "Devices & Networks" editor.



HMI connection in the "Devices & Networks" editor

You configure the HMI connection between the PLC and the HMI device via PROFIBUS in the "Devices & Networks" editor.

Connection in the "Connections" editor

Alternatively, you configure the connection between the PLC and HMI device in the "Connections" editor of the HMI device.

See also

Communication with SIMATIC S7-1500 (Page 77)

Configuring an HMI connection via PROFIBUS (Page 98)

Configuring an HMI connection via PROFIBUS (Basic Panels)

Introduction

You configure an HMI connection over PROFIBUS between HMI devices and a SIMATIC S7 1500 in the "Devices & Networks" editor.

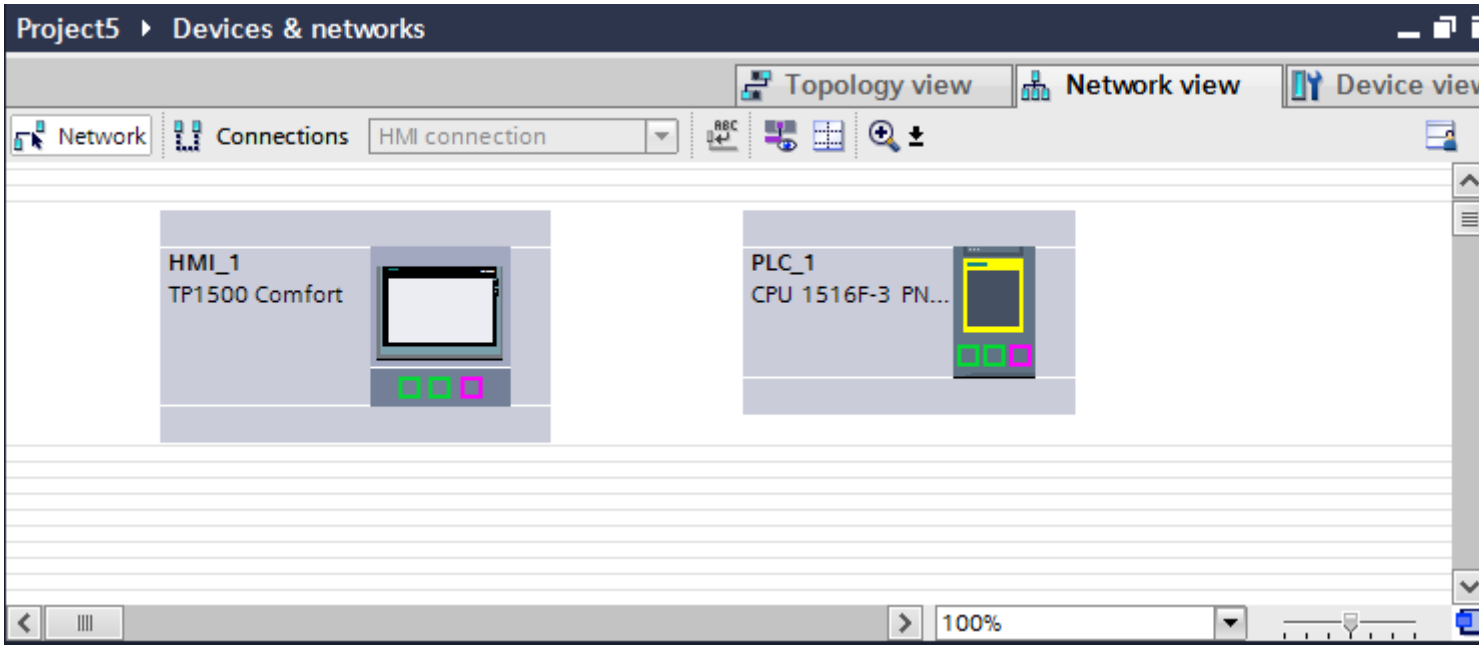
Requirements

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

- HMI device with MPI/DP interface
- SIMATIC S7 1500 with PROFIBUS interface

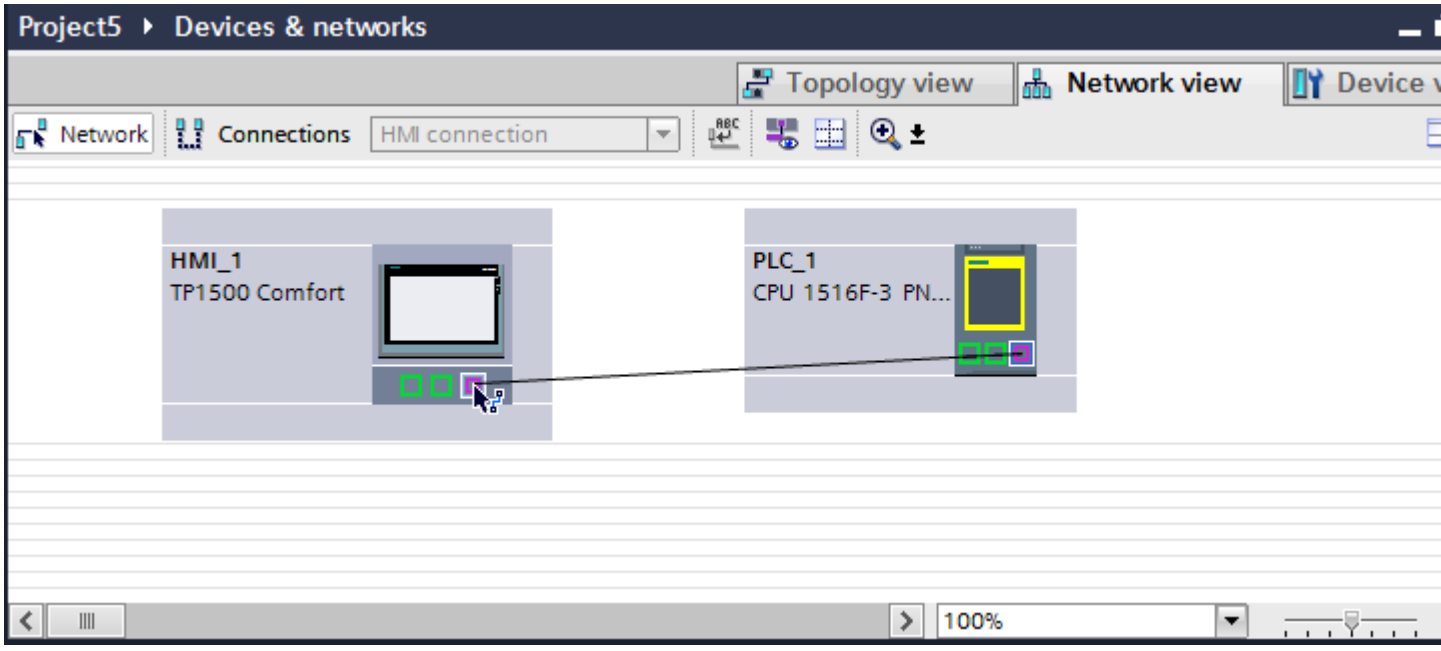
Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed graphically in the network view.



2. Click the "Connections" button.
   The devices available for connection are highlighted in color.
3. Click the HMI device interface.

4. Select the "PROFIBUS" interface type in the Inspector window under "Properties > General > HMI MPIDP > Parameters".
5. Click the interface of the PLC and use a drag-and-drop operation to draw a connection to the HMI device.



6. Click the connecting line.
7. Click "Highlight HMI connection" and select the HMI connection.
   The connection is displayed graphically in the Inspector window.
8. Click the communication partners in the "Network view" and change the PROFIBUS parameters in the Inspector window according to the requirements of your project.
   For additional information on this, see section "PROFIBUS parameters (Page 100)".

Note

The created HMI connection is also shown in the tabular area of the editor on 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 an HMI connection over PROFIBUS between an HMI device and a SIMATIC S7 1500.

See also

Communication with SIMATIC S7-1500 (Page 77)

Communication via PROFIBUS (Page 97)

PROFIBUS parameters (Page 100)

1.6.3.2 PROFIBUS parameters (Basic Panels)

PROFIBUS parameters for the HMI connection (Basic Panels)

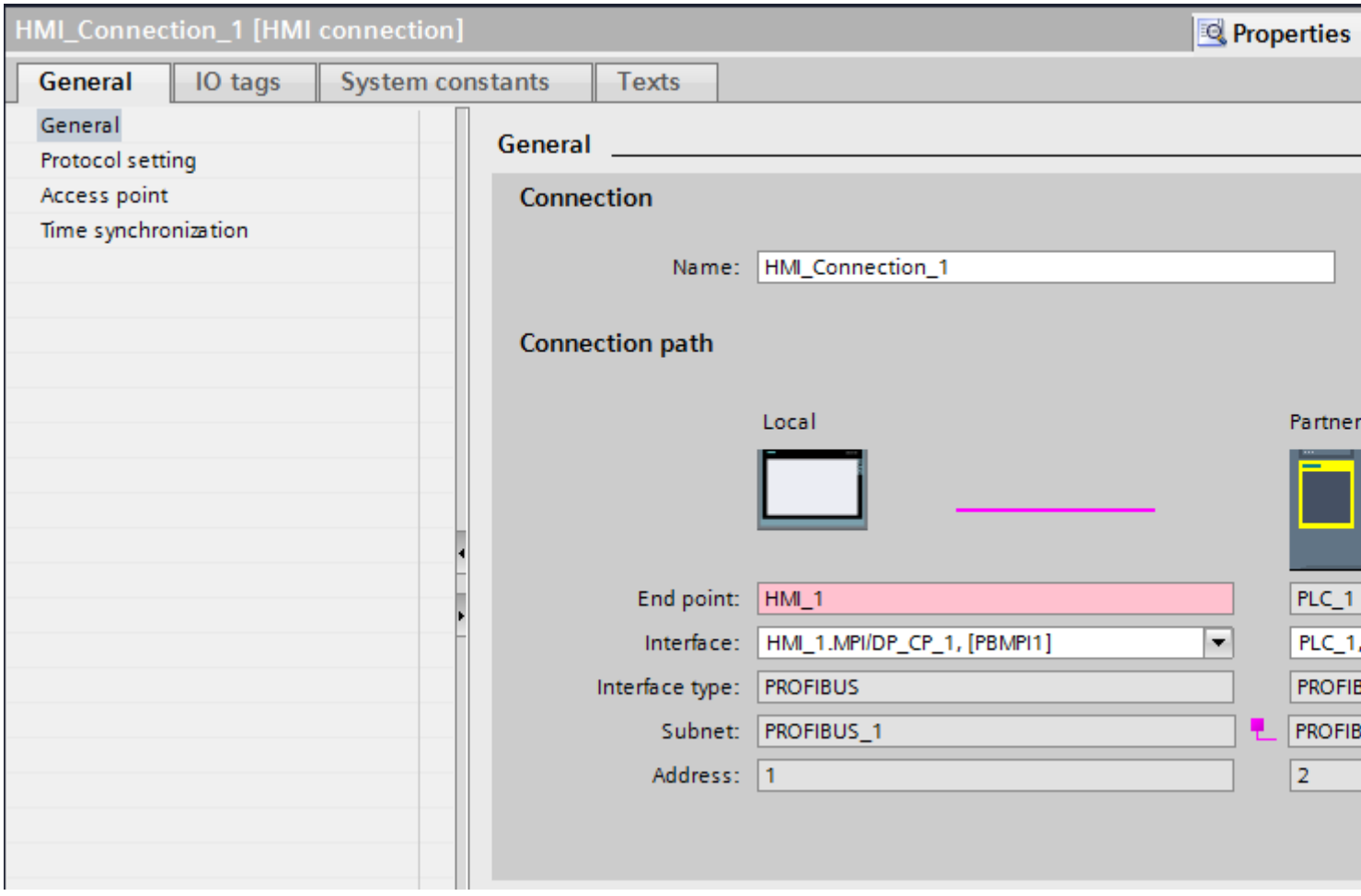
PROFIBUS parameters for the HMI connection

An overview of the configured HMI connection parameters can be found in the properties for the HMI connection.

Only limited changes are possible in this Inspector window.

Displaying and changing the HMI connection parameters

1. Click the HMI connection in the "Devices & Networks" editor.
2. Change the parameters of the HMI connection in the Inspector window under "Properties > General > General".



"Connection"

Displays whether the devices are networked together.

 - displayed if the devices are networked together.

 - displayed if the devices are not networked together.

"Connection path"

The communication partners of the selected HMI connection and the associated PROFIBUS parameters are displayed in the "Connection path" area. Some of the areas displayed cannot be edited in this dialog.

- "End point"
  Displays the device name. This area cannot be edited.
- "Interface"
  Displays the selected interface of the device. You can choose between several interfaces, depending on the device.
- "Interface type"
  Displays the selected interface type. This area cannot be edited.
- "Subnet"
  Displays the selected subnet. This area cannot be edited.
- "Address"
  Displays the PROFIBUS address of the device. This area cannot be edited.
- "Find connection path" button
  Enables the subsequent specification of connections.

See also

Communication with SIMATIC S7-1500 (Page 77)

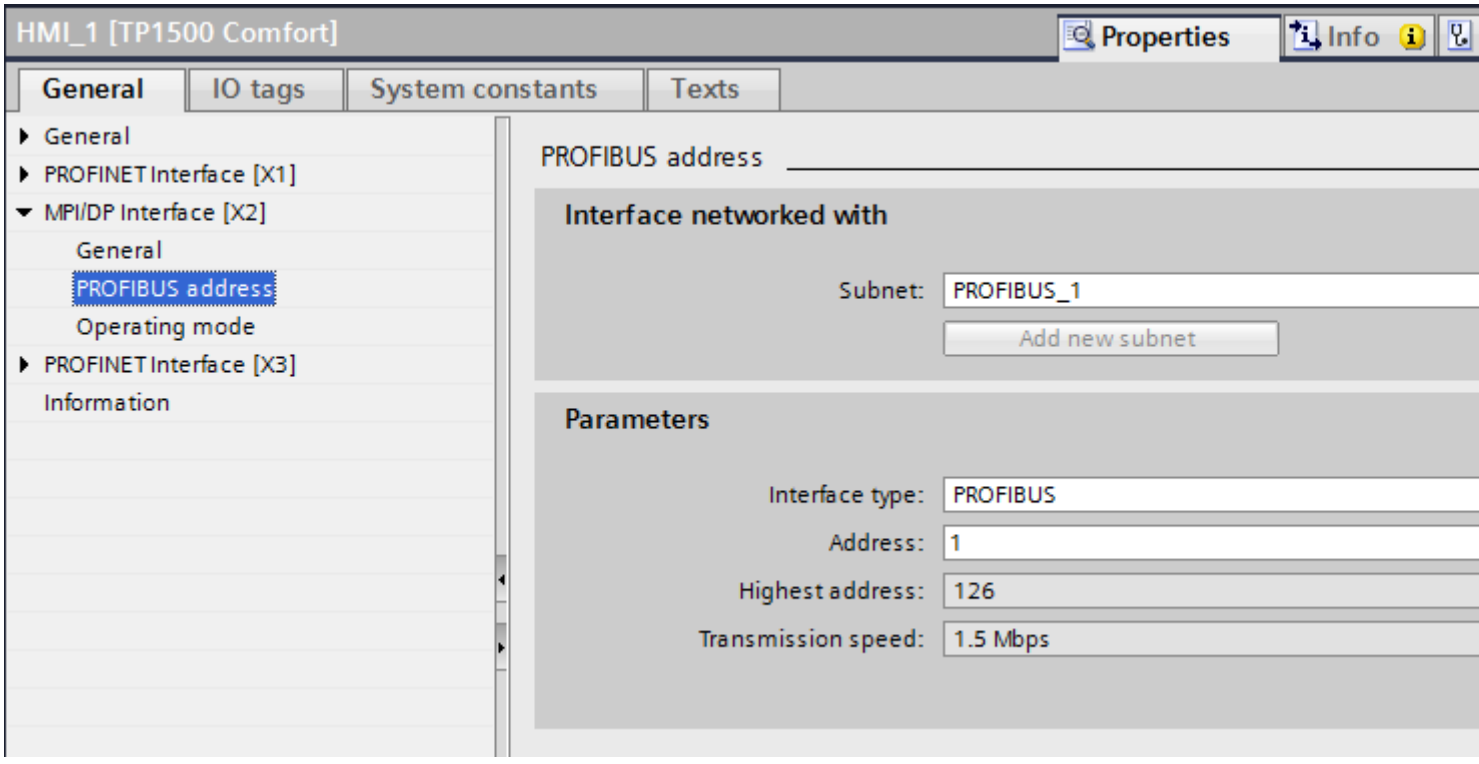
PROFIBUS parameters for the HMI device (Basic Panels)

PROFIBUS parameters for the HMI device

An overview of the configured HMI device parameters can be found in the properties for the HMI device.

Displaying and changing PROFINET parameters of the HMI device

1. Click the HMI device in the "Devices & Networks" editor.
2. Change the parameters of the HMI device in the Inspector window under "Properties > General > General".



"Interface networked with"

In the "Interface networked with" area, select the subnet of the HMI connection via which the HMI device is connected to the network. You use the "Add new subnet" button to create a new subnet.

"Parameters"

- "Interface type"
  You assign the interface type in the "Interface type" area. Depending on the HMI device type, you have various interfaces to choose from.
- "Address"
  You assign the PROFIBUS address of the HMI device in the "Address" area. The PROFIBUS address must be unique throughout the PROFIBUS network.
- "Highest address"
  The "Highest address" area displays the highest address of the PROFIBUS network.
- "Transmission speed"
  The "Transmission speed" is determined by the slowest device connected to the network. The setting is identical throughout the network.

See also

Communication with SIMATIC S7-1500 (Page 77)

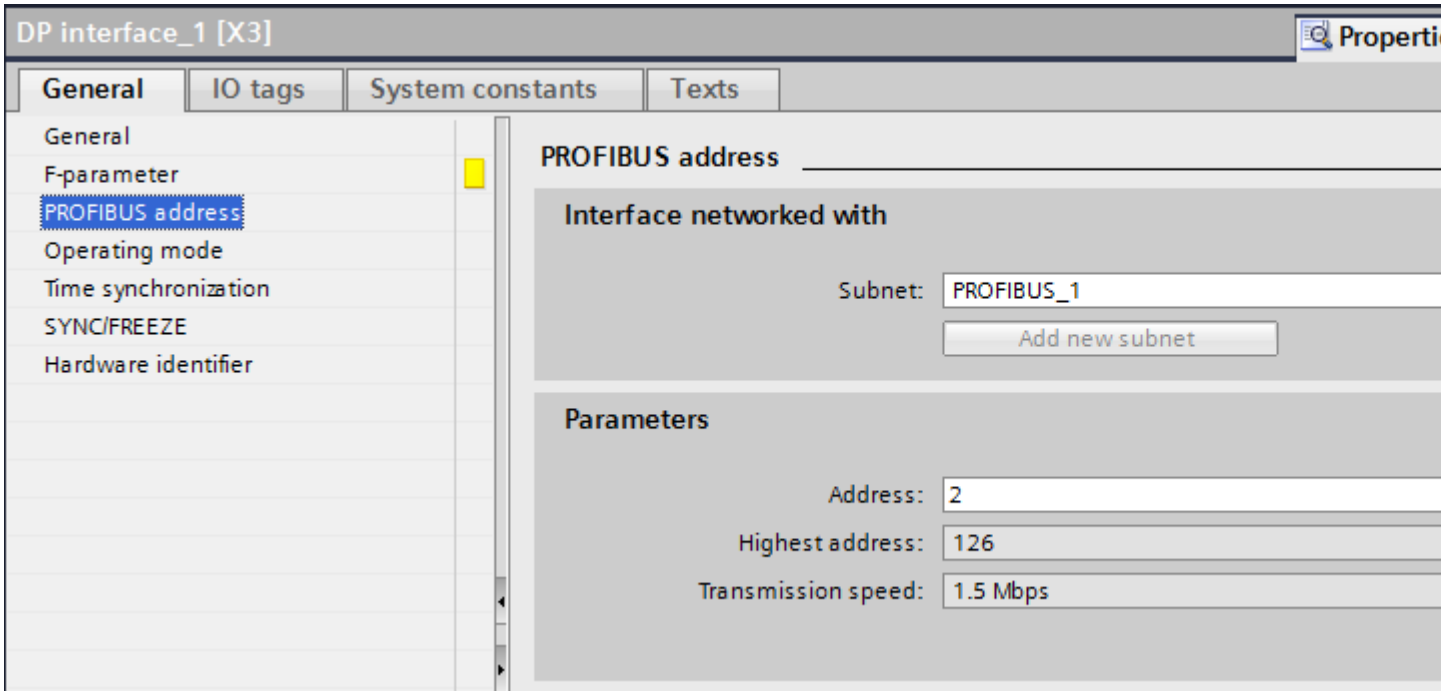
PROFIBUS parameters for the PLC (Basic Panels)

PROFIBUS parameters for the PLC

An overview of the configured parameters can be found in the properties for the PLC.

Displaying and changing PROFIBUS parameters of the PLC

1. Click the PLC in the "Devices & Networks" editor.
2. Change the parameters of the PLC in the Inspector window under "Properties > General > General".



"Interface networked with"

In the "Subnet" area, select the subnet of the HMI connection via which the PLC is connected to the network. You use the "Add new subnet" button to create a new subnet.

"Parameters"

- "Interface type"
  Depending on the HMI device type, you have various interfaces to choose from.
- "Address"
  You assign the PROFIBUS address of the HMI device in the "Address" area. The PROFIBUS address must be unique throughout the PROFIBUS network.
- "Highest address"
  The "Highest address" area displays the highest address of the PROFIBUS network.
- "Transmission speed"
  The "Transmission speed" is determined by the slowest device connected to the network. The setting is identical throughout the network.

See also

Communication with SIMATIC S7-1500 (Page 77)

Protection of communication (Basic Panels)

Security levels (Basic Panels)

If you want to protect the controller and HMI device communication, you can assign protection levels for the communication.

For a SIMATIC S7-1500 CPU, you can enter multiple passwords and thereby set up different access rights for various user groups.

The passwords are entered in a table, so that exactly one protection level is assigned to each password.

The effect of the password is given in the "Protection" column.

For the SIMATIC S7-1500 controller, several aspects need to be considered when setting protection levels.

For additional information on this, see:

Setting options for the protection (Page 105)

Example

When configuring the controller, you select the "Complete protection" protection level for a standard CPU (i.e., not an F-CPU).

Afterwards, you enter a separate password for every protection level above it in the table.

For users who do not know any of the passwords, the CPU is completely protected. Not even HMI access is possible.

For users who know one of the assigned passwords, the effect depends on the table row in which the password occurs:

- The password in row 1 (no protection) allows access as if the CPU were completely unprotected. Users who know this password have unrestricted access to the CPU.
- The password in row 2 (write protection) allows access as if the CPU were write-protected. Despite knowing the password, users who know this password only have read access to the CPU.
- The password in row 3 (read and write protection) allows access as if the CPU were read-protected and write-protected, so that only HMI access is possible for users who know this password.

See also

Access password for the HMI connection (Page 109)

Setting options for the protection (Page 105)

Setting options for the protection (Basic Panels)

Access levels

The following section describes how to use the various access levels of the S7-1500 CPUs.

S7-1500 CPUs provide various access levels to limit the access to specific functions.

The individual access levels as well as their associated passwords are specified in the object properties of the CPU. You assign parameters for the access level in a table.

Access level	Access			Access permission
	HMI	Read	Write	Password
Full access (no protection)	✓	✓	✓	
Read access	✓	✓		
HMI access	✓			
No access (complete protection)				

The green checkmarks in the columns to the right of the respective access level specify which operations are possible without knowing the password of this access level.

If you want to use the functions of fields that are not selected in the "Access" column, a password has to be entered:

Example:

You set the access level "Read access". You can see from the table that write access is not possible during operation without entering a password.

The table also shows that full access is necessary for the "Write" function.

To use a function requiring write access during operation, the password for full access must therefore be entered.

NOTICE
Configuring an access level does not replace know-how protection
Configuring access levels offers a high degree of protection against unauthorized changes to the CPU by restricting download privileges to the CPU. However, blocks on the memory card are not write- or read-protected. Use know-how protection to protect the code of blocks on the memory card.

Default characteristics

The default access level is "Full access (no protection)". Every user can read and change the hardware configuration and the blocks. A password is not set and is also not required for online access.

The access levels in detail

Below you can find an explanation of the existing access levels and which functions are possible at the respective access level.

- Full access (no protection):
  The hardware configuration and the blocks can be read and changed by all users.
- Read access for F-blocks (F-CPUs only):
  F-blocks of the safety program cannot be changed without authorization by the password of this access level or a higher level.
  Further information is available in the programming and operating manual *SIMATIC Safety - Configuring and Programming.*
- 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, as is changing the operating state (RUN/STOP).
  You cannot load blocks or a hardware configuration into the CPU. Moreover, no writing test functions and firmware updates are possible.

- HMI access:
  Only HMI access and access to diagnostics data is possible. Tags can be read and written via a HMI device.
  At this access level, you can neither load blocks and the hardware configuration into the CPU nor load blocks and the hardware configuration from the CPU into the programming device.
  In addition, the following is **not** possible: Writing test functions, changing the operating state (RUN/STOP) and firmware updates.
- No access (complete protection):
  Only identification data can be read, via "Accessible devices", for example.
  Neither read nor 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).
  Legitimation with a configured password provides you with access in accordance with the associated protection level.

Behavior of functions at different access levels

The table below describes which online functions are possible in the various protection levels.

Function	Full access	Read access	HMI access	No access
Identification of the device, via "Accessible devices", for example	Yes	Yes	Yes	Yes
HMI diagnostics view	Yes	Yes	Yes	No
Monitoring tags (M, I, Q, DB content) via HMI device	Yes	Yes	Yes	No
Modifying tags (M, I, Q, DB content) via HMI device	Yes	Yes	Yes	No
Diagnostics display (for example, device information, connection display, alarm display, diagnostic buffer)	Yes	Yes	Yes	No
Reading cycle time statistics (Online & Diagnostics)	Yes	Yes	Yes	No
Reading information from the hardware configuration (Online & Diagnostics)	Yes	Yes	Yes	No
Reading time-of-day	Yes	Yes	Yes	No
Executing online functions within the hardware configuration (Online & Diagnostics)	Yes	Yes	Yes	No
Acknowledging alarms	Yes	Yes	Yes	No
Receiving alarms	Yes	Yes	Yes	No
Enabling/disabling alarms	Yes	Yes	No	No
Reading tags via test function (STEP 7, tag table or watch table)	Yes	Yes	No	No

Requesting operating state change online (RUN/STOP/ warm restart)	Yes	Yes	No	No
Downloading data blocks, code blocks, hardware configuration to PG/PC	Yes	Yes	No	No
Set time-of-day	Yes	Yes	No	No
Deleting data blocks, code blocks, hardware configuration in the CPU	Yes	No	No	No
Downloading individual data blocks, code blocks, hardware configurations to the CPU	Yes	No	No	No
Loading PLC program to the device and resetting	Yes	No	No	No
Firmware update of CPUs or I/O modules	Yes	No	No	No
Modifying tags via test function (STEP 7, watch table)	Yes	No	No	No
Reading tags in the program status	Yes	No	No	No
Online editing of blocks	Yes	No	No	No
Modifying outputs in STOP mode	Yes	No	No	No

Behavior of a password-protected module during operation

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

Before an online function is executed, the necessary permission is checked and, if necessary, the user is prompted to enter a password.

Example: The module was configured with read access and you want to execute the "Modify tags" function. This requires write access to a test function; therefore, the assigned password must be entered to execute the function.

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. If the online connection is restored after an interruption, you do not have to enter the access data again. To manually cancel access authorization, click "Online > Delete access rights".

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

Access to a password-protected S7-1500 CPU can be restricted locally on the display. The restriction is only in effect when the operating mode switch is set to RUN.

See also

Security levels (Page 104)

Access password for the HMI connection (Basic Panels)

Introduction

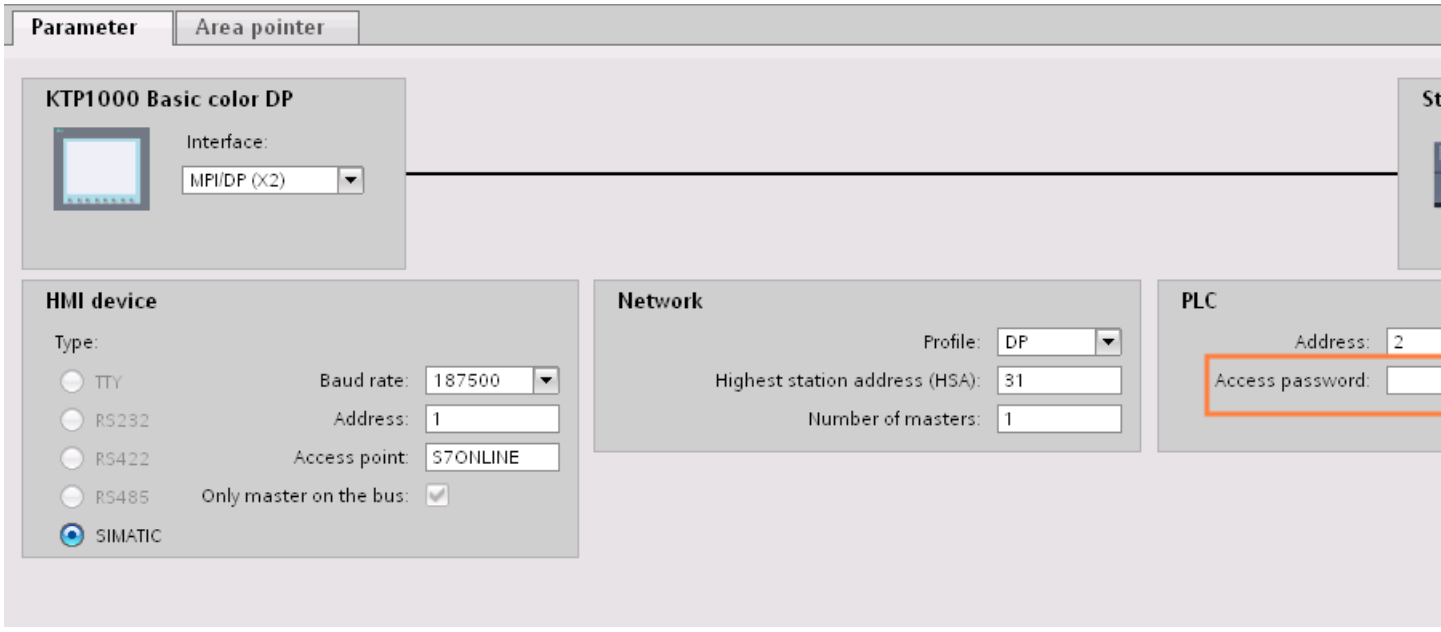
You secure access to a PLC by assigning a password.

You assign the password when you configure the password.

A password is mandatory as of protection level "Complete protection". Communication cannot be established with the PLC if you enter no password or an incorrect password.

Assigning password

You enter the password for the PLC in the "Connections" editor.



See also

Security levels (Page 104)

1.6.4 Data exchange (Basic Panels)

1.6.4.1 Data exchange using area pointers (Basic Panels)

General information on area pointers (Basic Panels)

Introduction

You use an area pointer to access a data area in the PLC. During communication, the PLC and the HMI device alternately access these data areas for read and write operations.

The PLC and the HMI device trigger defined interactions based on the evaluation of stored data.

Configuration of area pointers

Before you use an area pointer, you enable it under "Connections > Area pointer". You then assign the area pointer parameters.

You can find more detailed information on configuring area pointers in:

Data exchange using area pointers (Page 61)

Restrictions

You can only configure the following data types for communication with SIMATIC S7 1500 for data exchange using area pointers:

- UInt and array of UInt
- Word and array of Word
- Int and array of Int
- "Array[0..15] of Bool" for area pointer "Coordination"
- Date_And_Time
- DTL and LDT

See also

Data exchange using area pointers (Page 61)

"Screen number" area pointer (Page 111)

"Date/time" area pointer (Page 112)

"Date/time PLC" area pointer (Page 113)

"Coordination" area pointer (Page 115)

"Project ID" area pointer (Page 116)

"Job mailbox" area pointer (Page 117)

"Data record" area pointer (Page 119)

"Screen number" area pointer (Basic Panels)

Function

The HMI devices store information about the screen called up on the HMI device in the "Screen number" area pointer.

This allows the transfer of the current screen contents from the HMI device to the PLC. The PLC can trigger specific reactions such as the call of a different screen.

Use

Configure and enable the area pointer in "Communication > Connections" before you put it into use. You can create only **one** instance of the "Screen number" area pointer and only on **one** PLC.

The screen number is always transferred to the PLC when a new screen is activated or when the focus within a screen changes from one screen object to another.

Structure

The area pointer is a data area in the memory of the PLC with a fixed length of 5 words.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st word	Current screen type															
2nd word	Current screen number															
3rd word	Reserved															
4th word	Current field number															
5th word	Reserved															

- Current screen type
  "1" for root screen or
  "4" for permanent area
- Current screen number
  1 to 32767
- Current field number
  1 to 32767

See also

General information on area pointers (Page 110)

"Date/time" area pointer (Basic Panels)

Function

This area pointer is used to transfer the date and time from the HMI device to the PLC.

The PLC writes control job "41" or "40" to the job mailbox.

When it evaluates the control job, the HMI device writes its current date and the time in the data area configured in the "Date/time" area pointer. All definitions are coded in BCD format.

Note

You cannot use the "Date/Time PLC" area pointer if you have configured the "Date/Time" area pointer.

The "Date/Time" area pointer when used in a project which contains multiple connections must be enabled for each configured connection.

"DTL" data type

If you configure the "Date/Time" area pointer, use the DTL data type.

A tag of the "DTL" data type has a length of 12 bytes and saves information on date and time in a predefined structure.

The "DTL" data type has the following structure:

Byte	Component	Data type	Value range
0	Year	UINT	1970 to 2554
1			
2	Month	USINT	0 to 12
3	Day	USINT	1 to 31
4	Day of the week	USINT	1(Sunday) to 7(Saturday) The day of the week is not considered in the value entry.
5	Hour	USINT	0 to 23
6	Minute	USINT	0 to 59
7	Second	USINT	0 to 59
8	Nanoseconds	UDINT	0 to 999 999 999
9			
10			
11			

The "DTL" data type supports time information down to the nanosecond range. Because panels only support time information down to the millisecond, you may encounter the following restriction when using the area pointers:

For the transmission of time information from a panel to the controller, the smallest unit of time is 1 millisecond. The value range from microseconds to nanoseconds of the "DTL" data type is filled with zeros.

Using data types

The data types "Date_And_Time, DTL" and "LDT" can only be used with the "Date/time" and "Date/time PLC" area pointers.

The data format of the "Date/time" area pointer depends on job mailbox 40/41.

If there are no control tags linked to the area pointer, or a control tag is linked with the data type "Array[0..5] of UInt/Word/Int", the following applies:

The configuration of the "Date/time" area pointer is only used for job mailbox 41.

If job mailbox 40 is used, the data format "DATE_AND_TIME (BCD-encoded)" is used (shown in the next section).

If the "Date/time" and "Date/time PLC" area pointers are linked to a control tag with the data type "DATE_AND_TIME", "DTL" or "LDT", the associated data format is used in the corresponding area pointer.

See also

General information on area pointers (Page 110)

"Date/time PLC" area pointer (Basic Panels)

Function

This area pointer is used to transfer the date and time from the PLC to the HMI device. Use this area pointer if the PLC is the time master.

The PLC loads the data area of the area pointer. All definitions are coded in BCD format.

The HMI device reads the data cyclically within the configured acquisition cycle and synchronizes itself.

Note

Set an acquisition cycle of sufficient length for the Date/time area pointer in order to avoid any negative impact on HMI device performance.
Recommended: Acquisition cycle of 1 minute, if the process allows this.

"Date/Time PLC" is a global area pointer and may be configured only once per project.

Note

You cannot use the "Date/Time" area pointer if you have configured the "Date/Time PLC" area pointer.

"DTL" data type

If you configure the "Date/Time" area pointer, use the DTL data type.

A tag of the "DTL" data type has a length of 12 bytes and saves information on date and time in a predefined structure.

The "DTL" data type has the following structure:

Byte	Component	Data type	Value range
0	Year	UINT	1970 to 2554
1			
2	Month	USINT	0 to 12
3	Day	USINT	1 to 31
4	Day of the week	USINT	1(Sunday) to 7(Saturday) The day of the week is not considered in the value entry.
5	Hour	USINT	0 to 23
6	Minute	USINT	0 to 59
7	Second	USINT	0 to 59
8	Nanoseconds	UDINT	0 to 999 999 999
9			
10			
11			

The "DTL" data type supports time information down to the nanosecond range. Because panels only support time information down to the millisecond, you may encounter the following restriction when using the area pointers:

For the transmission of time information from a panel to the controller, the smallest unit of time is 1 millisecond. The value range from microseconds to nanoseconds of the "DTL" data type is filled with zeros.

Using data types

The data types "Date_And_Time, DTL" and "LDT" can only be used with the "Date/time" and "Date/time PLC" area pointers.

The data format of the "Date/time" area pointer depends on job mailbox 40/41.

If there are no control tags linked to the area pointer, or a control tag is linked with the data type "Array[0..5] of UInt/Word/Int", the following applies:

The configuration of the "Date/time" area pointer is only used for job mailbox 41.

If job mailbox 40 is used, the data format "DATE_AND_TIME (BCD-encoded)" is used (shown in the next section).

If the "Date/time" and "Date/time PLC" area pointers are linked to a control tag with the data type "DATE_AND_TIME", "DTL" or "LDT", the associated data format is used in the corresponding area pointer.

See also

General information on area pointers (Page 110)

"Coordination" area pointer (Basic Panels)

Function

The "Coordination" area pointer is used to implement the following functions:

- Detecting the startup of the HMI device in the control program
- Detecting the current operating mode of the HMI device in the control program
- Detecting whether the HMI device is ready to communicate in the control program

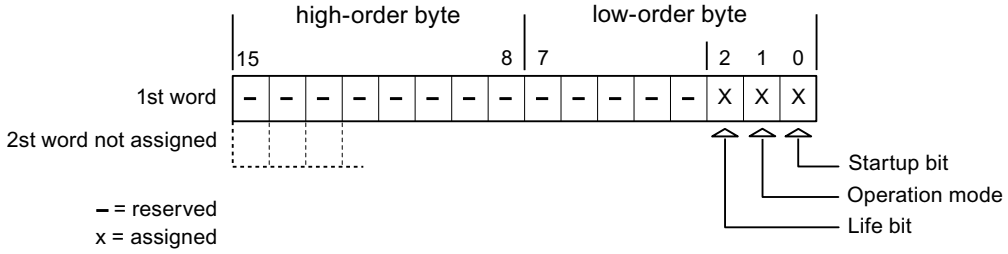
By default, the "Coordination" area pointer has the length of one word and cannot be changed.

Usage

Note

The HMI device always writes the entire coordination area when updating the area pointer. The control program can for this reason not make changes to the coordination area.

Assignment of the bits in the "Coordination" area pointer



Startup bit

The startup bit is set briefly to "0" by the HMI device during startup. It sets the bit permanently to "1" when startup is completed.

Operating mode

The operating mode bit is set to 1 as soon as the user switches the HMI device offline. The status of the operating mode bit is "0" during normal operation of the HMI device. You can determine the current operating mode of the HMI device by reading this bit in the control program.

Life bit

The HMI device inverts the life bit at intervals of approximately one second. You can check whether or not there is still a connection to the HMI device by querying this bit in the control program.

See also

General information on area pointers (Page 110)

"Project ID" area pointer (Basic Panels)

Function

When Runtime starts, a check can be carried out as to whether the HMI device is connected to the correct PLC. This check is important when operating with several HMI devices.

For this, the HMI device compares a value stored on the PLC with the value specified in configuration. This ensures compatibility of the configuration data with the PLC program. If there is no concordance, a system event is given on the HMI device and Runtime is stopped.

Use

Note

HMI connections cannot be switched "online".

The HMI connection in which the "Project ID" area pointer is used must be switched "online".

To use this area pointer, set up the following during the configuration:

- Define the version of the configuration. Values between 1 and 255 are possible.
  You enter the version in the editor "Runtime settings > General" in the "Identification" area.
- Data address of the value for the version that is stored in the PLC:
  You enter the data address in the editor "Communication > Connections" under "Address".

Connection failure

A connection failure to a device on which the "project ID" area pointer is configured results in all the other connections of the device being switched to "offline".

This behavior has the following requirements:

- You have configured several connections in a project.
- You are using the "project ID" area pointer in at least one connection.

Causes which may set connections "offline":

- The PLC is not available.
- The connection has been switched offline in the engineering system.

See also

General information on area pointers (Page 110)

"Job mailbox" area pointer (Basic Panels)

Function

The PLC can use the job mailbox to transfer jobs to the HMI device to trigger corresponding actions on the HMI device. These functions include, for example:

- Display screen
- Set date and time

Data structure

The first word of the job mailbox contains the job number. Depending on the job mailbox, up to three parameters can be transferred.

Word	Most significant byte	Least significant byte
n+0	0	Job number
n+1	Parameter 1	
n+2	Parameter 2	
n+3	Parameter 3	

The HMI device evaluates the job mailbox if the first word of this job is not equal to zero. This means that the parameters must be entered in the job mailbox first, followed by the job number.

When the HMI device accepts the job mailbox, the first word is set to 0 again. The execution of the job mailbox is generally not completed at this point in time.

Job mailboxes

All job mailboxes and their parameters are listed below. The "No." column contains the job number of the job mailbox. Job mailboxes can only be triggered by the PLC when the HMI device is online.

No.	Function	
14	**Set time (BCD-coded)**	
	Parameter 1	Left byte: - Right byte: hours (0-23)
	Parameter 2	Left byte: minutes (0-59) Right byte: seconds (0-59)
	Parameter 3	-
15	**Setting the date (BCD coded)** [3) 4)]	
	Parameter 1	Left byte: - Right byte: weekday (1-7: Sunday-Saturday)
	Parameter 2	Left byte: day (1-31) Right byte: month (1-12)
	Parameter 3	Left byte: year
23	**User logon**	

No.	Function	
14	**Set time (BCD-coded)**	
	Logs the user on with the name "PLC user" at the HMI device with the group number transferred in Parameter 1. The logon is possible only when the transferred group number exists in the project.	
	Parameter 1	Group number 1 to 255
	Parameter 2, 3	-
24	**User logoff**	
	Logs off the current user. (The function corresponds to the "logoff" system function)	
	Parameter 1, 2, 3	-
40	**Transfer date/time to PLC**	
	(in the S7 format DATE_AND_TIME) An interval of at least 5 seconds must be maintained between two successive jobs to prevent overload of the HMI device.	
	Parameter 1, 2, 3	-
41	**Transfer date/time to PLC**	
	(In OP/MP format) An interval of at least 5 seconds must be maintained between successive jobs to prevent overload of the HMI device.	
	Parameter 1, 2, 3	-
46	**Update tag**	
	Causes the HMI device to read the current value of the tags from the PLC whose update ID matches the value transferred in Parameter 1. (Function corresponds to the "UpdateTag" system function.)	
	Parameter 1	1 - 100
49	**Delete alarm buffer**	
	Deletes all analog alarms and discrete alarms of the "Warnings" class from the alarm buffer.	
	Parameter 1, 2, 3	-
50	**Delete alarm buffer**	
	Deletes all analog alarms and discrete alarms of the "Errors" class from the alarm buffer.	
	Parameter 1, 2, 3	-
51	**Screen selection** [2]	
	Parameter 1	Screen number
	Parameter 2	-
	Parameter 3	Field number
69	**Reading data record from PLC** [1]	
	Parameter 1	Recipe number (1-999)
	Parameter 2	Data record number (1-65535)
	Parameter 3	0: Do not overwrite existing data record 1: Overwrite existing data record
70	**Writing data record to PLC** [1]	
	Parameter 1	Recipe number (1-999)
	Parameter 2	Data record number (1-65535)
	Parameter 3	-

1)	Only for devices supporting recipes.
2)	OP 73, OP 77A and TP 177A HMI devices also execute the "Screen selection" job mailbox if the on-screen keyboard is active.
3)	The weekday is ignored on HMI device KTP 600 BASIC PN.
4)	The weekday is ignored when you configure the "Date/Time PLC" area pointer.

See also

General information on area pointers (Page 110)

"Data record" area pointer (Basic Panels)

"Data record" area pointer (Basic Panels)

Function

When data records are transferred between the HMI device and PLC, both partners access common communications areas on the PLC.

Data transfer types

There are two ways of transferring data records between the HMI device and PLC:

- Transfer without synchronization
- Transfer with synchronization via the data mailbox

Data records are always transferred directly, which means that the tag values are read straight from an address or written straight to an address configured for this tag without being redirected via an interim memory.

Initiating the transfer of data records

There are three ways of triggering the transfer:

- Operator input in the recipe view
- Job mailboxes
  The transfer of data records can also be triggered by the PLC.
- Triggering by configured functions

If the transfer of data records is triggered by a configured function or by a job mailbox, the recipe view on the HMI device remains operable. The data records are transferred in the background.

Simultaneous processing of several transfer requests is, however, not possible. In this case, the HMI device rejects the other transfer requests with a system alarm.

See also

General information on area pointers (Page 110)
Transfer without synchronization (Page 120)
Transfer with synchronization (Page 121)
Sequence of a transfer triggered by a job mailbox (Page 122)
Sequence of the transfer when triggered by a configured function (Page 123)
Possible causes of error when transferring data records (Page 124)

Transfer without synchronization (Basic Panels)

If you select asynchronous transfer of data records between the HMI device and PLC, there is no coordination over the common data areas. It is therefore unnecessary to set up a data area during configuration.

Asynchronous data record transfer can be a useful alternative, for example when:

- The system is capable of excluding the risk of uncontrolled overwriting of data by the communication peer.
- The PLC does not require information about the recipe number and data record number.
- The transfer of data records is triggered by the operator of the HMI device.

Reading values

When a read job is triggered, the values are read from the PLC addresses and transferred to the HMI device.

- Triggering by the operator in the recipe view:
  The values are downloaded to the HMI device. You can then, for example, process, edit, or save these values in the HMI device.
- Triggering by a function or job mailbox:
  The values are saved immediately to the data volume.

Writing values

When a write job is triggered, the values are written to the PLC addresses.

- Triggering by the operator in the recipe view:
  The current values are written to the PLC.
- Triggering by a function or job mailbox:
  The current values are written to the PLC from the data medium.

See also

"Data record" area pointer (Page 119)

Transfer with synchronization (Basic Panels)

If you select synchronous transfer, both communication partners set status bits in the common data area. You can use this mechanism to prevent uncontrolled overwriting of data in either direction in your control program.

Application

Synchronous data record transfer can be a useful solution, for example, when:

- The PLC is the "active partner" in the transfer of data records.
- The PLC evaluates the information about the recipe number and data record number.
- The transfer of data records is triggered by means of a Job mailbox.

Requirements

In order to synchronize transfer of data records between the HMI device and the PLC, the following requirements must be met during configuration:

- An area pointer has been set up: "Communication > Connections" editor in "Area pointer".
- The PLC with which the HMI device synchronizes transfer of data records is specified in the recipe:
  "Recipes" editor in the inspector window the option "Coordinated transfer of data records" under "General > Synchronization > Settings"

Structure of the data area

The data area has a fixed length of 5 words. Structure of the data area:

	15		0
1. Word	Current recipe number (1 - 999)		
2. Word	Current data record number (0 - 65535)		
3. Word	Reserved		
4. Word	Status (0, 2, 4, 12)		
5. Word	Reserved		

- Status
  The status word (word 4) can adopt the following values:

Value		Meaning
Decimal	Binary	
0	0000 0000	Transfer permitted, data mailbox free
2	0000 0010	Transferring
4	0000 0100	Transfer completed without error
12	0000 1100	Transfer completed with error

See also

"Data record" area pointer (Page 119)

Sequence of a transfer triggered by a job mailbox (Basic Panels)

The transfer of data records between the HMI device and the PLC can be initiated by either one of these stations.

The two job mailboxs No. 69 and No. 70 are available for this type of transfer.

No. 69: Read data record from PLC ("PLC → DAT")

Job mailbox no. 69 transfers data records from the PLC to the HMI device. The job mailbox is structured as follows:

	Left byte (LB)	Right byte (RB)
Word 1	0	69
Word 2	Recipe number (1-999)	
Word 3	Data record number (1 to 65535)	
Word 4	Do not overwrite existing data record: 0 Overwrite existing data record: 1	

No. 70: Write data record to PLC ("DAT → PLC")

Job mailbox No. 70 transfers data records from the HMI device to the PLC. The job mailbox is structured as follows:

	Left byte (LB)	Right byte (RB)
Word 1	0	70
Word 2	Recipe number (1-999)	
Word 3	Data record number (1-65,535)	
Word 4	—	

Sequence when reading from the PLC with job mailbox "PLC → DAT" (no. 69)

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the job and the status "Transferring" in the data record.	Abort without return message.
3	The HMI device reads the values from the PLC and saves these to the data record defined in the job mailbox.	

Step	Action	
4	• If "Overwrite" was selected in the job, an existing data record is overwritten without any prompt for confirmation. The HMI device sets the status "Transfer completed." • If "Do not overwrite" was selected in the job, and the data record already exists, the HMI device aborts the job and enters 0000 1100 in the status word of the data mailbox.	
5	The control program must reset the status word to zero in order to enable further transfers.	

Sequence of writing to the PLC with job mailbox "DAT → PLC" (no. 70)

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the job and the status "Transferring" in the data mailbox.	Abort without return message.
3	The HMI device fetches the values of the data record specified in the function from the data medium and writes the values to the PLC.	
4	The HMI device sets the status "Transfer completed."	
5	The control program can now evaluate the transferred data. The control program must reset the status word to zero in order to enable further transfers.	

See also

"Data record" area pointer (Page 119)

Sequence of the transfer when triggered by a configured function (Basic Panels)

Reading from the PLC using a configured function

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the function and the status "Transferring" in the data mailbox.	Abort with system event.
3	The HMI device reads the values from the PLC and stores them in the data record specified in the function.	

Step	Action	
4	• If "Yes" was selected for the "Overwrite" function, an existing data record is overwritten without any prompt for confirmation. The HMI device sets the status "Transfer completed." • If "No" was selected for the "Overwrite" function and the data record already exists, the HMI device aborts the job and enters 0000 1100 in the status word of the data mailbox.	
5	The control program must reset the status word to zero in order to enable further transfers.	

Writing to the PLC by means of configured function

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the function and the status "Transferring" in the data mailbox.	Abort with system event.
3	The HMI device fetches the values of the data record specified in the function from the data medium and transfers the values to the PLC.	
4	The HMI device sets the status "Transfer completed."	
5	The control program can now evaluate the transferred data. The control program must reset the status word to zero in order to enable further transfers.	

See also

"Data record" area pointer (Page 119)

Possible causes of error when transferring data records (Basic Panels)

Possible causes of error

The section below shows possible error causes which lead to the cancellation of data record transfer:

- Tag address not set up on the PLC
- Overwriting data records not possible

- Recipe number does not exist
- Data record number does not exist

Note

The status word may only be set by the HMI device. The PLC may only reset the status word to zero.

Note

The PLC may only evaluate the recipe and data record numbers when data inconsistency is detected if one of the conditions outlined below has been met:

- The data mailbox status is set to "Transfer completed".
- The data mailbox status is set to "Transfer completed with error".

Reaction to an aborted transfer due to errors

If the transfer of data records is aborted due to errors, the HMI device reacts as follows:

- Triggering by the operator in the recipe view
  Information in the status bar of the recipe view and output of system alarms
- Triggered by function
  Output of system alarms
- Triggering by job mailbox
  No return message on the HMI device

You can nonetheless evaluate the status of the transfer by querying the status word in the data record.

See also

"Data record" area pointer (Page 119)

1.6.4.2 Trends (Basic Panels)

Trends

A trend is the graphical representation of one or more values from the PLC. The value is read out time-triggered for Basic Panels.

For additional information see:

AUTOHOTSPOT

Time-triggered trends

The HMI device reads in the trend values cyclically at an interval specified in the configuration.

Time-triggered trends are suitable for continuous curves, such as the operating temperature of a motor.

1.6.4.3 Alarms (Basic Panels)

Configuring alarms (Basic Panels)

Configure alarms

Several steps are needed to configure alarms, such as operational messages, error alarms, and acknowledgement.

- Step 1: Create tags
- Step 2: Configure alarms
- Step 3: Configure acknowledgment

You can find additional information in the section:

AUTOHOTSPOT

Distinctive features when configuring alarms

If you are configuring connections of HMI devices to PLCs of other manufacturers, note the following distinctive features when configuring:

- Data types of the tags
- Addressing of tags
- How the bit positions are counted

Data types

For connections with a SIMATIC communication driver, the following data types are supported:

PLC	Permitted data types	
	Discrete alarms	Analog alarms
SIMATIC S7 PLCs	WORD, INT	BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, TIMER

How the bit positions are counted

For connections with a SIMATIC communication driver, the following counting method applies:

How the bit positions are counted	Byte 0								Byte 1							
	Most significant byte								Least significant byte							
In SIMATIC S7 PLCs	7							0	7							0
In WinCC you configure:	15							8	7							0

See also

Acknowledgment of alarms (Page 127)

Acknowledgment of alarms (Basic Panels)

Procedure

Create suitable tags on the PLC to acknowledge an error alarm. You assign these tags to an alarm in the "Bit messages" editor. You make the assignment in "Properties > Acknowledgment".

Distinction in terms of acknowledgment:

- Acknowledgment by the PLC
- Acknowledgment on the HMI device

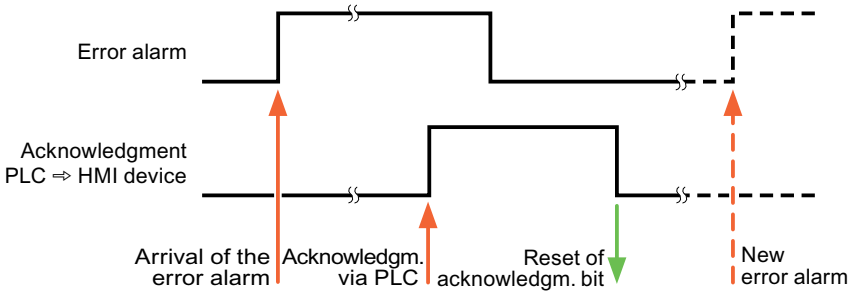
Acknowledgment by the PLC

In "PLC acknowledgment tag", configure the tag or array tag and the bit number that the HMI device uses to identify a PLC acknowledgment.

A bit set in the tag triggers acknowledgment of the assigned error alarm bit at the HMI device. This tag bit returns a function similar to acknowledgment on the HMI device which is triggered by pressing the "ACK" button, for example.

The acknowledgment bit must be located in the same tag as the bit for the error alarm.

Reset the acknowledgment bit before setting the bit in the alarm area again. The figure below shows the pulse diagram.



Acknowledgment on the HMI device

In the "HMI acknowledgment tag" area, configure the tag or array tag as well as the bit number that the HMI device writes to the PLC after acknowledgment. Make sure when you use an array tag that it is not longer than 6 words.

To always create a signal change when setting an assigned acknowledgment bit of a discrete alarm that must be acknowledged, the HMI device will reset the acknowledgment bit assigned to the alarm as soon as it detects an alarm subject to acknowledgment and write the acknowledgment tag in the PLC. There will be a certain delay between detecting the message

and writing the acknowledgment tag in the PLC because the HMI device has to process the operations.

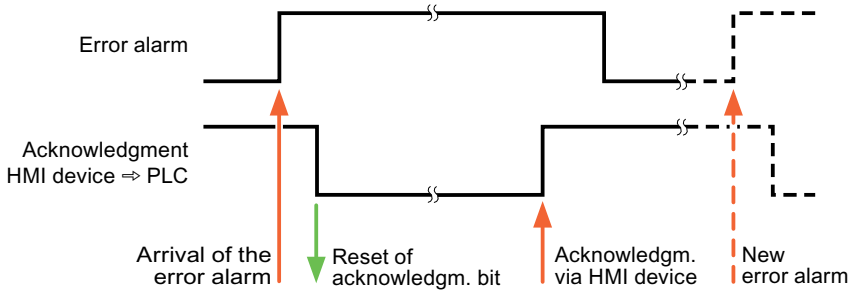
If a discrete alarm subject to acknowledgment is acknowledged by the HMI device, then the corresponding bit in the assigned acknowledgment tag will be set. The entire acknowledgment tag is then written to the PLC by the HMI device. This allows the PLC to recognize that a certain alarm message has been acknowledged at the HMI device.

Note

All alarm bits acknowledged since the last Runtime start will remain in the acknowledgment tag until a new incoming of the respective discrete alarms is detected.

This area should only be read by the PLC because the entire section of the HMI device will be overwritten once the next acknowledgment tag is written.

The figure below shows the pulse diagram.



See also

Configuring alarms (Page 126)

1.6.5 Performance features of communication (Basic Panels)

1.6.5.1 SIMATIC S7-1500 device dependency (Basic Panels)

Device dependency

If you use devices from an earlier version of the TIA Portal with TIA Portal V14, it may not be possible to configure connections to certain HMI devices.

Basic Panels V11.0

HMI devices	SIMATIC S7-1500
KP300 Basic	No
KP400 Basic	No
KTP400 Basic PN	No
KTP600 Basic DP	No
KTP600 Basic PN	No
KTP1000 Basic DP	No
KTP1000 Basic PN	No
TP1500 Basic PN	No

Basic Panels V12.0

HMI devices	SIMATIC S7-1500
KP300 Basic	Yes
KP400 Basic	Yes
KTP400 Basic PN	Yes
KTP600 Basic DP	Yes
KTP600 Basic PN	Yes
KTP1000 Basic DP	Yes
KTP1000 Basic PN	Yes
TP1500 Basic PN	Yes

Basic Panels V13.0

HMI devices	SIMATIC S7-1500
KTP400 Basic	Yes
KTP700 Basic	Yes
KTP900 Basic	Yes
KTP1200 Basic	Yes

Basic Panels V13.0.1

HMI devices	SIMATIC S7-1500
KTP400 Basic PN	Yes
KTP700 Basic PN	Yes
KTP700 Basic DP	Yes
KTP900 Basic PN	Yes
KTP1200 Basic PN	Yes
KTP1200 Basic DP	Yes

Basic Panels V14.0

HMI devices	SIMATIC S7-1500
KTP400 Basic PN	Yes
KTP700 Basic PN	Yes
KTP700 Basic DP	Yes
KTP900 Basic PN	Yes
KTP1200 Basic PN	Yes
KTP1200 Basic DP	Yes

See also

Communication with SIMATIC S7-1500 (Page 77)
Valid data types for SIMATIC S7 1500 (Page 130)

1.6.5.2 Valid data types for SIMATIC S7 1500 (Basic Panels)

Valid data types for connections with SIMATIC S7 1500

The table lists the data types that can be used when configuring tags and area pointers.

Data type	Length	
BOOL	1 bit	
BYTE	1 byte	
WORD	2 bytes	
DWORD	4 bytes	
CHAR	1 byte	
WCHAR	2 bytes	RT Professional
Array	(Element number * data type length) bytes [1]	
INT	2 bytes	
DINT	4 bytes	
REAL	4 bytes	
TIME	4 bytes	
DATE	2 bytes	
TIME_OF_DAY	4 bytes	
S5TIME	2 bytes	
COUNTER	2 bytes	
TIMER	2 bytes	
DATE_AND_TIME	8 bytes	
STRING	(2+n) bytes, n = 0 to 254	
WSTRING	(4+2*n) bytes, n = 0 to 254	Basic Panels
	(4+2*n) bytes, n = 0 to 4094	Panels, RT Advanced
	(4+2*n) bytes, n = 0 to 65534	RT Professional

Data type	Length
DTL	12 bytes
LDT	8 bytes
LINT	8 bytes
LREAL	8 bytes
LTIME	8 bytes
LTIME_OF_DAY	8 bytes
SINT	1 byte
UDINT	4 bytes
UINT	2 bytes
ULINT	8 bytes
USINT	1 byte

[1] Example of "Length of an array": The length is 400 bytes (100 * 4) for 100 elements of the data type REAL

See also

Communication with SIMATIC S7-1500 (Page 77)

SIMATIC S7-1500 device dependency (Page 128)

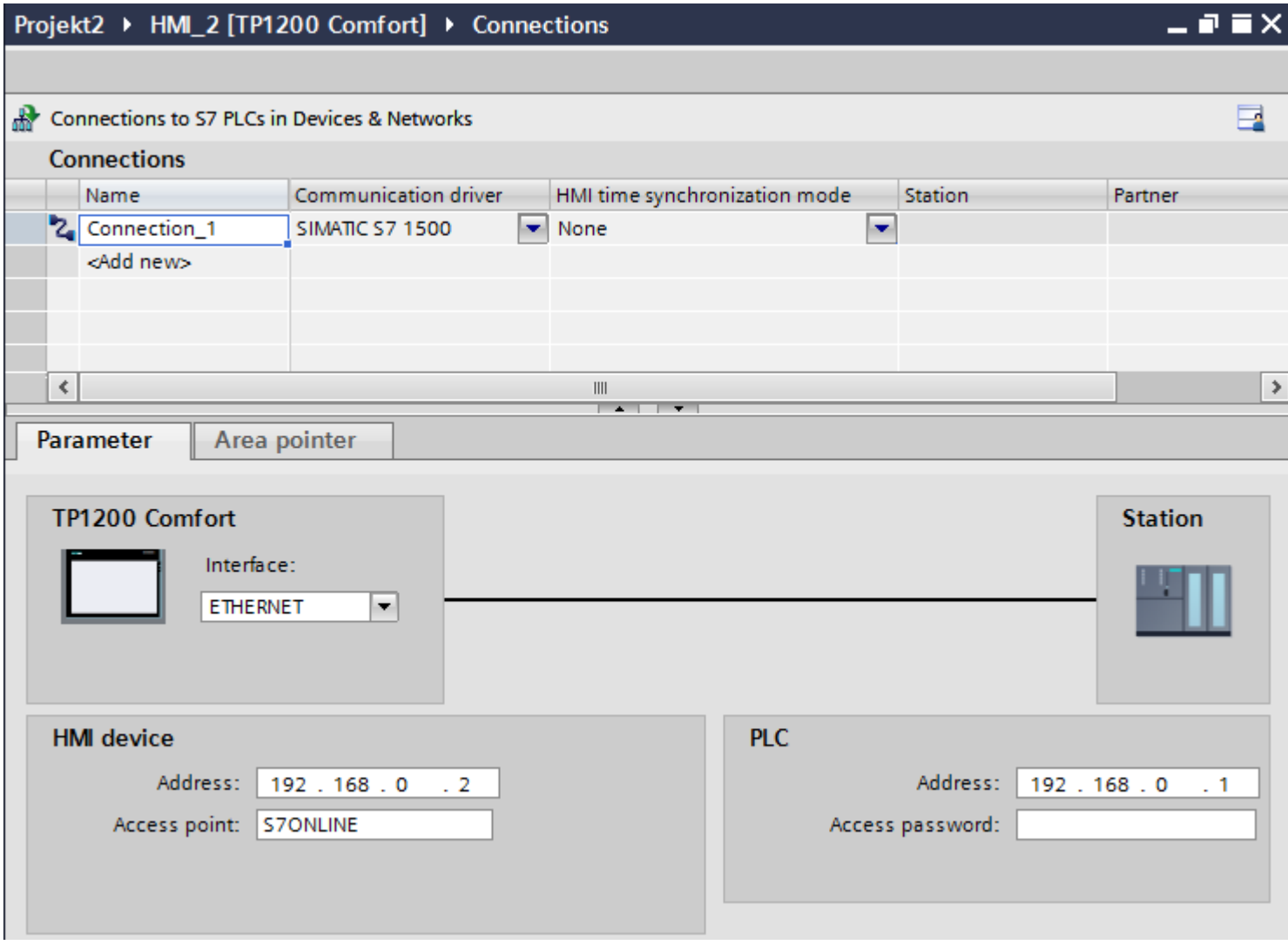
1.6.6 Configuring connections in the "Connections" editor (Basic Panels)

1.6.6.1 Parameters for the connection (SIMATIC S7 1500) (Basic Panels)

Parameters to be set

To set the connection parameters, such as addresses and profiles, click the connection that you have created in the "Connections" editor.

The communication partners are displayed schematically in the Inspector window under "Parameters". The "HMI device", "Network", and "PLC" areas are available for assigning parameters according to the interface used.



See also

Communication with SIMATIC S7-1500 (Page 77)

1.6.6.2 Ethernet parameters (Basic Panels)

Parameters for the HMI device

You set the parameters for the HMI device in the network under "HMI device".. The changes are not transferred automatically to the HMI device. You must change the settings in the Control Panel of the HMI device.

- "Interface"
  If you are directly connected to the HMI device during configuration, you can set up the IP address of the HMI device in WinCC.

  Note

  The IP address in the Control Panel will be overwritten upon subsequent loading if you have already set up the IP address in the HMI device control panel.

  The IP address already set up in the Control Panel will be retained upon subsequent loading if you activate "Set IP address using a different method".

  The IP address is transferred to the HMI device during project transfer.
  To set up the IP address of the HMI device:

  – Click the HMI device.
  – Open the "Device configuration" editor.
  – Click the Ethernet interface.
  – Assign the IP address in the inspector window under:
    "General > PROFINET interface > Ethernet addresses"

- "Address"
  You assign the IP address of the HMI device in the "Address" area.
  When you transfer the WinCC project to the HMI device, this IP address is set up directly in the HMI device.

- "Access point"
  Specifies the access point for the PG/PC interface that can be used to reach the communication partner.

Parameters for the PLC

Under "PLC", you address the S7 module with which the HMI device will exchange data. Assign a name for the connection for each communication partner.

- "Address"
  Under "Address", set the IP address of the S7 module to which the HMI device is connected.
- "Access password"
  Enter a password in the "Access password" field. This password must match the one you saved to the PLC.

Note

You only need a password if you have set "Complete protection" at the PLC.

No connection is set up to the PLC if the "Complete protection" security level is stored on the PLC and you do not enter a password.

See also

Communication with SIMATIC S7-1500 (Page 77)

1.6.6.3 PROFIBUS parameters (Basic Panels)

Parameters for the HMI device

You assign the parameters for the HMI device in the network once under "HMI device". The change applies to each communication partner.

- "Type"
  Specifies the physical connection used.
- "Interface"
  For "Interface", you select the HMI device interface via which the HMI device is connected to the PROFIBUS network.
- "Baud rate"
  For "Baud rate", you set the transmission speed of the data in the network. The baud rate is determined by the slowest HMI device connected to the network. The setting must be identical throughout the network.

Note

If you set a baud rate of 1.5 Mbaud for OP 73 or OP 77A, the highest station address must be less than or equal to 63.

- "Address"
  You set the PROFIBUS address of the HMI device under "Address". The PROFIBUS address must be unique in the PROFIBUS network.

- "Only master on bus"
  Disables an additional safety feature against bus faults when the HMI device is connected to the network. A passive station (slave) can only send data if it is requested to do so by an active station (master).
  In S7-200, you must set an HMI device as the master.
- "Access point"
  The access point defines a logical device name through which the communication partner can be reached.

Parameters for the network

Under "Network", you set the parameters for the PROFIBUS network to which the HMI device is linked.

- "Profile"
  For "Profile", you select the network profile that is used in the network. In "Profile", set "DP", "Universal", or "Standard". The setting must be identical throughout the network.
- "Highest address"
  For "Highest station address", set the highest station address. The highest station address must be greater than or equal to the highest actual PROFIBUS address. The setting must be identical throughout the network.

  Note

  If you set a baud rate of 1.5 Mbaud in the OP 73 or the OP 77A, the highest station address must be less than or equal 63.

- "Number of masters"
  For "Number of masters", set the number of masters in the PROFIBUS network. This information is necessary to correctly calculate the bus parameters.

Parameters for the PLC

Under "PLC", you address the S7 module with which the HMI device will exchange data. Assign a name for the connection for each communication partner.

- "Address"
  For "Address", set the PROFIBUS address of the S7 module (CPU, FM, or CP) to which the HMI device is connected.
- "Access password"
  Enter a password in the "Access password" field. This password must match the one you saved to the PLC.

  Note

  You only need a password if you have set "Complete protection" at the PLC.

  No connection is set up to the PLC if the "Complete protection" security level is stored on the PLC and you do not enter a password.

See also

Communication with SIMATIC S7-1500 (Page 77)

1.6.7 Configuring time synchronization (Basic Panels)

1.6.7.1 Time synchronization restrictions (Basic Panels)

Approved HMI devices

You can configure the time synchronization between a SIMATIC S7-1200 or SIMATIC S7-1500 and an HMI device with the following HMI devices:

Device	Operating system
Basic Panels	-
TP177 4"	Windows CE 5.0
Multi Panel 177	Windows CE 5.0
Multi Panel 277	Windows CE 5.0
Multi Panel 377	Windows CE 5.0
Mobile Panel 277	Windows CE 5.0
Mobile 277 IWLAN V2	Windows CE 5.0
Comfort Panels	Windows CE 6.0
PC systems with WinCC RT Advanced	Microsoft Windows

Configuration restrictions

- If an HMI device has several connections to SIMATIC S7-1200 or SIMATIC S7-1500, you can only configure one connection as "slave".
- If you have enabled time synchronization for the HMI device as "slave", you can no longer use the global area pointer "Date/time PLC".
- An HMI device can only request the time from a PLC with "Complete protection" security type configuration if the correct "Access password" is configured.
  Configure the "Access password" for communication with a PLC with "Complete protection" security type in the "Connections" editor of the HMI device.
  This "Access password" must match the password configured on the PLC. The PLC password is assigned in the PLC properties at: "General > Security"
- Basic Panels can only be configured as "Slave".
- If you use Basic Panels for the configuration, it is not possible to use time synchronization via NTP and the "Date/time PLC" area pointer simultaneously.
- Time synchronization with SIMATIC S7-1200 (V1.0) controllers is not possible.

- Time synchronization between the HMI device TP177 4" and SIMATIC S7-1200 (V4.0) controllers is not possible.
- Time synchronization between the HMI device TP177 4" and SIMATIC S7-1500 is not possible.

See also

Time synchronization (Page 140)

Configuring time synchronization for integrated connections (Page 137)

Configuring time synchronization for non-integrated connections (Page 138)

1.6.7.2 Configuring time synchronization for integrated connections

Introduction

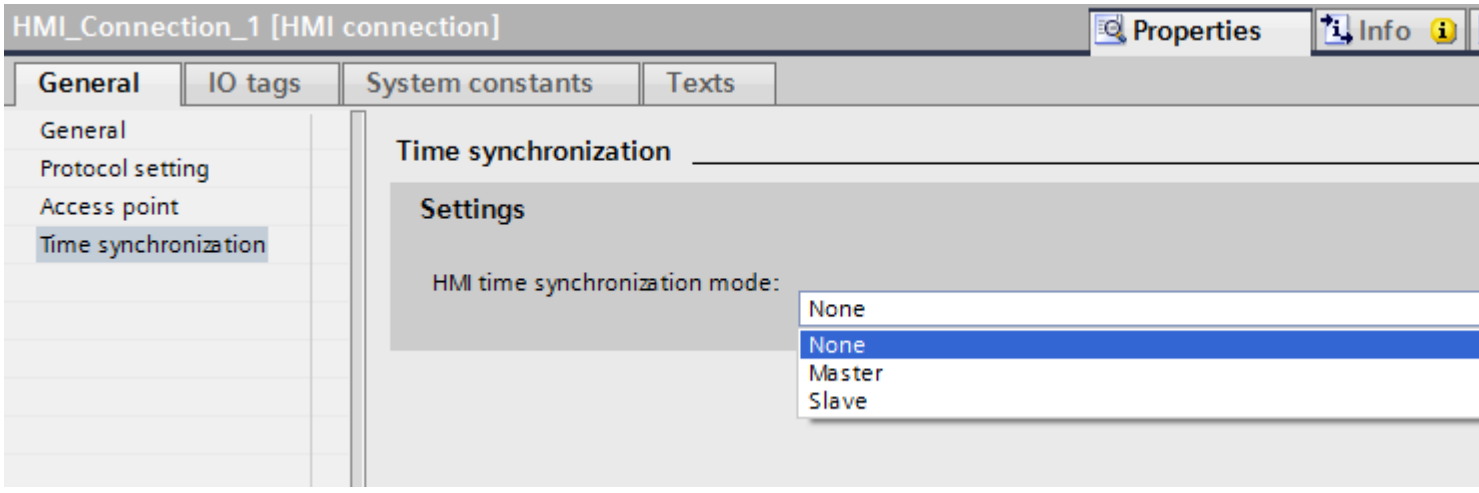
You configure time synchronization for an integrated connection in the "Devices & Networks" editor.

Requirements

- An HMI connection between an HMI device and a SIMATIC S7 1200 or SIMATIC S7 1500 has been configured.
- The HMI device must support the "time synchronization" function.
- The "Devices & Networks" editor is open.

Procedure

1. Click the line of the HMI connection in the "Devices & networks" editor.
2. Select the following in the inspector window under "General > Time synchronization > Settings":
   - None: No time synchronization is used.
   - Master: The HMI device sets the time.
   - Slave: The PLC sets the time.



See also

Time synchronization (Page 140)

Time synchronization restrictions (Page 136)

Configuring time synchronization for non-integrated connections (Page 138)

1.6.7.3 Configuring time synchronization for non-integrated connections (Basic Panels)

Introduction

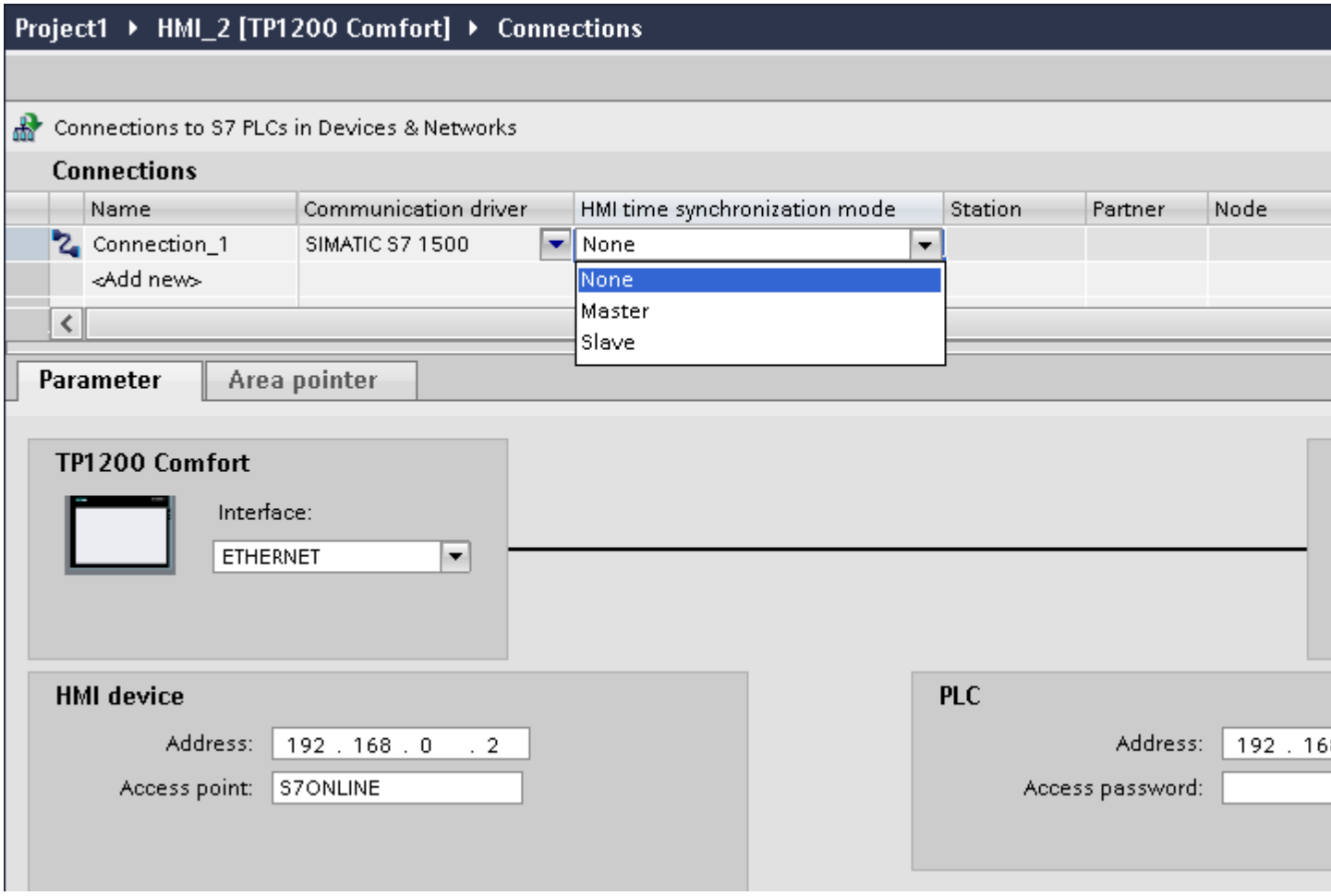
You configure time synchronization for a non-integrated connection in the "Connections" editor.

Requirements

- An HMI device which supports the "time synchronization" function has been created.
- "Connections" editor is open.

Procedure

1. Double-click "<Add>".
2. In the "Communication drivers" column, select the "SIMATIC S7 1500" PLC.
3. Select the following in the "HMI time synchronization mode" column:
   - None: No time synchronization is used.
   - Master: The HMI device sets the time.
   - Slave: The PLC sets the time.



See also

Time synchronization (Page 140)

Time synchronization restrictions (Page 136)

Configuring time synchronization for integrated connections (Page 137)

1.6.7.4 Time synchronization (Basic Panels)

Introduction

To have the same time of day throughout the plant, you can synchronize the time on various plant components using time synchronization. WinCC time synchronization is operated as a master-slave system.

One system component must be a clock for all components of a plant to work with identical time. The component functioning as the clock is referred to as the time master. The components that receive the time are time slaves.

Properties of time synchronization

- The HMI device can define the time as master or can accept the time of the PLC as slave.
- In "master mode", the time is synchronized at each connection setup.
- In "slave mode", the time is synchronized at each connection setup and than at cyclic intervals of 10 minutes.
- The first time synchronization is performed on the HMI device immediately after the start of runtime.
- Time synchronization is only performed on the HMI device during operation of runtime.

See also

Time synchronization restrictions (Page 136)

Configuring time synchronization for integrated connections (Page 137)

Configuring time synchronization for non-integrated connections (Page 138)

1.7 Communicating with SIMATIC S7 1200 (Basic Panels)

1.7.1 Communication with SIMATIC S7-1200 (Basic Panels)

Introduction

This section describes communication between the HMI device and the SIMATIC S7-1200 controller.

You can configure the following communication channels for the SIMATIC S7-1200 controller:

- PROFINET
- PROFIBUS

HMI connection for communication

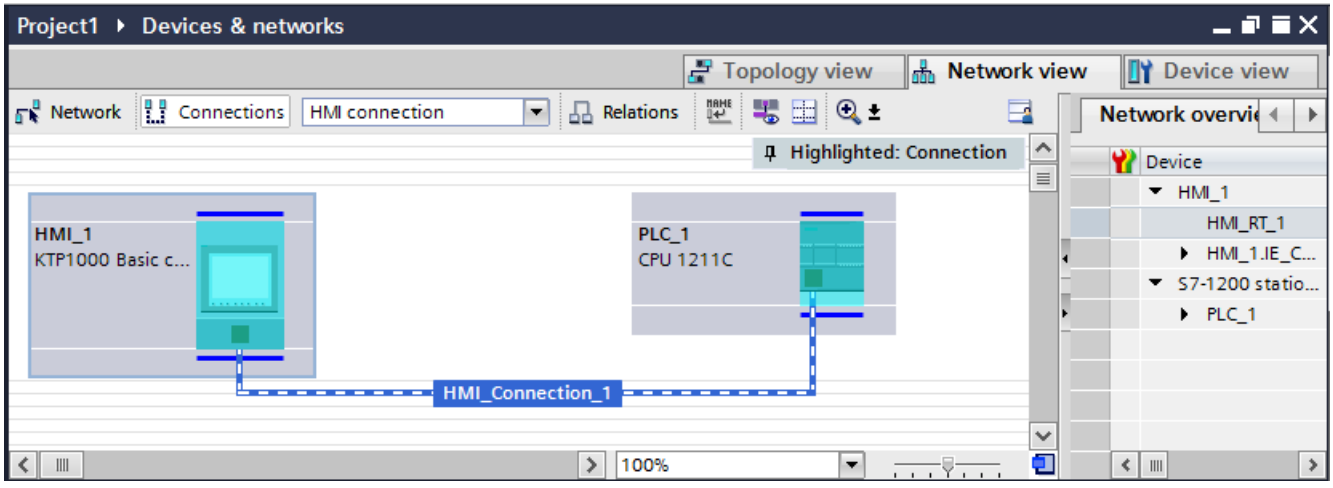
Connections between the HMI device and SIMATIC S7-1200 are configured in the "Devices & Networks" editor. If you have configured an HMI device with a serial port, you must configure a PROFIBUS-capable communication module for the SIMATIC S7-1200.

1.7.2 Communication via PROFINET (Basic Panels)

1.7.2.1 Communication via PROFINET (Basic Panels)

HMI connections via PROFINET

If you have inserted an HMI device and a SIMATIC S7 1200 into the project, you interconnect the two PROFINET interfaces in the "Devices & Networks" editor.



You can also connect multiple HMI devices to one SIMATIC S7 1200 and multiple SIMATIC S7 1200s to one HMI device. The maximum number of communication partners that you can connect to an HMI device is dependent on the HMI device used.

Additional information is available in the documentation for the respective HMI device.

HMI connection in the "Devices & Networks" editor

You configure the HMI connection between the PLC and the HMI device via PROFINET in the "Devices & Networks" editor.

Connection in the "Connections" editor

Alternatively, you configure the connection between the PLC and HMI device via PROFINET in the "Connections" editor of the HMI device.

See also

Communication with SIMATIC S7-1200 (Page 140)

Configuring an HMI connection via PROFINET (Page 142)

1.7.2.2 Configuring an HMI connection via PROFINET (Basic Panels)

Introduction

You configure an HMI connection between HMI devices and a SIMATIC S7 1200 via PROFINET or Ethernet in the "Devices & Networks" editor.

⚠ 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

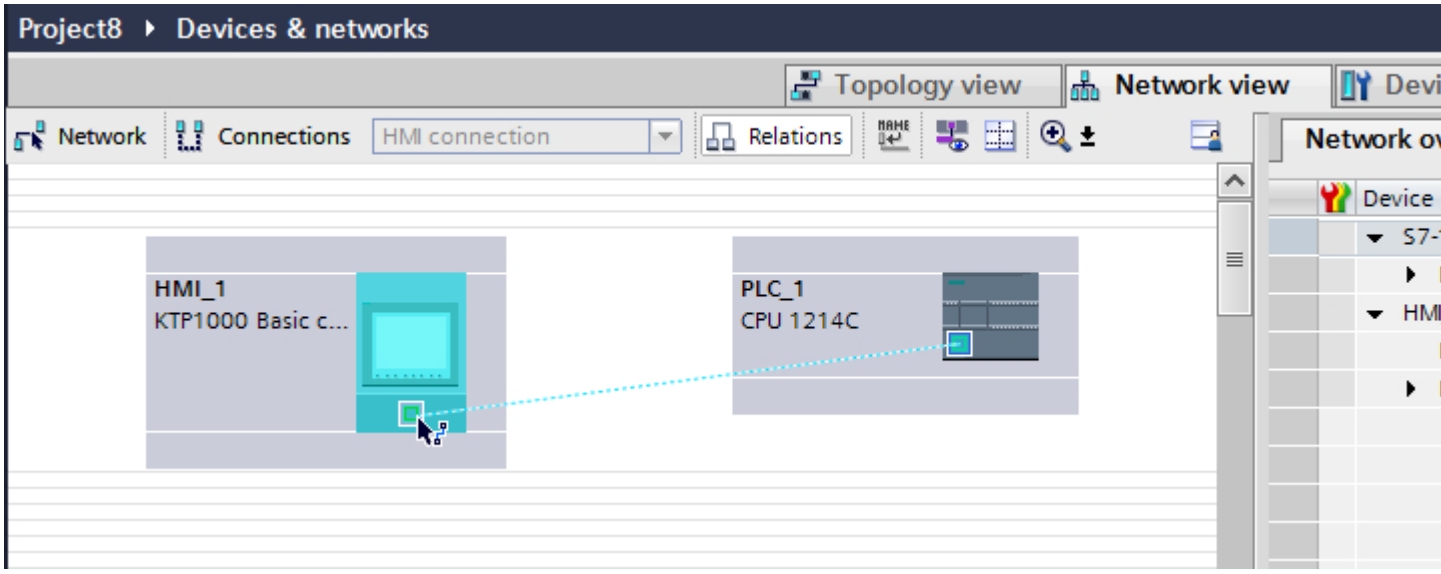
The following communication partners are created in the "Devices & Networks" editor:

- SIMATIC S7 1200
- HMI device with PROFINET or Ethernet interface

Procedure

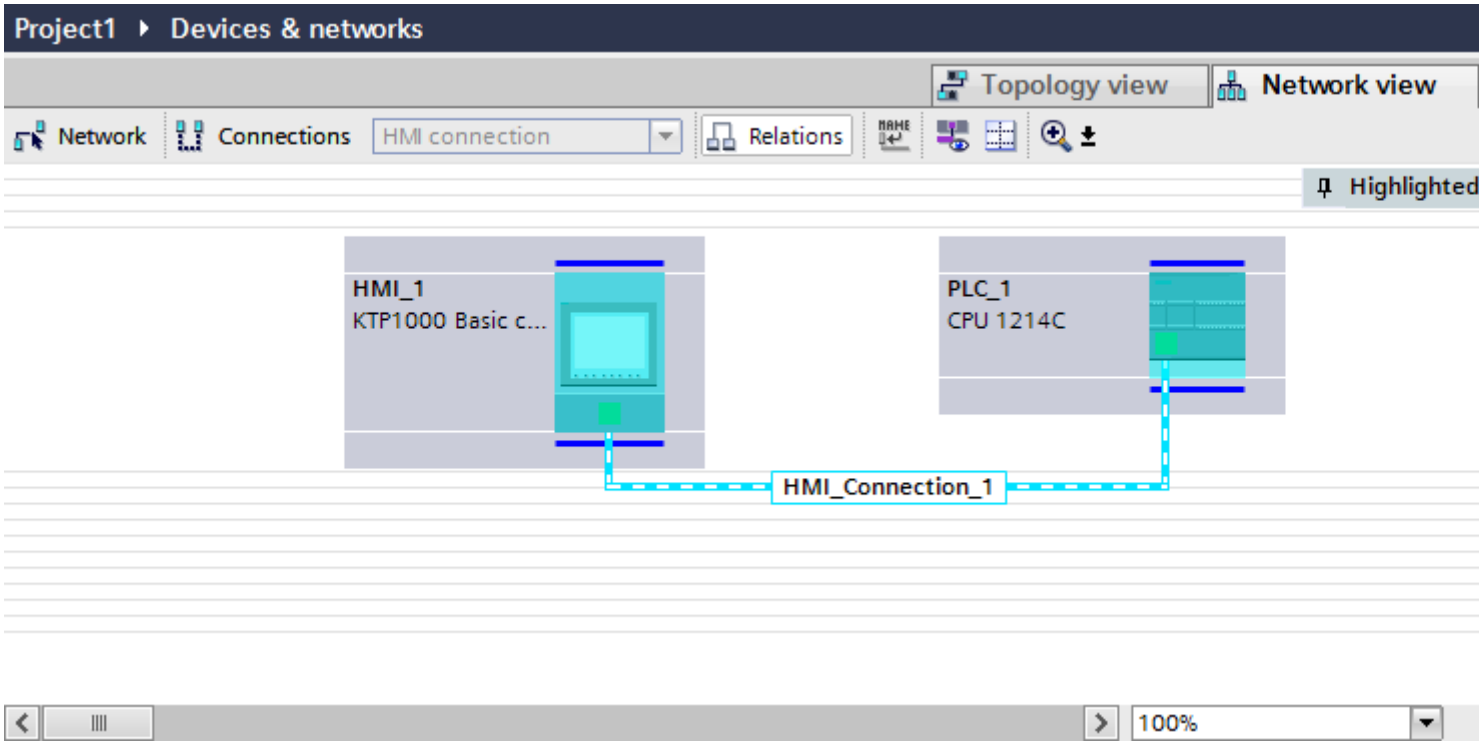
1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed graphically in the network view.
2. Click the "Connections" button and select "HMI connection" for the connection type.
   The devices available for connection are highlighted in color.

3. Click the PROFINET interface of the PLC and use a drag-and-drop operation to draw a connection to the PROFINET or Ethernet interface of the HMI device.



4. Click the connecting line.

5. Click "Highlight HMI connection" and select the HMI connection.



The connection is displayed graphically in the Inspector window.

6. Click the communication partners in the "Network view" and change the PROFINET parameters in the Inspector window according to the requirements of your project. For additional information on this, see section "PROFINET parameters (Page 145)".

Note

The created HMI connection is also shown in the tabular area of the editor on 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 a SIMATIC S7 1200. The IP address and subnet mask connection parameters are configured.

See also

Communication with SIMATIC S7-1200 (Page 140)

Communication via PROFINET (Page 141)

PROFINET parameters (Page 145)

1.7.2.3 PROFINET parameters (Basic Panels)

PROFINET parameters for the HMI connection (Basic Panels)

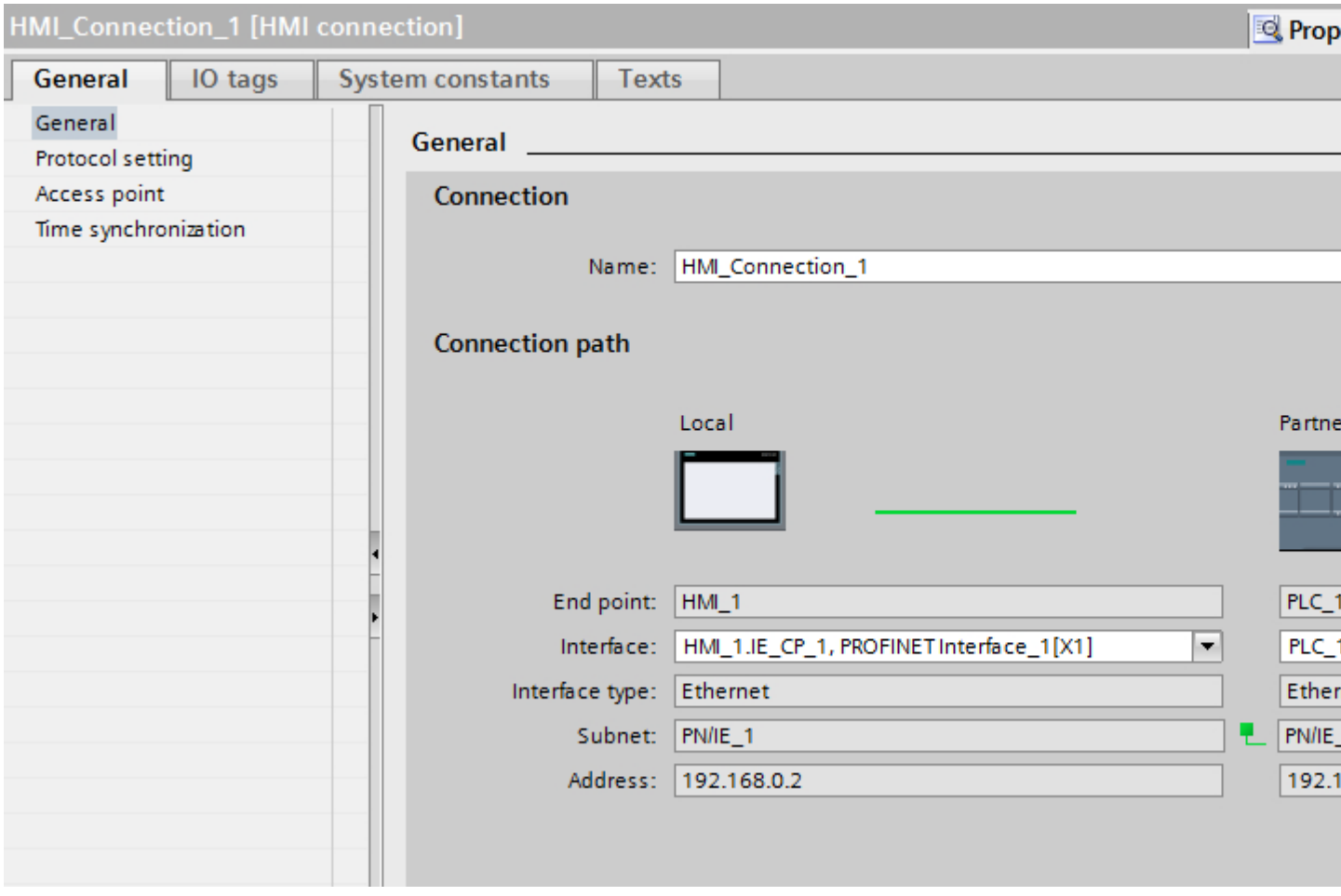
PROFINET parameters for the HMI connection

An overview of the configured HMI connection parameters can be found in the properties for the HMI connection.

Only limited changes are possible in this Inspector window.

Displaying and editing HMI connection parameters

1. Click the HMI connection in the "Devices & Networks" editor.
2. Change the parameters of the HMI connection in the Inspector window under "Properties > General > General".



Connection

The "Connection" area displays the HMI connection created for communication between the devices.

You can edit the name of the HMI connection in this area.

"Connection path"

The communication partners of the selected HMI connection and the associated PROFINET parameters are displayed in the "Connection path" area. Some of the areas displayed cannot be edited in this dialog.

- "End point"
  Displays the device name. This area cannot be edited.
- "Interface"
  Displays the selected interface of the device. You can choose between several interfaces, depending on the device.
- "Interface type"
  Displays the selected interface type. This area cannot be edited.
- "Subnet"
  Displays the selected subnet. This area cannot be edited.
- "Address"
  Displays the selected IP address of the device. This area cannot be edited.
- "Find connection path" button
  Enables the subsequent specification of connections.

See also

Communication via PROFINET (Page 141)

Communication with SIMATIC S7-1200 (Page 140)

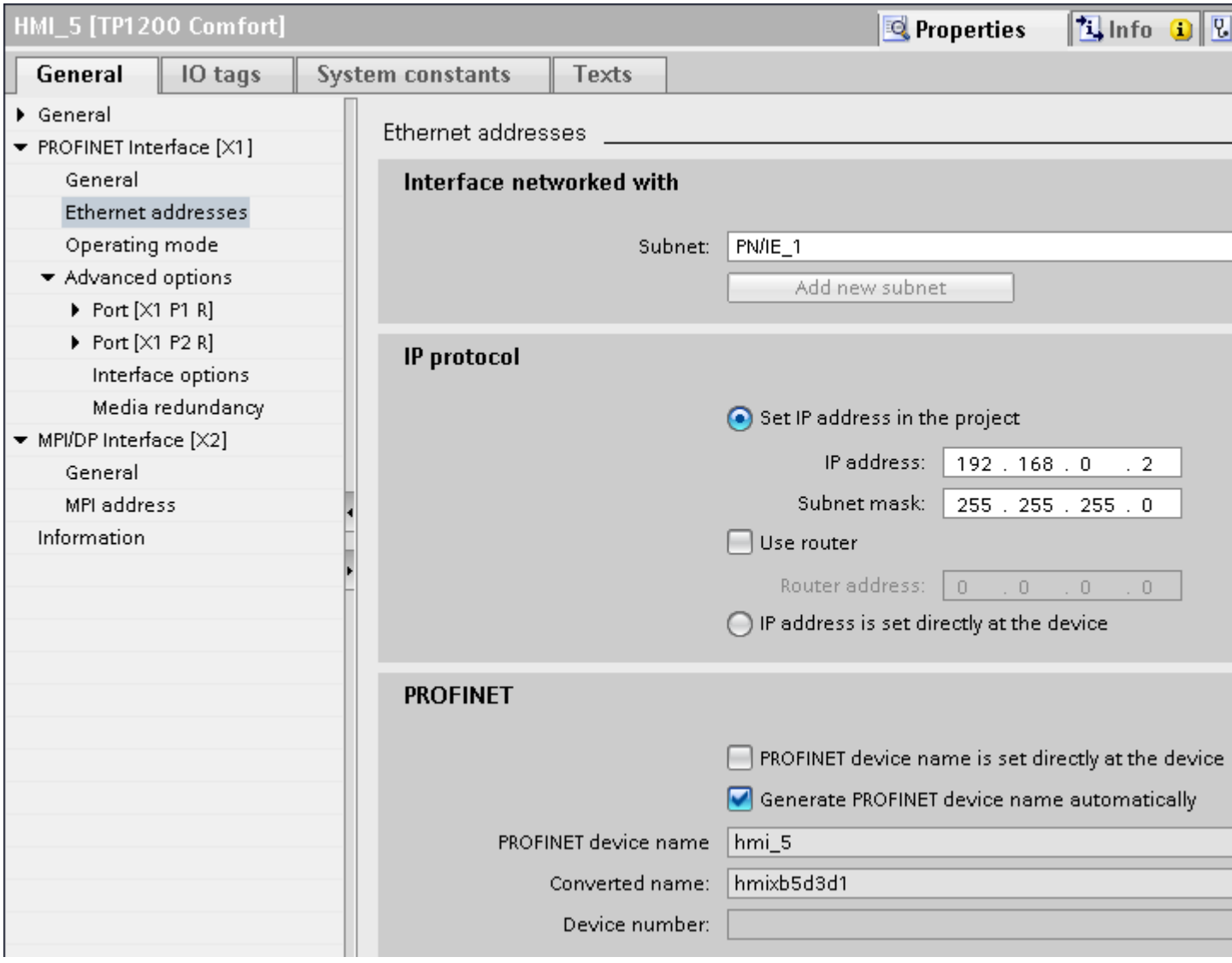
PROFINET parameters for the HMI device (Basic Panels)

PROFINET parameters for the HMI device

An overview of the configured HMI device parameters can be found in the properties for the HMI device.

Displaying and changing PROFINET parameters of the HMI device

1. Click the HMI device in the "Devices & Networks" editor.
2. Change the parameters of the HMI device in the Inspector window under "Properties > General > General".



"Interface networked with"

In the "Interface networked with" area, select the subnet of the HMI connection via which the HMI device is connected to the network. You use the "Add new subnet" button to create a new subnet.

"IP protocol"

- "Set IP address in the project"
  When you transfer the WinCC project to the HMI device, this IP address is set up directly in the HMI device.

  Note

  The device is automatically restarted in the case of HMI devices with the Windows CE 3.0 operating system.

  HMI devices with Windows CE 3.0:
  - OP 77B
  - TP 177B color PN/DP
  - TP 177B mono DP
  - OP 177B color PN/DP
  - OP 177B mono DP
  - Mobile Panel 177 PN
  - Mobile Panel 177 DP
  - TP 277 6"
  - OP 277 6"

- "Subnet mask"
  You assign data of the subnet mask in the "Subnet mask" area.
- "Use IP router"
  If you are using an IP router, select "Use IP router" and enter the router address in the "Router address" field.
- "Set IP address using a different method"
  If the function "Set IP address using a different method" is activated, the IP address is not taken from the project. You have to enter the IP address directly in the Control Panel of the HMI device.

See also

Communication via PROFINET (Page 141)

Communication with SIMATIC S7-1200 (Page 140)

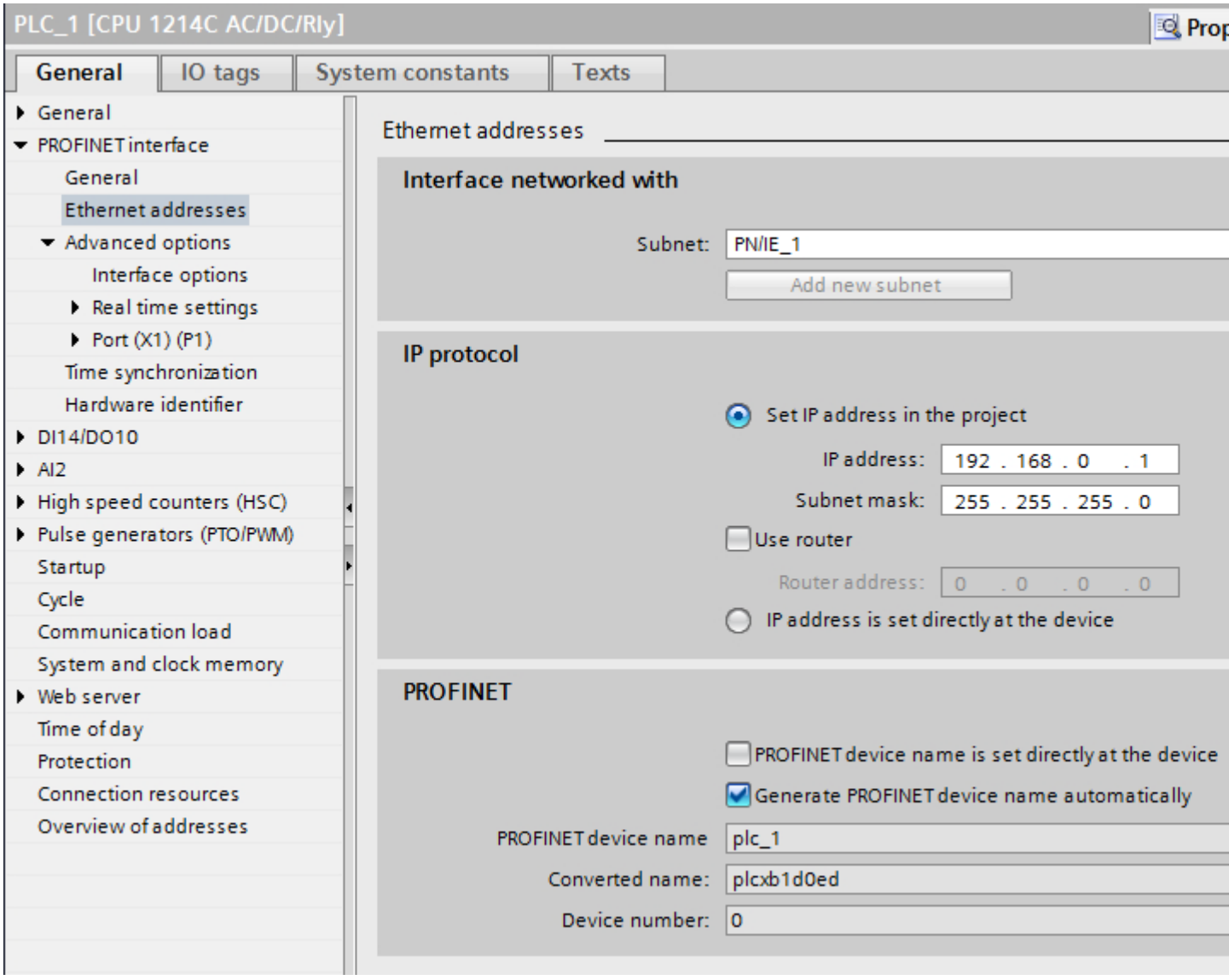
PROFINET parameters for the PLC (Basic Panels)

PROFINET parameters for the PLC

An overview of the configured parameters can be found in the properties for the PLC.

Displaying and changing PROFINET parameters of the PLC

1. Click the PLC in the "Devices & Networks" editor.
2. Change the parameters of the PLC in the Inspector window under "Properties > General > General".



"Interface networked with"

In the "Subnet" area, select the subnet of the HMI connection via which the PLC is connected to the network. You use the "Add new subnet" button to create a new subnet.

"IP protocol"

- "Interface type"
  Depending on the HMI device type, you have various interfaces to choose from.
- "IP address"
  You assign the IP address of the HMI device in the "IP address" area.
- "Subnet mask"
  You assign data of the subnet mask in the "Subnet mask" area.
  If you are using an IP router, select "Use IP router" and enter the router address in the field.

See also

Communication via PROFINET (Page 141)

Communication with SIMATIC S7-1200 (Page 140)

Configuring Industrial Ethernet (Basic Panels)

Rules for the network configuration

The Ethernet interfaces of the devices have a default IP address that you can change.

IP address

The IP parameters are visible if the communication-capable devices support the TCP/IP protocol.

The IP address consists of 4 decimal figures in the range of 0 to 255. The decimal figures are separated from one another by a dot.

Example: 140.80.0.2

The IP address consists of the following:

- The address of the (sub) net
- The address of the node (generally also called host or network node)

Subnet mask

The subnet mask splits these two addresses. It determines which part of the IP address addresses the network and which part of the IP address addresses the node.

The set bits of the subnet mask determine the network part of the IP address.

Example:

Subnet mask: 255.255.0.0 = 11111111.11111111.00000000.00000000

In the example given for the above IP address, the subnet mask shown here has the following meaning:

The first 2 bytes of the IP address identify the subnet - i.e. 140.80. The last two bytes address the node, thus 0.2.

It is generally true that:

- The network address results from AND linking the IP address and subnet mask.
- The node address results from AND NOT linking the IP address and subnet mask.

Relation between IP address and default subnet mask

An agreement exists relating to the assignment of IP address ranges and so-called "Default subnet masks". The first decimal number (from the left) in the IP address determines the structure of the default subnet mask. It determines the number of "1" values (binary) as follows:

IP address (decimal)	IP address (binary)	Address class	Default subnet mask
0 to 126	0xxxxxxx.xxxxxxxx....	A	255.0.0.0
128 to 191	10xxxxxx.xxxxxxxx...	B	255.255.0.0
192 to 223	110xxxxx.xxxxxxxx...	C	255.255.255.0

Note

Range of values for the first decimal point

A value of between 224 and 255 is also possible for the first decimal number of the IP address (address class D etc). This is, however, not recommended because there is no address check for these values.

Masking other subnets

You can use the subnet mask to add further structures and form "private" subnets for a subnet that is assigned one of the address classes A, B or C. This is done by setting other lower points of the subnet mask to "1". For each bit set to "1", the number of "private" networks doubles and the number of nodes they contain is halved. Externally, the network functions like an individual network as it did previously.

Example:

You have a subnet of address class B (e.g. IP address 129.80.xxx.xxx) and change the default subnet mask as follows:

Masks	Decimal	Binary
Default subnet mask	255.255.0.0	11111111.11111111.00000000.00000000
Subnet mask	255.255.128.0	11111111.11111111.10000000.00000000

Result:

All nodes with addresses between 129.80.001.xxx and 129.80.127.xxx are on one subnet, all nodes with addresses between 129.80.128.xxx and 129.80.255.xxx are on another subnet.

Router

The job of the routers is to connect the subnets. If an IP datagram is to be sent to another network, it first has to be conveyed to a router. To make this possible, in this case you have to enter the address of the router for each node in the subnet.

The IP address of a node in the subnet and the address of the router may only differ at the points at which there is a "0" in the subnet mask.

See also

Communication via PROFINET (Page 141)

Protection of communication (Basic Panels)

Security levels (Basic Panels)

If you want to protect the controller and HMI device communication, you can assign protection levels for the communication.

For a SIMATIC S7-1500 CPU, you can enter multiple passwords and thereby set up different access rights for various user groups.

The passwords are entered in a table, so that exactly one protection level is assigned to each password.

The effect of the password is given in the "Protection" column.

For the SIMATIC S7-1500 controller, several aspects need to be considered when setting protection levels.

For additional information on this, see:

Setting options for the protection (FW as of V4) (Page 153)

Example

When configuring the controller, you select the "Complete protection" protection level for a standard CPU (i.e., not an F-CPU).

Afterwards, you enter a separate password for every protection level above it in the table.

For users who do not know any of the passwords, the CPU is completely protected. Not even HMI access is possible.

For users who know one of the assigned passwords, the effect depends on the table row in which the password occurs:

- The password in row 1 (no protection) allows access as if the CPU were completely unprotected. Users who know this password have unrestricted access to the CPU.
- The password in row 2 (write protection) allows access as if the CPU were write-protected. Despite knowing the password, users who know this password only have read access to the CPU.
- The password in row 3 (read and write protection) allows access as if the CPU were read-protected and write-protected, so that only HMI access is possible for users who know this password.

See also

Setting options for the protection (FW as of V4) (Page 153)

Setting options for the protection level (FW V1 to V3) (Page 155)

Access password for the HMI connection (Page 156)

Setting options for the protection (FW as of V4) (Basic Panels)

Protection level

The following section describes how to use the various access levels of the S7-1200 CPUs as of V4.

S7-1200 CPUs provide various access levels to limit the access to specific functions.

The parameters for the access levels are assigned in a table. The green checkmarks in the columns to the right of the respective access level specify which operations are possible without knowing the password of this access level. If you want to use the functions of check boxes that are not selected, a password has to be entered.

NOTICE
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 memory card are not write- or read-protected. Use know-how protection to protect the code of blocks on the memory card.

Default characteristics

The default access level is "Full access (no protection)". Every user can read and change the hardware configuration and the blocks. A password is not set and is also not required for online access.

The access levels in detail

With an S7-1200 CPU, you can configure the following access levels:

- Full access (no protection): The hardware configuration and the blocks can be read and changed by all users.
- Read access: With this access level, only read access to the hardware configuration and the blocks is possible without entering a password - meaning that you can load the hardware configuration and blocks into the programming device. In addition, HMI access and access to diagnostics data is possible.
  You cannot load blocks or a hardware configuration into the CPU without entering the password. Moreover, writing test functions and firmware updates are **not** possible without a password.
- 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. In addition, the following is **not** possible without a password: Writing test functions, changing the operating state (RUN/STOP) and firmware updates.
- 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).
  Authorization with the password again provides you full access to the CPU.

Behavior of a password-protected module during operation

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

Validity is checked before the online function is executed. If password protection is in place, you are prompted to enter a password.

Example: The module was configured with read access and you want to execute the "Modify tags" function. This requires write access; therefore, the assigned password must be entered to execute the function.

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".

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

See also

Security levels (Page 152)

Setting options for the protection level (FW V1 to V3) (Page 155)

Access password for the HMI connection (Page 156)

Setting options for the protection level (FW V1 to V3) (Basic Panels)

Protection level

The following section describes how to use the various protection levels of the S7-1200 CPUs V1 to V3.

Effects of the protection level setting

You can choose between the following protection levels:

- No protection: This corresponds to the default behavior. You cannot enter a password. Read and write access is always permitted.
- Write protection: Only read-only access is possible. You cannot change any data on the CPU and cannot load any blocks or a configuration. HMI access and communication between CPUs are excluded from the write protection. Assignment of a password is required to select this protection level.
- Write/read protection: No write or read access is possible in the "Accessible devices" area or in the project for devices that are switched online. Only the CPU type and the identification data can be displayed in the project tree under "Accessible devices". Display of online information or blocks under "Accessible devices", or in the project for devices interconnected online, is possible.
  HMI access and communication between CPUs are excluded from the write protection. Assignment of a password is required to select this protection level.

Behavior of a password-protected CPU during operation

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

Validity is checked before the online function is executed. If password protection is in place, you are prompted to enter a password.

Example: The module was assigned write protection and you want to execute the "Modify tags" function. This requires write access; therefore, the assigned password must be entered to execute the function.

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 with a password.

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 authorization will also expire when the project is closed.

Note

You can not restrict functions for process control, monitoring, and communications.

Some functions are still protected due to their use as online data. RUN/STOP in the "Online Tools" task card or "Set the time" in the diagnostics and online editor is therefore write-protected.

See also

Security levels (Page 152)
Setting options for the protection (FW as of V4) (Page 153)
Access password for the HMI connection (Page 156)

Access password for the HMI connection (Basic Panels)

Introduction

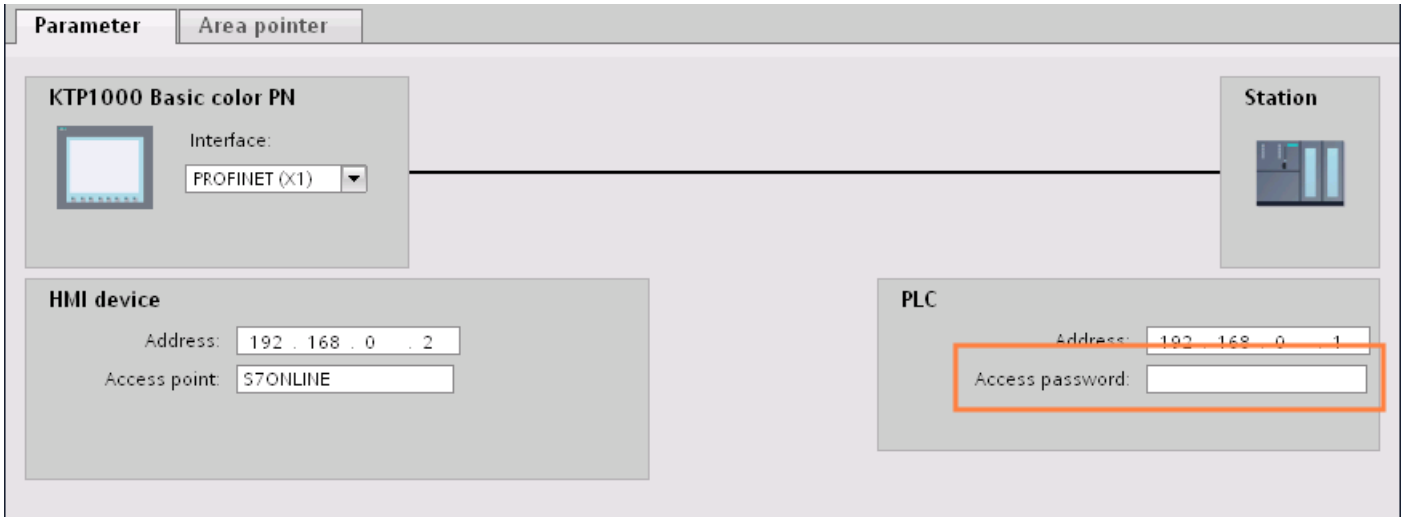
You secure access to a PLC by assigning a password.

You assign the password when you configure the password.

A password is mandatory as of protection level "Complete protection". Communication to the PLC is denied if an incorrect password or no password is entered.

Entering access password

You enter the password for the PLC in the "Connections" editor.



See also

Security levels (Page 152)
Setting options for the protection (FW as of V4) (Page 153)
Setting options for the protection level (FW V1 to V3) (Page 155)

1.7.2.4 Setting port options (Basic Panels)

Setting the port options (Basic Panels)

Changing connection settings for the PROFINET IO port

You can change the network settings for the PROFINET IO port as required. By default, the settings are made automatically. In normal situations, this guarantees problem-free communication.

Possible settings for transmission rate / duplex

Depending on the selected device, you can make the following settings for "Transmission rate / duplex":

- Automatic setting
  Recommended default setting of the port. The transmission settings are automatically "negotiated" with the peer port. The "Enable autonegotiation" option is also enabled as a default, in other words, you can use cross cables or patch cables for the connection.
- TP/ITP at x Mbps full duplex (half duplex)
  Setting of the transmission rate and the full duplex/half duplex mode. The effectiveness depends on the "Enable autonegotiation" setting:
  - Autonegotiation enabled
    You can use both cross cable and patch cable.
  - Autonegotiation disabled
    Make sure that you use the correct cable (patch cable or cross cable)! The port is also monitored with this setting.
- Deactivated
  Depending on the module type, the drop down list box can contain the "- Disabled -" option. This option, for example, allows you to prevent access to an unused port for security reasons. With this setting, diagnostic events are not generated.

"Monitor" option

This option enables or disables port diagnostics. Examples of port diagnostics: The link status is monitored, in other words, the diagnostics are generated during link-down and the system reserve is monitored in the case of fiber optic ports.

Option "Enable autonegotiation "

The autonegotiation setting can only be changed if a concrete medium (for example, TP 100 Mbps full duplex) is selected. Whether or not a concrete medium can be set depends on the properties of the module.

If autonegotiation is disabled, this causes the port to be permanently specified, as for example, is necessary for a prioritized startup of the IO device.

You must make sure the partner port has the same settings because with this option the operating parameters of the connected network are not detected and the data transmission rate and transmission mode can accordingly not be optimally set.

Note

When a local port is connected, STEP 7 makes the setting for the partner port if the partner port supports the setting. If the partner port does not accept the setting, an error message is generated.

See also

Wiring rules for disabled autonegotiation (Page 158)

Boundaries at the port (Page 159)

Wiring rules for disabled autonegotiation (Basic Panels)

Requirements

You have made the following settings for the port in question, for example, to accelerate the startup time of the IO device:

- Fixed transmission rate
- Autonegotiation incl. autocrossing disabled

The time for negotiating the transmission rate during startup has been saved.

If you have disabled autonegotiation, you must observe the wiring rules.

Wiring rules for disabled autonegotiation

PROFINET devices have the following two types of ports:

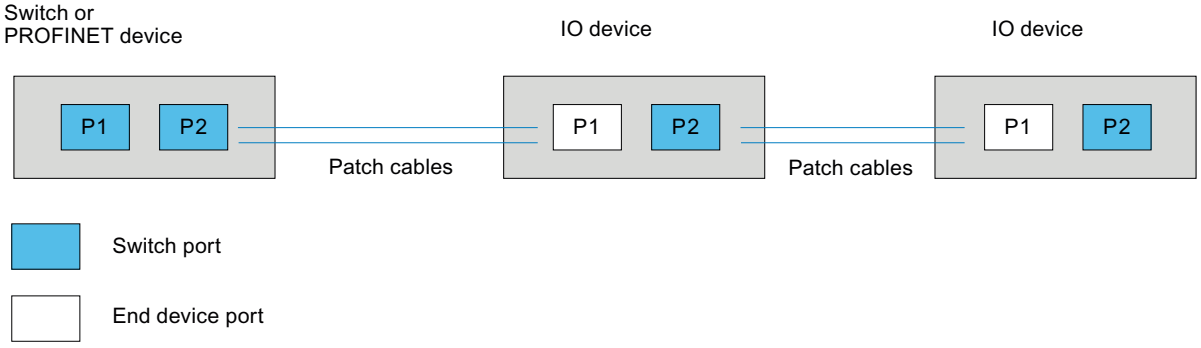
Type of port	PROFINET devices	Note
Switch port with crossed pin assignment	For IO devices: Port 2 For S7 CPUs with 2 ports: Ports 1 and 2	Crossed pin assignment means that the pin assignment for the ports for sending and receiving between the respective PROFINET devices is exchanged internally.
End device port with uncrossed pin assignment	For IO devices: Port 1 For S7 CPUs with one port: Port 1	-

Validity of the wiring rules

The cabling rules described in the following paragraph apply exclusively for the situation in which you have specified a fixed port setting.

Rules for cabling

You can connect several IO devices in line using a single cable type (patch cable). To do this, you connect port 2 of the IO device (distributed I/O) with port 1 of the next IO device. The following graphic gives an example with two IO devices.



See also

Setting the port options (Page 157)

Boundaries at the port (Page 159)

Boundaries at the port (Basic Panels)

Requirements

The respective device must support boundaries settings in order to work with boundaries. If the device for PROFINET does not support boundaries settings, the corresponding parameters are disabled.

Enable boundaries

"Boundaries" are limits for transmission of certain Ethernet frames. The following boundaries can be set at a port:

- "End of discovery of accessible devices"
  No forwarding of DCP frames to identify accessible devices. Devices downstream from this port cannot be reached by the project tree under "Accessible devices". Devices downstream from this port cannot be reached by the CPU.
- "End of topology discovery"
  LLDP frames (Link Layer Discovery Protocol) are not forwarded for topology detection.
- "End of sync domain"
  No forwarding of sync frames transmitted to synchronize nodes within a sync domain.
  If you operate, for example, a PROFINET device with more than two ports in a ring, you should prevent the sync frame from being fed into the ring by setting a sync boundary (at the ports not inside the ring).
  Additional example: If you want to use several sync domains, configure a sync domain boundary for the port connected to a PROFINET device from the other sync domain.

Restrictions

The following restrictions must be observed:

- The individual check boxes can only be used if the port supports the function in question.
- If a partner port has been determined for the port, the following check boxes cannot be used:
  - "End of discovery of accessible devices"
  - "End of topology discovery"

See also

Setting the port options (Page 157)

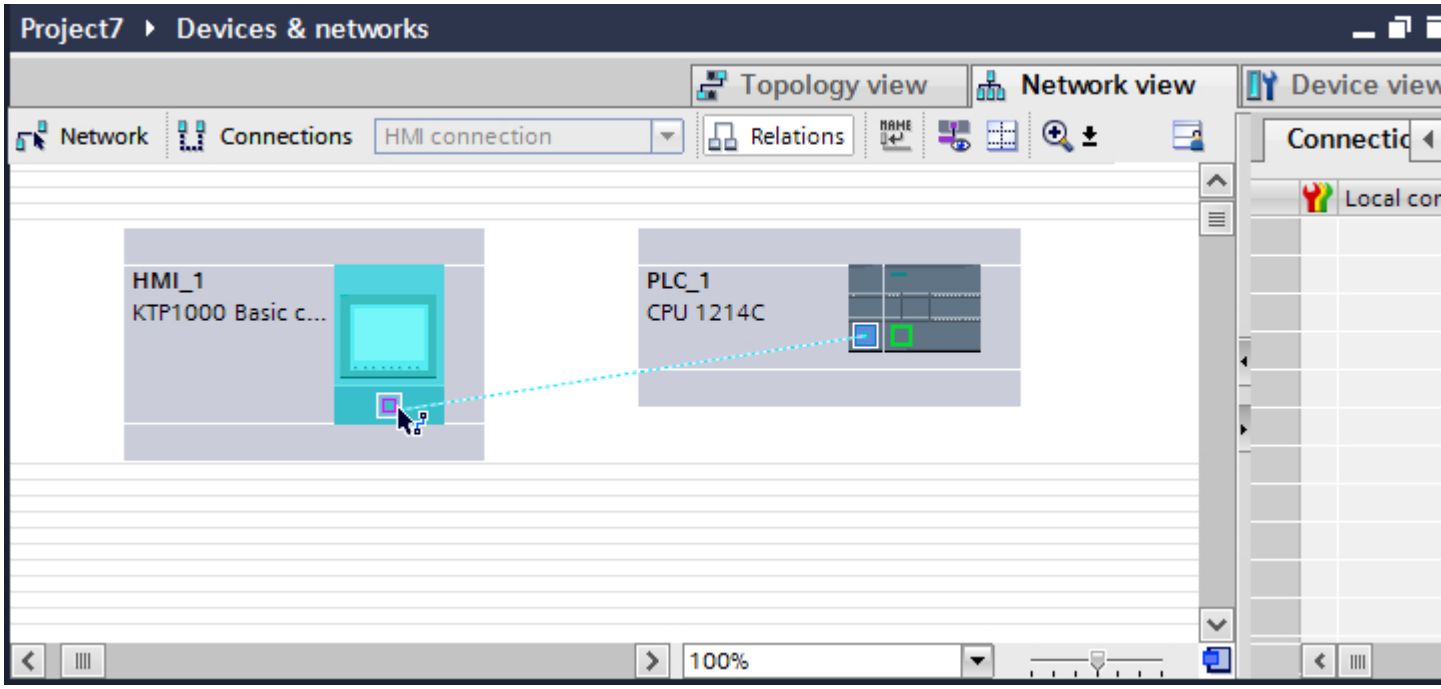
Wiring rules for disabled autonegotiation (Page 158)

1.7.3 Communication via PROFIBUS (Basic Panels)

1.7.3.1 Communication via PROFIBUS (Basic Panels)

HMI connections via PROFIBUS

If you want to connect a SIMATIC S7 1200 to a HMI device via PROFIBUS, you must configure a PROFIBUS-capable communication module to a slot of the controller first.



HMI connection in the "Devices & Networks" editor

You configure the HMI connection between the PLC and the HMI device via PROFIBUS in the "Devices & Networks" editor.

Connection in the "Connections" editor

Alternatively, you configure the connection between the PLC and HMI device in the "Connections" editor of the HMI device.

See also

Communication with SIMATIC S7-1200 (Page 140)

Configuring an HMI connection via PROFIBUS (Page 162)

1.7.3.2 Configuring an HMI connection via PROFIBUS (Basic Panels)

Introduction

You configure an HMI connection between HMI devices and a SIMATIC S7-1200 via PROFIBUS in the "Devices & Networks" editor.

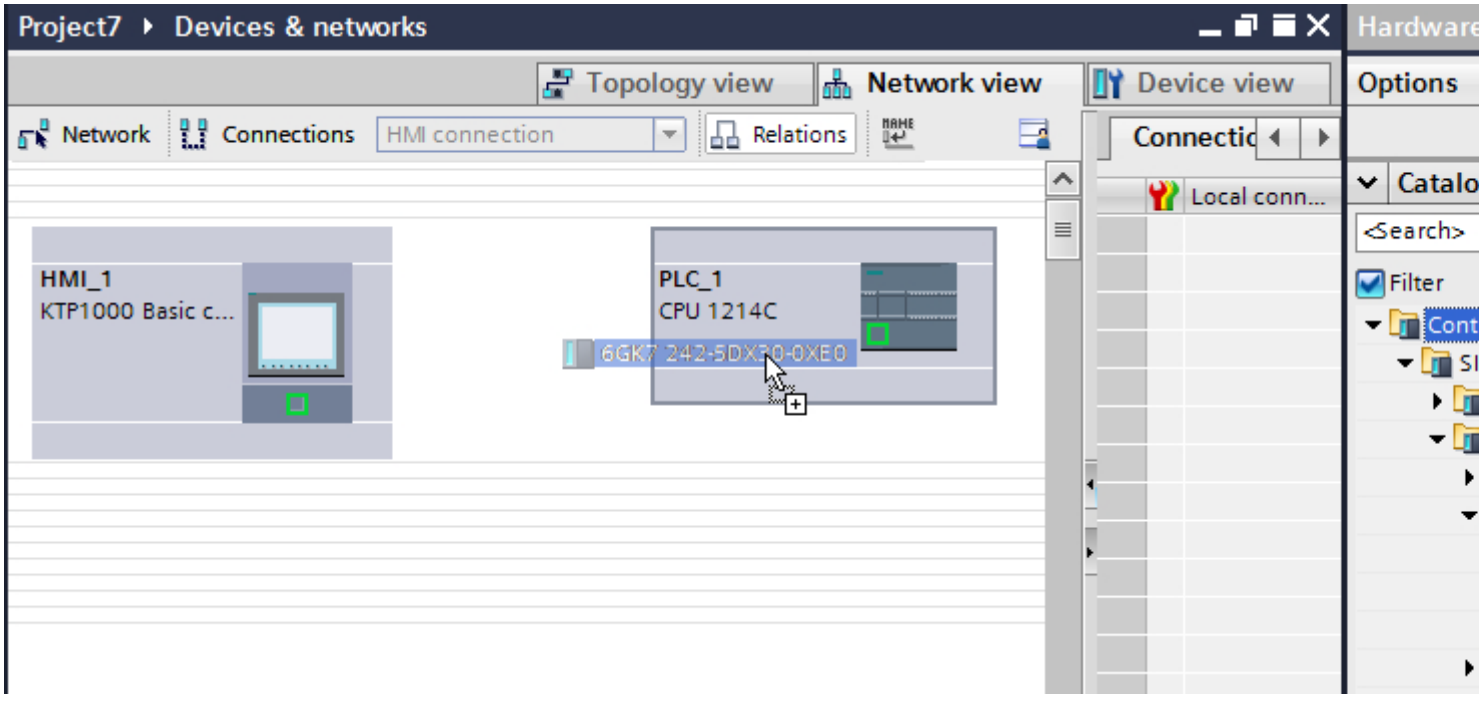
Requirements

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

- HMI device with MPI/DP interface
- SIMATIC S7-1200

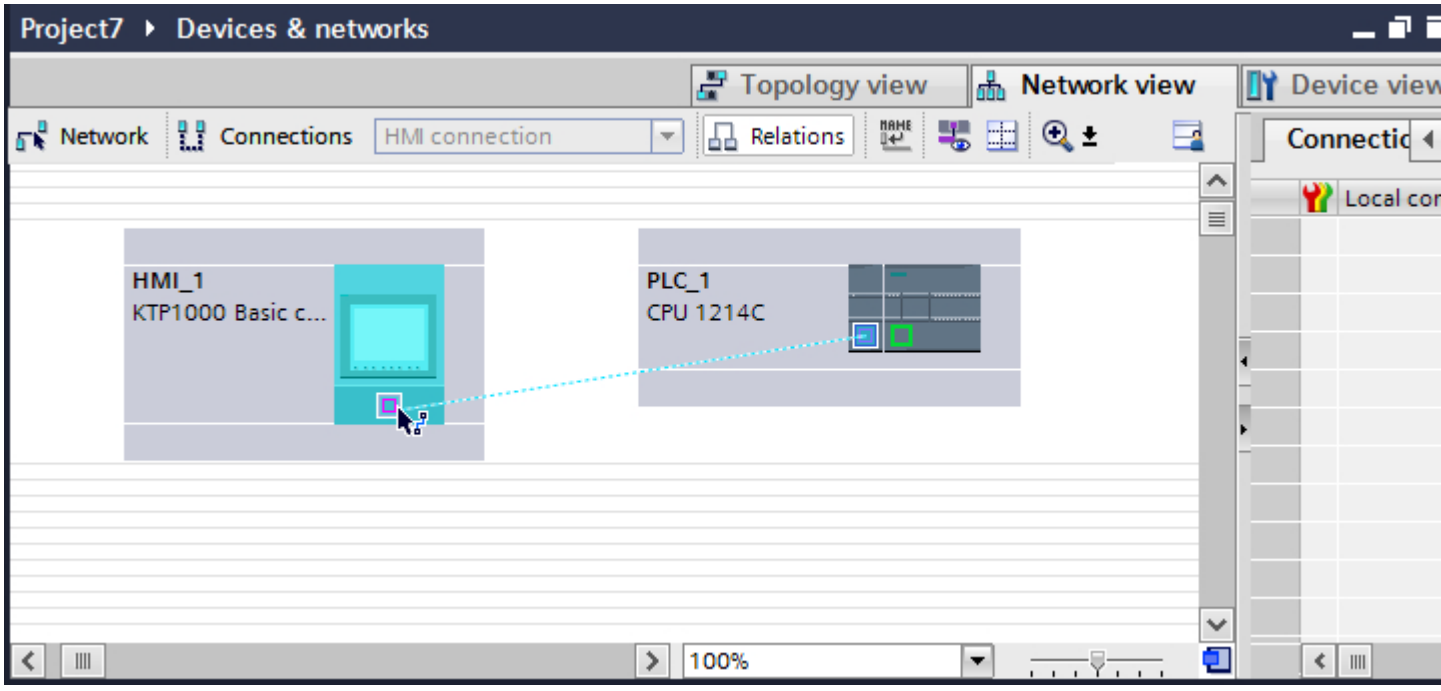
Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed graphically in the network view.
2. Click the "Connections" button.
   The devices available for connection are highlighted in color.
3. Use a drag-and-drop operation to move a PROFIBUS-capable communication module from the hardware catalog to the PLC.



4. Click the HMI device interface.
5. Select the "PROFIBUS" interface type in the Inspector window under "Properties > General > PROFIBUS address/ MPI address > Parameters".

6. Click the interface of the communication module and use a drag-and-drop operation to draw a connection to the HMI device.



7. Click the name of the connection.
   The connection is displayed graphically in the Inspector window.
8. Click "Highlight HMI connection" and select the HMI connection.
9. Click the communication partners in the "Network view" and change the PROFIBUS parameters in the Inspector window according to the requirements of your project.
   See the "PROFIBUS parameters (Page 164)" section for more detailed information.

Note

The created HMI connection is also shown in the tabular area of the editor on 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 an HMI connection between an HMI device and a SIMATIC S7-1200 via PROFIBUS.

See also

Communication with SIMATIC S7-1200 (Page 140)

Communication via PROFIBUS (Page 161)

PROFIBUS parameters (Page 164)

1.7.3.3 PROFIBUS parameters (Basic Panels)

PROFIBUS parameters for the HMI connection (Basic Panels)

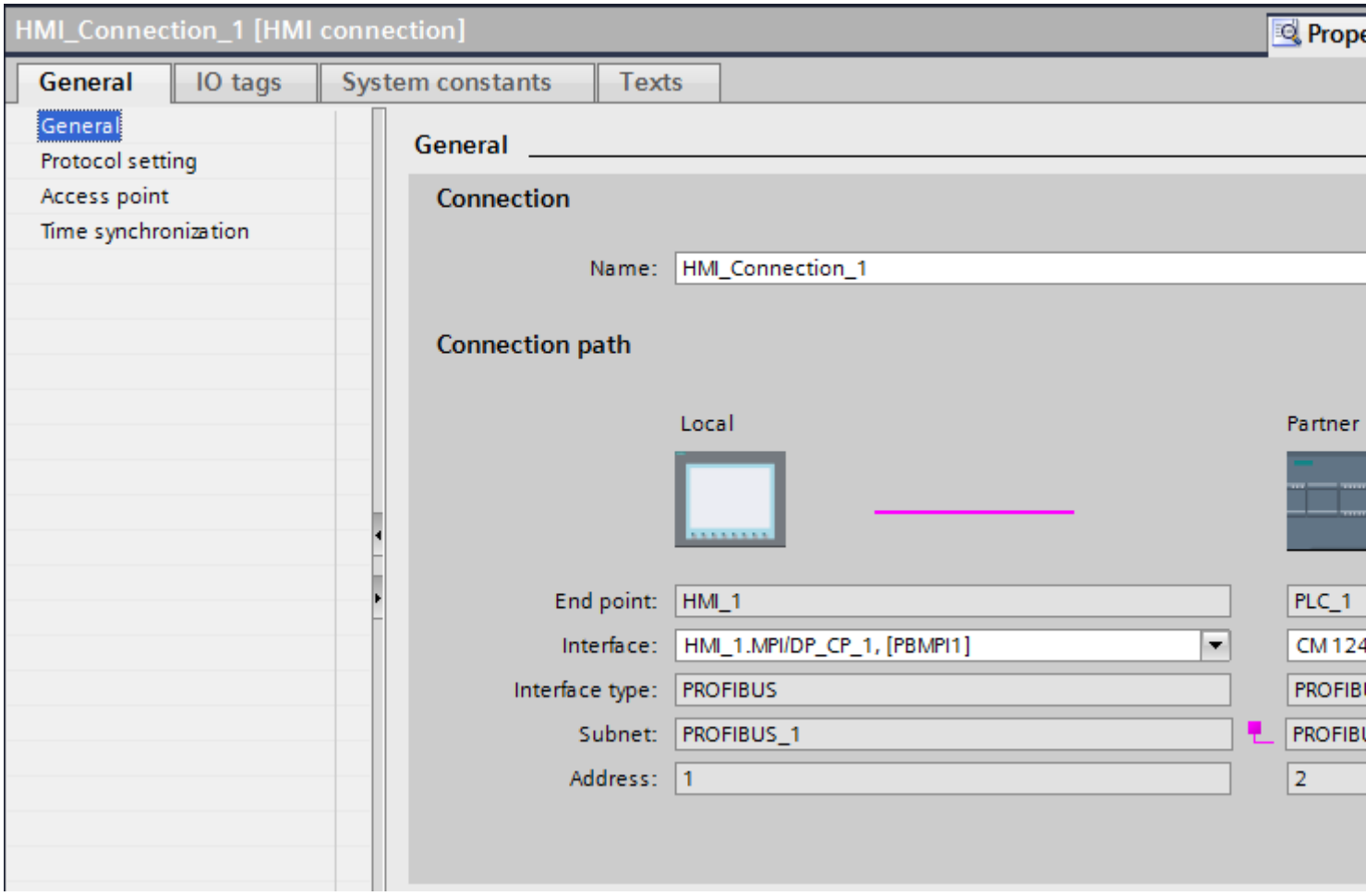
PROFIBUS parameters for the HMI connection

An overview of the configured HMI connection parameters can be found in the properties for the HMI connection.

Only limited changes are possible in this Inspector window.

Displaying and editing HMI connection parameters

1. Click the HMI connection in the "Devices & Networks" editor.
2. Change the parameters of the HMI connection in the Inspector window under "Properties > General > General".



"Connection"

The "Connection" area displays the HMI connection created for communication between the devices.

You can edit the name of the HMI connection in this area.

"Connection path"

The communication partners of the selected HMI connection and the associated PROFIBUS parameters are displayed in the "Connection path" area. Some of the areas displayed cannot be edited in this dialog.

- "End point"
  Displays the device name. This area cannot be edited.
- "Interface"
  Displays the selected interface of the device. You can choose between several interfaces, depending on the device.
- "Interface type"

Displays the selected interface type. This area cannot be edited.

- "Subnet"
  Displays the selected subnet. This area cannot be edited.
- "Address"
  Displays the PROFIBUS address of the device. This area cannot be edited.
- "Find connection path" button
  Enables the subsequent specification of connections.

See also

Communication via PROFIBUS (Page 161)

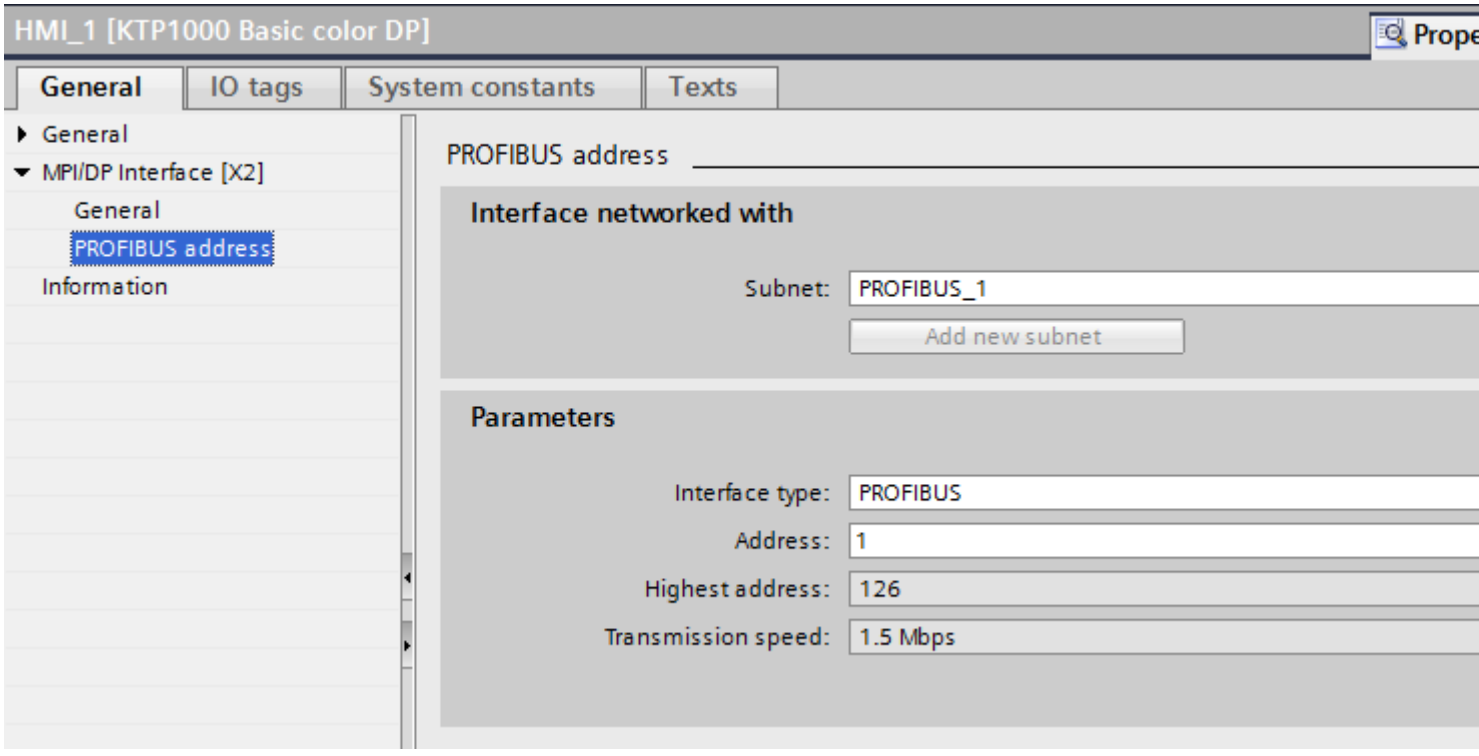
PROFIBUS parameters for the HMI device (Basic Panels)

PROFIBUS parameters for the HMI device

An overview of the configured HMI device parameters can be found in the properties for the HMI device.

Displaying and changing PROFIBUS parameters of the HMI device

1. Click the HMI device in the "Devices & Networks" editor.
2. Change the parameters of the HMI device in the Inspector window under "Properties > General > General".



"Interface networked with"

In the "Interface networked with" area, select the subnet of the HMI connection via which the HMI device is connected to the network. You use the "Add new subnet" button to create a new subnet.

"Parameters"

- "Interface type"
  Depending on the HMI device type, you have various interfaces to choose from.
- "Address"
  You assign the PROFIBUS address of the HMI device in the "Address" area. The PROFIBUS address must be unique throughout the PROFIBUS network.
- "Highest address"
  The "Highest address" area displays the highest address of the PROFIBUS network.
- "Transmission speed"
  The "Transmission speed" is determined by the slowest device connected to the network. The setting is identical throughout the network.

See also

Communication via PROFIBUS (Page 161)

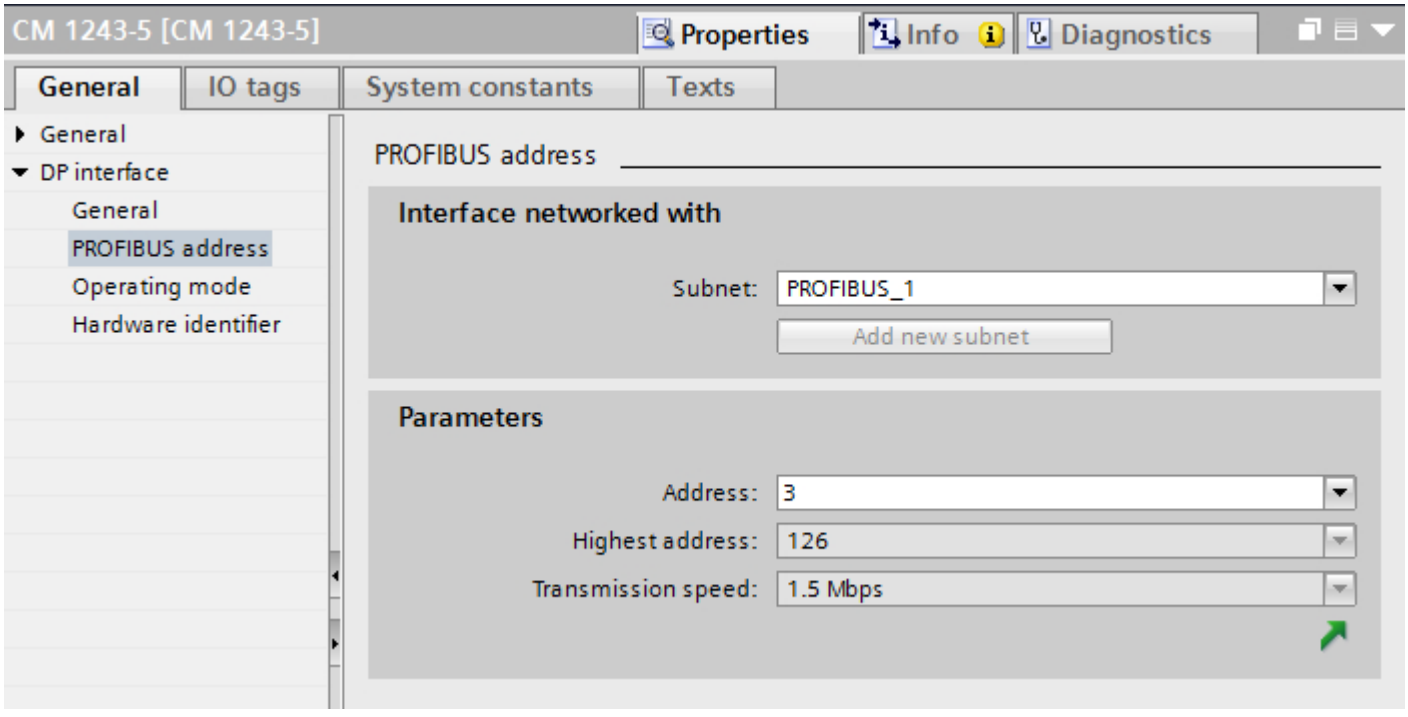
PROFIBUS parameters for the PLC (Basic Panels)

PROFIBUS parameters for the PLC

An overview of the configured parameters can be found in the properties for the PLC.

Displaying and changing PROFIBUS parameters of the PLC

1. Click the PLC in the "Devices & Networks" editor.
2. Change the parameters of the PLC in the Inspector window under "Properties > General > General".



"Interface networked with"

In the "Subnet" area, select the subnet of the HMI connection via which the PLC is connected to the network. You use the "Add new subnet" button to create a new subnet.

"Parameters"

- "Interface type"
  Depending on the HMI device type, you have various interfaces to choose from.
- "Address"
  You assign the PROFIBUS address of the HMI device in the "Address" area. The PROFIBUS address must be unique throughout the PROFIBUS network.
- "Highest address"
  The "Highest address" area displays the highest address of the PROFIBUS network.
- "Transmission speed"
  The "Transmission speed" is determined by the slowest device connected to the network. The setting is identical throughout the network.

See also

Communication via PROFIBUS (Page 161)

Bus profiles with PROFIBUS (Basic Panels)

Introduction

Depending on the device types connected and protocols used on the PROFIBUS, different profiles are available. The profiles differ in terms of the setting options and calculation of bus parameters. The profiles are explained below.

Devices with different profiles on the same PROFIBUS subnet

The PROFIBUS subnet only functions without problem if the bus parameters of all devices have the same values.

Profiles and transmission rates

Profiles	Supported transmission speeds in Kbits/s
DP	9,6 19,2 45,45 93,75 187,5 500 1500 3000 6000 12000
Standard	9,6 19,2 45,45 93,75 187,5 500 1500 3000 6000 12000
Universal	9,6 19,2 93,75 187,5 500 1500

Meaning of profiles

Profile	Meaning
DP	Select the "DP" bus profile when the only devices connected to the PROFIBUS subnet are those which satisfy the requirements of standard EN 50170 Volume 2/3, Part 8-2 PROFIBUS. The bus parameter setting is optimized on these devices. This includes devices with DP master and DP slave interfaces of the SIMATIC S7 and distributed I/Os of other manufacturers.
Standard	Compared to the "DP" profile, the "Standard" profile also offers scope for devices of another project or devices which have not been configured here to be taken into account when calculating the bus parameters. The bus parameters are then calculated following a simple, non-optimized algorithm.
Universal	Select the "Universal" bus profile when individual devices on the PROFIBUS subnet use the PROFIBUS-FMS service. This includes the following devices for example: • CP 343-5 • PROFIBUS-FMS devices of other manufacturers As with the "Standard" profile, this profile allows you to take other devices into account when calculating the bus parameters.

See also

Communication via PROFIBUS (Page 161)

Protection of communication (Basic Panels)

Security levels (Basic Panels)

If you want to protect the controller and HMI device communication, you can assign protection levels for the communication.

For a SIMATIC S7-1200 CPU, you can enter multiple passwords and thereby set up different access rights for various user groups.

The passwords are entered in a table, so that exactly one protection level is assigned to each password.

The effect of the password is given in the "Protection" column.

For the SIMATIC S7-1200 controller, several aspects need to be considered when setting protection levels.

For additional information on this, see:

Setting options for the protection (FW as of V4) (Page 170)

Example

When configuring the controller, you select the "Complete protection" protection level for a standard CPU (i.e., not an F-CPU).

Afterwards, you enter a separate password for every protection level above it in the table.

For users who do not know any of the passwords, the CPU is completely protected. Not even HMI access is possible.

For users who know one of the assigned passwords, the effect depends on the table row in which the password occurs:

- The password in row 1 (no protection) allows access as if the CPU were completely unprotected. Users who know this password have unrestricted access to the CPU.
- The password in row 2 (write protection) allows access as if the CPU were write-protected. Despite knowing the password, users who know this password only have read access to the CPU.
- The password in row 3 (read and write protection) allows access as if the CPU were read-protected and write-protected, so that only HMI access is possible for users who know this password.

See also

Setting options for the protection (FW as of V4) (Page 170)

Setting options for the protection (FW as of V4) (Basic Panels)

Protection level

The following section describes how to use the various access levels of the S7-1200 CPUs as of V4.

S7-1200 CPUs provide various access levels to limit the access to specific functions.

The parameters for the access levels are assigned in a table. The green checkmarks in the columns to the right of the respective access level specify which operations are possible without knowing the password of this access level. If you want to use the functions of check boxes that are not selected, a password has to be entered.

NOTICE
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 memory card are not write- or read-protected. Use know-how protection to protect the code of blocks on the memory card.

Default characteristics

The default access level is "Full access (no protection)". Every user can read and change the hardware configuration and the blocks. A password is not set and is also not required for online access.

The access levels in detail

With an S7-1200 CPU, you can configure the following access levels:

- Full access (no protection): The hardware configuration and the blocks can be read and changed by all users.
- Read access: With this access level, only read access to the hardware configuration and the blocks is possible without entering a password - meaning that you can load the hardware configuration and blocks into the programming device. In addition, HMI access and access to diagnostics data is possible.
  You cannot load blocks or a hardware configuration into the CPU without entering the password. Moreover, writing test functions and firmware updates are **not** possible without a password.
- 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. In addition, the following is **not** possible without a password: Writing test functions, changing the operating state (RUN/STOP) and firmware updates.
- 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).
  Authorization with the password again provides you full access to the CPU.

Behavior of a password-protected module during operation

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

Validity is checked before the online function is executed. If password protection is in place, you are prompted to enter a password.

Example: The module was configured with read access and you want to execute the "Modify tags" function. This requires write access; therefore, the assigned password must be entered to execute the function.

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".

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

See also

Security levels (Page 169)

Access password for the HMI connection (Basic Panels)

Introduction

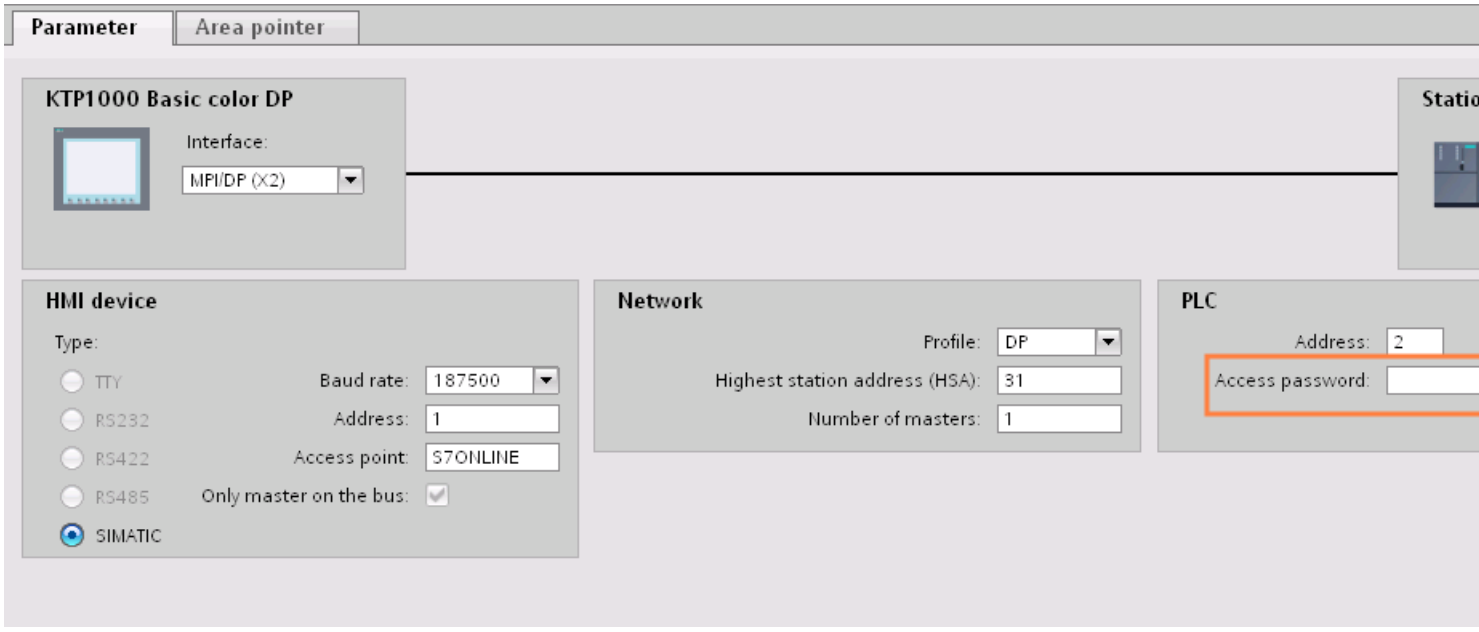
You secure access to a PLC by assigning a password.

You assign the password when you configure the password.

A password is mandatory as of protection level "Complete protection". Communication cannot be established with the PLC if you enter no password or an incorrect password.

Assigning password

You enter the password for the PLC in the "Connections" editor.



See also

Communication with SIMATIC S7-1200 (Page 140)

1.7.4 Data exchange (Basic Panels)

1.7.4.1 Data exchange using area pointers (Basic Panels)

General information on area pointers (Basic Panels)

Introduction

You use an area pointer to access a data area in the PLC. During communication, the PLC and the HMI device alternately access these data areas for read and write operations.

The PLC and the HMI device trigger defined interactions based on the evaluation of stored data.

Configuration of area pointers

Before you use an area pointer, you enable it under "Connections > Area pointer". You then assign the area pointer parameters.

You can find more detailed information on configuring area pointers in:

AUTOHOTSPOT

See also

Area pointer "Date/time" (Page 173)

"Screen number" area pointer (Page 175)

"Date/time PLC" area pointer (Page 176)

Area pointer "Coordination" (Page 177)

Area pointer "Screen number" (Page 178)

Area pointer "Project ID" (Page 180)

Area pointer "Job mailbox" (Page 181)

"Data mailbox" area pointer (Page 183)

Area pointer "Date/time" (Basic Panels)

Function

This area pointer is used to transfer the date and time from the HMI device to the PLC.

The PLC writes control job "41" to the job mailbox.

When it evaluates the control job, the HMI device writes its current date and the time in the data area configured in the "Date/time" area pointer.

Note

You cannot use the "Date/Time PLC" area pointer if you have configured the "Date/Time" area pointer.

The "Date/Time" area pointer when used in a project which contains multiple connections must be enabled for each configured connection.

"DTL" data type

If you configure the "Date/Time" area pointer, use the DTL data type.

A tag of the "DTL" data type has a length of 12 bytes and saves information on date and time in a predefined structure.

The "DTL" data type has the following structure:

Byte	Component	Data type	Value range
0	Year	UINT	1970 to 2554
1			
2	Month	USINT	0 to 12
3	Day	USINT	1 to 31
4	Weekday	USINT	1(Sunday) to 7(Saturday) The weekday is not considered in the value entry.
5	Hour	USINT	0 to 23
6	Minute	USINT	0 to 59
7	Second	USINT	0 to 59
8	Nanoseconds	UDINT	0 to 999 999 999
9			
10			
11			

The "DTL" data type supports time information down to the nanosecond range. Because panels only support time information down to the millisecond, you may encounter the following restriction when using the area pointers:

For the transmission of time information from a panel to the controller, the smallest unit of time is 1 millisecond. The value range from microseconds to nanoseconds of the "DTL" data type is filled with zeros.

See also

General information on area pointers (Page 173)

"Screen number" area pointer (Basic Panels)

Function

The HMI devices store information about the screen called up on the HMI device in the "Screen number" area pointer.

This allows the transfer of the current screen contents from the HMI device to the PLC. The PLC can trigger specific reactions such as the call of a different screen.

Use

Configure and enable the area pointer in "Communication > Connections" before you put it into use. You can create only **one** instance of the "Screen number" area pointer and only on **one** PLC.

The screen number is always transferred to the PLC when a new screen is activated or when the focus within a screen changes from one screen object to another.

Structure

The area pointer is a data area in the memory of the PLC with a fixed length of 5 words.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st word	Current screen type															
2nd word	Current screen number															
3rd word	Reserved															
4th word	Current field number															
5th word	Reserved															

- Current screen type
  "1" for root screen or
  "4" for permanent area
- Current screen number
  1 to 32767
- Current field number
  1 to 32767

See also

General information on area pointers (Page 173)

"Date/time PLC" area pointer (Basic Panels)

Function

This area pointer is used to transfer the date and time from the PLC to the HMI device. Use this area pointer if the PLC is the time master.

The PLC loads the data area of the area pointer.

The HMI device reads the data cyclically within the configured acquisition cycle and synchronizes itself.

Note

Set an acquisition cycle of sufficient length for the date/time area pointer PLC to avoid any negative impact on HMI device performance.
Recommended: Acquisition cycle of 1 minute, if the process allows this.

"Date/Time PLC" is a global area pointer and may be configured only once per project.

Note

You cannot use the "Date/Time" area pointer if you have configured the "Date/Time PLC" area pointer.

"DTL" data type

If you configure the "Date/Time PLC" area pointer, use the DTL data type.

A tag of the "DTL" data type has a length of 12 bytes and saves information on date and time in a predefined structure.

The "DTL" data type has the following structure:

Byte	Component	Data type	Value range
0	Year	UINT	1970 to 2554
1			
2	Month	USINT	0 to 12
3	Day	USINT	1 to 31
4	Day of the week	USINT	1(Sunday) to 7(Saturday) The day of the week is not considered in the value entry.
5	Hour	USINT	0 to 23
6	Minute	USINT	0 to 59
7	Second	USINT	0 to 59
8	Nanoseconds	UDINT	0 to 999 999 999
9			
10			
11			

The HMI devices do not support the use of nanoseconds. Values in the nanosecond range will be ignored during processing in Runtime.

The "DTL" data type supports time information down to the nanosecond range. Because panels only support time information down to the millisecond, you may encounter the following restriction when using the area pointers:

For the transmission of time information from a PLC to a panel, the range from microseconds to nanoseconds is ignored. The time information is processed on the panel down to milliseconds.

See also

General information on area pointers (Page 173)

Area pointer "Coordination" (Basic Panels)

Function

The "Coordination" area pointer is used to implement the following functions:

- Detecting the startup of the HMI device in the control program
- Detecting the current operating mode of the HMI device in the control program
- Detecting whether the HMI device is ready to communicate in the control program

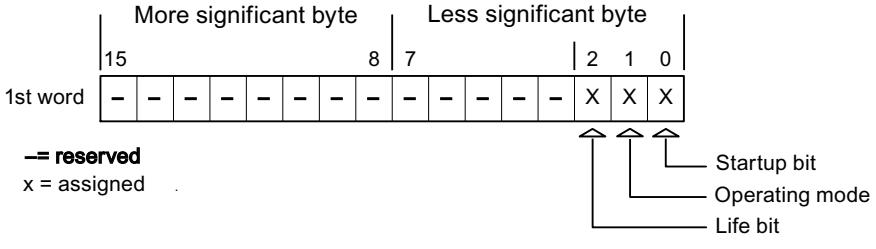
By default, the "Coordination" area pointer has the length of one word and cannot be changed.

Usage

Note

The HMI device always writes the entire coordination area when updating the area pointer. The control program can for this reason not make changes to the coordination area.

Assignment of the bits in the "Coordination" area pointer



Startup bit

The startup bit is set briefly to "0" by the HMI device during startup. It sets the bit permanently to "1" when startup is completed.

Operating mode

The operating mode bit is set to 1 as soon as the user switches the HMI device offline. The status of the operating mode bit is "0" during normal operation of the HMI device. You can determine the current operating mode of the HMI device by reading this bit in the control program.

Life bit

The HMI device inverts the life bit at intervals of approximately one second. You can check whether or not there is still a connection to the HMI device by querying this bit in the control program.

Processing in the PLC

For a simpler evaluation in the PLC program, use a Bool array for this area pointer when using the SIMATIC S7 1200 communication driver. You will have to map the complete 16-bit word of the area pointer. Configure a tag of the data type "Array [0 .. 15] of bool" for this purpose.

Permitted data types

You can use the following data types when you configure the "Coordination" area pointer.

- Word
- UInt
- Bool

See also

General information on area pointers (Page 173)

Area pointer "Screen number" (Basic Panels)

Function

The HMI devices store information about the screen called up on the HMI device in the "Screen number" area pointer.

This allows the transfer of the current screen contents from the HMI device to the PLC. Certain reactions can be triggered in the PLC, such as the call of a different screen.

Use

Before the "Screen number" area pointer can be used, it must be set up and activated by selecting "Communication ▸ Area pointer". You can create only **one** instance of the "Screen number" area pointer and only on **one** PLC.

The screen number is always transferred to the PLC when a new screen is activated or when the focus within a screen changes from one screen object to another.

Structure

The area pointer is a data area in the memory of the PLC with a fixed length of 5 words.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st word	Current screen type															
2nd word	Current screen number															
3rd word	Reserved															
4th word	Current field number															
5th word	Reserved															

- Current screen type
  "1" for root screen or
  "4" for permanent area
- Current screen number
  1 to 32767
- Current field number
  1 to 32767

Note

Device dependency

Permanent areas are not available on Basic Panels.

Permitted data types

You can use the following data types when you configure the "Screen number" area pointer.

- Word
- UInt

See also

General information on area pointers (Page 173)

Area pointer "Project ID" (Basic Panels)

Function

When Runtime starts, a check can be carried out as to whether the HMI device is connected to the correct PLC. This check is important when operating with several HMI devices.

For this, the HMI device compares a value stored on the PLC with the value specified in configuration. This ensures compatibility of the configuration data with the PLC program.

A missing compatibility results in a corresponding alarm and Runtime will not be started.

Use

Note

HMI connections cannot be switched "online".

The HMI connection in which the "Project ID" area pointer is used must be switched "online".

To use this area pointer, set up the following during the configuration:

- Define the version of the configuration. Values between 1 and 255 are possible.
  You enter the version in the editor "Runtime settings > General" in the "Identification" area.
- This is where you select the PLC tag or the tag array that you have configured as the data area for the area pointer.

Connection failure

A connection failure to a device on which the "project ID" area pointer is configured results in all the other connections of the device being switched to "offline".

This behavior has the following requirements:

- You have configured several connections in a project.
- You are using the "project ID" area pointer in at least one connection.

Causes which may set connections "offline":

- The PLC is not available.
- The connection has been switched offline in the engineering system.

Permitted data types

You can use the following data types when you configure the "Project ID" area pointer.

- Word
- UInt

See also

General information on area pointers (Page 173)

Area pointer "Job mailbox" (Basic Panels)

Function

The PLC can use the job mailbox to transfer jobs to the HMI device to trigger corresponding actions on the HMI device. These functions include, for example:

- Display screen
- Set date and time

Data structure

The first word of the job mailbox contains the job number. Depending on the job mailbox, up to three parameters can be transferred.

Word	Most significant byte	Least significant byte
n+0	0	Job number
n+1	Parameter 1	
n+2	Parameter 2	
n+3	Parameter 3	

The HMI device evaluates the job mailbox if the first word of this job is not equal to zero. This means that the parameters must be entered in the job mailbox first, followed by the job number.

When the HMI device accepts the job mailbox, the first word is set to 0 again. The execution of the job mailbox is generally not completed at this point in time.

Job mailboxes

All job mailboxes and their parameters are listed below. The "No." column contains the job number of the job mailbox. Job mailboxes can only be triggered by the PLC when the HMI device is online.

No.	Function	
14	**Set time (BCD-coded)**	
	Parameter 1	Left byte: - Right byte: hours (0-23)
	Parameter 2	Left byte: minutes (0-59) Right byte: seconds (0-59)
	Parameter 3	-
15	**Set date (BCD-coded)**	
	Parameter 1	Left byte: - Right byte: weekday (1-7: Sunday-Saturday)
	Parameter 2	Left byte: day (1-31) Right byte: month (1-12)
	Parameter 3	Left byte: year
23	**User logon**	

No.	Function	
14	**Set time (BCD-coded)**	
	Logs the user on with the name "PLC user" at the HMI device with the group number transferred in Parameter 1. The logon is possible only when the transferred group number exists in the project.	
	Parameter 1	Group number 1 to 255
	Parameter 2, 3	-
24	**User logoff**	
	Logs off the current user. (The function corresponds to the "logoff" system function)	
	Parameter 1, 2, 3	-
40	**Transfer date/time to PLC**	
	An interval of at least 5 seconds must be maintained between two successive jobs to prevent overload of the HMI device.	
	Parameter 1, 2, 3	-
41	**Transfer date/time to PLC**	
	An interval of at least 5 seconds must be maintained between two successive jobs to prevent overload of the HMI device.	
	Parameter 1, 2, 3	-
46	**Update tag**	
	Causes the HMI device to read the current value of the tags from the PLC whose update ID matches the value transferred in Parameter 1. (Function corresponds to the "UpdateTag" system function.)	
	Parameter 1	1 - 100
49	**Clear event buffer**	
	Deletes all analog alarms and discrete alarms of the "Warnings" class from the alarm buffer.	
	Parameter 1, 2, 3	-
50	**Clear error alarm buffer**	
	Deletes all analog alarms and discrete alarms of the "Errors" class from the alarm buffer.	
	Parameter 1, 2, 3	-
51	**Display selection**	
	Parameter 1	Screen number
	Parameter 2	-
	Parameter 3	Field number
69	**Reading data record from PLC** [1)]	
	Parameter 1	Recipe number (1-999)
	Parameter 2	Data record number (1-65535)
	Parameter 3	0: Do not overwrite existing data record 1: Overwrite existing data record
70	**Writing data record to PLC** [1)]	
	Parameter 1	Recipe number (1-999)
	Parameter 2	Data record number (1-65535)
	Parameter 3	-

1)	Only devices supporting recipes
2)	OP 73, OP 77A and TP 177A HMI devices also execute the "Screen selection" job mailbox if the on-screen keyboard is active.
3)	The weekday is ignored on HMI device KTP 600 BASIC PN.

Permitted data types

You can use the following data types when you configure the "Screen number" area pointer:

- Word
- UInt

See also

General information on area pointers (Page 173)

"Data record" area pointer (Basic Panels)

"Data mailbox" area pointer (Basic Panels)

Function

When data records are transferred between the HMI device and PLC, both partners access common communications areas on the PLC.

Data transfer types

There are two ways of transferring data records between the HMI device and PLC:

- Transfer without synchronization
- Transfer with synchronization over the data mailbox

Data records are always transferred directly. That is, the tag values are read from an address or written to an address configured for this tag directly, without redirecting the values by means of interim memory.

Initiating the transfer of data records

There are three ways of triggering the transfer:

- Operator input in the recipe view
- Job mailboxes
  The transfer of data records can also be triggered by the PLC.
- Triggering by configured functions

If the transfer of data records is triggered by a job mailbox, the data in the recipe view will be updated as well. Avoid operating the recipe view while job mailboxes for transfer of data records are being triggered. If you have already started editing a data record and a job mailbox is triggered for transfer of data records, then this job mailbox will be rejected.

Permitted data types

You can use the following data types when you configure the "Data record" area pointer.

- Word
- UInt

See also

General information on area pointers (Page 173)

Transfer without synchronization (Page 184)

Sequence of a transfer started by the operator in the recipe display (Page 185)

Sequence of the transfer triggered by a job mailbox (Page 186)

Sequence of the transfer when triggered by a configured function (Page 187)

Possible causes of error when transferring data records (Page 188)

Transfer without synchronization (Basic Panels)

If you select asynchronous transfer of data records between the HMI device and PLC, there is no coordination over the common data areas. It is therefore unnecessary to set up a data area during configuration.

Asynchronous data record transfer can be a useful alternative, for example, when:

- The system is capable of excluding the risk of uncontrolled overwriting of data by the communication peer.
- The PLC does not require information about the recipe number and data record number.
- The transfer of data records is triggered by the operator of the HMI device.

Reading values

When a read job is triggered, the values are read from the PLC addresses and transferred to the HMI device.

- Triggering by the operator in the recipe view:
  The values are downloaded to the HMI device. You can then process, edit, or save these values, for example.
- Triggering by a function or job mailbox:
  The values are saved immediately to the data volume.

Writing values

When a write job is triggered, the values are written to the PLC addresses.

- Triggering by the operator in the recipe view:
  The current values are written to the PLC.
- Triggering by a function or job mailbox:
  The current values are written to the PLC from the data medium.

See also

"Data mailbox" area pointer (Page 183)

Sequence of a transfer started by the operator in the recipe display (Basic Panels)

Reading from the PLC started by the operator in the recipe view

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe number to be read and the status "Transferring" in the data mailbox and sets the data record number to 0.	Abort with system alarm.
3	The HMI device reads the values from the PLC and displays them in the recipe view. If the recipes have synchronized tags, the values from the PLC are also written to the tags.	
4	The HMI device sets the status "Transfer completed."	
5	The control program must reset the status word to zero in order to enable further transfers.	

Writing to the PLC started by the operator in the recipe view

Step	Action	
	Check: Status word = 0?	
1	Yes	No
	The HMI device enters the recipe and data record number to be written and the status "Transferring" in the data mailbox.	Abort with system alarm.
2	The HMI device writes the current values to the PLC. If the recipes have synchronized tags, the changed values are synchronized between the recipe view and tags and then written to the PLC.	
3	The HMI device sets the status "Transfer completed."	
4	If required, the control program can now evaluate the transferred data.	
5	The control program must reset the status word to zero in order to enable further transfers.	

Note

The status word may only be set by the HMI device. The PLC may only reset the status word to zero.

Note

The PLC may only evaluate the recipe and data record numbers when data inconsistency is detected if one of the conditions outlined below has been met:

- The data mailbox status is set to "Transfer completed".
- The data mailbox status is set to "Transfer completed with error".

See also

"Data mailbox" area pointer (Page 183)

Sequence of the transfer triggered by a job mailbox (Basic Panels)

The transfer of data records between the HMI device and the PLC can be initiated by either one of these stations.

The two job mailboxes No. 69 and No. 70 are available for this type of transfer.

No. 69: Read data record from PLC ("PLC → DAT")

Job mailbox no. 69 transfers data mailboxes from the PLC to the HMI device. The job mailbox is structured as follows:

	Most significant byte	Least significant byte
Word 1	0	69
Word 2	Recipe number (1-999)	
Word 3	Data record number (1-65,535)	
Word 4	Do not overwrite existing data record: 0 Overwrite existing data record: 1	

No. 70: Write data record to PLC ("DAT → PLC")

Job mailbox no. 70 transfers data mailboxes from the HMI device to the PLC. The job mailbox is structured as follows:

	Most significant byte	Least significant byte
Word 1	0	70
Word 2	Recipe number (1-999)	
Word 3	Data record number (1-65,535)	
Word 4	—	

Sequence when reading from the PLC with job mailbox "PLC → DAT" (no. 69)

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the job and the status "Transferring" in the data mailbox.	Abort without return message.
3	The HMI device reads the values and stores the values in the data record specified in the job mailbox.	
4	• If "Overwrite" was selected in the job, an existing data record is overwritten without any prompt for confirmation. The HMI device sets the status "Transfer completed". • If "Do not overwrite" was selected in the job, and the data record already exists, the HMI device aborts the job and enters 0000 1100 in the status word of the data mailbox.	
5	The control program must reset the status word to zero in order to enable further transfers.	

Sequence writing to the PLC with job mailbox "DAT → PLC" (no. 70)

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the job and the status "Transferring" in the data mailbox.	Abort without return message.
3	The HMI device fetches the values of the data record specified in the job from the data medium and writes the values to the PLC.	
4	The HMI device sets the status "Transfer completed."	
5	The PLC program can now evaluate the transferred data. To allow further transfers, the PLC program must set the status word to 0 again.	

See also

"Data mailbox" area pointer (Page 183)

Sequence of the transfer when triggered by a configured function (Basic Panels)

Reading from the PLC using a configured function

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the function and the status "Transferring" in the data mailbox.	Abort with system alarm.

Step	Action	
3	The HMI device reads the values from the PLC and stores them in the data record specified in the function.	
4	• If "Yes" was selected for the "Overwrite" function, an existing data record is overwritten without any prompt for confirmation. The HMI device sets the status "Transfer completed." • If "No" was selected for the "Overwrite" function and the data record already exists, the HMI device aborts the job and enters 0000 1100 in the status word of the data mailbox.	
5	The control program must reset the status word to zero in order to enable further transfers.	

Writing to the PLC by means of configured function

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the function and the status "Transferring" in the data mailbox.	Abort with system alarm.
3	The HMI device fetches the values of the data record specified in the function from the data medium and transfers the values to the PLC.	
4	The HMI device sets the status "Transfer completed."	
5	The control program can now evaluate the transferred data. The control program must reset the status word to zero in order to enable further transfers.	

See also

"Data mailbox" area pointer (Page 183)

Possible causes of error when transferring data records (Basic Panels)

Possible causes of error

The section below shows possible error causes which lead to the cancellation of data record transfer:

- Tag address not set up on the PLC
- Overwriting data records not possible

- Recipe number does not exist
- Data record number does not exist

Note

The status word may only be set by the HMI device. The PLC may only reset the status word to zero.

Note

The PLC may only evaluate the recipe and data record numbers when data inconsistency is detected if one of the conditions outlined below has been met:

- The data mailbox status is set to "Transfer completed".
- The data mailbox status is set to "Transfer completed with error".

Reaction to an aborted transfer due to errors

If the transfer of data records is aborted due to errors, the HMI device reacts as follows:

- Triggering by the operator in the recipe view
  Information in the status bar of the recipe view and output of system alarms
- Triggered by function
  Output of system alarms
- Triggering by job mailbox
  No return message on the HMI device

You can nonetheless evaluate the status of the transfer by querying the status word in the data mailbox.

Note

Availability for specific devices

Notes in the status bar of the recipe view are not available in Basic Panels.

See also

"Data mailbox" area pointer (Page 183)

1.7.4.2 Trends (Basic Panels)

Trends

A trend is the graphical representation of one or more values from the PLC. The value is read out time-triggered for Basic Panels.

For additional information see:

AUTOHOTSPOT

Time-triggered trends

The HMI device reads in the trend values cyclically at an interval specified in the configuration.

Time-triggered trends are suitable for continuous curves, such as the operating temperature of a motor.

1.7.4.3 Alarms (Basic Panels)

Configuring alarms (Basic Panels)

Configure alarms

Several steps are needed to configure alarms, such as operational messages, error alarms, and acknowledgement.

- Step 1: Create tags
- Step 2: Configure alarms
- Step 3: Configure acknowledgment

You can find additional information in the section:

AUTOHOTSPOT

Distinctive features when configuring alarms

If you are configuring connections of HMI devices to PLCs of other manufacturers, note the following distinctive features when configuring:

- Data types of the tags
- Addressing of tags
- How the bit positions are counted

Data types

For connections with a SIMATIC communication driver, the following data types are supported:

PLC	Permitted data types	
	Discrete alarms	Analog alarms
SIMATIC S7 PLCs	WORD, INT	BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, TIMER

How the bit positions are counted

For connections with a SIMATIC communication driver, the following counting method applies:

How the bit positions are counted	Byte 0								Byte 1							
	Most significant byte								Least significant byte							
In SIMATIC S7 PLCs	7							0	7							0
In WinCC you configure:	15							8	7							0

See also

Acknowledgment of alarms (Page 191)

Acknowledgment of alarms (Basic Panels)

Procedure

Create suitable tags on the PLC to acknowledge an error alarm. You assign these tags to an alarm in the "Bit messages" editor. You make the assignment in "Properties > Acknowledgment".

Distinction in terms of acknowledgment:

- Acknowledgment by the PLC
- Acknowledgment on the HMI device

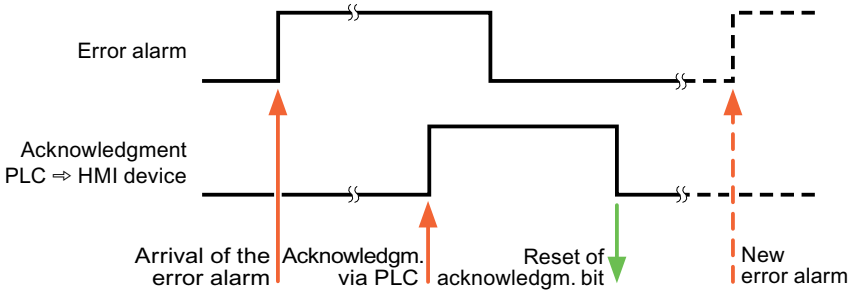
Acknowledgment by the PLC

In "PLC acknowledgment tag", configure the tag or array tag and the bit number that the HMI device uses to identify a PLC acknowledgment.

A bit set in the tag triggers acknowledgment of the assigned error alarm bit at the HMI device. This tag bit returns a function similar to acknowledgment on the HMI device which is triggered by pressing the "ACK" button, for example.

The acknowledgment bit must be located in the same tag as the bit for the error alarm.

Reset the acknowledgment bit before setting the bit in the alarm area again. The figure below shows the pulse diagram.



Acknowledgment on the HMI device

In the "HMI acknowledgment tag" area, configure the tag or array tag as well as the bit number that the HMI device writes to the PLC after acknowledgment. Make sure when you use an array tag that it is not longer than 6 words.

To always create a signal change when setting an assigned acknowledgment bit of a discrete alarm that must be acknowledged, the HMI device will reset the acknowledgment bit assigned to the alarm as soon as it detects an alarm subject to acknowledgment and write the acknowledgment tag in the PLC. There will be a certain delay between detecting the message and writing the acknowledgment tag in the PLC because the HMI device has to process the operations.

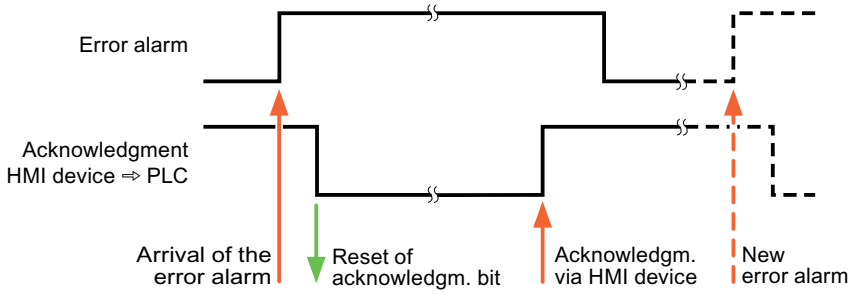
If a discrete alarm subject to acknowledgment is acknowledged by the HMI device, then the corresponding bit in the assigned acknowledgment tag will be set. The entire acknowledgment tag is then written to the PLC by the HMI device. This allows the PLC to recognize that a certain alarm message has been acknowledged at the HMI device.

Note

All alarm bits acknowledged since the last Runtime start will remain in the acknowledgment tag until a new incoming of the respective discrete alarms is detected.

This area should only be read by the PLC because the entire section of the HMI device will be overwritten once the next acknowledgment tag is written.

The figure below shows the pulse diagram.



See also

Configuring alarms (Page 190)

1.7.5 Performance features of communication (Basic Panels)

1.7.5.1 S7-1200 device dependency (Basic Panels)

Communication with the SIMATIC S7-1200 controller

If you use devices from an earlier version of the TIA Portal with TIA Portal V14, it may not be possible to configure integrated connections to certain HMI devices.

Basic Panels V11.0

HMI devices	SIMATIC S7-1200 (V1)	SIMATIC S7-1200 (V2)	SIMATIC S7-1200 (V2.2)	SIMATIC S7-1200 (V3)	SIMATIC S7-1200 (V4)
KP300 Basic	Yes	Yes	Yes	Yes	Yes
KP400 Basic	Yes	Yes	Yes	Yes	Yes
KTP400 Basic PN	Yes	Yes	Yes	Yes	Yes
KTP600 Basic DP	Yes	Yes	Yes	Yes	Yes
KTP600 Basic PN	Yes	Yes	Yes	Yes	Yes
KTP1000 Basic DP	Yes	Yes	Yes	Yes	Yes
KTP1000 Basic PN	Yes	Yes	Yes	Yes	Yes
TP1500 Basic PN	Yes	Yes	Yes	Yes	Yes

Basic Panels V12.0

HMI devices	SIMATIC S7-1200 (V1)	SIMATIC S7-1200 (V2)	SIMATIC S7-1200 (V2.2)	SIMATIC S7-1200 (V3)	SIMATIC S7-1200 (V4)
KP300 Basic	No	Yes	Yes	Yes	Yes
KP400 Basic	No	Yes	Yes	Yes	Yes
KTP400 Basic PN	No	Yes	Yes	Yes	Yes
KTP600 Basic DP	No	Yes	Yes	Yes	Yes
KTP600 Basic PN	No	Yes	Yes	Yes	Yes
KTP1000 Basic DP	No	Yes	Yes	Yes	Yes
KTP1000 Basic PN	No	Yes	Yes	Yes	Yes
TP1500 Basic PN	No	Yes	Yes	Yes	Yes

Basic Panels V13.0

HMI devices	SIMATIC S7-1200 (V1)	SIMATIC S7-1200 (V2)	SIMATIC S7-1200 (V2.2)	SIMATIC S7-1200 (V3)	SIMATIC S7-1200 (V4)
KTP400 Basic PN	No	Yes	Yes	Yes	Yes
KTP700 Basic PN	No	Yes	Yes	Yes	Yes
KTP700 Basic DP	No	Yes	Yes	Yes	Yes
KTP900 Basic PN	No	Yes	Yes	Yes	Yes
KTP1200 Basic PN	No	Yes	Yes	Yes	Yes
KTP1200 Basic DP	No	Yes	Yes	Yes	Yes

Basic Panels V13.0.1

HMI devices	SIMATIC S7-1200 (V1)	SIMATIC S7-1200 (V2)	SIMATIC S7-1200 (V2.2)	SIMATIC S7-1200 (V3)	SIMATIC S7-1200 (V4)
KTP400 Basic PN	No	Yes	Yes	Yes	Yes
KTP700 Basic PN	No	Yes	Yes	Yes	Yes
KTP700 Basic DP	No	Yes	Yes	Yes	Yes
KTP900 Basic PN	No	Yes	Yes	Yes	Yes
KTP1200 Basic PN	No	Yes	Yes	Yes	Yes
KTP1200 Basic DP	No	Yes	Yes	Yes	Yes

Basic Panels V14.0

HMI devices	SIMATIC S7-1200 (V1)	SIMATIC S7-1200 (V2)	SIMATIC S7-1200 (V2.2)	SIMATIC S7-1200 (V3)	SIMATIC S7-1200 (V4)
KTP400 Basic PN	No	Yes	Yes	Yes	Yes
KTP700 Basic PN	No	Yes	Yes	Yes	Yes
KTP700 Basic DP	No	Yes	Yes	Yes	Yes
KTP900 Basic PN	No	Yes	Yes	Yes	Yes
KTP1200 Basic PN	No	Yes	Yes	Yes	Yes
KTP1200 Basic DP	No	Yes	Yes	Yes	Yes

See also

Communication with SIMATIC S7-1200 (Page 140)
Valid data types for SIMATIC S7-1200 (Page 195)

1.7.5.2 Valid data types for SIMATIC S7-1200 (Basic Panels)

Valid data types for connections with SIMATIC S7-1200

The table lists the data types that can be used when configuring tags and area pointers.

Data type	Length	
BOOL	1 bit	
SINT	1 byte	
INT	2 bytes	
DINT	4 bytes	
USINT	1 byte	
UINT	2 bytes	
UDINT	4 bytes	
REAL	4 bytes	
LREAL	8 bytes	
TIME	4 bytes	
DATE	2 bytes	
DTL	12 bytes	Basic Panels, Panels, RT Advanced
	8 bytes	RT Professional
TIME_OF_DAY, TOD	4 bytes	
STRING	(2+n) bytes, n = 0 to 254	
WSTRING	(4+2*n) bytes, n = 0 to 254	Basic Panels
	(4+2*n) bytes, n = 0 to 4094	Panels, RT Advanced
	(4+2*n) bytes, n = 0 to 65534	RT Professional
CHAR	1 byte	
WCHAR	2 bytes	RT Professional
Array	(Element number * data type length) bytes [1)]	
BYTE	1 byte	
WORD	2 bytes	
DWORD	4 bytes	
LDT	8 bytes	RT Professional
DATE_AND_TIME	8 bytes	RT Professional

[1)] Example of "Length of an array": The length is 400 bytes (100 * 4) for 100 elements of the data type REAL

See also

Communication with SIMATIC S7-1200 (Page 140)

S7-1200 device dependency (Page 193)

1.7.6 Creating connections in the "Connections" editor (Basic Panels)

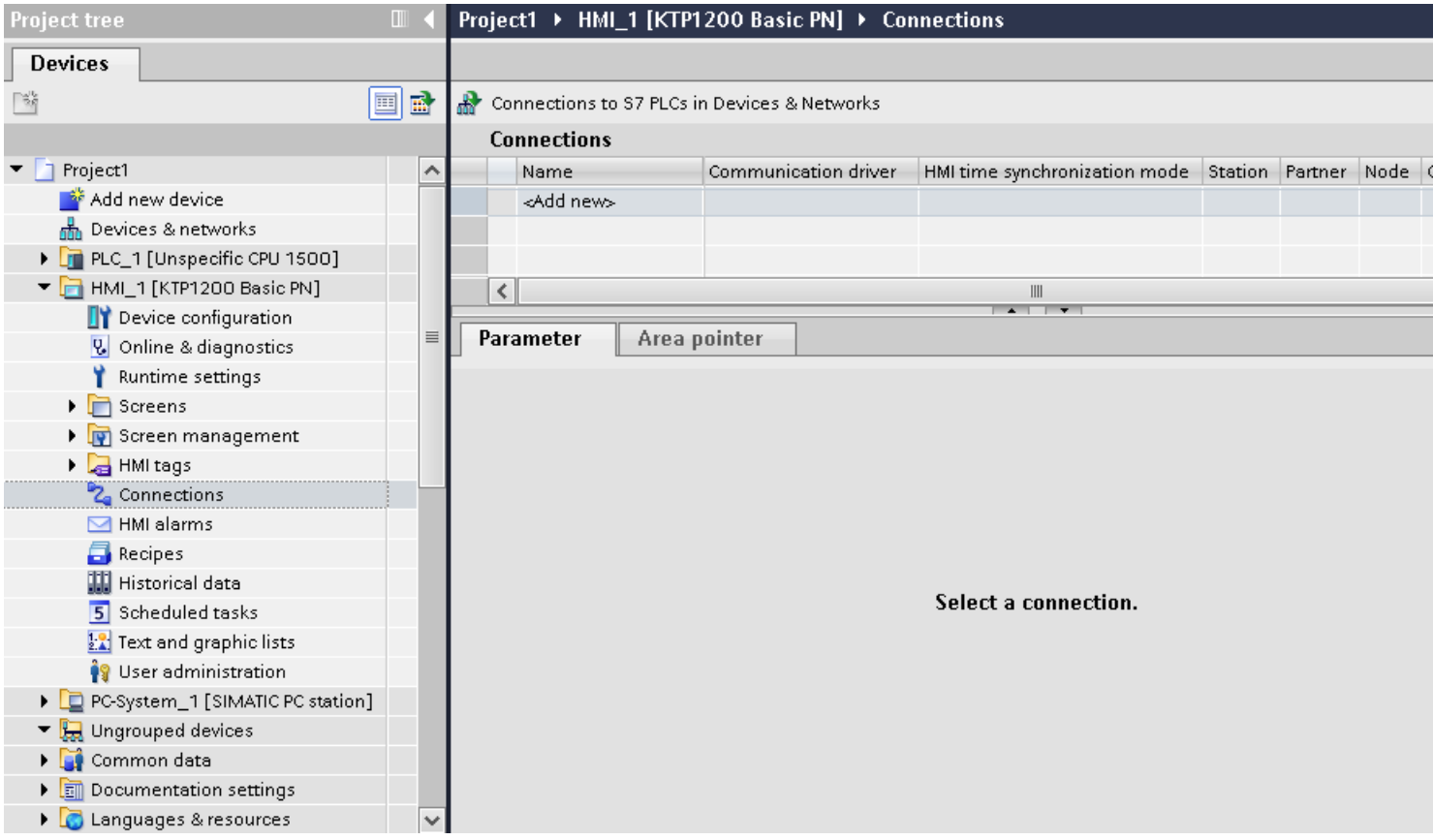
1.7.6.1 Creating a PROFINET connection (Basic Panels)

Requirements

- A project is open.
- An HMI device with a PROFINET interface has been created.

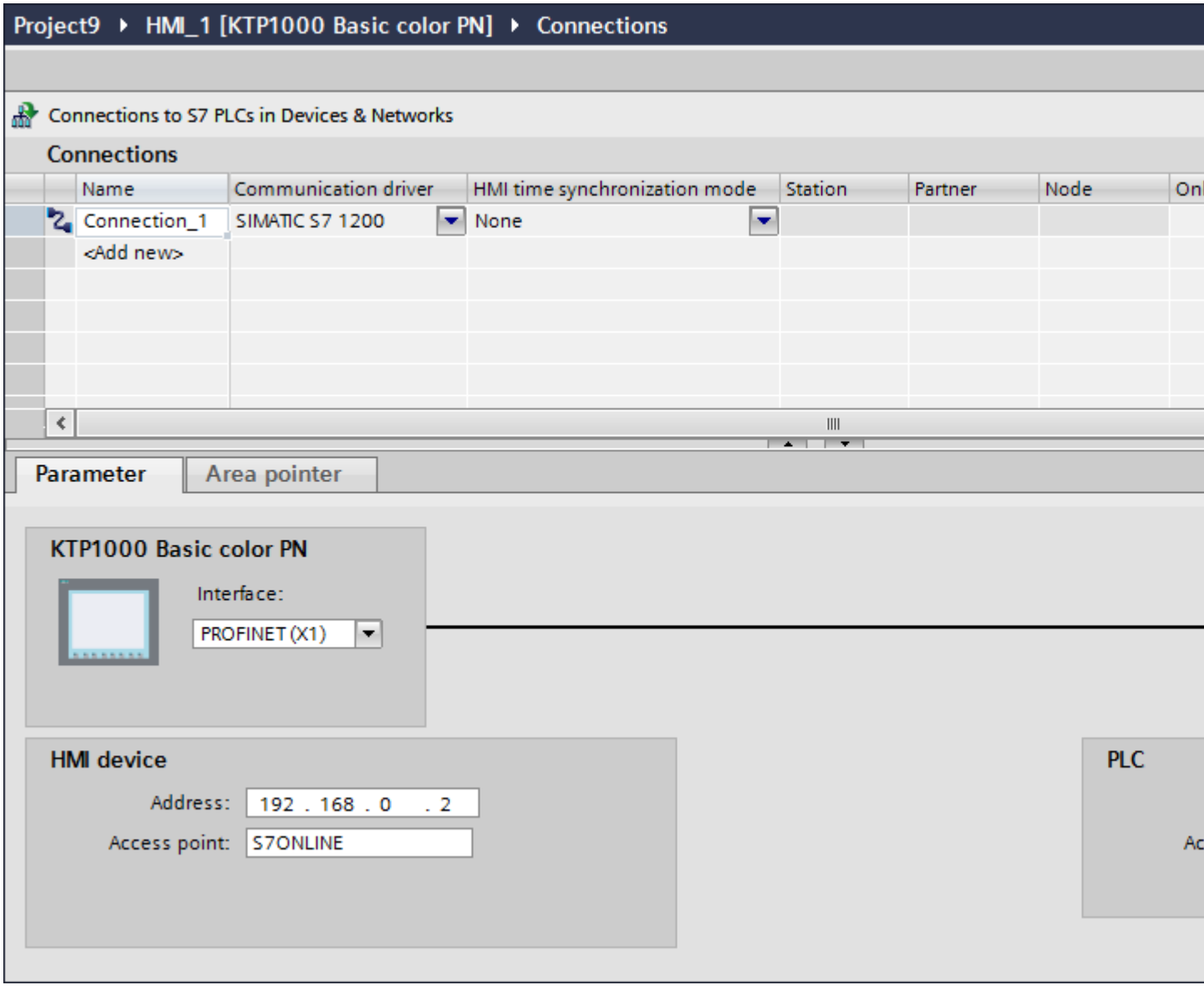
Procedure

1. Open the "Connections" editor of the HMI device.
2. Double-click "<Add>".



3. In the "Communication drivers" column, select the "SIMATIC S7 1200" driver.

4. Click the name of the connection.
5. Select a PROFINET interface of the HMI device in the Inspector window under "Parameters > Interface".



6. Set the IP addresses of the communication partners in the Inspector window:
   - HMI device: "Parameters > HMI device > Address"
   - PLC: "Parameters > PLC > Address"

See also

Communication with SIMATIC S7-1200 (Page 140)

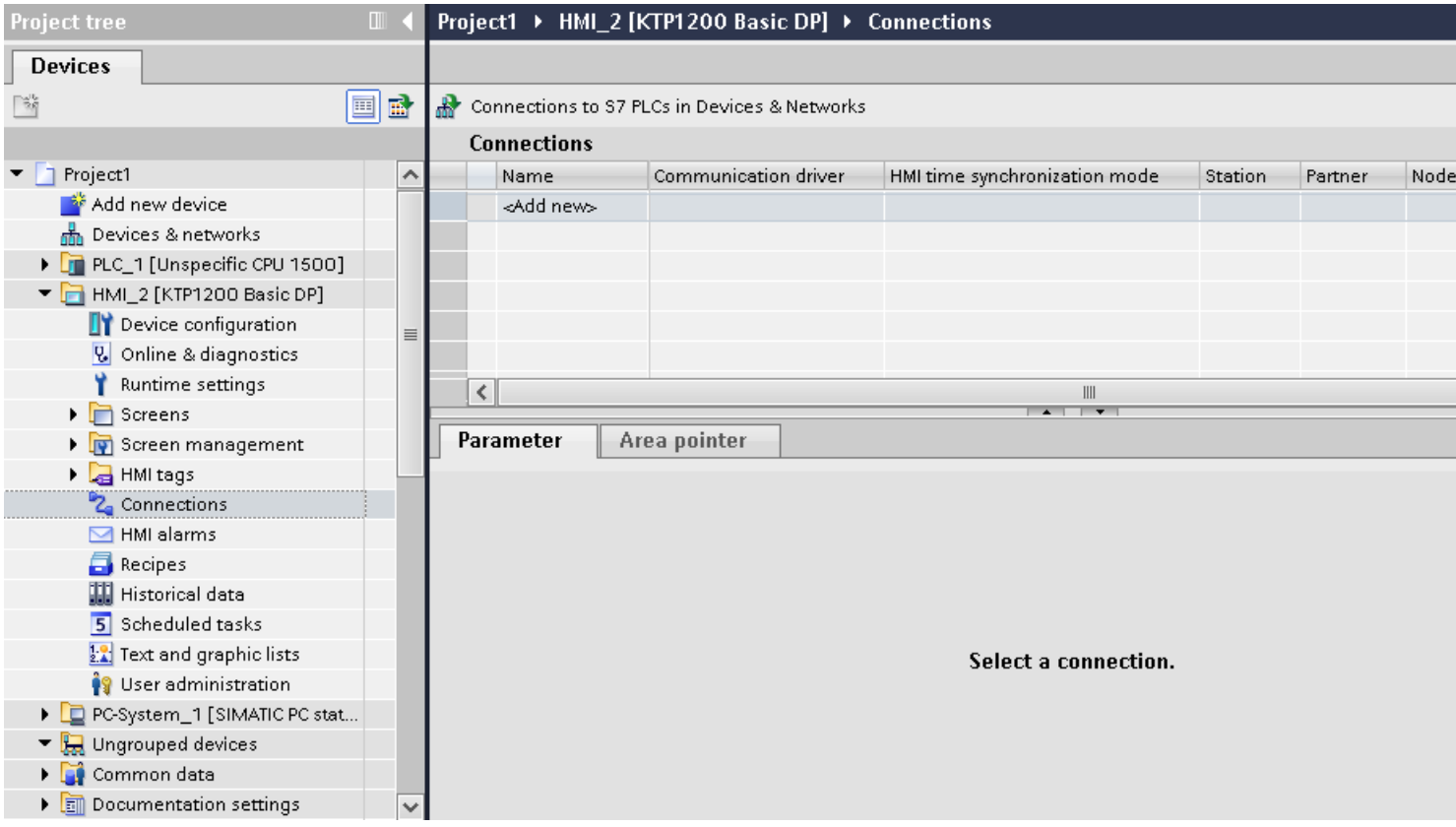
1.7.6.2 Creating a PROFIBUS DP connection (Basic Panels)

Requirements

- A project is open.
- An HMI device with a PROFIBUS interface has been created.

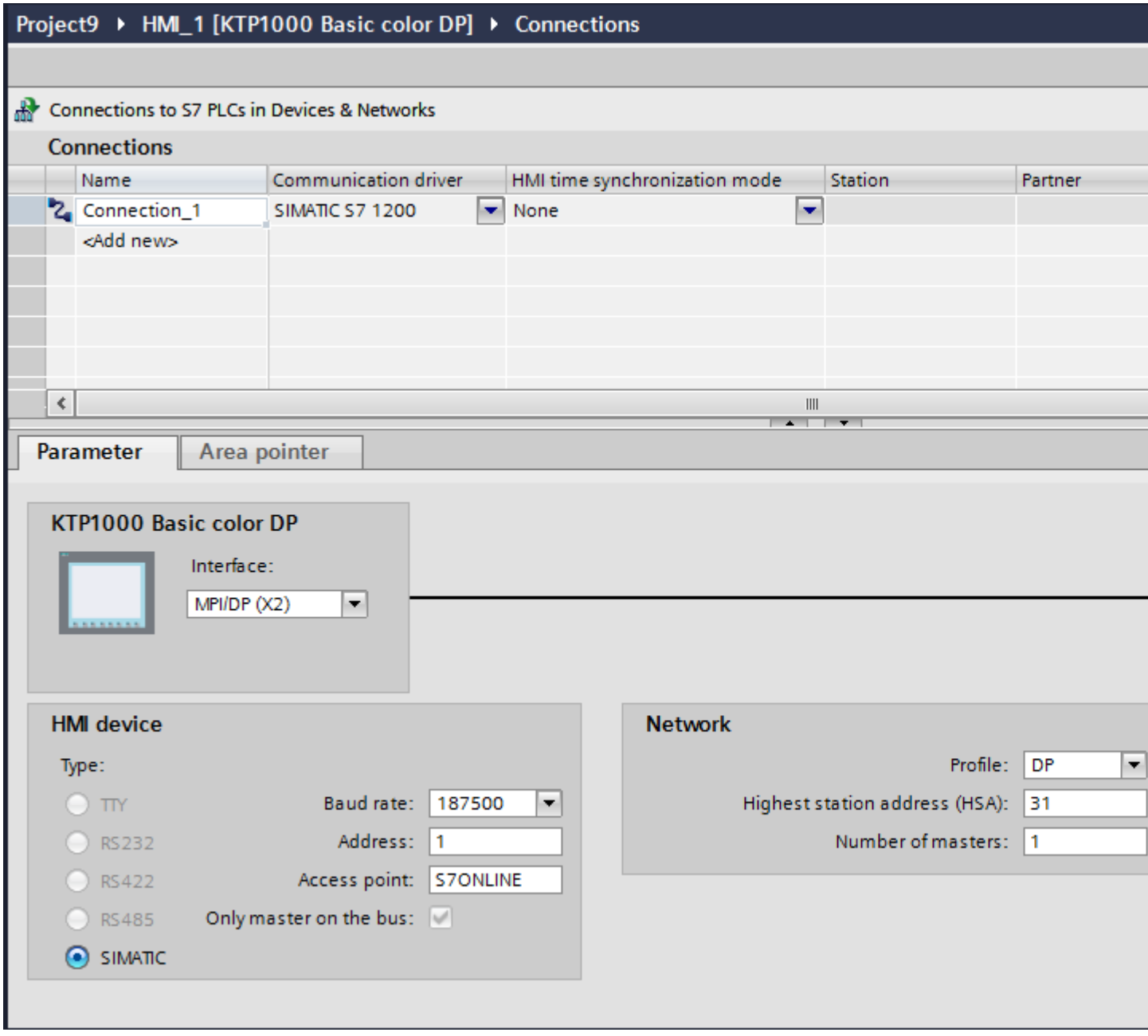
Procedure

1. Open the "Connections" editor of the HMI device.
2. Double-click "<Add>".



3. In the "Communication drivers" column, select the "SIMATIC S7 1200" driver.
4. Click the name of the connection.
5. Select the "MPI/DP" interface in the Inspector window under "Parameters".

6. Select the "DP" profile in the Inspector window under "Parameters > Network".



7. Set the addresses of the communication partners in the Inspector window:
   - HMI device: "Parameters > HMI device > Address"
   - PLC: "Parameters > PLC > Address"

See also

Communication with SIMATIC S7-1200 (Page 140)

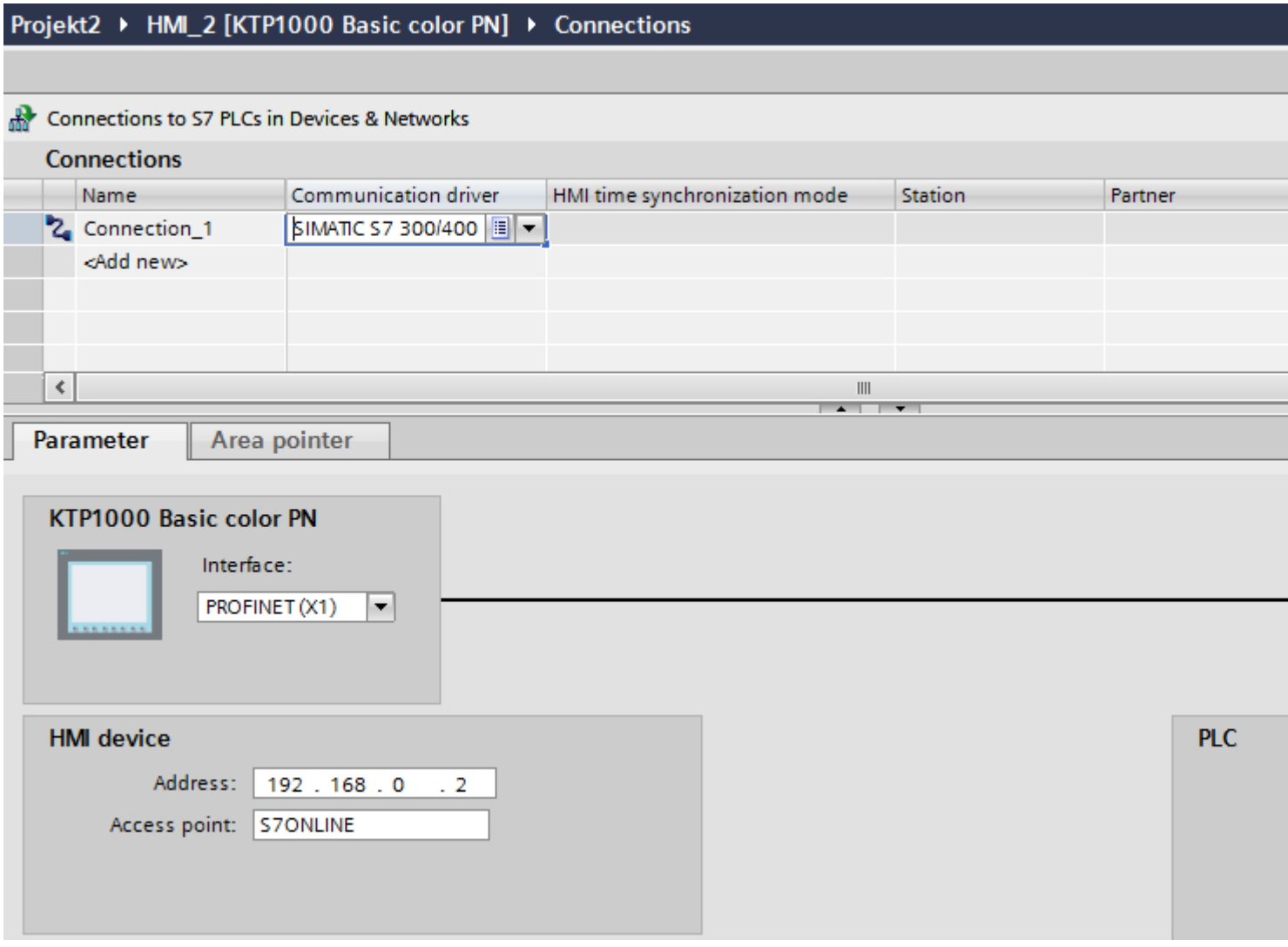
1.7.6.3 Parameters for the connection (Basic Panels)

Parameters for the connection (SIMATIC S7 1200) (Basic Panels)

Parameters to be set

To assign the connection parameters, such as addresses and profiles, click the connection that you have created in the "Connections" editor.

The communication partners are displayed schematically in the Inspector window under "Parameters". The "HMI device", "Network", and "PLC" areas are available for assigning parameters according to the interface used.



See also

Communication with SIMATIC S7-1200 (Page 140)

Ethernet parameters (Basic Panels)

Parameters for the HMI device

You set the parameters for the HMI device in the network under "HMI device".. The changes are not transferred automatically to the HMI device. You must change the settings in the Control Panel of the HMI device.

- "Interface"
  If you are directly connected to the HMI device during configuration, you can set up the IP address of the HMI device in WinCC.

  Note

  The IP address in the Control Panel will be overwritten upon subsequent loading if you have already set up the IP address in the HMI device control panel.

  The IP address already set up in the Control Panel will be retained upon subsequent loading if you activate "Set IP address using a different method".

  The IP address is transferred to the HMI device during project transfer.
  To set up the IP address of the HMI device:

  – Click the HMI device.
  – Open the "Device configuration" editor.
  – Click the Ethernet interface.
  – Assign the IP address in the inspector window under:
    "General > PROFINET interface > Ethernet addresses"

- "Address"
  You assign the IP address of the HMI device in the "Address" area.
  When you transfer the WinCC project to the HMI device, this IP address is set up directly in the HMI device.

- "Access point"
  Specifies the access point for the PG/PC interface that can be used to reach the communication partner.

Parameters for the PLC

Under "PLC", you address the S7 module with which the HMI device will exchange data. Assign a name for the connection for each communication partner.

- "Address"
  Under "Address", set the IP address of the S7 module to which the HMI device is connected.
- "Access password"
  Enter a password in the "Access password" field. This password must match the one you saved to the PLC.

Note

You only need a password if you have set "Complete protection" at the PLC.

No connection is set up to the PLC if the "Complete protection" security level is stored on the PLC and you do not enter a password.

See also

Communication with SIMATIC S7-1200 (Page 140)

PROFIBUS parameters (Basic Panels)

Parameters for the HMI device

You assign the parameters for the HMI device in the network once under "HMI device". The change applies to each communication partner.

- "Type"
  Specifies the physical connection used.
- "Interface"
  For "Interface", you select the HMI device interface via which the HMI device is connected to the PROFIBUS network.
- "Baud rate"
  For "Baud rate", you set the transmission speed of the data in the network. The baud rate is determined by the slowest HMI device connected to the network. The setting must be identical throughout the network.

Note

If you set a baud rate of 1.5 Mbaud for OP 73 or OP 77A, the highest station address must be less than or equal to 63.

- "Address"
  You set the PROFIBUS address of the HMI device under "Address". The PROFIBUS address must be unique in the PROFIBUS network.

- "Only master on bus"
  Disables an additional safety feature against bus faults when the HMI device is connected to the network. A passive station (slave) can only send data if it is requested to do so by an active station (master).
  In S7-200, you must set an HMI device as the master.
- "Access point"
  The access point defines a logical device name through which the communication partner can be reached.

Parameters for the network

Under "Network", you set the parameters for the PROFIBUS network to which the HMI device is linked.

- "Profile"
  For "Profile", you select the network profile that is used in the network. In "Profile", set "DP", "Universal", or "Standard". The setting must be identical throughout the network.
- "Highest address"
  For "Highest station address", set the highest station address. The highest station address must be greater than or equal to the highest actual PROFIBUS address. The setting must be identical throughout the network.

  Note

  If you set a baud rate of 1.5 Mbaud in the OP 73 or the OP 77A, the highest station address must be less than or equal 63.

- "Number of masters"
  For "Number of masters", set the number of masters in the PROFIBUS network. This information is necessary to correctly calculate the bus parameters.

Parameters for the PLC

Under "PLC", you address the S7 module with which the HMI device will exchange data. Assign a name for the connection for each communication partner.

- "Address"
  For "Address", set the PROFIBUS address of the S7 module (CPU, FM, or CP) to which the HMI device is connected.
- "Access password"
  Enter a password in the "Access password" field. This password must match the one you saved to the PLC.

  Note

  You only need a password if you have set "Complete protection" at the PLC.

  No connection is set up to the PLC if the "Complete protection" security level is stored on the PLC and you do not enter a password.

See also

Communication with SIMATIC S7-1200 (Page 140)

1.7.7 Configuring time synchronization (Basic Panels)

1.7.7.1 Time synchronization (Basic Panels)

Introduction

To have the same time of day throughout the plant, you can synchronize the time on various plant components using time synchronization. WinCC time synchronization is operated as a master-slave system.

One system component must be a clock for all components of a plant to work with identical time. The component functioning as the clock is referred to as the time master. The components that receive the time are time slaves.

Properties of time synchronization

- The HMI device can define the time as master or can accept the time of the PLC as slave.
- In "master mode", the time is synchronized at each connection setup.
- In "slave mode", the time is synchronized at each connection setup and than at cyclic intervals of 10 minutes.
- The first time synchronization is performed on the HMI device immediately after the start of runtime.
- Time synchronization is only performed on the HMI device during operation of runtime.

See also

Time synchronization restrictions (Page 205)

Configuring time synchronization for integrated connections (Page 206)

Configuring time synchronization for non-integrated connections (Page 207)

1.7.7.2 Time synchronization restrictions (Basic Panels)

Approved HMI devices

You can configure the time synchronization between a SIMATIC S7-1200 or SIMATIC S7-1500 and an HMI device with the following HMI devices:

Device	Operating system
Basic Panels	-
TP177 4"	Windows CE 5.0
Multi Panel 177	Windows CE 5.0
Multi Panel 277	Windows CE 5.0
Multi Panel 377	Windows CE 5.0
Mobile Panel 277	Windows CE 5.0
Mobile 277 IWLAN V2	Windows CE 5.0
Comfort Panels	Windows CE 6.0
PC systems with WinCC RT Advanced	Microsoft Windows

Configuration restrictions

- If an HMI device has several connections to SIMATIC S7-1200 or SIMATIC S7-1500, you can only configure one connection as "slave".
- If you have enabled time synchronization for the HMI device as "slave", you can no longer use the global area pointer "Date/time PLC".
- An HMI device can only request the time from a PLC with "Complete protection" security type configuration if the correct "Access password" is configured.
  Configure the "Access password" for communication with a PLC with "Complete protection" security type in the "Connections" editor of the HMI device.
  This "Access password" must match the password configured on the PLC. The PLC password is assigned in the PLC properties at: "General > Security"
- Basic Panels can only be configured as "Slave".
- If you use Basic Panels for the configuration, it is not possible to use time synchronization via NTP and the "Date/time PLC" area pointer simultaneously.
- Time synchronization with SIMATIC S7-1200 (V1.0) controllers is not possible.
- Time synchronization between the HMI device TP177 4" and SIMATIC S7-1200 (V4.0) controllers is not possible.
- Time synchronization between the HMI device TP177 4" and SIMATIC S7-1500 is not possible.

See also

Time synchronization (Page 204)

Configuring time synchronization for integrated connections (Page 206)

Configuring time synchronization for non-integrated connections (Page 207)

1.7.7.3 Configuring time synchronization for integrated connections

Introduction

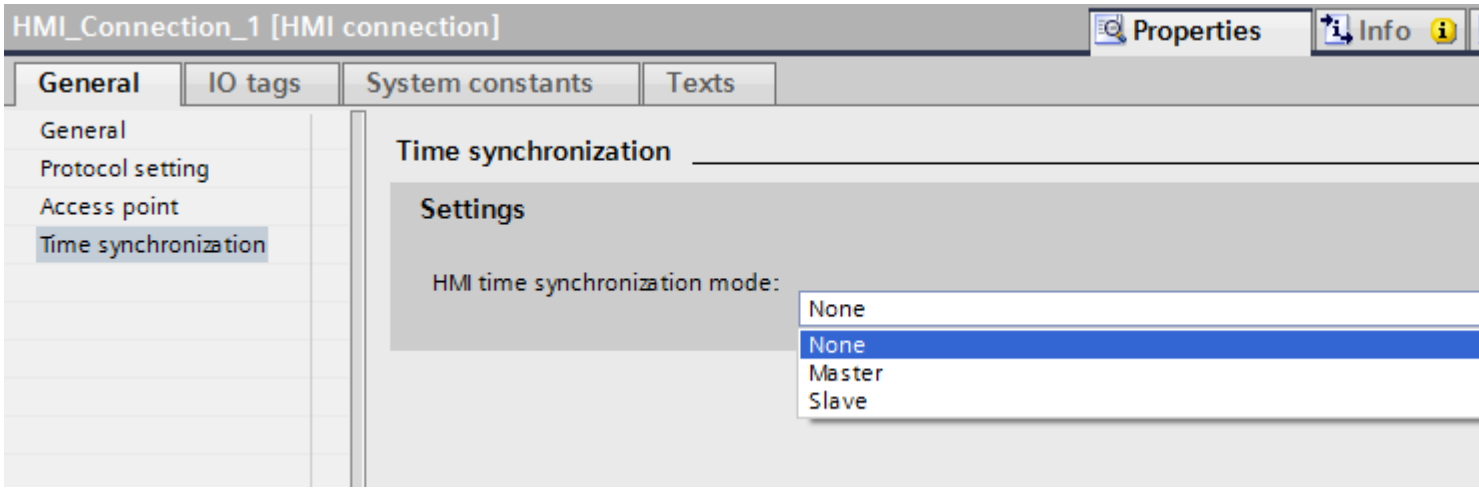
You configure time synchronization for an integrated connection in the "Devices & Networks" editor.

Requirements

- An HMI connection between an HMI device and a SIMATIC S7 1200 or SIMATIC S7 1500 has been configured.
- The HMI device must support the "time synchronization" function.
- The "Devices & Networks" editor is open.

Procedure

1. Click the line of the HMI connection in the "Devices & networks" editor.
2. Select the following in the inspector window under "General > Time synchronization > Settings":
   - None: No time synchronization is used.
   - Master: The HMI device sets the time.
   - Slave: The PLC sets the time.



See also

Time synchronization (Page 204)

Time synchronization restrictions (Page 205)

Configuring time synchronization for non-integrated connections (Page 207)

1.7.7.4 Configuring time synchronization for non-integrated connections (Basic Panels)

Introduction

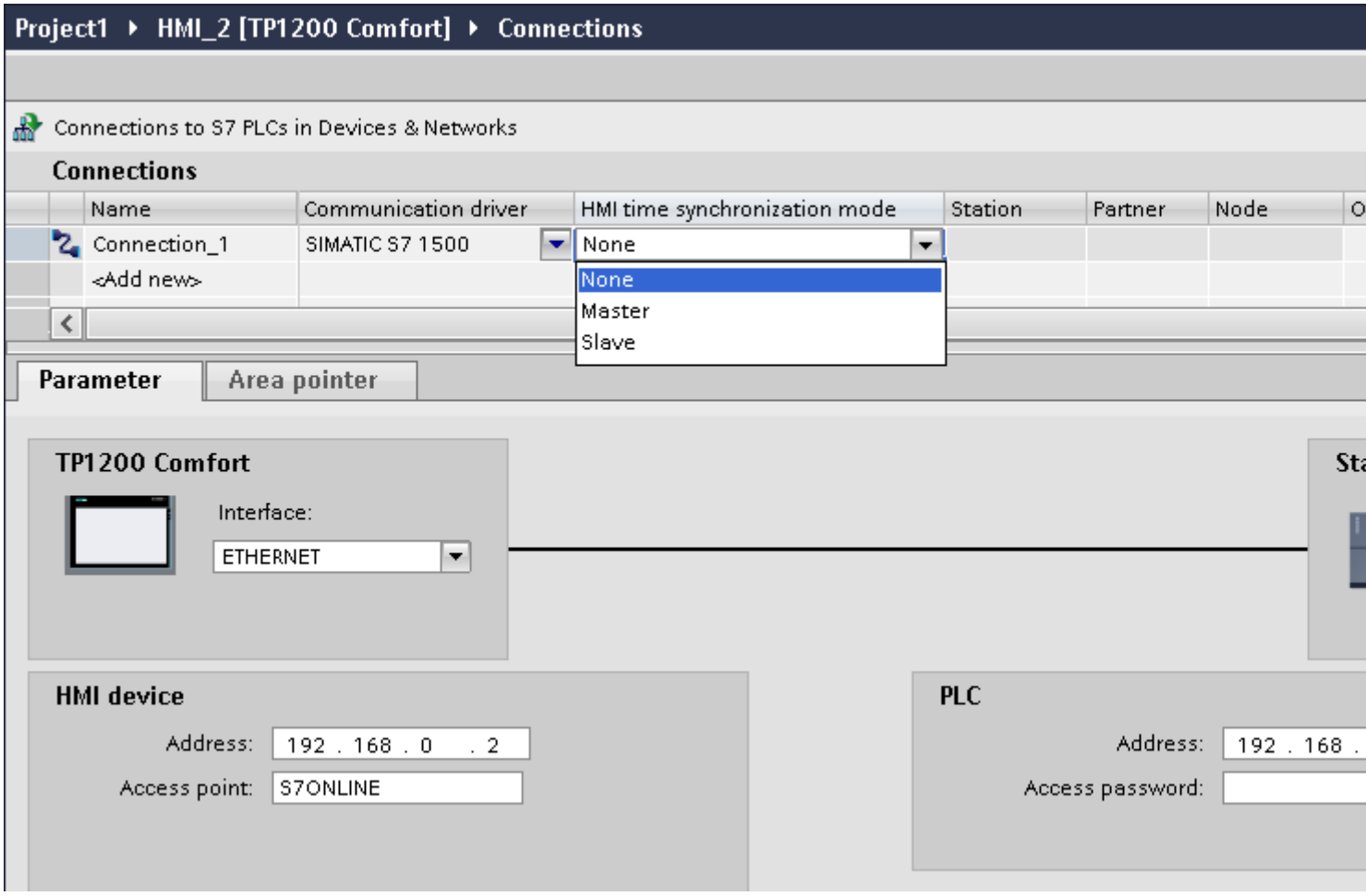
You configure time synchronization for a non-integrated connection in the "Connections" editor.

Requirements

- An HMI device which supports the "time synchronization" function has been created.
- "Connections" editor is open.

Procedure

1. Double-click "<Add>".
2. In the "Communication drivers" column, select the "SIMATIC S7 1500" PLC.
3. Select the following in the "HMI time synchronization mode" column:
   - None: No time synchronization is used.
   - Master: The HMI device sets the time.
   - Slave: The PLC sets the time.



See also

Time synchronization (Page 204)

Time synchronization restrictions (Page 205)

Configuring time synchronization for integrated connections (Page 206)

1.8 Communicating with SIMATIC S7 300/400 (Basic Panels)

1.8.1 Communication with SIMATIC S7 300/400 (Basic Panels)

Introduction

This section describes the communication between an HMI device and the SIMATIC S7 300 and S7 400 PLCs. These two PLCs will be referred to jointly as SIMATIC S7 300/400.

You can configure the following communication channels for the SIMATIC S7 300/400 PLC:

- PROFINET
- PROFIBUS
- MPI

HMI connection for communication

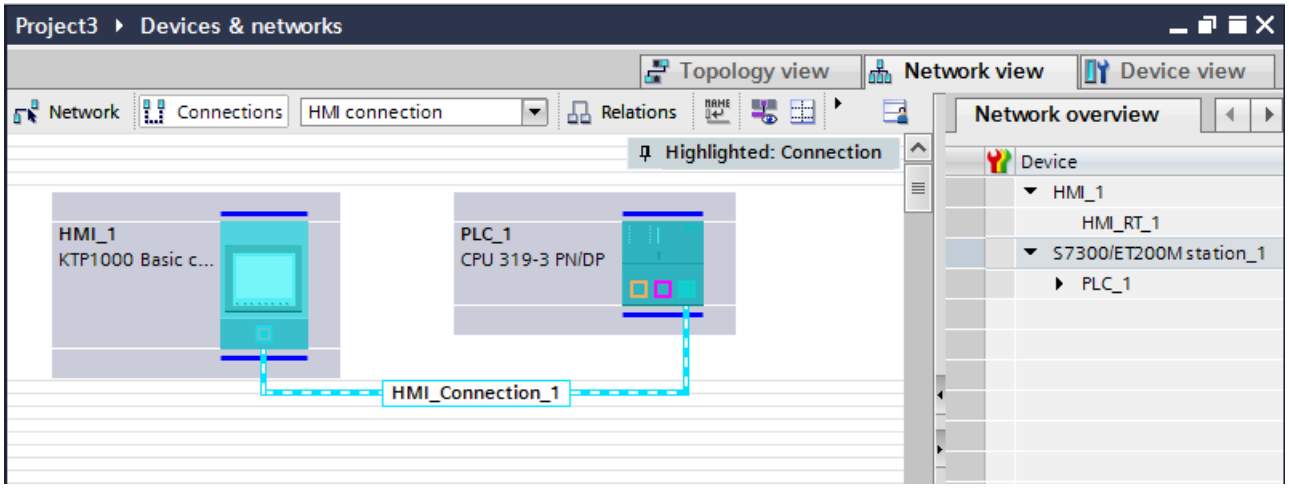
You configure connections between the HMI device and a SIMATIC S7 300/400 in the "Devices & Networks" editor.

1.8.2 Communication via PROFINET (Basic Panels)

1.8.2.1 Communication via PROFINET (Basic Panels)

HMI connections via PROFINET

If you have inserted an HMI device and a SIMATIC S7 300/400 into the project, you interconnect the two PROFINET interfaces in the "Devices & Networks" editor.



You can also connect multiple HMI devices to one SIMATIC S7 300/400 and multiple SIMATIC S7 300/400s to one HMI device. The maximum number of communication partners that you can connect to an HMI device is dependent on the HMI device used.

Additional information is available in the documentation for the respective HMI device.

HMI connection in the "Devices & Networks" editor

You configure the HMI connection between the PLC and the HMI device via PROFINET in the "Devices & Networks" editor.

Connection in the "Connections" editor

Alternatively, you configure the connection between the PLC and HMI device via PROFINET in the "Connections" editor of the HMI device.

See also

Communication with SIMATIC S7 300/400 (Page 209)

Configuring an HMI connection via PROFINET (Page 210)

1.8.2.2 Configuring an HMI connection via PROFINET (Basic Panels)

Introduction

You configure an HMI connection between HMI devices and a SIMATIC S7 300/400 via PROFINET or Ethernet in the "Devices & Networks" editor.

⚠ 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.

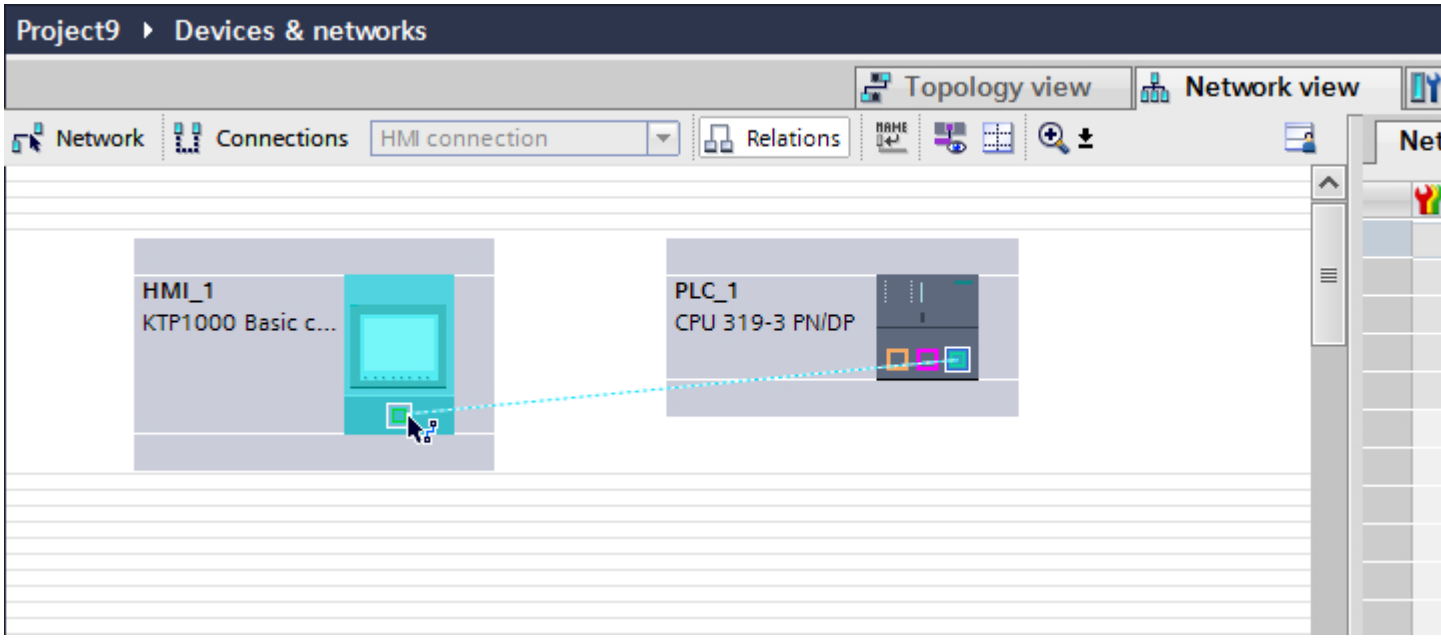
Requirements

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

- HMI device with PROFINET or Ethernet interface
- SIMATIC S7 300/400 with PROFINET interface.

Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed graphically in the network view.
2. Click the "Connections" button and select "HMI connection" for the connection type.
   The devices available for connection are highlighted in color.
3. Click the PROFINET interface of the PLC and use a drag-and-drop operation to draw a connection to the PROFINET or Ethernet interface of the HMI device.



4. Click the connecting line.
5. Click "Highlight HMI connection" and select the HMI connection.
   The connection is displayed graphically in the Inspector window.
6. Click the communication partners in the "Network view" and change the PROFINET parameters in the Inspector window according to the requirements of your project.
   For additional information on this, see section "PROFINET parameters (Page 212)".

Note

The created HMI connection is also shown in the tabular area of the editor on 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 a SIMATIC S7 300/400. The IP address and subnet mask connection parameters are configured.

See also

Communication with SIMATIC S7 300/400 (Page 209)

Communication via PROFINET (Page 209)

PROFINET parameters (Page 212)

1.8.2.3 PROFINET parameters (Basic Panels)

PROFINET parameters for the HMI connection (Basic Panels)

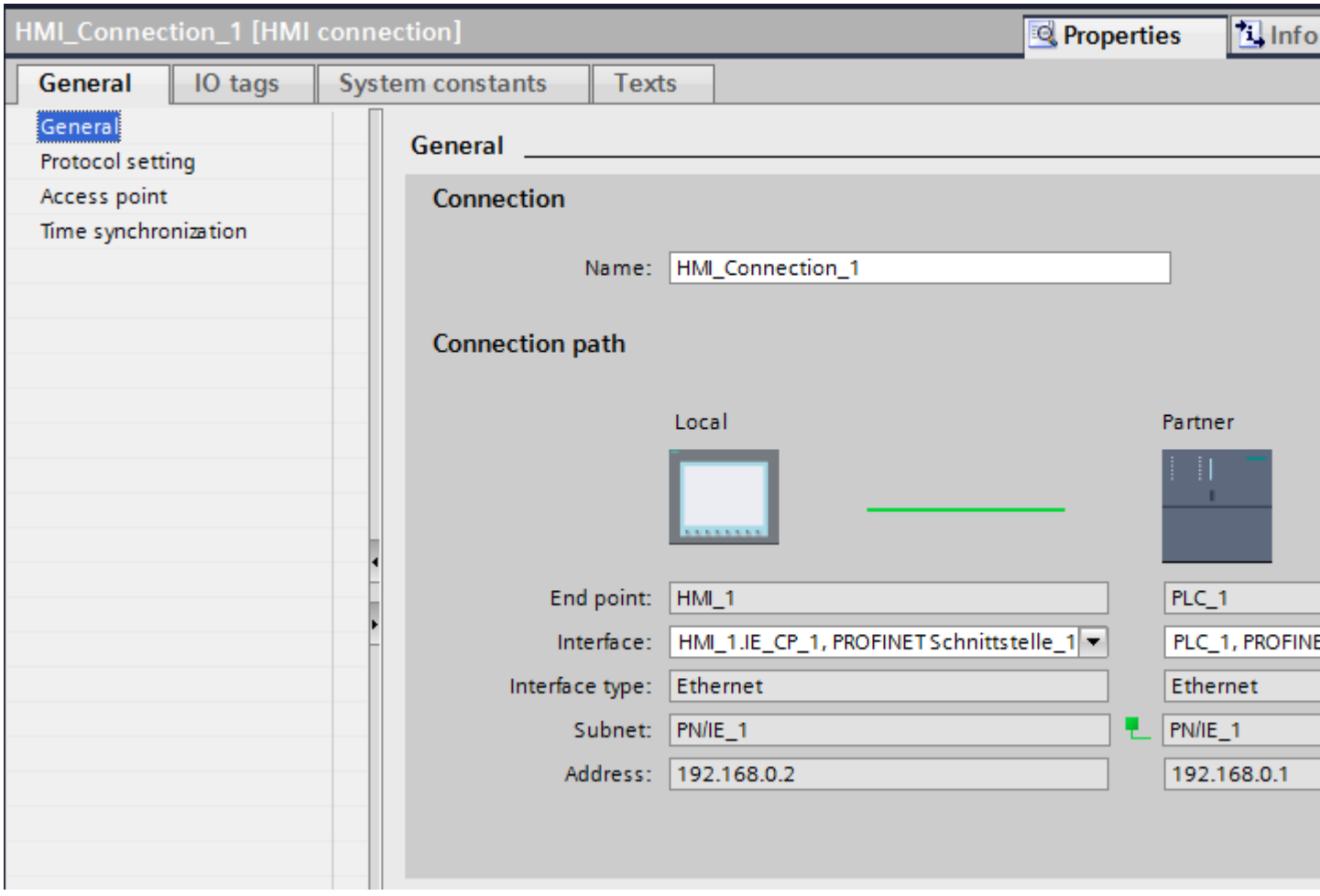
PROFINET parameters for the HMI connection

An overview of the configured HMI connection parameters can be found in the properties for the HMI connection.

Only limited changes are possible in this Inspector window.

Displaying and changing the HMI connection parameters

1. Click the HMI connection in the "Devices & Networks" editor.
2. Change the parameters of the HMI connection in the Inspector window under "Properties > General > General".



"Connection"

The "Connection" area displays the HMI connection created for communication between the devices.

You can edit the name of the HMI connection in this area.

"Connection path"

The communication partners of the selected HMI connection and the associated PROFINET parameters are displayed in the "Connection path" area. Some of the areas displayed cannot be edited in this dialog.

- "End point"
  Displays the device name. This area cannot be edited.
- "Interface"
  Displays the selected interface of the device. You can choose between several interfaces, depending on the device.
- "Interface type"
  Displays the selected interface type. This area cannot be edited.
- "Subnet"
  Displays the selected subnet. This area cannot be edited.
- "Address"
  Displays the selected IP address of the device. This area cannot be edited.
- "Find connection path" button
  Enables the subsequent specification of connections.

See also

Communication via PROFINET (Page 209)

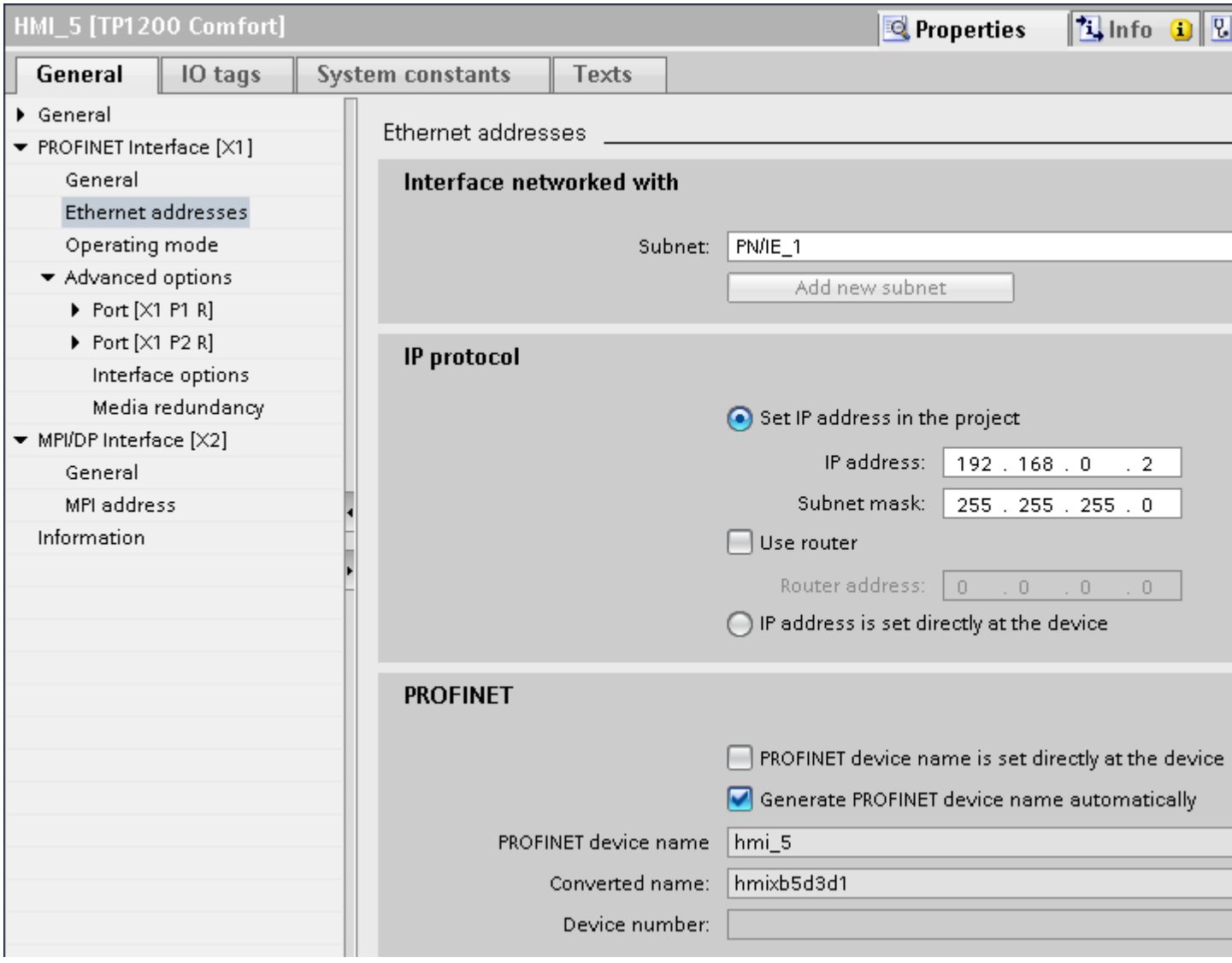
PROFINET parameters for the HMI device (Basic Panels)

PROFINET parameters for the HMI device

An overview of the configured HMI device parameters can be found in the properties for the HMI device.

Displaying and changing PROFINET parameters of the HMI device

1. Click the HMI device in the "Devices & Networks" editor.
2. Change the parameters of the HMI device in the Inspector window under "Properties > General".



"Interface networked with"

In the "Interface networked with" area, select the subnet of the HMI connection via which the HMI device is connected to the network. You use the "Add new subnet" button to create a new subnet.

"IP protocol"

- "Set IP address in the project"
  When you transfer the WinCC project to the HMI device, this IP address is set up directly in the HMI device.

  Note

  The device is automatically restarted in the case of HMI devices with the Windows CE 3.0 operating system.

  HMI devices with Windows CE 3.0:
  - OP 77B
  - TP 177B color PN/DP
  - TP 177B mono DP
  - OP 177B color PN/DP
  - OP 177B mono DP
  - Mobile Panel 177 PN
  - Mobile Panel 177 DP
  - TP 277 6"
  - OP 277 6"

- "Subnet mask"
  You assign data of the subnet mask in the "Subnet mask" area.
- "Use IP router"
  If you are using an IP router, select "Use IP router" and enter the router address in the "Router address" field.
- "Set IP address using a different method"
  If the function "Set IP address using a different method" is activated, the IP address is not taken from the project. You have to enter the IP address directly in the Control Panel of the HMI device.

See also

Communication via PROFINET (Page 209)

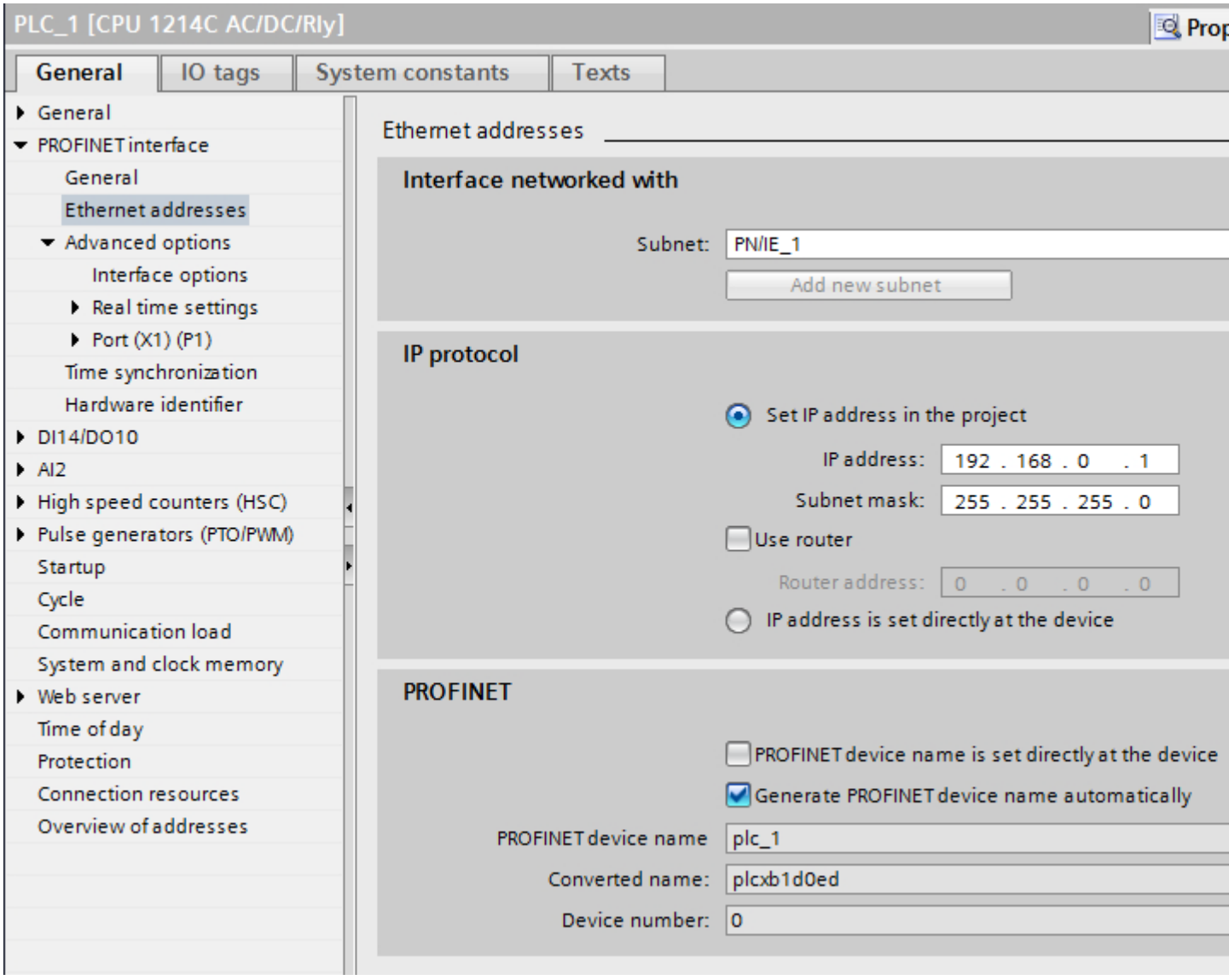
PROFINET parameters for the PLC (Basic Panels)

PROFINET parameters for the PLC

An overview of the configured parameters can be found in the properties for the PLC.

Displaying and changing PROFINET parameters of the PLC

1. Click the PLC in the "Devices & Networks" editor.
2. Change the parameters of the PLC in the Inspector window under "Properties > General > General".



"Interface networked with"

In the "Subnet" area, select the subnet of the HMI connection via which the PLC is connected to the network. You use the "Add new subnet" button to create a new subnet.

"IP protocol"

- "Interface type"
  Depending on the HMI device type, you have various interfaces to choose from.
- "IP address"
  You assign the IP address of the HMI device in the "IP address" area.
- "Subnet mask"
  You assign data of the subnet mask in the "Subnet mask" area.
  If you are using an IP router, select "Use IP router" and enter the router address in the field.

See also

Communication via PROFINET (Page 209)

Configuring Industrial Ethernet (Basic Panels)

Rules for the network configuration

The Ethernet interfaces of the devices have a default IP address that you can change.

IP address

The IP parameters are visible if the communication-capable devices support the TCP/IP protocol.

The IP address consists of 4 decimal figures in the range of 0 to 255. The decimal figures are separated from one another by a dot.

Example: 140.80.0.2

The IP address consists of the following:

- The address of the (sub) net
- The address of the node (generally also called host or network node)

Subnet mask

The subnet mask splits these two addresses. It determines which part of the IP address addresses the network and which part of the IP address addresses the node.

The set bits of the subnet mask determine the network part of the IP address.

Example:

Subnet mask: 255.255.0.0 = 11111111.11111111.00000000.00000000

In the example given for the above IP address, the subnet mask shown here has the following meaning:

The first 2 bytes of the IP address identify the subnet - i.e. 140.80. The last two bytes address the node, thus 0.2.

It is generally true that:

- The network address results from AND linking the IP address and subnet mask.
- The node address results from AND NOT linking the IP address and subnet mask.

Relation between IP address and default subnet mask

An agreement exists relating to the assignment of IP address ranges and so-called "Default subnet masks". The first decimal number (from the left) in the IP address determines the structure of the default subnet mask. It determines the number of "1" values (binary) as follows:

IP address (decimal)	IP address (binary)	Address class	Default subnet mask
0 to 126	0xxxxxxx.xxxxxxxx....	A	255.0.0.0
128 to 191	10xxxxxx.xxxxxxxx...	B	255.255.0.0
192 to 223	110xxxxx.xxxxxxxx...	C	255.255.255.0

Note

Range of values for the first decimal point

A value of between 224 and 255 is also possible for the first decimal number of the IP address (address class D etc). This is, however, not recommended because there is no address check for these values.

Masking other subnets

You can use the subnet mask to add further structures and form "private" subnets for a subnet that is assigned one of the address classes A, B or C. This is done by setting other lower points of the subnet mask to "1". For each bit set to "1", the number of "private" networks doubles and the number of nodes they contain is halved. Externally, the network functions like an individual network as it did previously.

Example:

You have a subnet of address class B (e.g. IP address 129.80.xxx.xxx) and change the default subnet mask as follows:

Masks	Decimal	Binary
Default subnet mask	255.255.0.0	11111111.11111111.00000000.00000000
Subnet mask	255.255.128.0	11111111.11111111.10000000.00000000

Result:

All nodes with addresses between 129.80.001.xxx and 129.80.127.xxx are on one subnet, all nodes with addresses between 129.80.128.xxx and 129.80.255.xxx are on another subnet.

Router

The job of the routers is to connect the subnets. If an IP datagram is to be sent to another network, it first has to be conveyed to a router. To make this possible, in this case you have to enter the address of the router for each node in the subnet.

The IP address of a node in the subnet and the address of the router may only differ at the points at which there is a "0" in the subnet mask.

See also

Communication via PROFINET (Page 209)

1.8.2.4 Setting port options (Basic Panels)

Setting the port options (Basic Panels)

Changing connection settings for the PROFINET IO port

You can change the network settings for the PROFINET IO port as required. By default, the settings are made automatically. In normal situations, this guarantees problem-free communication.

Possible settings for transmission rate / duplex

Depending on the selected device, you can make the following settings for "Transmission rate / duplex":

- Automatic setting
  Recommended default setting of the port. The transmission settings are automatically "negotiated" with the peer port. The "Enable autonegotiation" option is also enabled as a default, in other words, you can use cross cables or patch cables for the connection.
- TP/ITP at x Mbps full duplex (half duplex)
  Setting of the transmission rate and the full duplex/half duplex mode. The effectiveness depends on the "Enable autonegotiation" setting:
  - Autonegotiation enabled
    You can use both cross cable and patch cable.
  - Autonegotiation disabled
    Make sure that you use the correct cable (patch cable or cross cable)! The port is also monitored with this setting.
- Deactivated
  Depending on the module type, the drop down list box can contain the "- Disabled -" option. This option, for example, allows you to prevent access to an unused port for security reasons. With this setting, diagnostic events are not generated.

"Monitor" option

This option enables or disables port diagnostics. Examples of port diagnostics: The link status is monitored, in other words, the diagnostics are generated during link-down and the system reserve is monitored in the case of fiber optic ports.

Option "Enable autonegotiation "

The autonegotiation setting can only be changed if a concrete medium (for example, TP 100 Mbps full duplex) is selected. Whether or not a concrete medium can be set depends on the properties of the module.

If autonegotiation is disabled, this causes the port to be permanently specified, as for example, is necessary for a prioritized startup of the IO device.

You must make sure the partner port has the same settings because with this option the operating parameters of the connected network are not detected and the data transmission rate and transmission mode can accordingly not be optimally set.

Note

When a local port is connected, STEP 7 makes the setting for the partner port if the partner port supports the setting. If the partner port does not accept the setting, an error message is generated.

See also

Wiring rules for disabled autonegotiation (Page 221)

Boundaries at the port (Page 223)

Wiring rules for disabled autonegotiation (Basic Panels)

Requirements

You have made the following settings for the port in question, for example, to accelerate the startup time of the IO device:

- Fixed transmission rate
- Autonegotiation incl. autocrossing disabled

The time for negotiating the transmission rate during startup has been saved.

If you have disabled autonegotiation, you must observe the wiring rules.

Wiring rules for disabled autonegotiation

PROFINET devices have the following two types of ports:

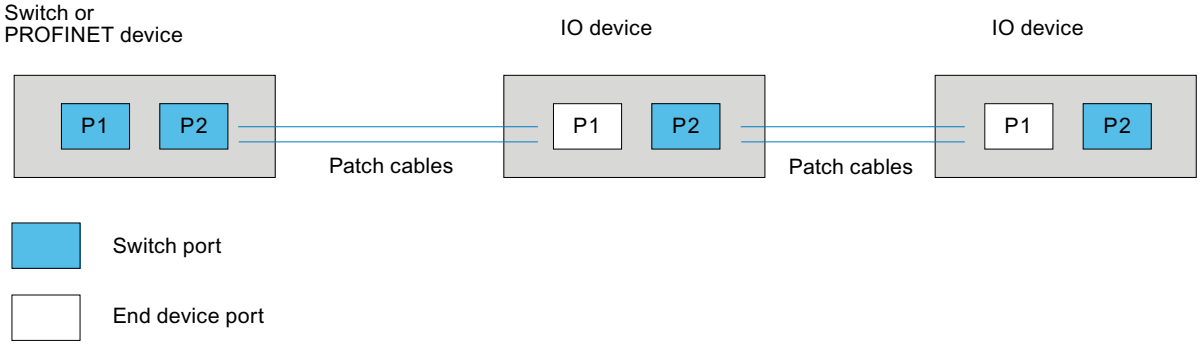
Type of port	PROFINET devices	Note
Switch port with crossed pin assignment	For IO devices: Port 2 For S7 CPUs with 2 ports: Ports 1 and 2	Crossed pin assignment means that the pin assignment for the ports for sending and receiving between the respective PROFINET devices is exchanged internally.
End device port with uncrossed pin assignment	For IO devices: Port 1 For S7 CPUs with one port: Port 1	-

Validity of the wiring rules

The cabling rules described in the following paragraph apply exclusively for the situation in which you have specified a fixed port setting.

Rules for cabling

You can connect several IO devices in line using a single cable type (patch cable). To do this, you connect port 2 of the IO device (distributed I/O) with port 1 of the next IO device. The following graphic gives an example with two IO devices.



See also

Setting the port options (Page 220)

Boundaries at the port (Page 223)

Boundaries at the port (Basic Panels)

Requirements

The respective device must support boundaries settings in order to work with boundaries. If the device for PROFINET does not support boundaries settings, the corresponding parameters are disabled.

Enable boundaries

"Boundaries" are limits for transmission of certain Ethernet frames. The following boundaries can be set at a port:

- "End of discovery of accessible devices"
  No forwarding of DCP frames to identify accessible devices. Devices downstream from this port cannot be reached by the project tree under "Accessible devices". Devices downstream from this port cannot be reached by the CPU.
- "End of topology discovery"
  LLDP frames (Link Layer Discovery Protocol) are not forwarded for topology detection.
- "End of sync domain"
  No forwarding of sync frames transmitted to synchronize nodes within a sync domain.
  If you operate, for example, a PROFINET device with more than two ports in a ring, you should prevent the sync frame from being fed into the ring by setting a sync boundary (at the ports not inside the ring).
  Additional example: If you want to use several sync domains, configure a sync domain boundary for the port connected to a PROFINET device from the other sync domain.

Restrictions

The following restrictions must be observed:

- The individual check boxes can only be used if the port supports the function in question.
- If a partner port has been determined for the port, the following check boxes cannot be used:
  - "End of discovery of accessible devices"
  - "End of topology discovery"

See also

Setting the port options (Page 220)

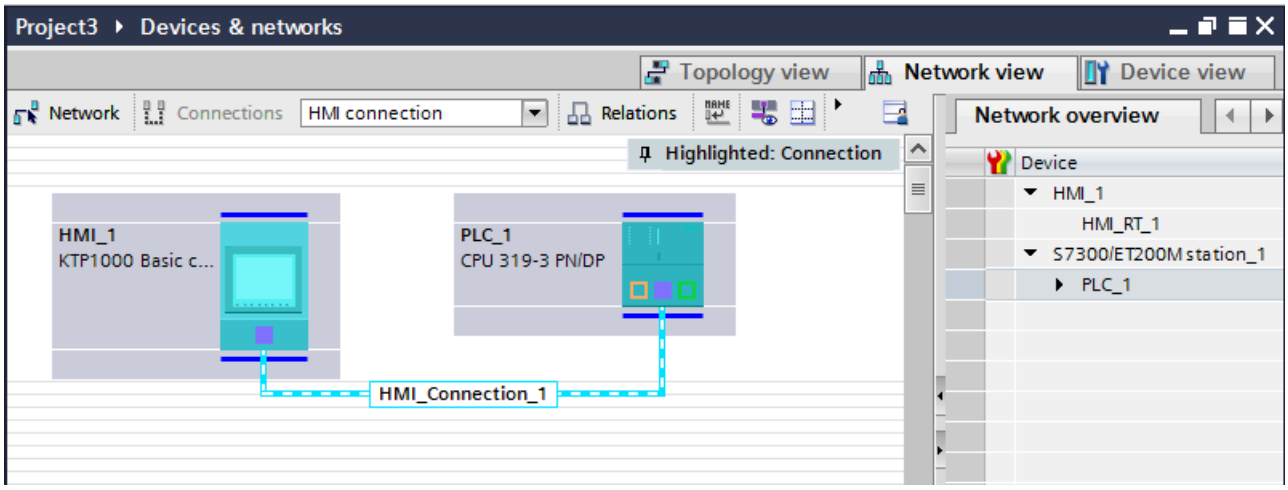
Wiring rules for disabled autonegotiation (Page 221)

1.8.3 Communication via PROFIBUS (Basic Panels)

1.8.3.1 Communication via PROFIBUS (Basic Panels)

HMI connections via PROFIBUS

If you have inserted an HMI device and a SIMATIC S7 300/400 into the project, you interconnect the two PROFIBUS interfaces in the "Devices & Networks" editor.



HMI connection in the "Devices & Networks" editor

You configure the HMI connection between the PLC and the HMI device via PROFIBUS in the "Devices & Networks" editor.

Connection in the "Connections" editor

Alternatively, you configure the connection between the PLC and HMI device in the "Connections" editor of the HMI device.

See also

Communication with SIMATIC S7 300/400 (Page 209)

Configuring an HMI connection via PROFIBUS (Page 224)

1.8.3.2 Configuring an HMI connection via PROFIBUS (Basic Panels)

Introduction

You configure an HMI connection between HMI devices and a SIMATIC S7 300/400 via PROFIBUS in the "Devices & Networks" editor.

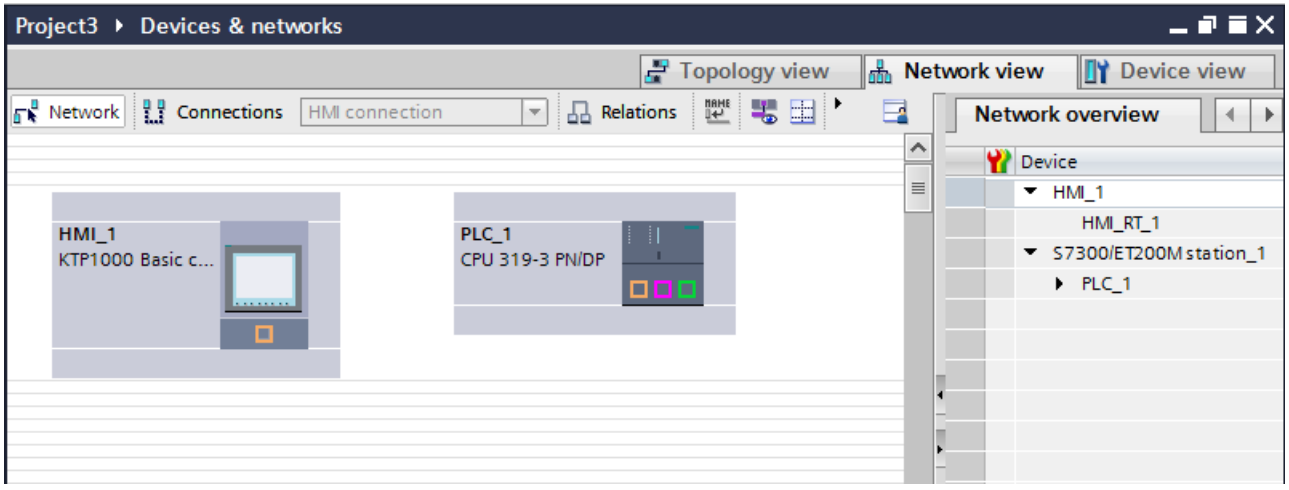
Requirements

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

- HMI device with MPI/DP interface
- SIMATIC S7 300/400 with PROFIBUS interface

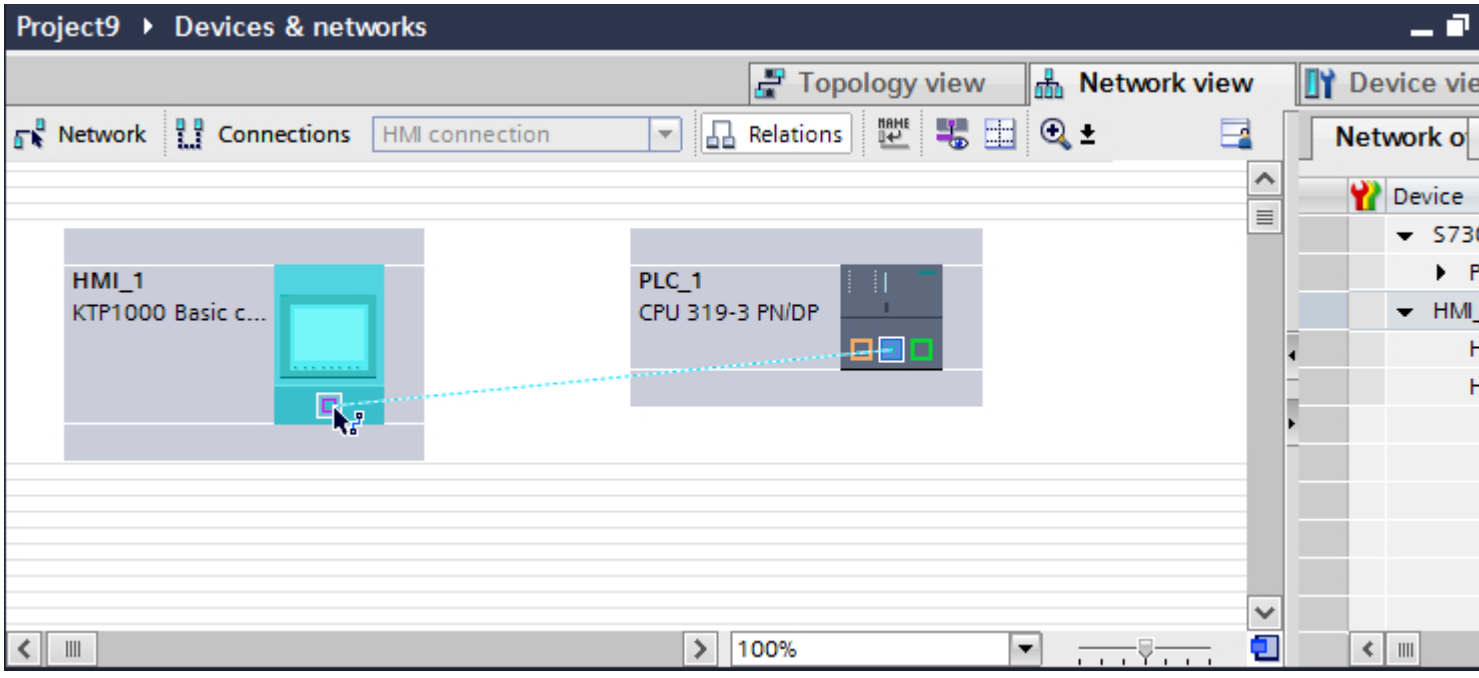
Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed graphically in the network view.



2. Click the "Connections" button.
   The devices available for connection are highlighted in color.
3. Click the HMI device interface.
4. Select the "PROFIBUS" interface type in the Inspector window under "Properties > General > HMI MPIDP > Parameters".

5. Click the interface of the PLC and use a drag-and-drop operation to draw a connection to the HMI device.



6. Click the connecting line.
7. Click "Highlight HMI connection" and select the HMI connection.
   The connection is displayed graphically in the Inspector window.
8. Click the communication partners in the "Network view" and change the PROFINET parameters in the Inspector window according to the requirements of your project.
   See the "PROFIBUS parameters (Page 227)" section for more detailed information.

Note

The created HMI connection is also shown in the tabular area of the editor on 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 an HMI connection between an HMI device and a SIMATIC S7 300/400 via PROFIBUS.

See also

Communication with SIMATIC S7 300/400 (Page 209)

Communication via PROFIBUS (Page 224)

PROFIBUS parameters (Page 227)

1.8.3.3 PROFIBUS parameters (Basic Panels)

PROFIBUS parameters for the HMI connection (Basic Panels)

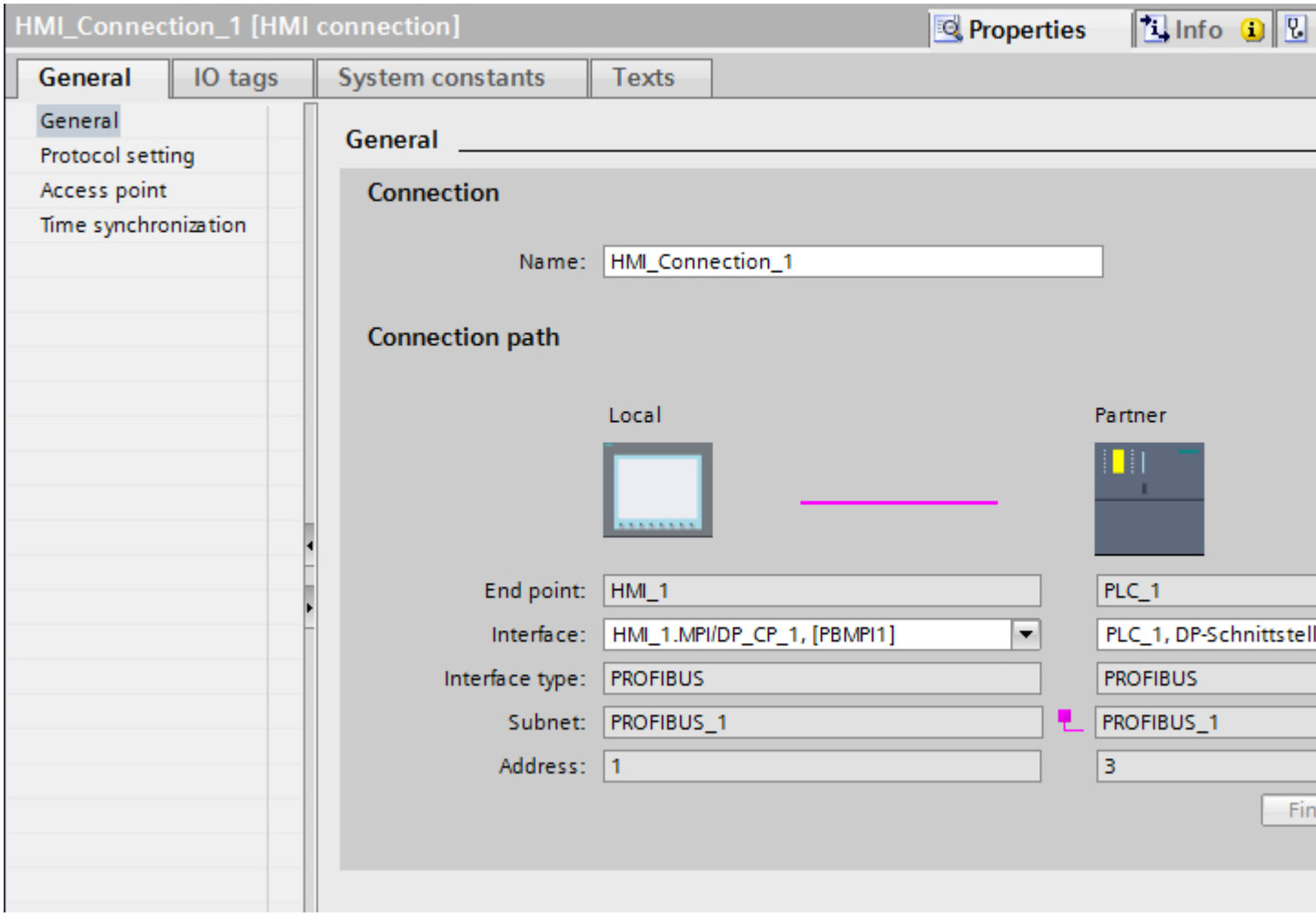
PROFIBUS parameters for the HMI connection

An overview of the configured HMI connection parameters can be found in the properties for the HMI connection.

Only limited changes are possible in this Inspector window.

Displaying and changing the HMI connection parameters

1. Click the HMI connection in the "Devices & Networks" editor.
2. Change the parameters of the HMI connection in the Inspector window under "Properties > General > General".



"Connection"

The name of the connection is displayed in the "Connection" area. The name is generated automatically when the connection is created. You can change the name as required.

"Connection path"

The communication partners of the selected HMI connection and the associated PROFIBUS parameters are displayed in the "Connection path" area. Some of the areas displayed cannot be edited in this dialog.

- "End point"
  Displays the device name. This area cannot be edited.
- "Interface"
  Displays the selected interface of the device. You can choose between several interfaces, depending on the device.
- "Interface type"
  Displays the selected interface type. This area cannot be edited.
- "Subnet"
  Displays the selected subnet. This area cannot be edited.
- "Address"
  Displays the PROFIBUS address of the device. This area cannot be edited.
- "Find connection path" button
  Enables the subsequent specification of connections.

See also

Communication via PROFIBUS (Page 224)

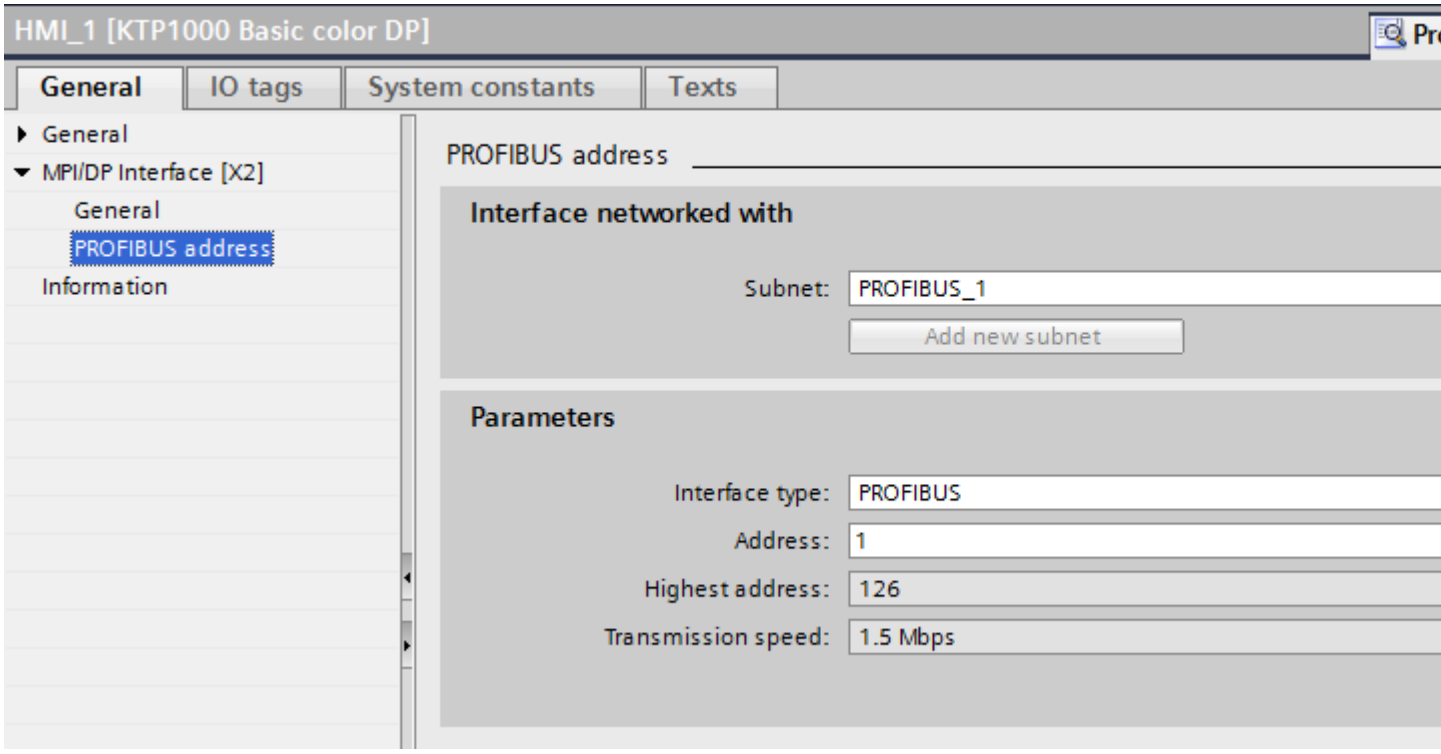
PROFIBUS parameters for the HMI device (Basic Panels)

PROFIBUS parameters for the HMI device

An overview of the configured HMI device parameters can be found in the properties for the HMI device.

Displaying and changing PROFINET parameters of the HMI device

1. Click the HMI device in the "Devices & Networks" editor.
2. Change the parameters of the HMI device in the Inspector window under "Properties > General > General".



"Interface networked with"

In the "Interface networked with" area, select the subnet of the HMI connection via which the HMI device is connected to the network. You use the "Add new subnet" button to create a new subnet.

"Parameters"

- "Interface type"
  You assign the interface type in the "Interface type" area. Depending on the HMI device type, you have various interfaces to choose from.
- "Address"
  You assign the PROFIBUS address of the HMI device in the "Address" area. The PROFIBUS address must be unique throughout the PROFIBUS network.
- "Highest address"
  The "Highest address" area displays the highest address of the PROFIBUS network.
- "Transmission speed"
  The "Transmission speed" is determined by the slowest device connected to the network. The setting is identical throughout the network.

See also

Communication via PROFIBUS (Page 224)

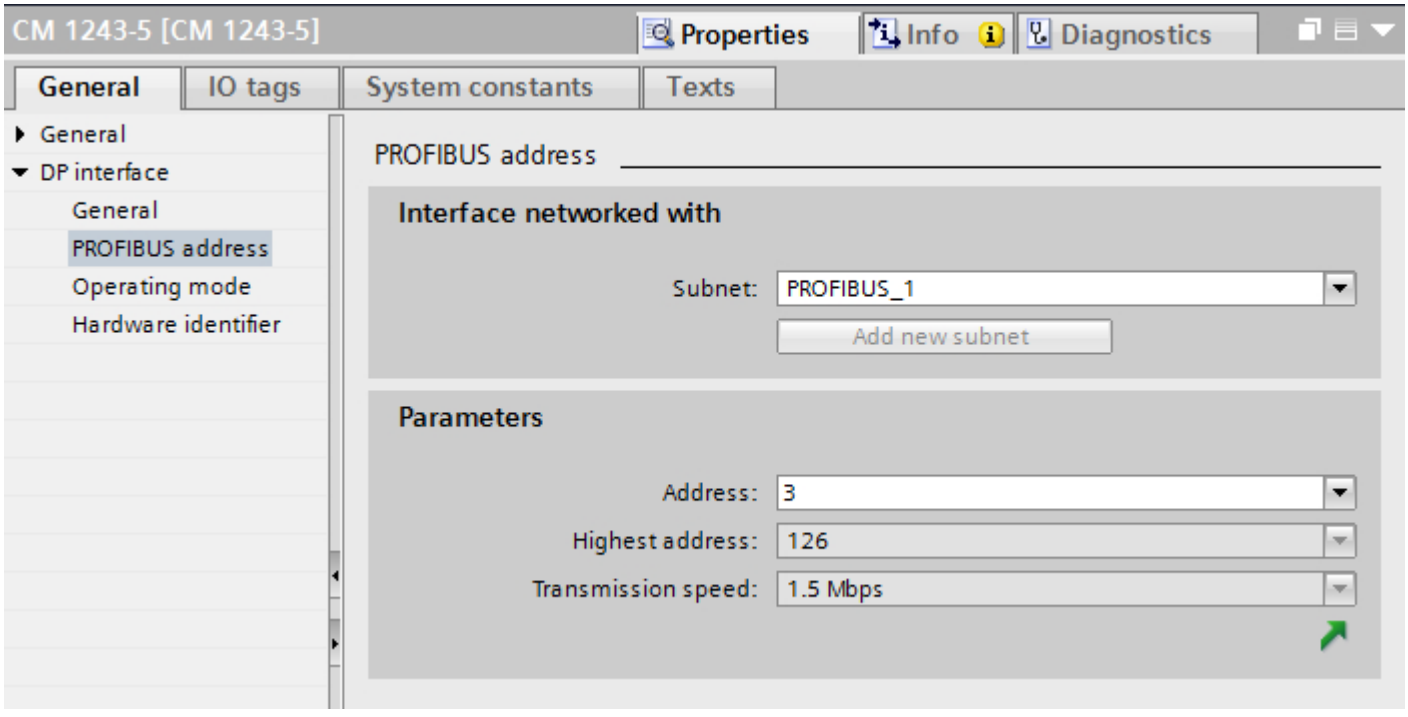
PROFIBUS parameters for the PLC (Basic Panels)

PROFIBUS parameters for the PLC

An overview of the configured parameters can be found in the properties for the PLC.

Displaying and changing PROFIBUS parameters of the PLC

1. Click the PLC in the "Devices & Networks" editor.
2. Change the parameters of the PLC in the Inspector window under "Properties > General > General".



"Interface networked with"

In the "Subnet" area, select the subnet of the HMI connection via which the PLC is connected to the network. You use the "Add new subnet" button to create a new subnet.

"Parameters"

- "Interface type"
  Depending on the HMI device type, you have various interfaces to choose from.
- "Address"
  You assign the PROFIBUS address of the HMI device in the "Address" area. The PROFIBUS address must be unique throughout the PROFIBUS network.
- "Highest address"
  The "Highest address" area displays the highest address of the PROFIBUS network.
- "Transmission speed"
  The "Transmission speed" is determined by the slowest device connected to the network. The setting is identical throughout the network.

See also

Communication via PROFIBUS (Page 224)

Bus profiles with PROFIBUS (Basic Panels)

Introduction

Depending on the device types connected and protocols used on the PROFIBUS, different profiles are available. The profiles differ in terms of the setting options and calculation of bus parameters. The profiles are explained below.

Devices with different profiles on the same PROFIBUS subnet

The PROFIBUS subnet only functions without problem if the bus parameters of all devices have the same values.

Profiles and transmission rates

Profiles	Supported transmission speeds in Kbits/s
DP	9,6 19,2 45,45 93,75 187,5 500 1500 3000 6000 12000
Standard	9,6 19,2 45,45 93,75 187,5 500 1500 3000 6000 12000
Universal	9,6 19,2 93,75 187,5 500 1500

Meaning of profiles

Profile	Meaning
DP	Select the "DP" bus profile when the only devices connected to the PROFIBUS subnet are those which satisfy the requirements of standard EN 50170 Volume 2/3, Part 8-2 PROFIBUS. The bus parameter setting is optimized on these devices. This includes devices with DP master and DP slave interfaces of the SIMATIC S7 and distributed I/Os of other manufacturers.
Standard	Compared to the "DP" profile, the "Standard" profile also offers scope for devices of another project or devices which have not been configured here to be taken into account when calculating the bus parameters. The bus parameters are then calculated following a simple, non-optimized algorithm.
Universal	Select the "Universal" bus profile when individual devices on the PROFIBUS subnet use the PROFIBUS-FMS service. This includes the following devices for example: • CP 343-5 • PROFIBUS-FMS devices of other manufacturers As with the "Standard" profile, this profile allows you to take other devices into account when calculating the bus parameters.

See also

Communication via PROFIBUS (Page 224)

1.8.4 Communication via MPI (Basic Panels)

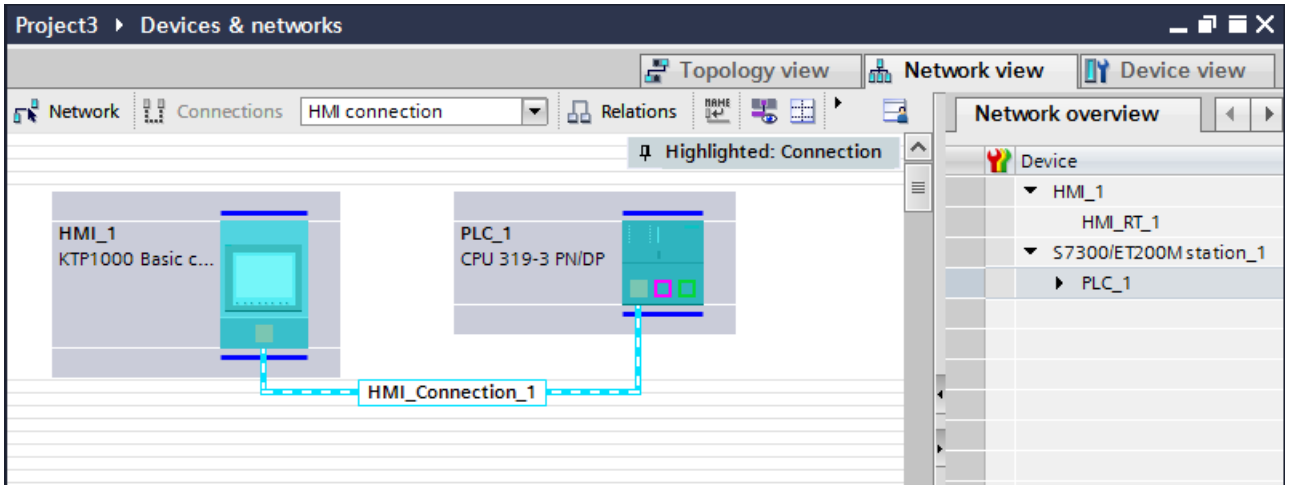
1.8.4.1 Communication via MPI (Basic Panels)

HMI connections via MPI

If you have inserted an HMI device and a SIMATIC S7 300/400 into the project, you interconnect the two MPI interfaces in the "Devices & Networks" editor.

HMI connection in the "Devices & Networks" editor

You configure the HMI connection between the PLC and the HMI device via PROFIBUS in the "Devices & Networks" editor.



Connection in the "Connections" editor

Alternatively, you configure the connection between the PLC and HMI device in the "Connections" editor of the HMI device.

See also

Communication with SIMATIC S7 300/400 (Page 209)

Configuring an HMI connection via MPI (Page 233)

1.8.4.2 Configuring an HMI connection via MPI (Basic Panels)

Introduction

You configure an HMI connection between HMI devices and a SIMATIC S7 300/400 via MPI in the "Devices & Networks" editor.

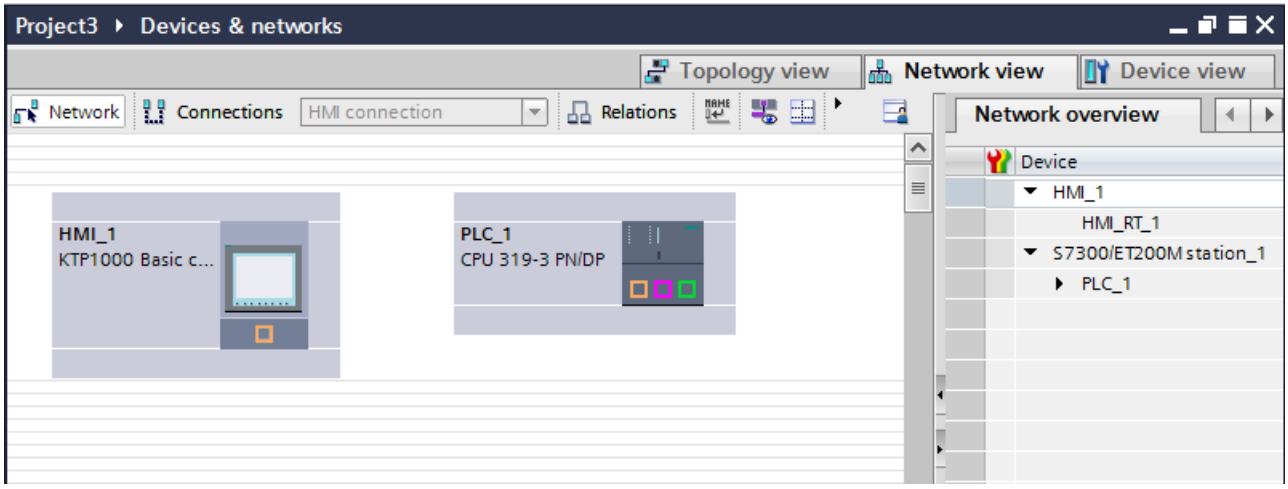
Requirements

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

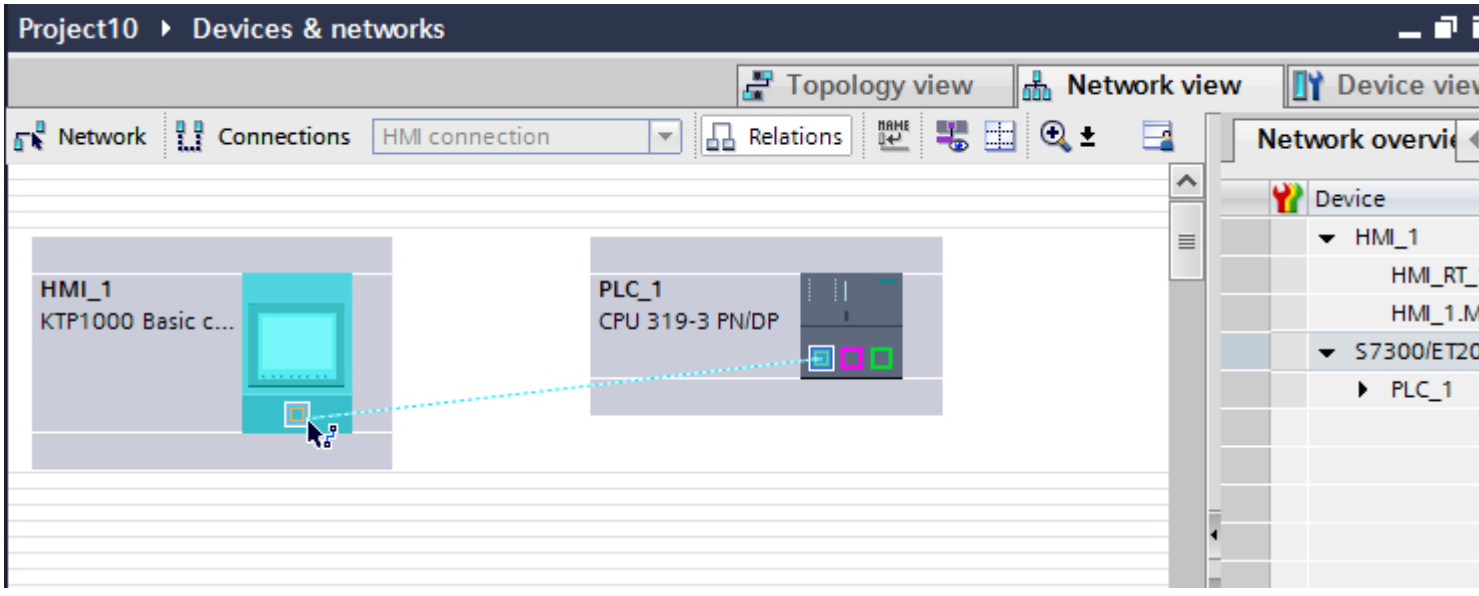
- HMI device with MPI/DP interface
- SIMATIC S7 300/400 with MPI/DP interface

Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed graphically in the network view.



2. Click the "Connections" button.
   The devices available for connection are highlighted in color.
3. Click the interface of the PLC and use a drag-and-drop operation to draw a connection to the HMI device.



4. Click the connecting line.
   The connection is displayed graphically in the Inspector window.

5. Click "Highlight HMI connection" and select the HMI connection.
6. Click the communication partners in the "Network view" and change the MPI parameters in the Inspector window according to the requirements of your project.
   See the "MPI parameters (Page 235)" section for more detailed information.

Note

The created HMI connection is also shown in the tabular area of the editor on the "Connections" tab. Use the table to monitor the connection parameters and change the connection partner. You can change the local name for the connection only in the table.

Result

You have created an HMI connection between an HMI device and a SIMATIC S7 300/400 via MPI.

See also

Communication with SIMATIC S7 300/400 (Page 209)

Communication via MPI (Page 232)

MPI parameters (Page 235)

1.8.4.3 MPI parameters (Basic Panels)

MPI parameters for the HMI connection (Basic Panels)

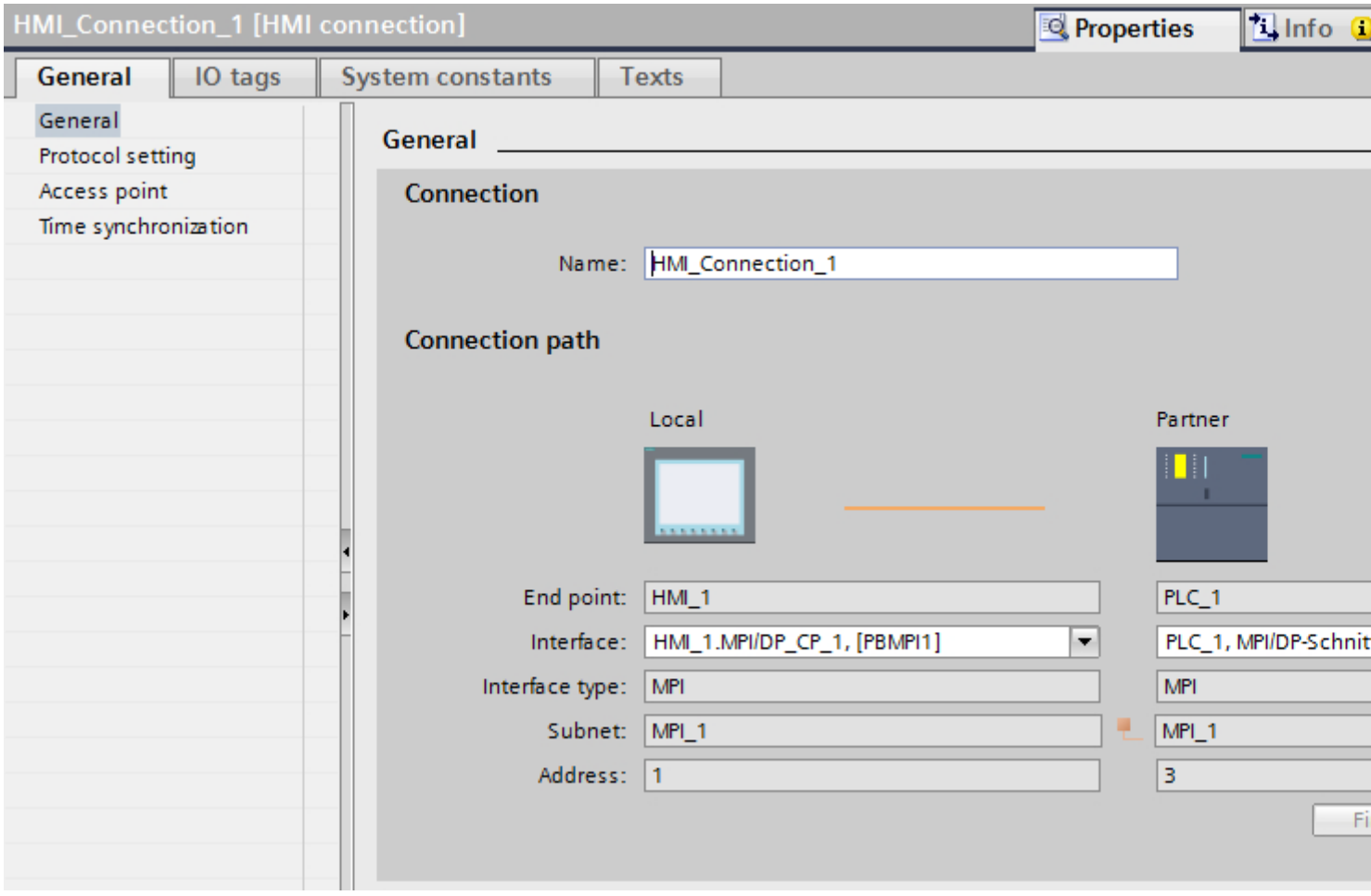
MPI parameters for the HMI connection

An overview of the configured HMI connection parameters can be found in the properties for the HMI connection.

Only limited changes are possible in this Inspector window.

Displaying and changing the HMI connection parameters

1. Click the HMI connection in the "Devices & Networks" editor.
2. Change the parameters of the HMI connection in the Inspector window under "Properties > General > General".



"Connection"

The "Connection" area displays the HMI connection created for communication between the devices.

You can edit the name of the HMI connection in this area.

"Connection path"

The communication partners of the selected HMI connection and the associated MPI parameters are displayed in the "Connection path" area. Some of the areas displayed cannot be edited in this dialog.

- "End point"
  Displays the name of the device. This area cannot be edited.
- "Interface"
  Displays the selected interface of the device. You can choose between several interfaces, depending on the device.
- "Interface type"
  Displays the selected interface type. This area cannot be edited.
- "Subnet"
  Displays the selected subnet. This area cannot be edited.
- "Address"
  Displays the MPI address of the device. This area cannot be edited.
- "Find connection path" button
  Enables the subsequent specification of connections.

See also

Communication via MPI (Page 232)

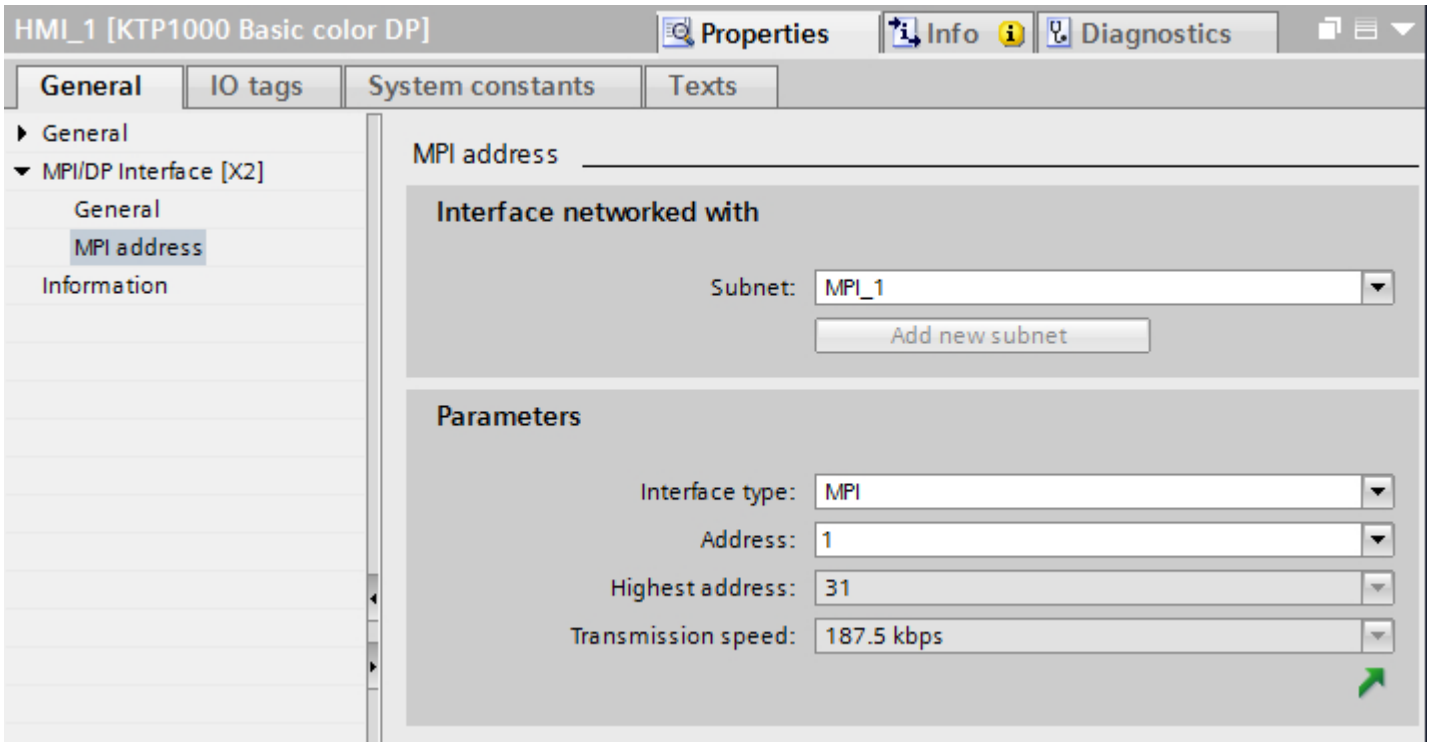
MPI parameters for the HMI device (Basic Panels)

MPI parameters for the HMI device

An overview of the configured HMI device parameters can be found in the properties for the HMI device.

Displaying and changing MPI parameters of the HMI device

1. Click the HMI device in the "Devices & Networks" editor.
2. Change the parameters of the HMI device in the Inspector window under "Properties > General > General".



"Interface networked with"

In the "Interface networked with" area, select the subnet of the HMI connection via which the HMI device is connected to the network. You use the "Add new subnet" button to create a new subnet.

"Parameters"

- "Interface type"
  You assign the interface type in the "Interface type" area. Depending on the HMI device type, you have various interfaces to choose from.
- "Address"
  You assign the MPI address of the HMI device in the "Address" area. The MPI address must be unique throughout the MPI network.
- "Highest address"
  The "Highest address" area displays the highest address of the MPI network.
- "Transmission speed"
  The "Transmission speed" is determined by the slowest device connected to the network. The setting is identical throughout the network.

See also

Communication via MPI (Page 232)

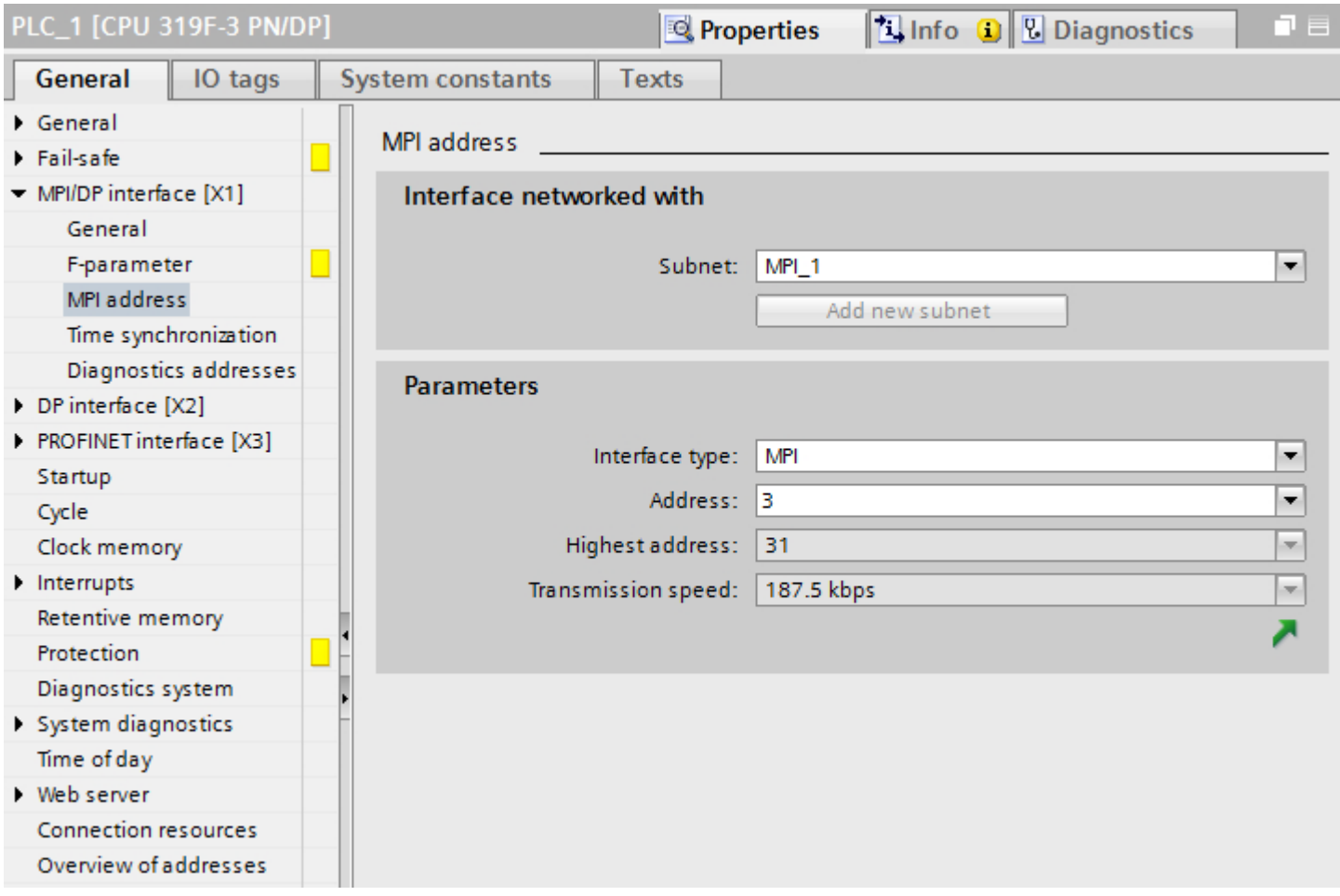
MPI parameters for the PLC (Basic Panels)

MPI parameters for the PLC

An overview of the configured parameters can be found in the properties for the PLC.

Displaying and changing PROFIBUS parameters of the PLC

1. Click the PLC in the "Devices & Networks" editor.
2. Change the parameters of the PLC in the Inspector window under "Properties > General > General".



"Interface networked with"

In the "Subnet" area, select the subnet of the HMI connection via which the HMI device is connected to the network. You use the "Add new subnet" button to create a new subnet.

"Parameters"

- "Interface type"
  Depending on the HMI device type, you have various interfaces to choose from.
- "Address"
  You assign the MPI address of the HMI device in the "Address" area. The MPI address must be unique throughout the MPI network.
- "Highest address"
  The "Highest address" area displays the highest address of the MPI network.
- "Transmission speed"
  The "Transmission speed" is determined by the slowest device connected to the network. The setting is identical throughout the network.

See also

Communication via MPI (Page 232)

Addressing of the PLC via MPI (Basic Panels)

Introduction

Each communication partner must be assigned an MPI network address.

Each S7 module which supports communication functions and is operated the SIMATIC S7-300/400 PLC is assigned a unique MPI address. Only one CPU may be used per rack.

Note

HMI devices cannot be operated with incorrect addressing

Always avoid redundant addressing on the MPI bus.

MPI address of the communication partner of a SIMATIC S7-300

When assigning addresses, you have to distinguish between communication partners with and without separate MPI address.

- If the communications partner has its own MPI address, you only need to define the MPI address.
- If the communication partners do not have a separate MPI address, specify the MPI address of the communications partner used for the connection. In addition, define the slot and rack of a communication partner without its own MPI address.

MPI address of the communication partner of a SIMATIC S7-400

Only S7 modules with an MPI connector are assigned an MPI address. Modules without an MPI connector are addressed indirectly:

- MPI address of the module to which the HMI is connected.
- The slot and the rack of the module with which the HMI device communicates.

See also

Communication via MPI (Page 232)

1.8.5 Data exchange (Basic Panels)

1.8.5.1 Data exchange using area pointers (Basic Panels)

General information on area pointers (Basic Panels)

Introduction

You use an area pointer to access a data area in the PLC. During communication, the PLC and the HMI device alternately access these data areas for read and write operations.

The PLC and the HMI device trigger defined interactions based on the evaluation of stored data.

Configuration of area pointers

Before you use an area pointer, you enable it under "Connections > Area pointer". You then assign the area pointer parameters.

You can find more detailed information on configuring area pointers in:

Configuring area pointers (Page 66)

See also

Configuring area pointers (Page 66)

"Screen number" area pointer (Page 242)

"Date/time" area pointer (Page 243)

"Date/time PLC" area pointer (Page 244)

"Coordination" area pointer (Page 245)

"Project ID" area pointer (Page 246)

"Job mailbox" area pointer (Page 247)

"Data record" area pointer (Page 249)

"Screen number" area pointer (Basic Panels)

Function

The HMI devices store information about the screen called up on the HMI device in the "Screen number" area pointer.

This allows the transfer of the current screen contents from the HMI device to the PLC. The PLC can trigger specific reactions such as the call of a different screen.

Use

Configure and enable the area pointer in "Communication > Connections" before you put it into use. You can create only **one** instance of the "Screen number" area pointer and only on **one** PLC.

The screen number is always transferred to the PLC when a new screen is activated or when the focus within a screen changes from one screen object to another.

Structure

The area pointer is a data area in the memory of the PLC with a fixed length of 5 words.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st word	Current screen type															
2nd word	Current screen number															
3rd word	Reserved															
4th word	Current field number															
5th word	Reserved															

- Current screen type
  "1" for root screen or
  "4" for permanent area
- Current screen number
  1 to 32767
- Current field number
  1 to 32767

See also

General information on area pointers (Page 241)

"Date/time" area pointer (Basic Panels)

Function

This area pointer is used to transfer the date and time from the HMI device to the PLC.

The PLC writes control job "41" or "40" to the job mailbox.

When it evaluates the control job, the HMI device writes its current date and the time in the data area configured in the "Date/time" area pointer. All definitions are coded in BCD format.

The "Date/Time" area pointer when used in a project which contains multiple connections must be enabled for each configured connection.

Note

You cannot use the "Date/Time PLC" area pointer if you have configured the "Date/Time" area pointer.

Note

Symbolic addressing is not possible if you are using the "Date/Time" area pointer.

The "Date/Time" area pointer when used in a project which contains multiple connections must be enabled for each configured connection.

The date/time data area has the following structure:

Data word	Most significant byte								Least significant byte								
	7							0	7							0	
n+0	Reserved								Hour (0 to 23)								Time
n+1	Minute (0 to 59)								Second (0 to 59)								
n+2	Reserved								Reserved								
n+3	Reserved								Weekday (1 to 7, 1=Sunday)								Date
n+4	Day (1 to 31)								Month (1 to 12)								
n+5	Year (80 to 99/0 to 29)								Reserved								

Note

When making entries in the "Year" data area, you should note that values 80 to 99 result in years 1980 through 1999, while the values 0 to 29 result in the years 2000 through 2029.

See also

General information on area pointers (Page 241)

"Date/time PLC" area pointer (Basic Panels)

Function

This area pointer is used to transfer the date and time from the PLC to the HMI device. Use this area pointer if the PLC is the time master.

The PLC loads the data area of the area pointer. All definitions are coded in BCD format.

The HMI device reads the data cyclically within the configured acquisition cycle and synchronizes itself.

Note

Set an acquisition cycle of sufficient length for the Date/time area pointer in order to avoid any negative impact on HMI device performance.
Recommended: Acquisition cycle of 1 minute, if the process allows this.

You cannot use the "Date/Time" area pointer if you have configured the "Date/Time PLC" area pointer.

"Date/Time PLC" is a global area pointer and may be configured only once per project.

Note

You cannot use the "Date/Time" area pointer if you have configured the "Date/Time PLC" area pointer.

The date/time data area has the following structure:

DATE_AND_TIME format (in BCD code)

Data word	Most significant byte			Least significant byte		
	7	. . . . .	0	7	. . . . .	0
n+0	Year (80 to 99/0 to 29)			Month (1 to 12)		
n+1	Day (1 to 31)			Hour (0 to 23)		
n+2	Minute (0 to 59)			Second (0 to 59)		
n+3	Reserved			Reserved		Weekday (1 to 7, 1=Sun-day)
n+4 [1)]	Reserved			Reserved		
n+5 [1)]	Reserved			Reserved		

1) The two data words must exist in the data area to ensure that the data format matches WinCC flexible and to avoid reading false information.

Note

When making entries in the "Year" data area, you should note that values 80 to 99 result in years 1980 through 1999, while the values 0 to 29 result in the years 2000 through 2029.

See also

General information on area pointers (Page 241)

"Coordination" area pointer (Basic Panels)

Function

The "Coordination" area pointer is used to implement the following functions:

- Detecting the startup of the HMI device in the control program
- Detecting the current operating mode of the HMI device in the control program
- Detecting whether the HMI device is ready to communicate in the control program

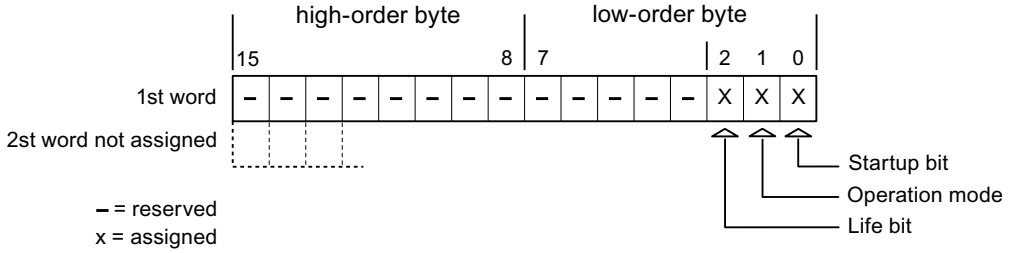
By default, the "Coordination" area pointer has the length of one word and cannot be changed.

Application

Note

The HMI device always writes the entire coordination area when updating the area pointer. The control program can for this reason not make changes to the coordination area.

Assignment of bits in the "Coordination" area pointer



Startup bit

The startup bit is set briefly to "0" by the HMI device during startup. It sets the bit permanently to "1" when startup is completed.

Operating mode

The operating mode bit is set to 1 as soon as the user switches the HMI device offline. The status of the operating mode bit is "0" during normal operation of the HMI device. You can determine the current operating mode of the HMI device by reading this bit in the control program.

Life bit

The HMI device inverts the life bit at intervals of approximately one second. You can check whether or not there is still a connection to the HMI device by querying this bit in the control program.

See also

General information on area pointers (Page 241)

"Project ID" area pointer (Basic Panels)

Function

When Runtime starts, a check can be carried out as to whether the HMI device is connected to the correct PLC. This check is important when operating with several HMI devices.

For this, the HMI device compares a value stored on the PLC with the value specified in configuration. This ensures compatibility of the configuration data with the PLC program. If there is no concordance, a system event is given on the HMI device and Runtime is stopped.

Use

Note

HMI connections cannot be switched "online".

The HMI connection in which the "Project ID" area pointer is used must be switched "online".

To use this area pointer, set up the following during the configuration:

- Define the version of the configuration. Values between 1 and 255 are possible.
  You enter the version in the editor "Runtime settings > General" in the "Identification" area.
- Data address of the value for the version that is stored in the PLC:
  You enter the data address in the editor "Communication > Connections" under "Address".

Connection failure

A connection failure to a device on which the "Project ID" area pointer is configured results in all the other connections of the device being switched to "offline".

This behavior has the following requirements:

- You have configured several connections in a project.
- You are using the "Project ID" area pointer in at least one connection.

Causes which may set connections "offline":

- The PLC is not available.
- The connection has been switched offline in the Engineering System.

See also

General information on area pointers (Page 241)

"Job mailbox" area pointer (Basic Panels)

Function

The PLC can use the job mailbox to transfer jobs to the HMI device to trigger corresponding actions on the HMI device. These functions include, for example:

- Display screen
- Set date and time

Data structure

The first word of the job mailbox contains the job number. Depending on the job mailbox, up to three parameters can be transferred.

Word	Most significant byte	Least significant byte
n+0	0	Job number
n+1	Parameter 1	
n+2	Parameter 2	
n+3	Parameter 3	

The HMI device evaluates the job mailbox if the first word of this job is not equal to zero. This means that the parameters must be entered in the job mailbox first, followed by the job number.

When the HMI device accepts the job mailbox, the first word is set to 0 again. The execution of the job mailbox is generally not completed at this point in time.

Job mailboxes

All job mailboxes and their parameters are listed below. The "No." column contains the job number of the job mailbox. Job mailboxes can only be triggered by the PLC when the HMI device is online.

No.	Function	
14	**Set time (BCD-coded)**	
	Parameter 1	Left byte: - Right byte: hours (0-23)
	Parameter 2	Left byte: minutes (0-59) Right byte: seconds (0-59)
	Parameter 3	-
15	**Setting the date (BCD coded)** [3) 4)]	
	Parameter 1	Left byte: - Right byte: weekday (1-7: Sunday-Saturday)
	Parameter 2	Left byte: day (1-31) Right byte: month (1-12)
	Parameter 3	Left byte: year
23	**User logon**	
	Logs the user on with the name "PLC user" at the HMI device with the group number transferred in Parameter 1. The logon is possible only when the transferred group number exists in the project.	
	Parameter 1	Group number 1 to 255
	Parameter 2, 3	-
24	**User logoff**	
	Logs off the current user. (The function corresponds to the "logoff" system function)	
	Parameter 1, 2, 3	-
40	**Transfer date/time to PLC**	
	(in the S7 format DATE_AND_TIME) An interval of at least 5 seconds must be maintained between two successive jobs to prevent overload of the HMI device.	
	Parameter 1, 2, 3	-
41	**Transfer date/time to PLC**	
	(In OP/MP format) An interval of at least 5 seconds must be maintained between successive jobs to prevent overload of the HMI device.	
	Parameter 1, 2, 3	-
46	**Update tag**	
	Causes the HMI device to read the current value of the tags from the PLC whose update ID matches the value transferred in Parameter 1. (Function corresponds to the "UpdateTag" system function.)	
	Parameter 1	1 - 100
49	**Delete alarm buffer**	
	Deletes all analog alarms and discrete alarms of the "Warnings" class from the alarm buffer.	
	Parameter 1, 2, 3	-

No.	Function	
14	**Set time (BCD-coded)**	
50	**Delete alarm buffer**	
	Deletes all analog alarms and discrete alarms of the "Errors" class from the alarm buffer.	
	Parameter 1, 2, 3	-
51	**Screen selection** [2]	
	Parameter 1	Screen number
	Parameter 2	-
	Parameter 3	Field number
69	**Reading data record from PLC** [1]	
	Parameter 1	Recipe number (1-999)
	Parameter 2	Data record number (1-65535)
	Parameter 3	0: Do not overwrite existing data record 1: Overwrite existing data record
70	**Writing data record to PLC** [1]	
	Parameter 1	Recipe number (1-999)
	Parameter 2	Data record number (1-65535)
	Parameter 3	-

[1]	Only for devices supporting recipes.
[2]	OP 73, OP 77A and TP 177A HMI devices also execute the "Screen selection" job mailbox if the on-screen keyboard is active.
[3]	The weekday is ignored on HMI device KTP 600 BASIC PN.
[4]	The weekday is ignored when you configure the "Date/Time PLC" area pointer.

See also

General information on area pointers (Page 241)

"Data record" area pointer (Basic Panels)

"Data record" area pointer (Basic Panels)

Function

When data records are transferred between the HMI device and PLC, both partners access common communications areas on the PLC.

Data transfer types

There are two ways of transferring data records between the HMI device and PLC:

- Transfer without synchronization
- Transfer with synchronization via the data mailbox

Data records are always transferred directly, which means that the tag values are read straight from an address or written straight to an address configured for this tag without being redirected via an interim memory.

Initiating the transfer of data records

There are three ways of triggering the transfer:

- Operator input in the recipe view
- Job mailboxes
  The transfer of data records can also be triggered by the PLC.
- Triggering by configured functions

If the transfer of data records is triggered by a configured function or by a job mailbox, the recipe view on the HMI device remains operable. The data records are transferred in the background.

Simultaneous processing of several transfer requests is, however, not possible. In this case, the HMI device rejects the other transfer requests with a system alarm.

See also

General information on area pointers (Page 241)

Transfer without synchronization (Page 250)

Transfer with synchronization (Page 251)

Sequence of a transfer started by the operator in the recipe view (Page 252)

Sequence of the transfer triggered by a job mailbox (Page 253)

Sequence of the transfer when triggered by a configured function (Page 255)

Possible causes of error when transferring data records (Page 256)

Transfer without synchronization (Basic Panels)

If you select asynchronous transfer of data records between the HMI device and PLC, there is no coordination over the common data areas. It is therefore unnecessary to set up a data area during configuration.

Asynchronous data record transfer can be a useful alternative, for example when:

- The system is capable of excluding the risk of uncontrolled overwriting of data by the communication peer.
- The PLC does not require information about the recipe number and data record number.
- The transfer of data records is triggered by the operator of the HMI device.

Reading values

When a read job is triggered, the values are read from the PLC addresses and transferred to the HMI device.

- Triggering by the operator in the recipe view:
  The values are downloaded to the HMI device. You can then, for example, process, edit, or save these values in the HMI device.
- Triggering by a function or job mailbox:
  The values are saved immediately to the data volume.

Writing values

When a write job is triggered, the values are written to the PLC addresses.

- Triggering by the operator in the recipe view:
  The current values are written to the PLC.
- Triggering by a function or job mailbox:
  The current values are written to the PLC from the data medium.

See also

"Data record" area pointer (Page 249)

Transfer with synchronization (Basic Panels)

If you select synchronous transfer, both communication partners set status bits in the common data area. You can use this mechanism to prevent uncontrolled overwriting of data in either direction in your control program.

Application

Synchronous data record transfer can be a useful solution, for example, when:

- The PLC is the "active partner" in the transfer of data records.
- The PLC evaluates the information about the recipe number and data record number.
- The transfer of data records is triggered by means of a Job mailbox.

Requirements

In order to synchronize transfer of data records between the HMI device and the PLC, the following requirements must be met during configuration:

- An area pointer has been set up: "Communication > Connections" editor in "Area pointer".
- The PLC with which the HMI device synchronizes transfer of data records is specified in the recipe:
  "Recipes" editor in the inspector window the option "Coordinated transfer of data records" under "General > Synchronization > Settings"

Structure of the data area

The data area has a fixed length of 5 words. Structure of the data area:

	15		0
1. Word	Current recipe number (1 - 999)		
2. Word	Current data record number (0 - 65535)		
3. Word	Reserved		
4. Word	Status (0, 2, 4, 12)		
5. Word	Reserved		

- Status
  The status word (word 4) can adopt the following values:

Value		Meaning
Decimal	Binary	
0	0000 0000	Transfer permitted, data mailbox free
2	0000 0010	Transferring
4	0000 0100	Transfer completed without error
12	0000 1100	Transfer completed with error

See also

"Data record" area pointer (Page 249)

Sequence of a transfer started by the operator in the recipe view (Basic Panels)

Reading from the PLC started by the operator in the recipe view

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe number to be read and the status "Transferring" in the data record and sets the data record number to 0.	Abort with system event.
3	The HMI device reads the values from the PLC and displays them in the recipe view. If the recipes have synchronized tags, the values from the PLC are also written to the tags.	
4	The HMI device sets the status "Transfer completed."	
5	The control program must reset the status word to zero in order to enable further transfers.	

Writing to the PLC started by the operator in the recipe view

Step	Action	
	Check: Status word = 0?	
1	Yes	No
	The HMI device enters the recipe and data record number to be written and the status "Transferring" in the data record.	Abort with system event.
2	The HMI device writes the current values to the PLC. If the recipes have synchronized tags, the changed values are synchronized between the recipe view and tags and then written to the PLC.	
3	The HMI device sets the status "Transfer completed."	
4	If required, the control program can now evaluate the transferred data.	
5	The control program must reset the status word to zero in order to enable further transfers.	

Note

The status word may only be set by the HMI device. The PLC may only reset the status word to zero.

Note

The PLC may only evaluate the recipe and data record numbers when data inconsistency is detected if one of the conditions outlined below has been met:

- The data mailbox status is set to "Transfer completed".
- The data mailbox status is set to "Transfer completed with error".

See also

"Data record" area pointer (Page 249)

Sequence of the transfer triggered by a job mailbox (Basic Panels)

The transfer of data records between the HMI device and the PLC can be initiated by either one of these stations.

The two job mailboxs No. 69 and No. 70 are available for this type of transfer.

No. 69: Read data record from PLC ("PLC → DAT")

Job mailbox no. 69 transfers data records from the PLC to the HMI device. The job mailbox is structured as follows:

	Left byte (LB)	Right byte (RB)
Word 1	0	69
Word 2	Recipe number (1-999)	

	Left byte (LB)	Right byte (RB)
Word 3	Data record number (1 to 65535)	
Word 4	Do not overwrite existing data record: 0 Overwrite existing data record: 1	

No. 70: Write data record to PLC ("DAT → PLC")

Job mailbox No. 70 transfers data records from the HMI device to the PLC. The job mailbox is structured as follows:

	Left byte (LB)	Right byte (RB)
Word 1	0	70
Word 2	Recipe number (1-999)	
Word 3	Data record number (1-65,535)	
Word 4	—	

Sequence when reading from the PLC with job mailbox "PLC → DAT" (no. 69)

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the job and the status "Transferring" in the data record.	Abort without return message.
3	The HMI device reads the values from the PLC and saves these to the data record defined in the job mailbox.	
4	• If "Overwrite" was selected in the job, an existing data record is overwritten without any prompt for confirmation. The HMI device sets the status "Transfer completed." • If "Do not overwrite" was selected in the job, and the data record already exists, the HMI device aborts the job and enters 0000 1100 in the status word of the data mailbox.	
5	The control program must reset the status word to zero in order to enable further transfers.	

Sequence of writing to the PLC with job mailbox "DAT → PLC" (no. 70)

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the job and the status "Transferring" in the data mailbox.	Abort without return message.
3	The HMI device fetches the values of the data record specified in the function from the data medium and writes the values to the PLC.	

Step	Action	
4	The HMI device sets the status "Transfer completed."	
5	The control program can now evaluate the transferred data. The control program must reset the status word to zero in order to enable further transfers.	

See also

"Data record" area pointer (Page 249)

Sequence of the transfer when triggered by a configured function (Basic Panels)

Reading from the PLC using a configured function

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the function and the status "Transferring" in the data mailbox.	Abort with system event.
3	The HMI device reads the values from the PLC and stores them in the data record specified in the function.	
4	• If "Yes" was selected for the "Overwrite" function, an existing data record is overwritten without any prompt for confirmation. The HMI device sets the status "Transfer completed." • If "No" was selected for the "Overwrite" function and the data record already exists, the HMI device aborts the job and enters 0000 1100 in the status word of the data mailbox.	
5	The control program must reset the status word to zero in order to enable further transfers.	

Writing to the PLC by means of configured function

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the function and the status "Transferring" in the data mailbox.	Abort with system event.
3	The HMI device fetches the values of the data record specified in the function from the data medium and transfers the values to the PLC.	
4	The HMI device sets the status "Transfer completed."	
5	The control program can now evaluate the transferred data. The control program must reset the status word to zero in order to enable further transfers.	

See also

"Data record" area pointer (Page 249)

Possible causes of error when transferring data records (Basic Panels)

Possible causes of error

The section below shows possible error causes which lead to the cancellation of data record transfer:

- Tag address not set up on the PLC
- Overwriting data records not possible
- Recipe number does not exist
- Data record number does not exist

Note

The status word may only be set by the HMI device. The PLC may only reset the status word to zero.

Note

The PLC may only evaluate the recipe and data record numbers when data inconsistency is detected if one of the conditions outlined below has been met:

- The data mailbox status is set to "Transfer completed".
- The data mailbox status is set to "Transfer completed with error".

Reaction to an aborted transfer due to errors

If the transfer of data records is aborted due to errors, the HMI device reacts as follows:

- Triggering by the operator in the recipe view
  Information in the status bar of the recipe view and output of system alarms
- Triggered by function
  Output of system alarms
- Triggering by job mailbox
  No return message on the HMI device

You can nonetheless evaluate the status of the transfer by querying the status word in the data record.

See also

"Data record" area pointer (Page 249)

1.8.5.2 Trends (Basic Panels)

Trends

A trend is the graphical representation of one or more values from the PLC. The value is read out time-triggered for Basic Panels.

For additional information see:

AUTOHOTSPOT

Time-triggered trends

The HMI device reads in the trend values cyclically at an interval specified in the configuration.

Time-triggered trends are suitable for continuous curves, such as the operating temperature of a motor.

1.8.5.3 Alarms (Basic Panels)

Configuring alarms (Basic Panels)

Configure alarms

Several steps are needed to configure alarms, such as operational messages, error alarms, and acknowledgement.

- Step 1: Create tags
- Step 2: Configure alarms
- Step 3: Configure acknowledgment

You can find additional information in the section:

AUTOHOTSPOT

Distinctive features when configuring alarms

If you are configuring connections of HMI devices to PLCs of other manufacturers, note the following distinctive features when configuring:

- Data types of the tags
- Addressing of tags
- How the bit positions are counted

Data types

For connections with a SIMATIC communication driver, the following data types are supported:

PLC	Permitted data types	
	Discrete alarms	Analog alarms
SIMATIC S7 300/400	WORD, INT	BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, COUNTER, TIME

How the bit positions are counted

For connections with a SIMATIC communication driver, the following counting method applies:

How the bit positions are counted	Byte 0								Byte 1							
	Most significant byte								Least significant byte							
In SIMATIC S7 PLCs	7							0	7							0
In WinCC you configure:	15							8	7							0

See also

Acknowledgment of alarms (Page 258)

Acknowledgment of alarms (Basic Panels)

Procedure

Create suitable tags on the PLC to acknowledge an error alarm. You assign these tags to an alarm in the "Bit messages" editor. You make the assignment in "Properties > Acknowledgment".

Distinction in terms of acknowledgment:

- Acknowledgment by the PLC
- Acknowledgment on the HMI device

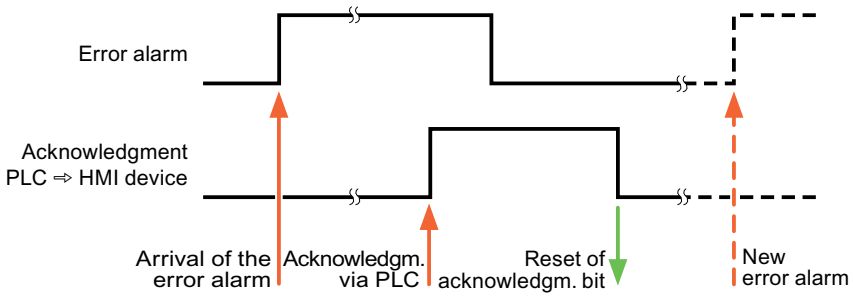
Acknowledgment by the PLC

In "PLC acknowledgment tag", configure the tag or array tag and the bit number that the HMI device uses to identify a PLC acknowledgment.

A bit set in the tag triggers acknowledgment of the assigned error alarm bit at the HMI device. This tag bit returns a function similar to acknowledgment on the HMI device which is triggered by pressing the "ACK" button, for example.

The acknowledgment bit must be located in the same tag as the bit for the error alarm.

Reset the acknowledgment bit before setting the bit in the alarm area again. The figure below shows the pulse diagram.



Acknowledgment on the HMI device

In the "HMI acknowledgment tag" area, configure the tag or array tag as well as the bit number that the HMI device writes to the PLC after acknowledgment. Make sure when you use an array tag that it is not longer than 6 words.

To always create a signal change when setting an assigned acknowledgment bit of a discrete alarm that must be acknowledged, the HMI device will reset the acknowledgment bit assigned to the alarm as soon as it detects an alarm subject to acknowledgment and write the acknowledgment tag in the PLC. There will be a certain delay between detecting the message and writing the acknowledgment tag in the PLC because the HMI device has to process the operations.

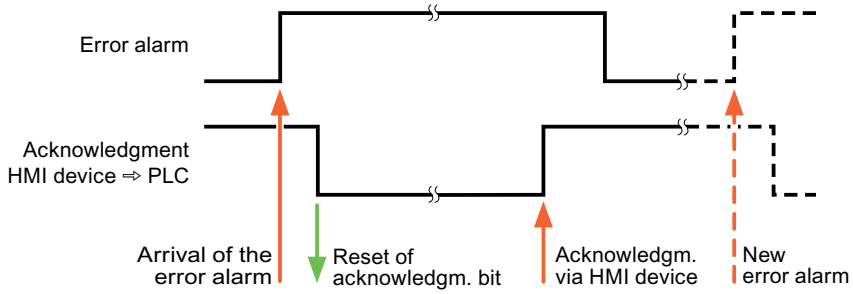
If a discrete alarm subject to acknowledgment is acknowledged by the HMI device, then the corresponding bit in the assigned acknowledgment tag will be set. The entire acknowledgment tag is then written to the PLC by the HMI device. This allows the PLC to recognize that a certain alarm message has been acknowledged at the HMI device.

Note

All alarm bits acknowledged since the last Runtime start will remain in the acknowledgment tag until a new incoming of the respective discrete alarms is detected.

This area should only be read by the PLC because the entire section of the HMI device will be overwritten once the next acknowledgment tag is written.

The figure below shows the pulse diagram.



See also

Configuring alarms (Page 257)

1.8.6 Performance features of communication (Basic Panels)

1.8.6.1 S7-300/400 device dependency (Basic Panels)

Communication with the SIMATIC S7-300/400 controller

If you use devices from an earlier version of the TIA Portal with TIA Portal V14, it may not be possible to configure integrated connections to certain HMI devices.

Basic Panels V11.0

HMI devices	SIMATIC S7-300/400
KP300 Basic	Yes
KP400 Basic	Yes
KTP400 Basic PN	Yes
KTP600 Basic DP	Yes
KTP600 Basic PN	Yes
KTP1000 Basic DP	Yes
KTP1000 Basic PN	Yes
TP1500 Basic PN	Yes

Basic Panels V12.0

HMI devices	SIMATIC S7-300/400
KP300 Basic	Yes
KP400 Basic	Yes
KTP400 Basic PN	Yes
KTP600 Basic DP	Yes
KTP600 Basic PN	Yes
KTP1000 Basic DP	Yes
KTP1000 Basic PN	Yes
TP1500 Basic PN	Yes

Basic Panels V13.0

HMI devices	SIMATIC S7-300/400
KTP400 Basic PN	Yes
KTP700 Basic PN	Yes
KTP700 Basic DP	Yes
KTP900 Basic PN	Yes
KTP1200 Basic PN	Yes
KTP1200 Basic DP	Yes

Basic Panels V13.0.1

HMI devices	SIMATIC S7-300/400
KTP400 Basic PN	Yes
KTP700 Basic PN	Yes
KTP700 Basic DP	Yes
KTP900 Basic PN	Yes
KTP1200 Basic PN	Yes
KTP1200 Basic DP	Yes

Basic Panels V14.0

HMI devices	SIMATIC S7-300/400
KTP400 Basic PN	Yes
KTP700 Basic PN	Yes
KTP700 Basic DP	Yes
KTP900 Basic PN	Yes
KTP1200 Basic PN	Yes
KTP1200 Basic DP	Yes

See also

Communication with SIMATIC S7 300/400 (Page 209)
Permitted data types for SIMATIC S7 300/400 (Page 262)

1.8.6.2 Permitted data types for SIMATIC S7 300/400 (Basic Panels)

Permitted data types for connections with SIMATIC S7 300/400

The table lists the data types that can be used when configuring tags and area pointers.

Data type	Length
BOOL	1 bit
BYTE	1 byte
WORD	2 bytes
DWORD	4 bytes
CHAR	1 byte
Array	(Element number * data type length) bytes [1]
INT	2 bytes
DINT	4 bytes
REAL	4 bytes
TIME	4 bytes
DATE	2 bytes
TIME_OF_DAY, TOD	4 bytes
S5TIME	2 bytes
COUNTER	2 bytes
TIMER	2 bytes
DATE_AND_TIME	8 bytes
STRING	(2+n) bytes, n = 0 to 254

[1] Example of "Length of an array": The length is 400 bytes (100 * 4) for 100 elements of the data type REAL

See also

Communication with SIMATIC S7 300/400 (Page 209)
S7-300/400 device dependency (Page 260)

1.8.7 Creating connections in the "Connections" editor (Basic Panels)

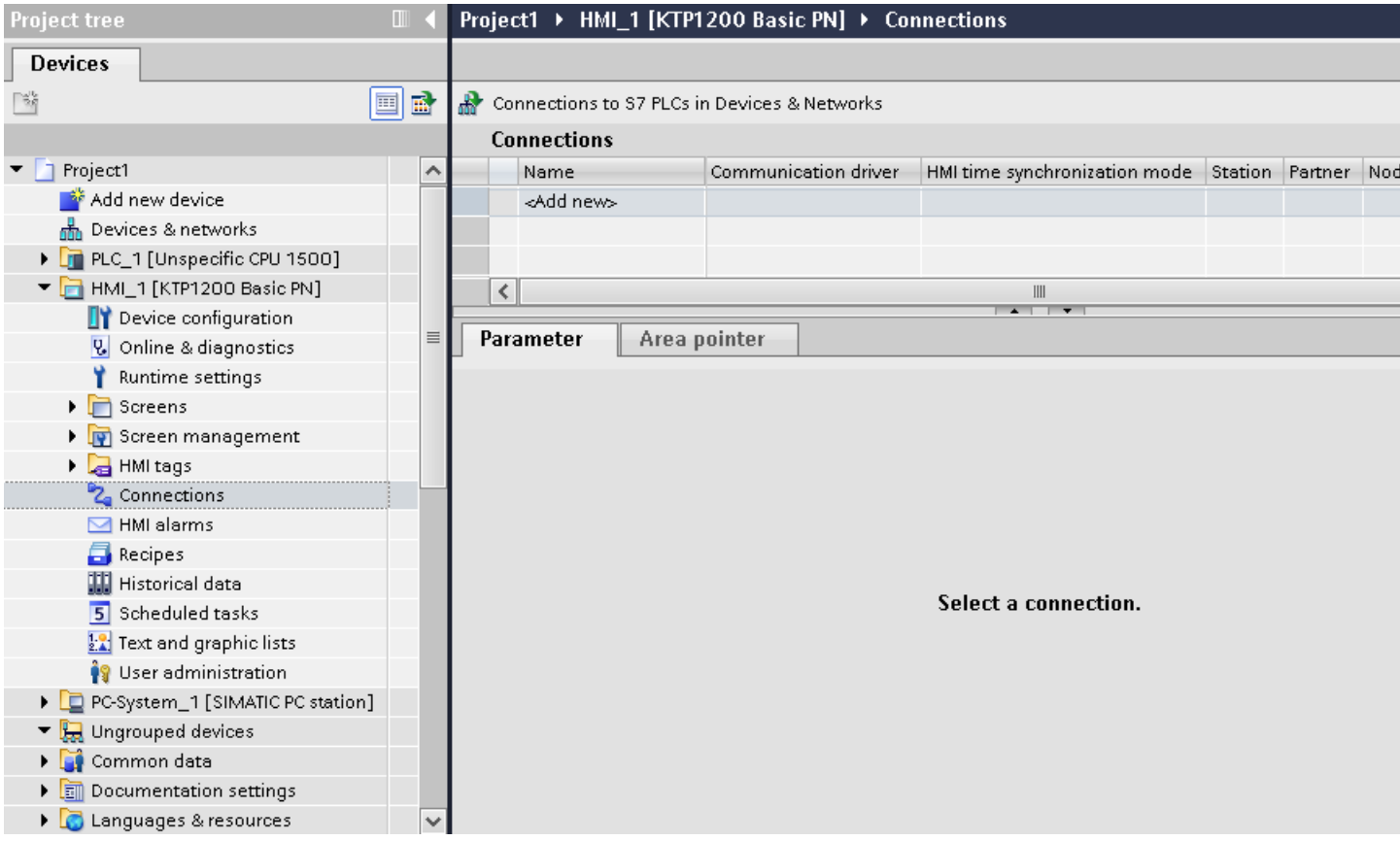
1.8.7.1 Creating a PROFINET connection (Basic Panels)

Requirements

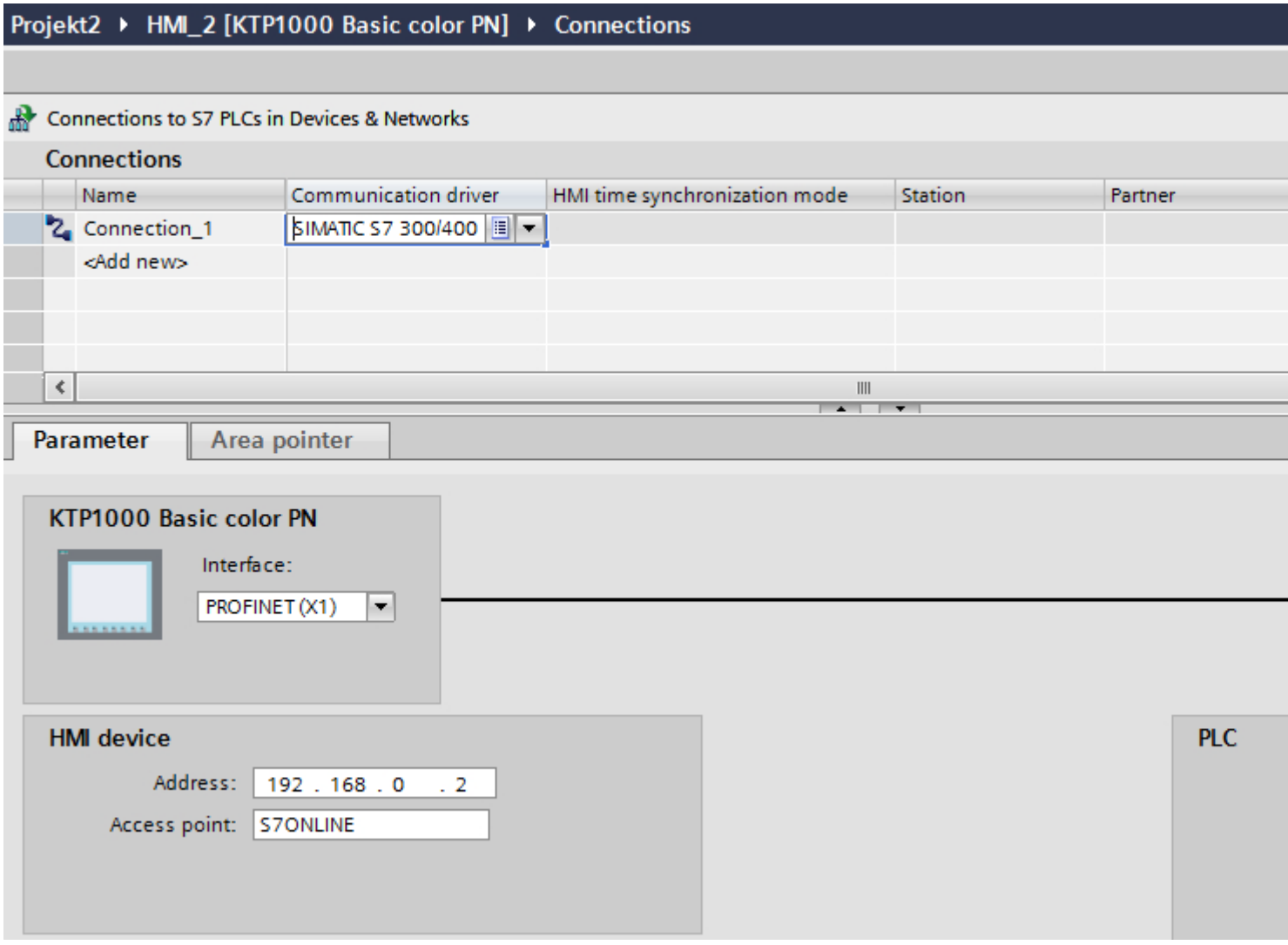
- A project is open.
- An HMI device with a PROFINET interface has been created.

Procedure

1. Open the "Connections" editor of the HMI device.
2. Double-click "<Add>".



3. Select the driver in the "Communication driver" column.



4. Click the name of the connection.
5. Select a PROFINET interface of the HMI device in the Inspector window under "Parameters > Interface".
6. Set the IP addresses of the communication partners in the Inspector window:
   - HMI device: "Parameters > HMI device > Address"
   - PLC: "Parameters > PLC > Address"

See also

Communication via PROFINET (Page 209)

Communication with SIMATIC S7 300/400 (Page 209)

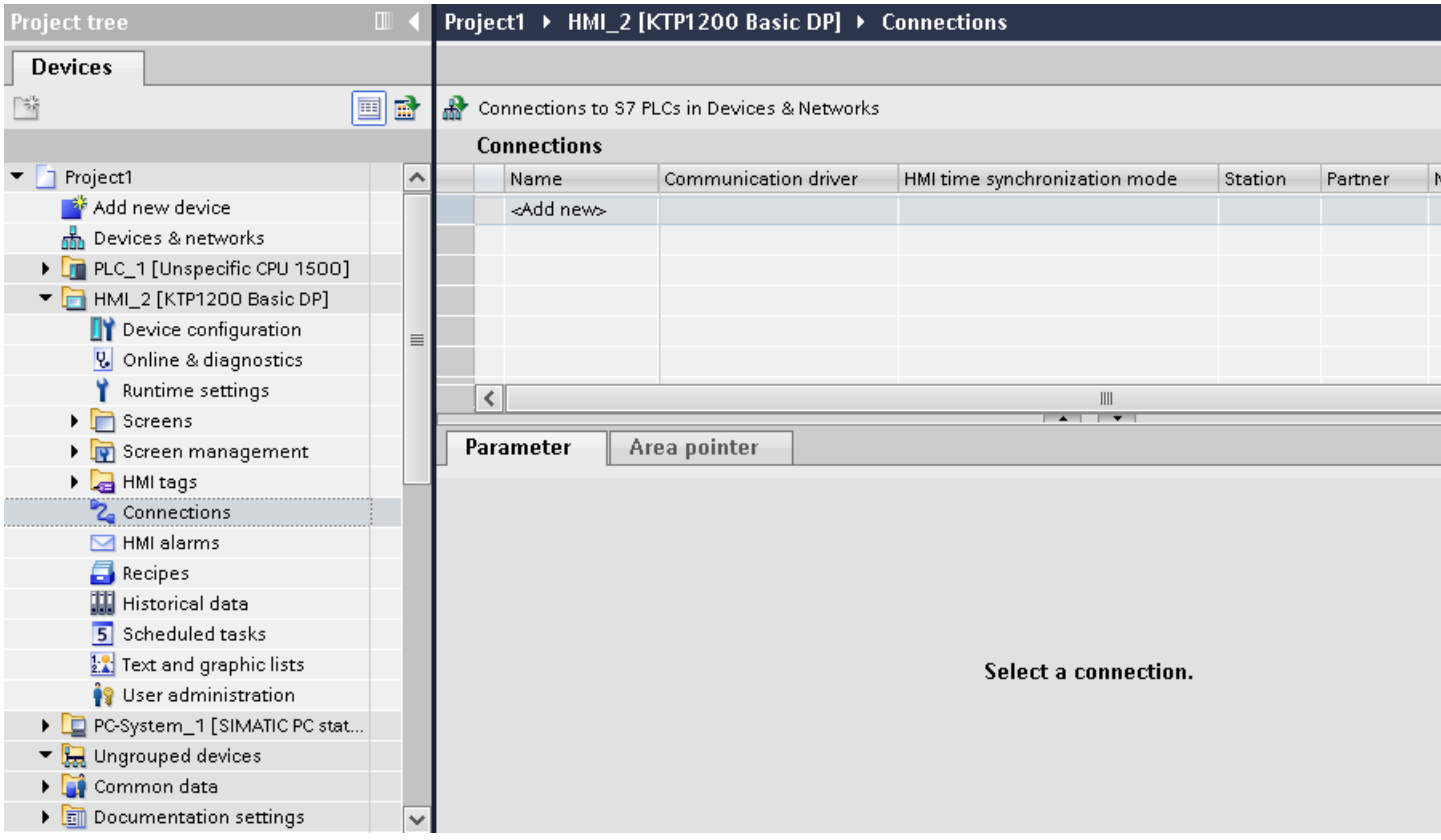
1.8.7.2 Creating a PROFIBUS connection (Basic Panels)

Requirements

- A project is open.
- An HMI device with a PROFIBUS interface has been created.

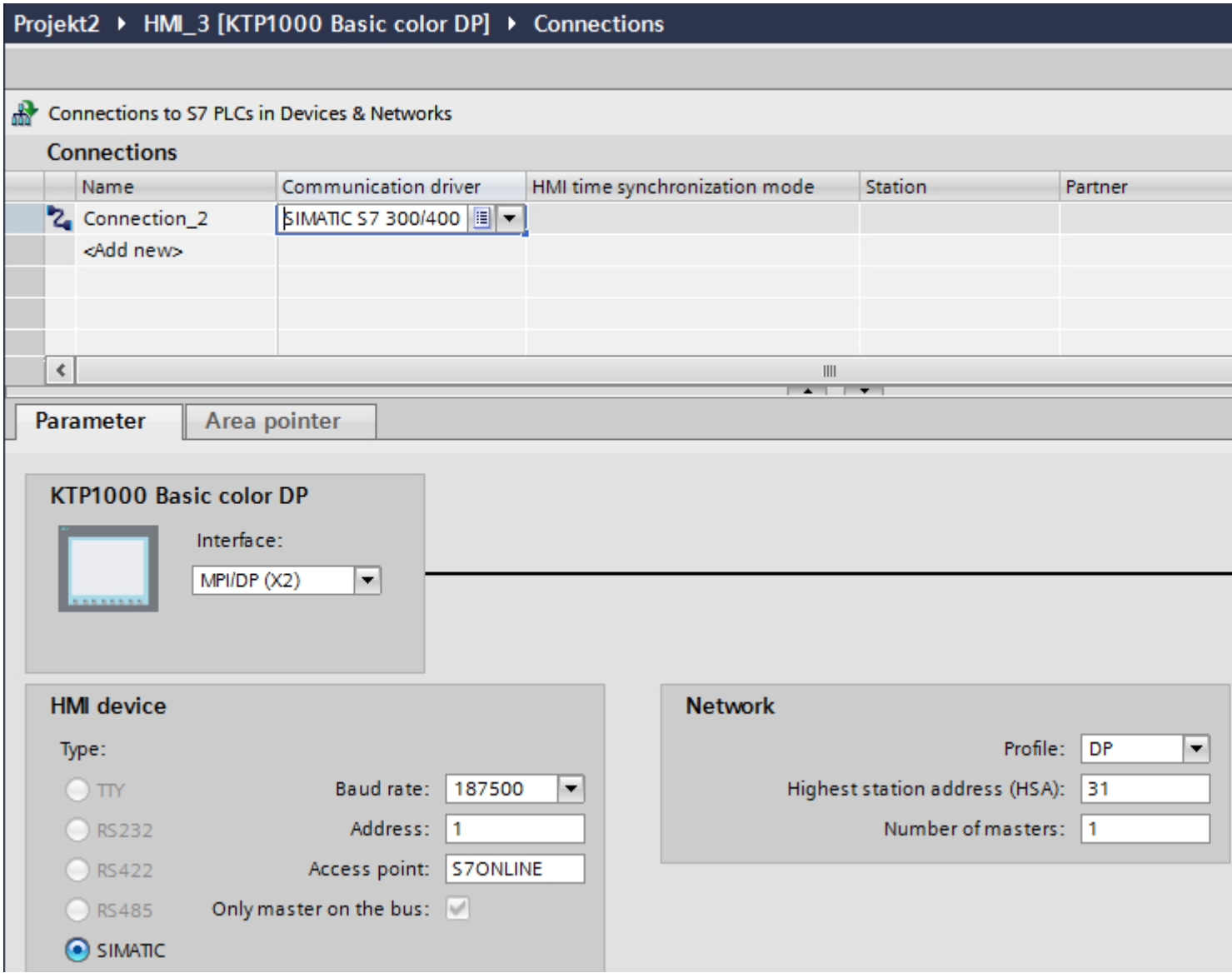
Procedure

1. Open the "Connections" editor of the HMI device.
2. Double-click "<Add>".



3. Select the driver in the "Communication driver" column.
4. Click the name of the connection.
5. Select the "MPI/DP" interface in the Inspector window under "Parameters > Interface".

6. Select the "DP" profile in the Inspector window under "Parameters > Network".



7. Set the addresses of the communication partners in the inspector window:
   - HMI device: "Parameters > HMI device > Address"
   - PLC: "Parameters > PLC > Address"

See also

Communication via PROFIBUS (Page 224)

Communication with SIMATIC S7 300/400 (Page 209)

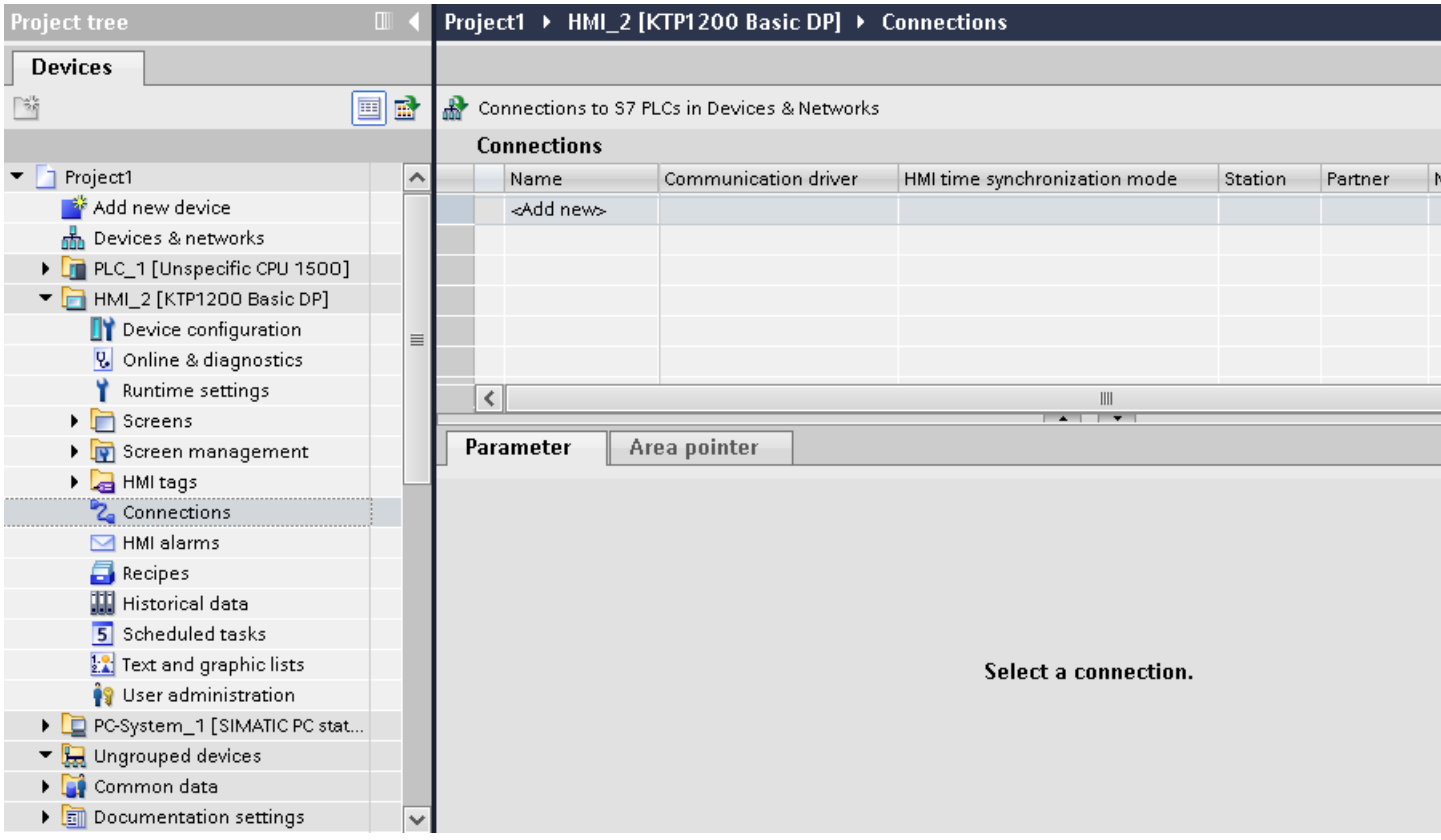
1.8.7.3 Creating an MPI connection (Basic Panels)

Requirements

- A project is open.
- An HMI device with an MPI interface has been created.

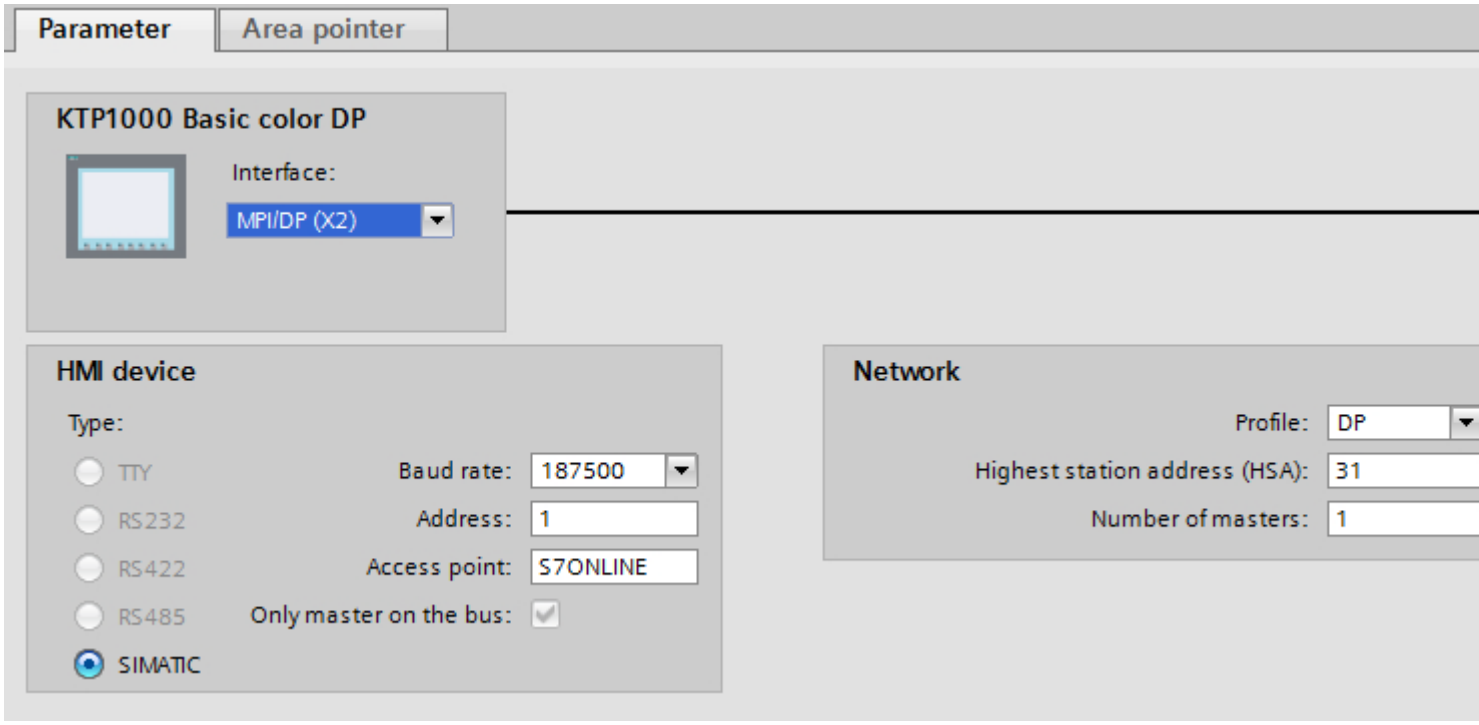
Procedure

1. Open the "Connections" editor of the HMI device.
2. Double-click "<Add>".



3. Select the driver in the "Communication driver" column.
4. Click the name of the connection.
5. Select the "MPI/DP" interface in the Inspector window under "Parameters > Interface".

6. Select the "MPI" profile in the Inspector window under "Parameters > Network".



7. Set the addresses of the communication partners in the inspector window:
   - HMI device: "Parameters > HMI device > Address"
   - PLC: "Parameters > PLC > Address"

See also

Communication via MPI (Page 232)

Communication with SIMATIC S7 300/400 (Page 209)

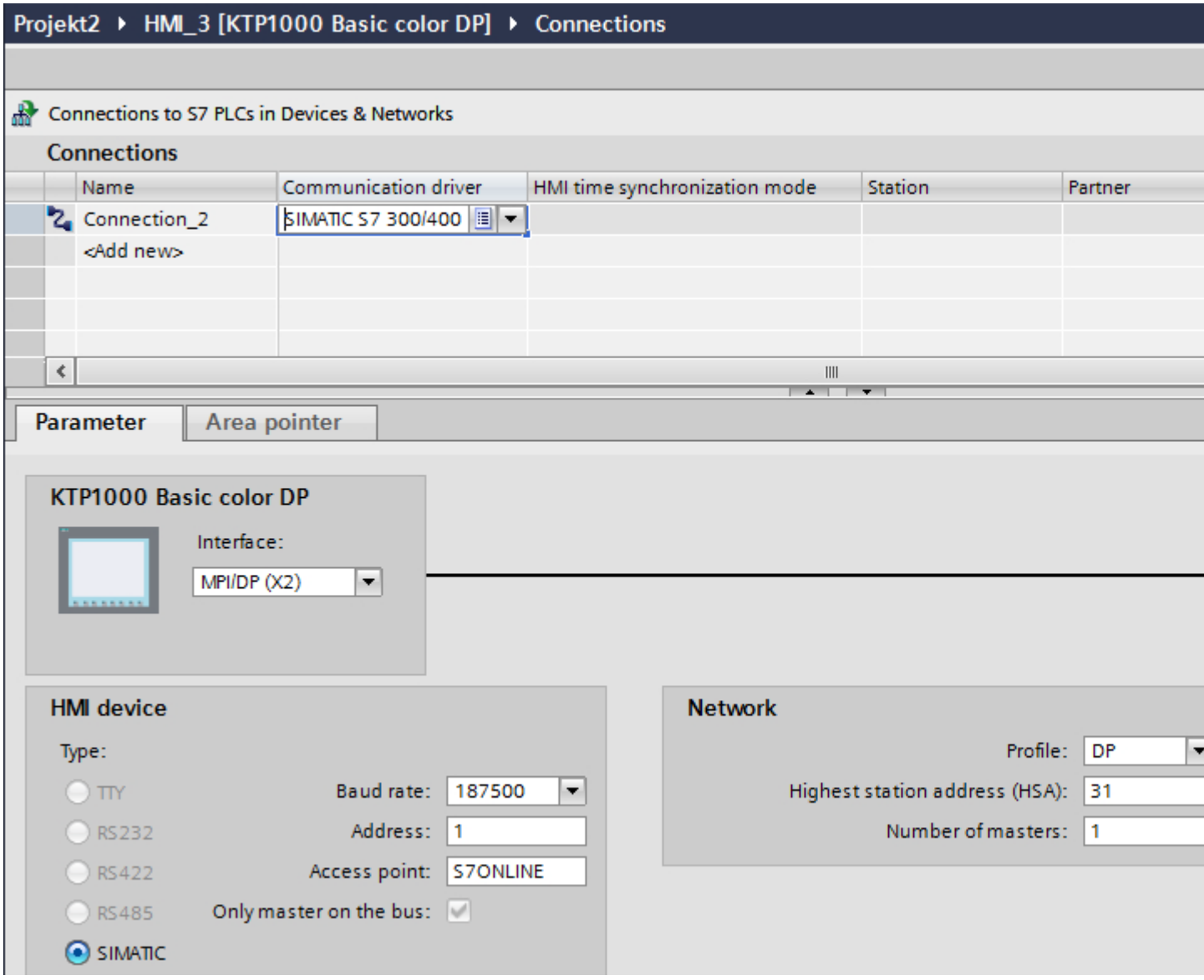
1.8.7.4 Parameters for the connection (Basic Panels)

Parameters for the connection (SIMATIC S7 300/400) (Basic Panels)

Parameters to be set

To set the connection parameters, such as addresses and profiles, click the connection that you have created in the "Connections" editor.

The communication partners are displayed schematically in the Inspector window under "Parameters". The "HMI device", "Network", and "PLC" areas are available for assigning parameters according to the interface used.



See also

Communication with SIMATIC S7 300/400 (Page 209)

Ethernet parameters (Basic Panels)

Parameters for the HMI device

You set the parameters for the HMI device in the network under "HMI device".. The changes are not transferred automatically to the HMI device. You must change the settings in the Control Panel of the HMI device.

- "Interface"
  If you are directly connected to the HMI device during configuration, you can set up the IP address of the HMI device in WinCC.

  Note

  The IP address in the Control Panel will be overwritten upon subsequent loading if you have already set up the IP address in the HMI device control panel.

  The IP address already set up in the Control Panel will be retained upon subsequent loading if you activate "Set IP address using a different method".

  The IP address is transferred to the HMI device during project transfer.
  To set up the IP address of the HMI device:

  – Click the HMI device.
  – Open the "Device configuration" editor.
  – Click the Ethernet interface.
  – Assign the IP address in the inspector window under:
    "General > PROFINET interface > Ethernet addresses"
- "Address"
  You assign the IP address of the HMI device in the "Address" area.
  When you transfer the WinCC project to the HMI device, this IP address is set up directly in the HMI device.
- "Access point"
  The access point defines a logical device name through which the communication partner can be reached.

Parameters for the PLC

Under "PLC", you address the S7 module with which the HMI device will exchange data. Assign a name for the connection for each communication partner.

- "Address"
  Under "Address", set the IP address of the S7 module to which the HMI device is connected.
- "Expansion slot"
  Defines the number of the expansion slot of the CPU to be addressed.

- "Rack"
  Defines the rack number of the CPU to be addressed.
- "Cyclic operation"

  Note

  The setting "Cyclic operation" cannot be configured for the SIMATIC S7 1200 PLC.

  When cyclic operation is enabled, the PLC optimizes the data exchange between the HMI device and the PLC. This increases system performance.
  Disable cyclic mode if you are operating several HMI devices in parallel.

See also

Communication with SIMATIC S7 300/400 (Page 209)

PROFIBUS parameters (Basic Panels)

Parameters for the HMI device

You assign the parameters for the HMI device in the network once under "HMI device". The change applies to each communication partner.

- "Type"
  Specifies the physical connection used.
- "Interface"
  For "Interface", you select the HMI device interface via which the HMI device is connected to the PROFIBUS network.
- "Baud rate"
  For "Baud rate", you set the transmission speed of the data in the network. The baud rate is determined by the slowest HMI device connected to the network. The setting must be identical throughout the network.

  Note

  If you set a baud rate of 1.5 Mbaud for OP 73 or OP 77A, the highest station address must be less than or equal to 63.

- "Address"
  You set the PROFIBUS address of the HMI device under "Address". The PROFIBUS address must be unique in the PROFIBUS network.
- "Sole master on bus"
  Disables an additional safety feature against bus faults when the HMI device is connected to the network. A passive station (slave) can only send data if it is requested to do so by an active station (master).
  In S7 200, you must set an HMI device as the master.
- "Access point"
  The access point defines a logical device name through which the communication partner can be reached.

Parameters for the network

Under "Network", you set the parameters for the PROFIBUS network to which the HMI device is linked.

- "Profile"
  For "Profile", you select the network profile that is used in the network. In "Profile", set "DP", "Universal", or "Standard". The setting must be identical throughout the network.
- "Highest address"
  For "Highest station address", set the highest station address. The highest station address must be greater than or equal to the highest actual PROFIBUS address. The setting must be identical throughout the network.

  Note

  If you set a baud rate of 1.5 Mbaud in the OP 73 or the OP 77A, the highest station address must be less than or equal 63.

- "Number of masters"
  For "Number of masters", set the number of masters in the PROFIBUS network. This information is necessary to correctly calculate the bus parameters.

Parameters for the PLC

Under "PLC", you address the S7 module with which the HMI device will exchange data. Assign a name for the connection for each communication partner.

- "Address"
  For "Address", set the PROFIBUS address of the S7 module (CPU, FM, or CP) to which the HMI device is connected.
- "Cyclic operation"

  Note

  The setting "Cyclic operation" cannot be configured for the SIMATIC S7 1200 PLC.

  When cyclic operation is enabled, the PLC optimizes the data exchange between the HMI device and the PLC. This increases system performance. Disable cyclic mode if you are operating several HMI devices in parallel. This setting is not required for SIMATIC S7-200 PLCs.

See also

Communication with SIMATIC S7 300/400 (Page 209)

MPI parameters (Basic Panels)

Parameters for the HMI device

You assign the parameters for the HMI device in the network once under "HMI device". The change applies to each communication partner.

- "Type"
  Specifies the physical connection used.
- "Interface"
  For "Interface", you select the HMI device interface via which the HMI device is connected to the MPI network.
- "Baud rate"
  For "Baud rate", you set the transmission speed of the data in the network. The baud rate is determined by the slowest HMI device connected to the network. The setting must be identical throughout the network.

  Note

  If you set a baud rate of 1.5 Mbaud for OP 73 or OP 77A, the highest station address must be less than or equal to 63.

- "Address"
  For "Address", you set the MPI address of the HMI device. The MPI address must be unique throughout the MPI network.
- "Sole master on bus"
  Disables an additional safety feature against bus faults when the HMI device is connected to the network. A passive station (slave) can only send data if it is requested to do so by an active station (master). If you have connected only slaves to the HMI device, you must therefore disable the "Sole master on bus" safety feature.
  In S7 200, you must set an HMI device as the master.

Parameters for the network

Under "Network", you set the parameters for the MPI network to which the HMI device is linked.

- "Profile"
  For "Profile", you select the network profile that is used in the network. In "Profile", set "MPI". The setting must be identical throughout the network.
- "Highest address"
  For "Highest station address", set the highest station address. The highest station address must be greater than or equal to the highest actual MPI address. The setting must be identical throughout the network.
- "Number of masters"
  This setting is not required for MPI.

Parameters for the PLC

Under "PLC", you address the S7 module with which the HMI device will exchange data. Assign a name for the connection for each communication partner.

- "Address"
  For "Address", set the MPI address of the S7 module (CPU, FM, or CP) to which the HMI device is connected.
- "Cyclic mode"
  When cyclic mode is enabled, the PLC optimizes the data exchange between the HMI device and the PLC. This improves system performance. Disable cyclic mode if you are operating several HMI devices in parallel. This setting is not required for SIMATIC S7-200.

See also

Communication with SIMATIC S7 300/400 (Page 209)

Cyclic operation (Basic Panels)

Handling the "Cyclic operation" selection

If "Cyclic operation" is enabled, the HMI device sends a message frame to the CPU at the beginning of communication indicating that certain tags are required on a recurring basis.

The CPU then always transmits the data at the same cyclic interval. This saves the HMI device from having to output new requests for the data.

If cyclic operation is disabled, the HMI device sends a request whenever information is required.

Additional properties:

- Cyclic operation reduces data transmission load at the HMI device. The PLC resources are used to relieve load on the HMI device.
- The PLC only supports a certain number of cyclic services. The HMI device handles the operation if the PLC cannot provide any further resources for cyclic services.
- The HMI device generates the cycle if the PLC does not support the cyclic mode.
- Screen tags are not integrated in cyclic operation.
- Cyclic mode is only set up at the restart of Runtime.
- The HMI device transfers several jobs to the PLC if cyclic mode is enabled, depending on the PLC.
- The HMI device only transfers one job to the PLC if cyclic mode is disabled.

See also

Communication with SIMATIC S7 300/400 (Page 209)

1.9 Communicating with the SIMATIC S7-1500 Software Controller (Basic Panels)

1.9.1 Communication with SIMATIC S7-1500 Software Controller (Basic Panels)

Introduction

This section describes the communication between a HMI device and a SIMATIC S7-1500 Software Controller.

You can configure the following communication channels for the SIMATIC S7-1500 Software Controller :

- PROFINET
- PROFIBUS

You can find additional information on communication via PROFIBUS at Communication via PROFIBUS (Page 97).

HMI connection for communication

You configure connections between the HMI device and SIMATIC S7-1500 Software Controller in the "Devices & Networks" editor.

See also

Configuring alarms (Page 321)

Communication via PROFIBUS (Page 290)

Configuring an HMI connection via PROFIBUS (Page 291)

PROFIBUS parameters for the HMI connection (Page 293)

PROFIBUS parameters for the HMI device (Page 294)

PROFIBUS parameters for the PLC (Page 296)

Communication via PROFIBUS (Page 97)

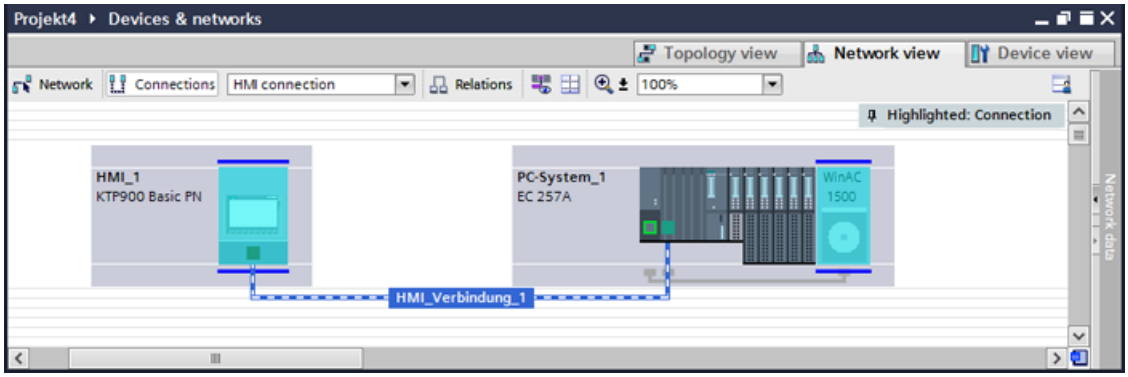
1.9.2 Communication via PROFINET (Basic Panels)

1.9.2.1 Configuring an HMI connection (Basic Panels)

Communication via PROFINET (Basic Panels)

HMI connections via PROFINET

If you have inserted an HMI device and a SIMATIC WinAC 1500 into the project, interconnect the two PROFINET interfaces in the "Devices & Networks" editor.



You can also connect multiple HMI devices to a single SIMATIC WinAC 1500 and multiple SIMATIC WinAC 1500 to a single HMI device.

The maximum number of communication partners that you can connect to an HMI device is dependent on the HMI device used.

Additional information is available in the documentation for the respective HMI device.

HMI connection in the "Devices & Networks" editor

You configure the HMI connection between the PLC and the HMI device via PROFINET in the "Devices & Networks" editor.

Connection in the "Connections" editor

Alternatively, you configure the connection between the PLC and HMI device via PROFINET in the "Connections" editor of the HMI device.

See also

Communication with SIMATIC S7-1500 Software Controller (Page 275)

Configuring an HMI connection via PROFINET (Page 277)

Configuring an HMI connection via PROFINET (Basic Panels)

Introduction

You configure a HMI connection between HMI devices and a SIMATIC S7-1500 Software Controller via PROFINET or Ethernet in the "Devices & Networks" editor.

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.

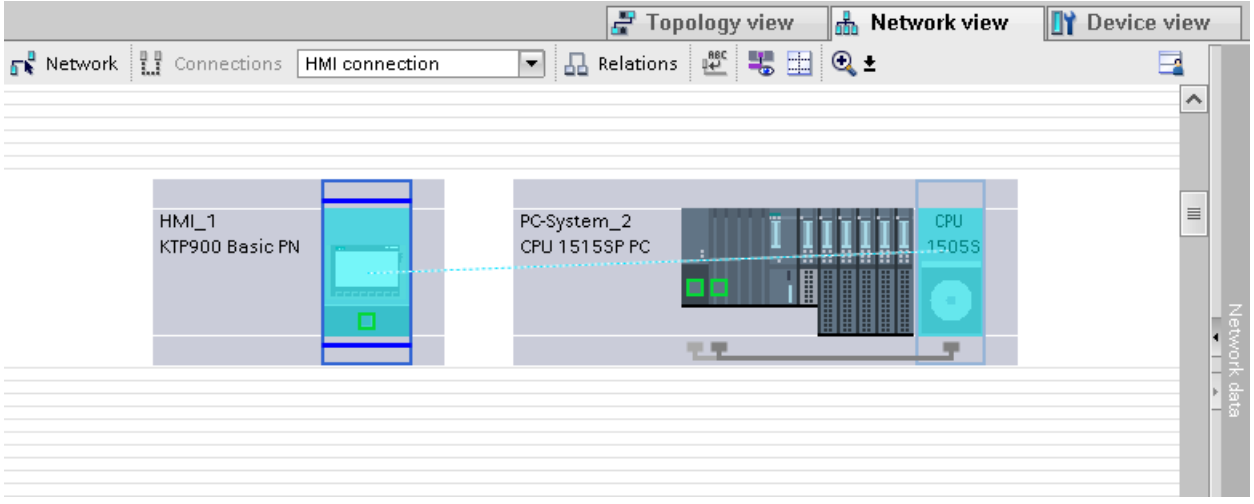
Requirements

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

- HMI device with PROFINET or Ethernet interface
- SIMATIC S7-1500 Software Controller on PC system with PROFINET interface.

Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed graphically in the network view.
2. Click the "Connections" button and select "HMI connection" for the connection type.
   The devices available for connection are highlighted in color.
3. Click the HMI device and use a drag-and-drop operation to draw a connection to the SIMATIC S7-1500 Software Controller.



4. Click the connecting line.

5. Click "Highlight HMI connection" and select the HMI connection.
   The connection is displayed graphically in the Inspector window.
6. Click the communication partners in the "Network view" and change the PROFINET parameters in the Inspector window according to the requirements of your project.
   For additional information on this, see section "PROFINET parameters (Page 278)".

Note

The created HMI connection is also shown in the tabular area of the editor on 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 a SIMATIC S7-1500 Software Controller. The IP address and subnet mask connection parameters are configured.

See also

Communication with SIMATIC S7-1500 Software Controller (Page 275)

Communication via PROFINET (Page 276)

PROFINET parameters (Page 278)

1.9.2.2 PROFINET parameters (Basic Panels)

PROFINET parameters for the HMI connection (Basic Panels)

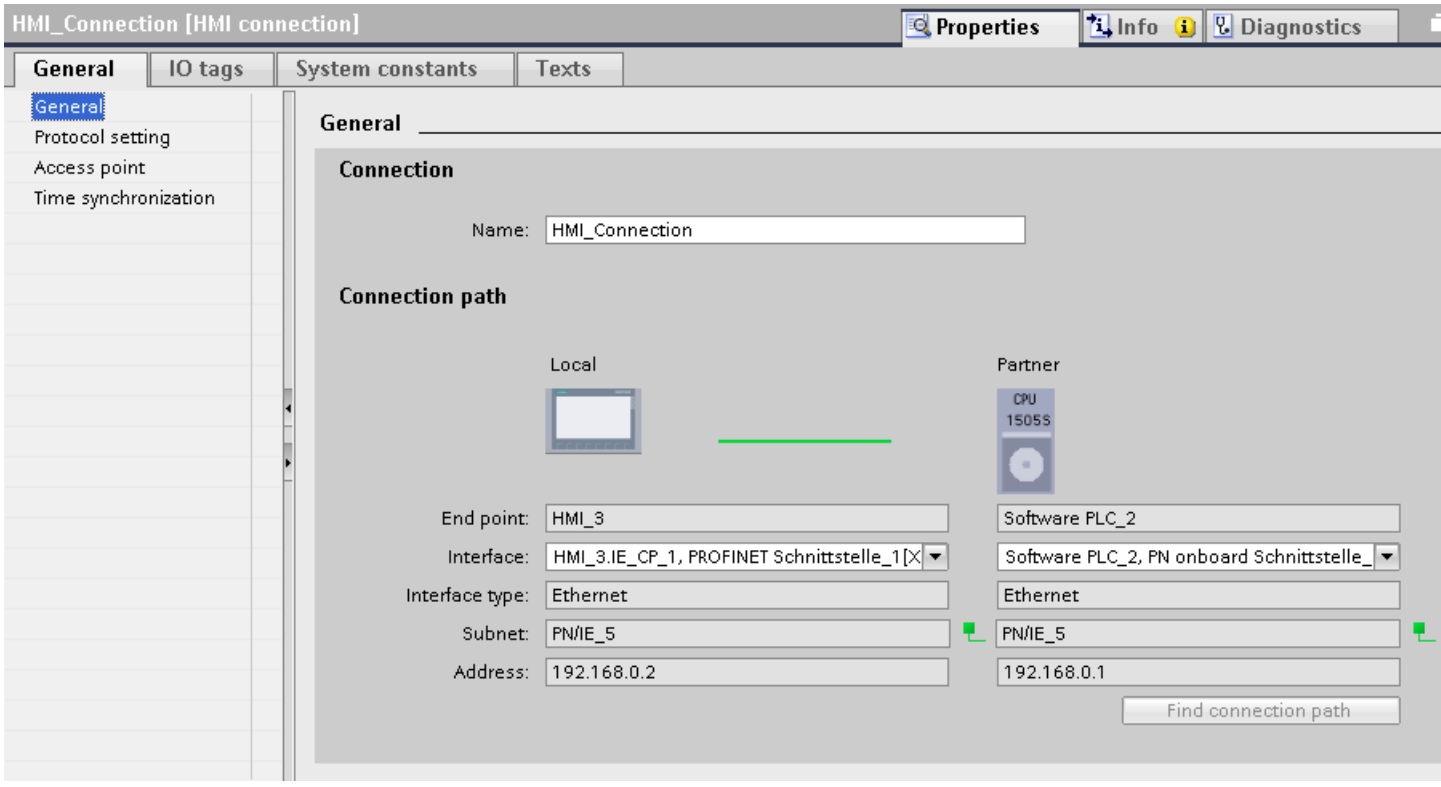
PROFINET parameters for the HMI connection

An overview of the configured HMI connection parameters can be found in the properties for the HMI connection.

Only limited changes are possible in this Inspector window.

Displaying and changing the HMI connection parameters

1. Click the HMI connection in the "Devices & Networks" editor.
2. Change the parameters of the HMI connection in the Inspector window under "Properties > General > General".



"Connection"

Shows the name of the HMI connection.

"Connection path"

The communication partners of the selected HMI connection and the associated PROFINET parameters are displayed in the "Connection path" area. Some of the areas displayed cannot be edited in this dialog.

- "End point"
  Displays the device name. This area cannot be edited.
- "Interface"
  Displays the selected interface of the device. You can choose between several interfaces, depending on the device.
- "Interface type"
  Displays the selected interface type. This area cannot be edited.
- "Subnet"
  Displays the selected subnet. This area cannot be edited.

- "Address"
  Displays the selected IP address of the device. This area cannot be edited.
- "Find connection path" button
  Enables the subsequent specification of connections.

See also

Communication via PROFINET (Page 276)

Communication with SIMATIC S7-1500 Software Controller (Page 275)

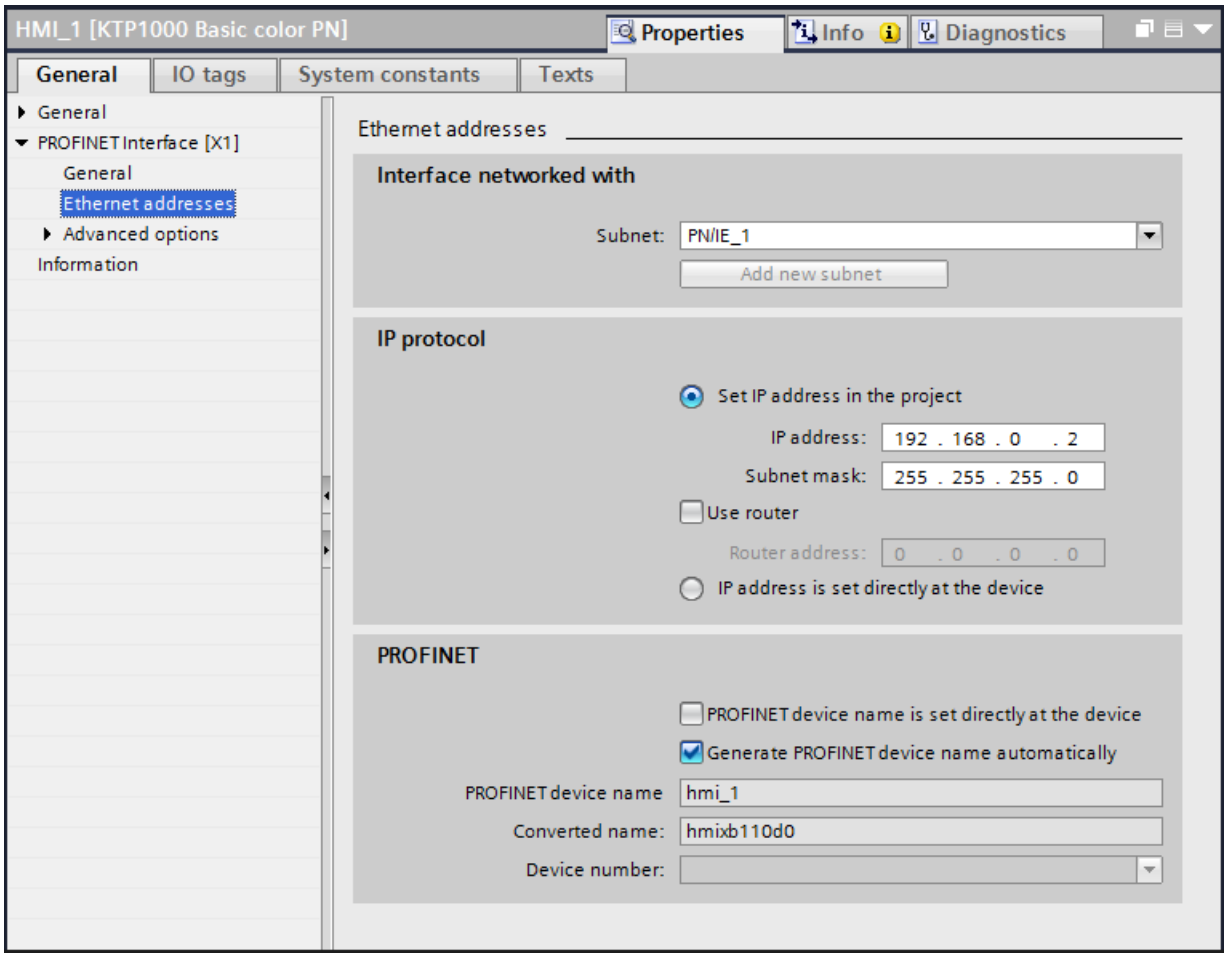
PROFINET parameters for the HMI device (Basic Panels)

PROFINET parameters for the HMI device

An overview of the configured HMI device parameters can be found in the properties for the HMI device.

Displaying and changing PROFINET parameters of the HMI device

1. Click the HMI device in the "Devices & Networks" editor.
2. Change the parameters of the HMI device in the Inspector window under "Properties > General".



"Interface networked with"

In the "Interface networked with" area, select the subnet of the HMI connection via which the HMI device is connected to the network. You use the "Add new subnet" button to create a new subnet.

"IP protocol"

- "Set IP address in the project"
  When you transfer the WinCC project to the HMI device, this IP address is set up directly in the HMI device.

  Note

  The device is automatically restarted in the case of HMI devices with the Windows CE 3.0 operating system.

  HMI devices with Windows CE 3.0:
  - OP 77B
  - TP 177B color PN/DP
  - TP 177B mono DP
  - OP 177B color PN/DP
  - OP 177B mono DP
  - Mobile Panel 177 PN
  - Mobile Panel 177 DP
  - TP 277 6"
  - OP 277 6"

- "Subnet mask"
  You assign data of the subnet mask in the "Subnet mask" area.
- "Use IP router"
  If you are using an IP router, select "Use IP router" and enter the router address in the "Router address" field.
- "IP address is set directly at the device"
  If the function "Set IP address using a different method" is activated, the IP address is not taken from the project. You have to enter the IP address directly in the Control Panel of the HMI device.

See also

Communication via PROFINET (Page 276)

Communication with SIMATIC S7-1500 Software Controller (Page 275)

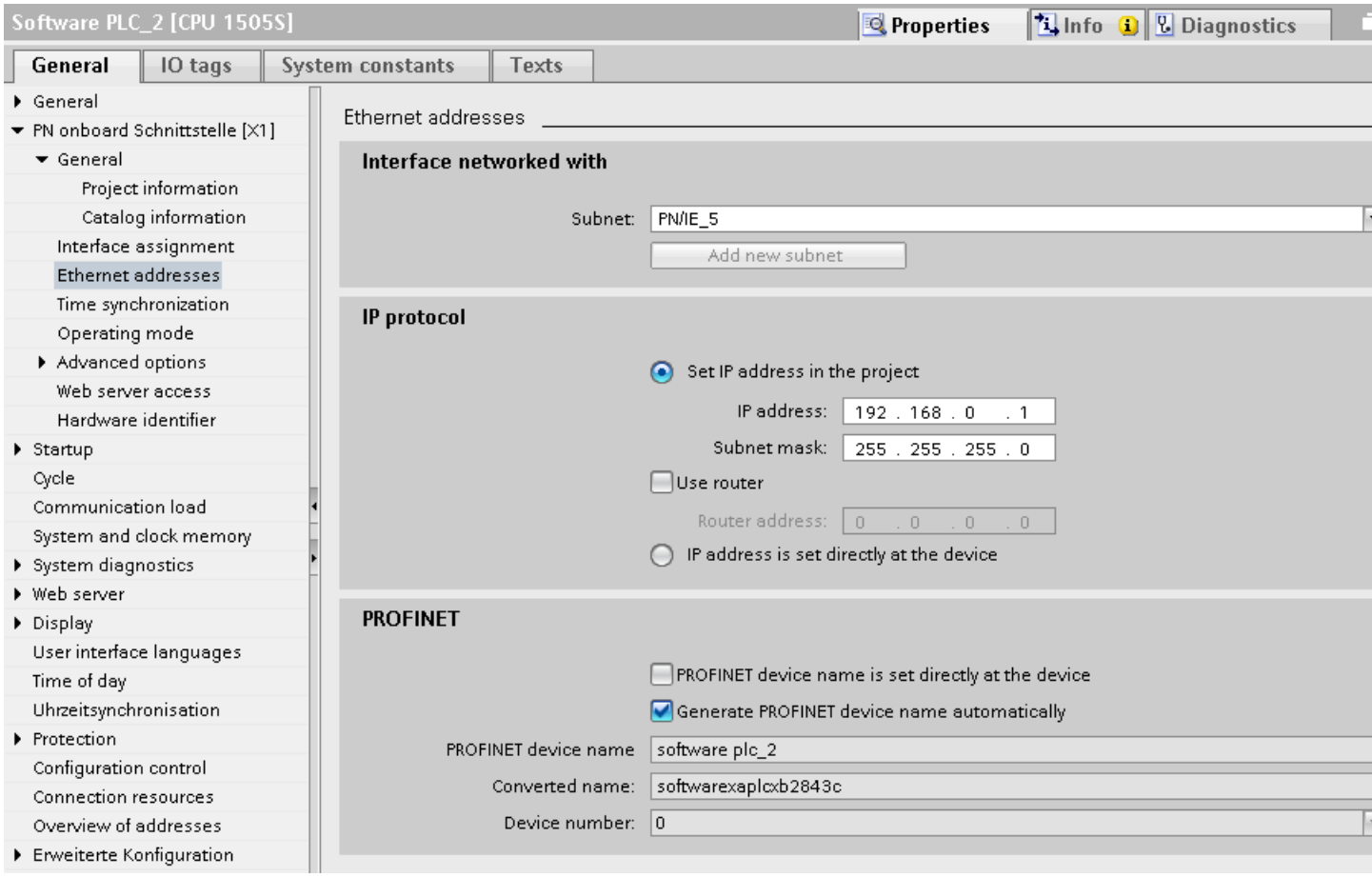
PROFINET parameters for the PLC (Basic Panels)

PROFINET parameters for the PLC

An overview of the configured parameters can be found in the properties for the PLC.

Displaying and changing PROFINET parameters of the PLC

1. Click the PLC in the "Devices & Networks" editor.
2. Change the parameters of the PLC in the Inspector window under "Properties > General > General".



"Interface networked with"

In the "Subnet" area, select the subnet of the HMI connection via which the PLC is connected to the network. You use the "Add new subnet" button to create a new subnet.

"IP protocol"

- "Interface type"
  Depending on the HMI device type, you have various interfaces to choose from.
- "IP address"
  You assign the IP address of the HMI device in the "IP address" area.
- "Subnet mask"
  You assign data of the subnet mask in the "Subnet mask" area.
  If you are using an IP router, select "Use IP router" and enter the router address in the field.

See also

Communication via PROFINET (Page 276)

Communication with SIMATIC S7-1500 Software Controller (Page 275)

Protection of communication (Basic Panels)

Security levels (Basic Panels)

If you want to protect the PLC and HMI device communication, you can assign protection levels for the communication.

For an SIMATIC S7-1500 Software Controller, you can enter multiple passwords to set up different access rights for different user groups.

The passwords are entered in a table, so that exactly one protection level is assigned to each password.

The effect of the password is given in the "Protection" column.

Example

You select the "Complete protection" protection level for a standard CPU (i.e., not an F-CPU) when configuring it.

Afterwards, you enter a separate password for every protection level above it in the table.

For users who do not know any of the passwords, the CPU is completely protected. Not even HMI access is possible.

For users who know one of the assigned passwords, the effect depends on the table row in which the password occurs:

- The password in row 1 (no protection) allows access as if the CPU were completely unprotected. Users who know this password have unrestricted access to the CPU.
- The password in row 2 (write protection) allows access as if the CPU were write-protected. Despite knowing the password, users who know this password only have read access to the CPU.
- The password in row 3 (read and write protection) allows access as if the CPU were read-protected and write-protected, so that only HMI access is possible for users who know this password.

See also

Access password for the HMI connection (Page 285)

Access password for the HMI connection (Basic Panels)

Introduction

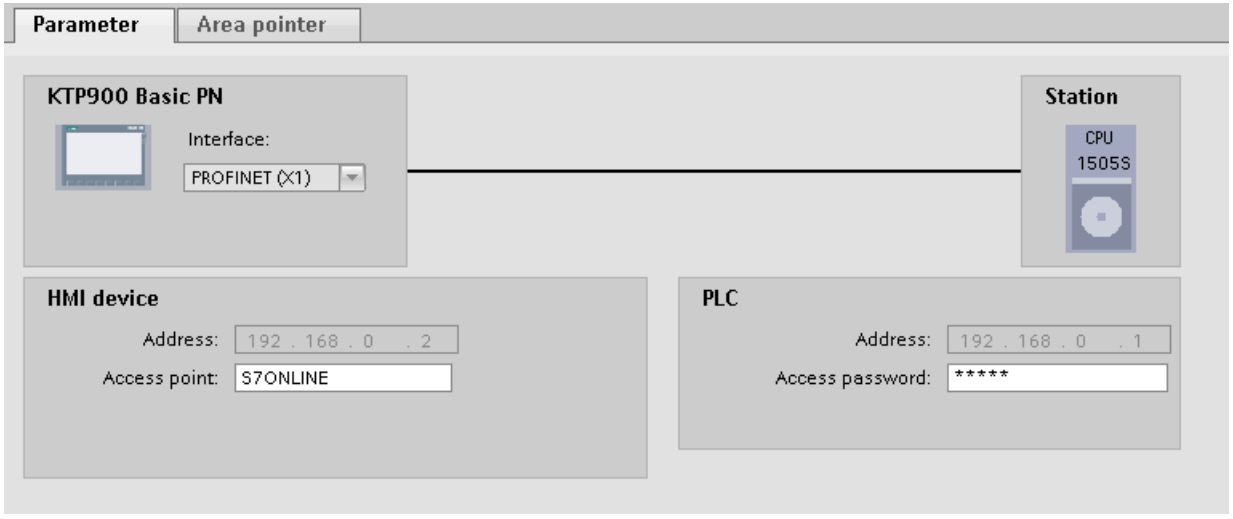
Communication with a PLC with the"Complete protection" protection level is protected by a password. The password is stored in the properties of the PLC.

Enter the password from the PLC in the "Access password" area.

Communication to the PLC is denied if an incorrect password or no password is entered.

Entering the access password

Enter the access password for the HMI connection in the "Connections" editor.



See also

Security levels (Page 284)

1.9.2.3 Setting port options (Basic Panels)

Setting the port options (Basic Panels)

Changing connection settings for the PROFINET IO port

You can change the network settings for the PROFINET IO port as required. By default, the settings are made automatically. In normal situations, this guarantees problem-free communication.

Possible settings for transmission rate / duplex

Depending on the selected device, you can make the following settings for "Transmission rate / duplex":

- Automatic setting
  Recommended default setting of the port. The transmission settings are automatically "negotiated" with the peer port. The "Enable autonegotiation" option is also enabled as a default, in other words, you can use cross cables or patch cables for the connection.
- TP/ITP at x Mbps full duplex (half duplex)
  Setting of the transmission rate and the full duplex/half duplex mode. The effectiveness depends on the "Enable autonegotiation" setting:
  - Autonegotiation enabled
    You can use both cross cable and patch cable.
  - Autonegotiation disabled
    Make sure that you use the correct cable (patch cable or cross cable)! The port is also monitored with this setting.
- Deactivated
  Depending on the module type, the drop down list box can contain the "- Disabled -" option. This option, for example, allows you to prevent access to an unused port for security reasons. With this setting, diagnostic events are not generated.

"Monitor" option

This option enables or disables port diagnostics. Examples of port diagnostics: The link status is monitored, in other words, the diagnostics are generated during link-down and the system reserve is monitored in the case of fiber optic ports.

Option "Enable autonegotiation "

The autonegotiation setting can only be changed if a concrete medium (for example, TP 100 Mbps full duplex) is selected. Whether or not a concrete medium can be set depends on the properties of the module.

If autonegotiation is disabled, this causes the port to be permanently specified, as for example, is necessary for a prioritized startup of the IO device.

You must make sure the partner port has the same settings because with this option the operating parameters of the connected network are not detected and the data transmission rate and transmission mode can accordingly not be optimally set.

Note

When a local port is connected, STEP 7 makes the setting for the partner port if the partner port supports the setting. If the partner port does not accept the setting, an error message is generated.

See also

Wiring rules for disabled autonegotiation (Page 287)

Boundaries at the port (Page 288)

Wiring rules for disabled autonegotiation (Basic Panels)

Requirements

You have made the following settings for the port in question, for example, to accelerate the startup time of the IO device:

- Fixed transmission rate
- Autonegotiation incl. autocrossing disabled

The time for negotiating the transmission rate during startup has been saved.

If you have disabled autonegotiation, you must observe the wiring rules.

Wiring rules for disabled autonegotiation

PROFINET devices have the following two types of ports:

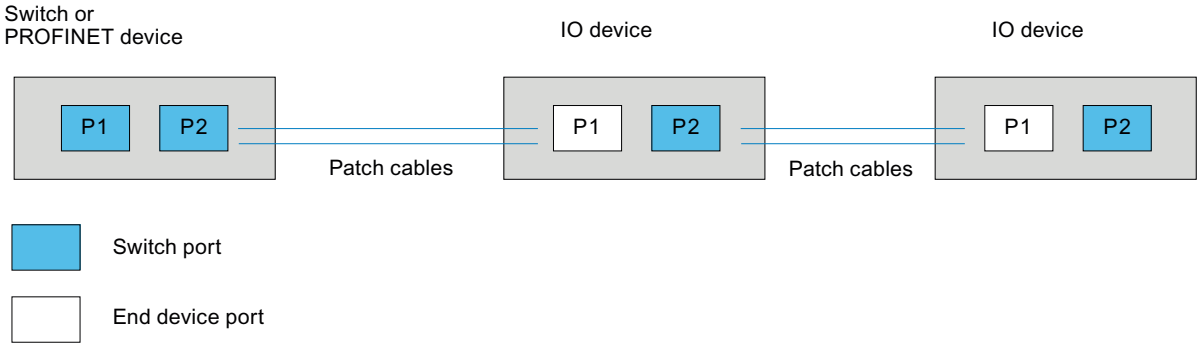
Type of port	PROFINET devices	Note
Switch port with crossed pin assignment	For IO devices: Port 2 For S7 CPUs with 2 ports: Ports 1 and 2	Crossed pin assignment means that the pin assignment for the ports for sending and receiving between the respective PROFINET devices is exchanged internally.
End device port with uncrossed pin assignment	For IO devices: Port 1 For S7 CPUs with one port: Port 1	-

Validity of the wiring rules

The cabling rules described in the following paragraph apply exclusively for the situation in which you have specified a fixed port setting.

Rules for cabling

You can connect several IO devices in line using a single cable type (patch cable). To do this, you connect port 2 of the IO device (distributed I/O) with port 1 of the next IO device. The following graphic gives an example with two IO devices.



See also

Setting the port options (Page 285)

Boundaries at the port (Page 288)

Boundaries at the port (Basic Panels)

Requirements

The respective device must support boundaries settings in order to work with boundaries. If the device for PROFINET does not support boundaries settings, the corresponding parameters are disabled.

Enable boundaries

"Boundaries" are limits for transmission of certain Ethernet frames. The following boundaries can be set at a port:

- "End of discovery of accessible devices"
  No forwarding of DCP frames to identify accessible devices. Devices downstream from this port cannot be reached by the project tree under "Accessible devices". Devices downstream from this port cannot be reached by the CPU.
- "End of topology discovery"
  LLDP frames (Link Layer Discovery Protocol) are not forwarded for topology detection.
- "End of sync domain"
  No forwarding of sync frames transmitted to synchronize nodes within a sync domain.
  If you operate, for example, a PROFINET device with more than two ports in a ring, you should prevent the sync frame from being fed into the ring by setting a sync boundary (at the ports not inside the ring).
  Additional example: If you want to use several sync domains, configure a sync domain boundary for the port connected to a PROFINET device from the other sync domain.

Restrictions

The following restrictions must be observed:

- The individual check boxes can only be used if the port supports the function in question.
- If a partner port has been determined for the port, the following check boxes cannot be used:
  - "End of discovery of accessible devices"
  - "End of topology discovery"

See also

Setting the port options (Page 285)

Wiring rules for disabled autonegotiation (Page 287)

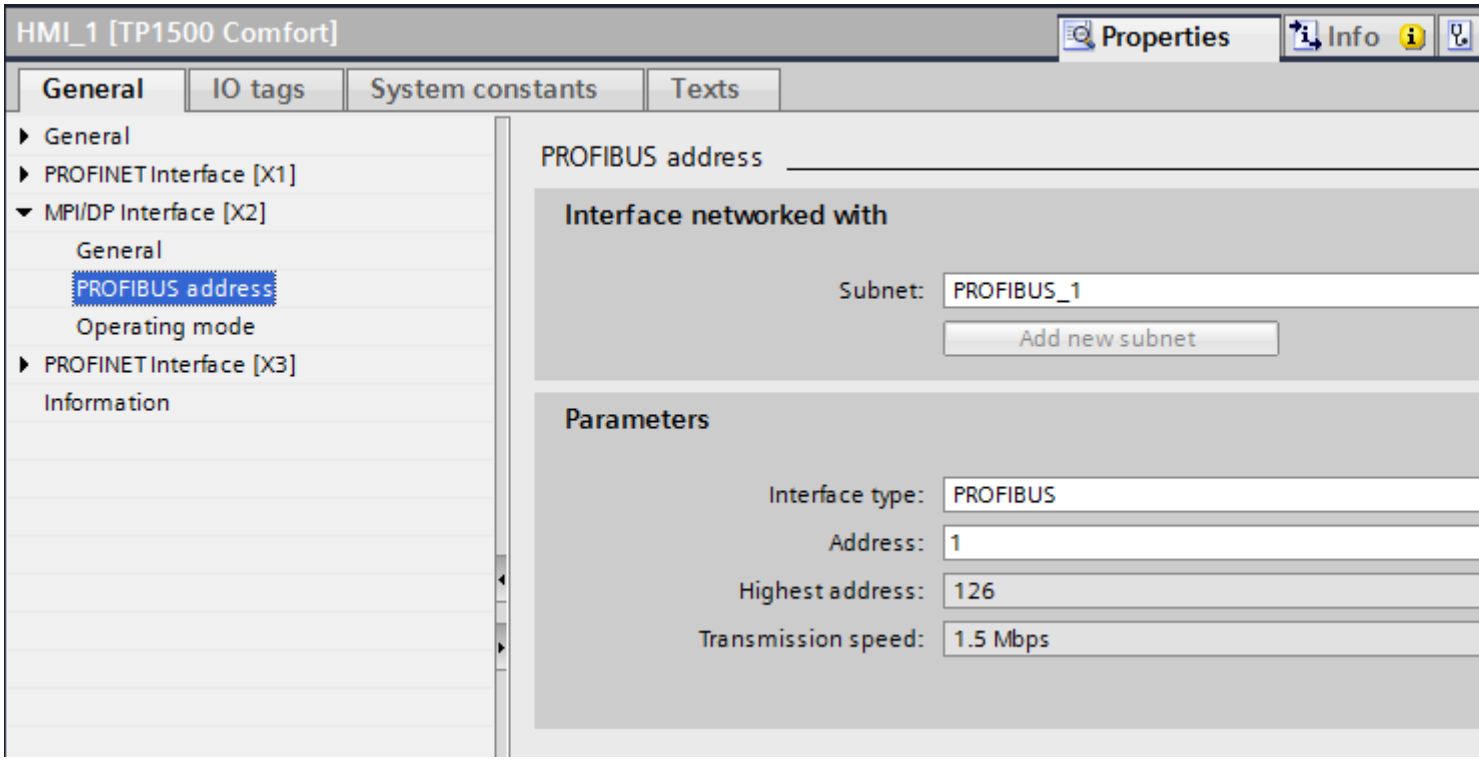
1.9.3 Communication via PROFIBUS (Basic Panels)

1.9.3.1 Configuring an HMI connection (Basic Panels)

Communication via PROFIBUS (Basic Panels)

HMI connections via PROFIBUS

If you have inserted an HMI device and a SIMATIC S7 1500 into the project, interconnect the two PROFIBUS interfaces in the "Devices & Networks" editor.



HMI connection in the "Devices & Networks" editor

You configure the HMI connection between the PLC and the HMI device via PROFIBUS in the "Devices & Networks" editor.

Connection in the "Connections" editor

Alternatively, you configure the connection between the PLC and HMI device in the "Connections" editor of the HMI device.

See also

Communication with SIMATIC S7-1500 Software Controller (Page 275)

Configuring an HMI connection via PROFIBUS (Page 291)

Configuring an HMI connection via PROFIBUS (Basic Panels)

Introduction

You configure an HMI connection over PROFIBUS between HMI devices and a SIMATIC S7 1500 in the "Devices & Networks" editor.

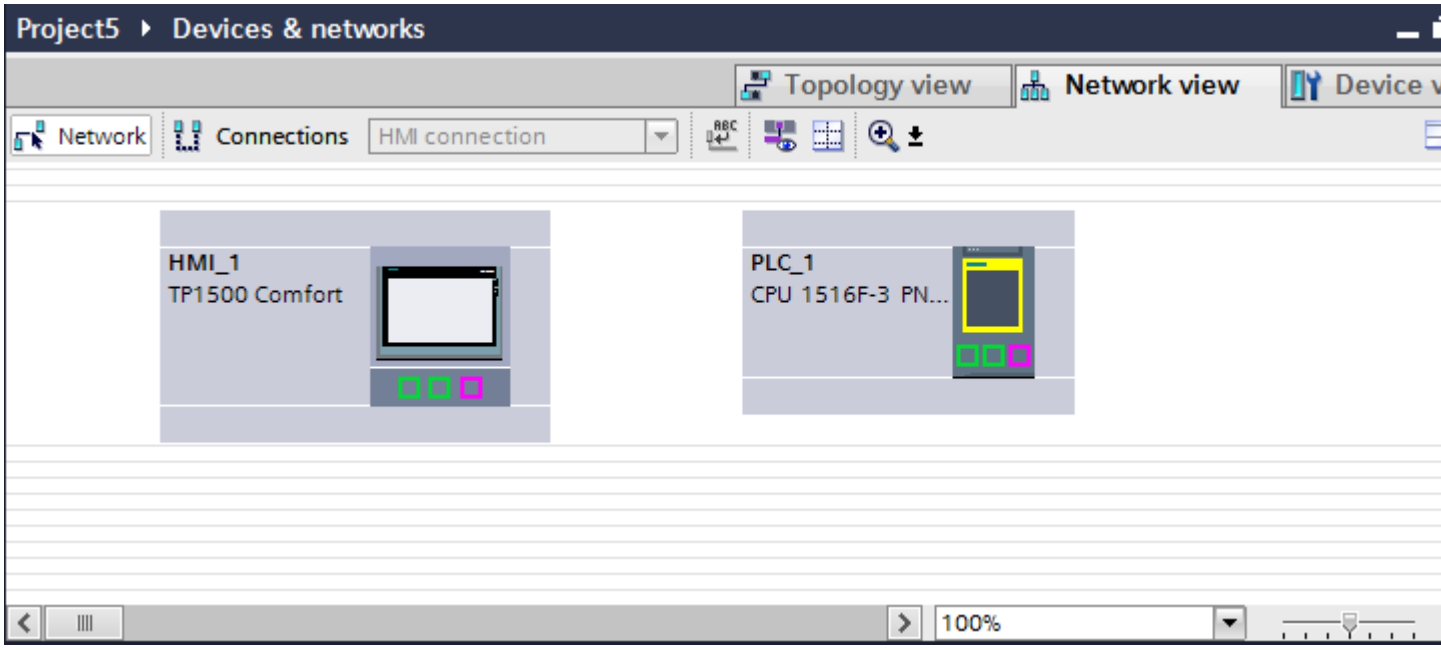
Requirements

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

- HMI device with MPI/DP interface
- SIMATIC S7 1500 with PROFIBUS interface

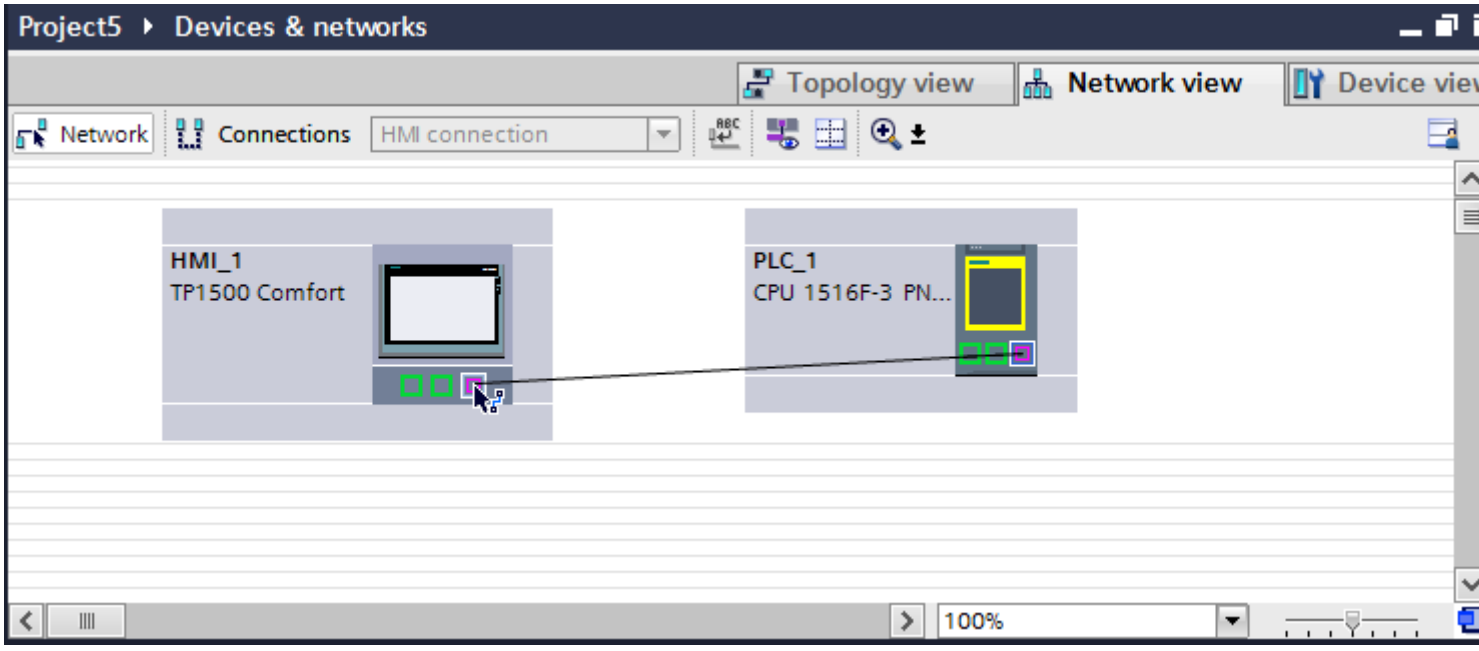
Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed graphically in the network view.



2. Click the "Connections" button.
   The devices available for connection are highlighted in color.
3. Click the HMI device interface.

4. Select the "PROFIBUS" interface type in the Inspector window under "Properties > General > HMI MPIDP > Parameters".
5. Click the interface of the PLC and use a drag-and-drop operation to draw a connection to the HMI device.



6. Click the connecting line.
7. Click "Highlight HMI connection" and select the HMI connection.
   The connection is displayed graphically in the Inspector window.
8. Click the communication partners in the "Network view" and change the PROFIBUS parameters in the Inspector window according to the requirements of your project.
   For additional information on this, see section "PROFIBUS parameters (Page 293)".

Note

The created HMI connection is also shown in the tabular area of the editor on 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 an HMI connection over PROFIBUS between an HMI device and a SIMATIC S7 1500.

See also

Communication via PROFIBUS (Page 290)

Communication with SIMATIC S7-1500 Software Controller (Page 275)

PROFIBUS parameters (Page 293)

1.9.3.2 PROFIBUS parameters (Basic Panels)

PROFIBUS parameters for the HMI connection (Basic Panels)

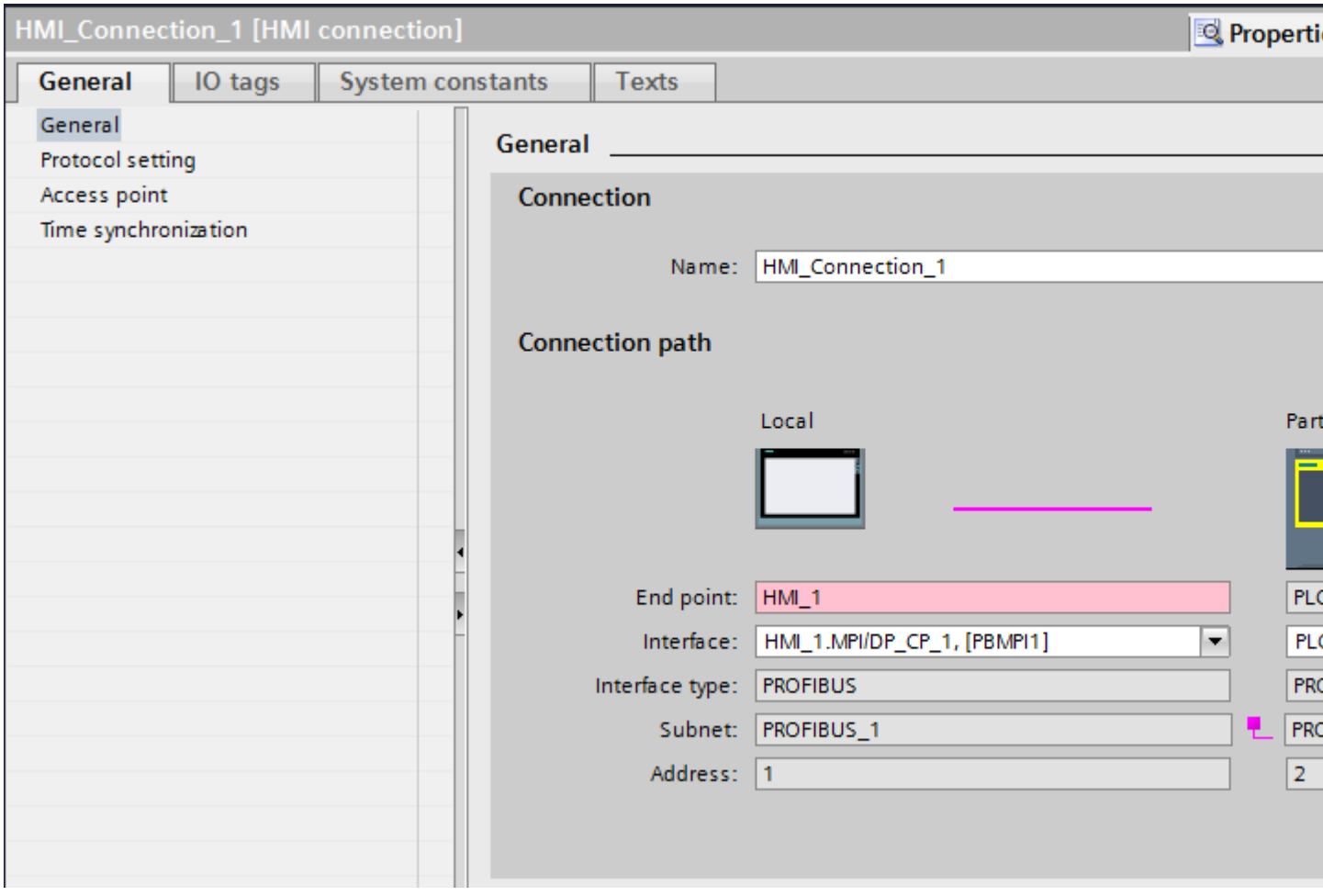
PROFIBUS parameters for the HMI connection

An overview of the configured HMI connection parameters can be found in the properties for the HMI connection.

Only limited changes are possible in this Inspector window.

Displaying and changing the HMI connection parameters

1. Click the HMI connection in the "Devices & Networks" editor.
2. Change the parameters of the HMI connection in the Inspector window under "Properties > General > General".



"Connection"

Displays whether the devices are networked together.

 - displayed if the devices are networked together.

 - displayed if the devices are not networked together.

"Connection path"

The communication partners of the selected HMI connection and the associated PROFIBUS parameters are displayed in the "Connection path" area. Some of the areas displayed cannot be edited in this dialog.

- "End point"
  Displays the device name. This area cannot be edited.
- "Interface"
  Displays the selected interface of the device. You can choose between several interfaces, depending on the device.
- "Interface type"
  Displays the selected interface type. This area cannot be edited.
- "Subnet"
  Displays the selected subnet. This area cannot be edited.
- "Address"
  Displays the PROFIBUS address of the device. This area cannot be edited.
- "Find connection path" button
  Enables the subsequent specification of connections.

See also

Communication with SIMATIC S7-1500 Software Controller (Page 275)

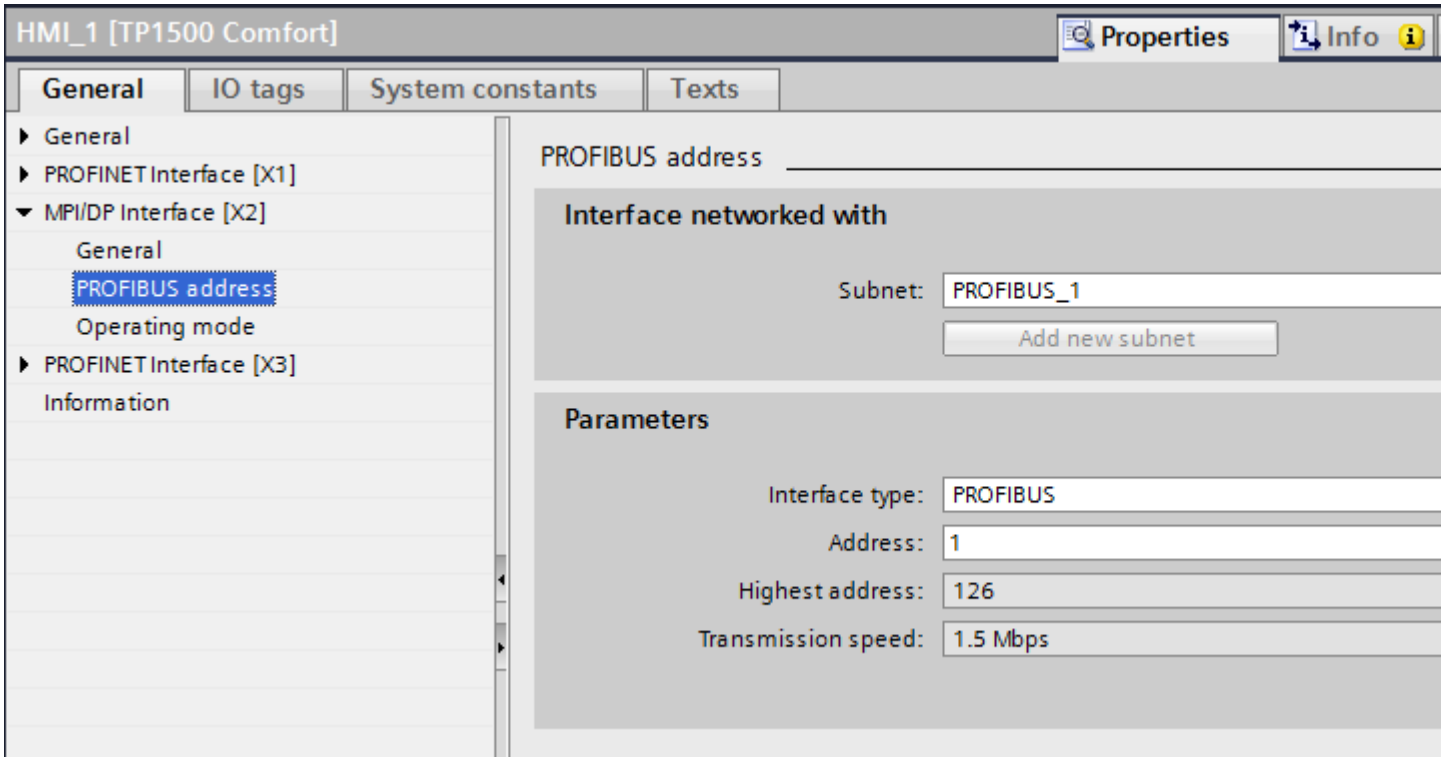
PROFIBUS parameters for the HMI device (Basic Panels)

PROFIBUS parameters for the HMI device

An overview of the configured HMI device parameters can be found in the properties for the HMI device.

Displaying and changing PROFINET parameters of the HMI device

1. Click the HMI device in the "Devices & Networks" editor.
2. Change the parameters of the HMI device in the Inspector window under "Properties > General > General".



"Interface networked with"

In the "Interface networked with" area, select the subnet of the HMI connection via which the HMI device is connected to the network. You use the "Add new subnet" button to create a new subnet.

"Parameters"

- "Interface type"
  You assign the interface type in the "Interface type" area. Depending on the HMI device type, you have various interfaces to choose from.
- "Address"
  You assign the PROFIBUS address of the HMI device in the "Address" area. The PROFIBUS address must be unique throughout the PROFIBUS network.
- "Highest address"
  The "Highest address" area displays the highest address of the PROFIBUS network.
- "Transmission speed"
  The "Transmission speed" is determined by the slowest device connected to the network. The setting is identical throughout the network.

See also

Communication with SIMATIC S7-1500 Software Controller (Page 275)

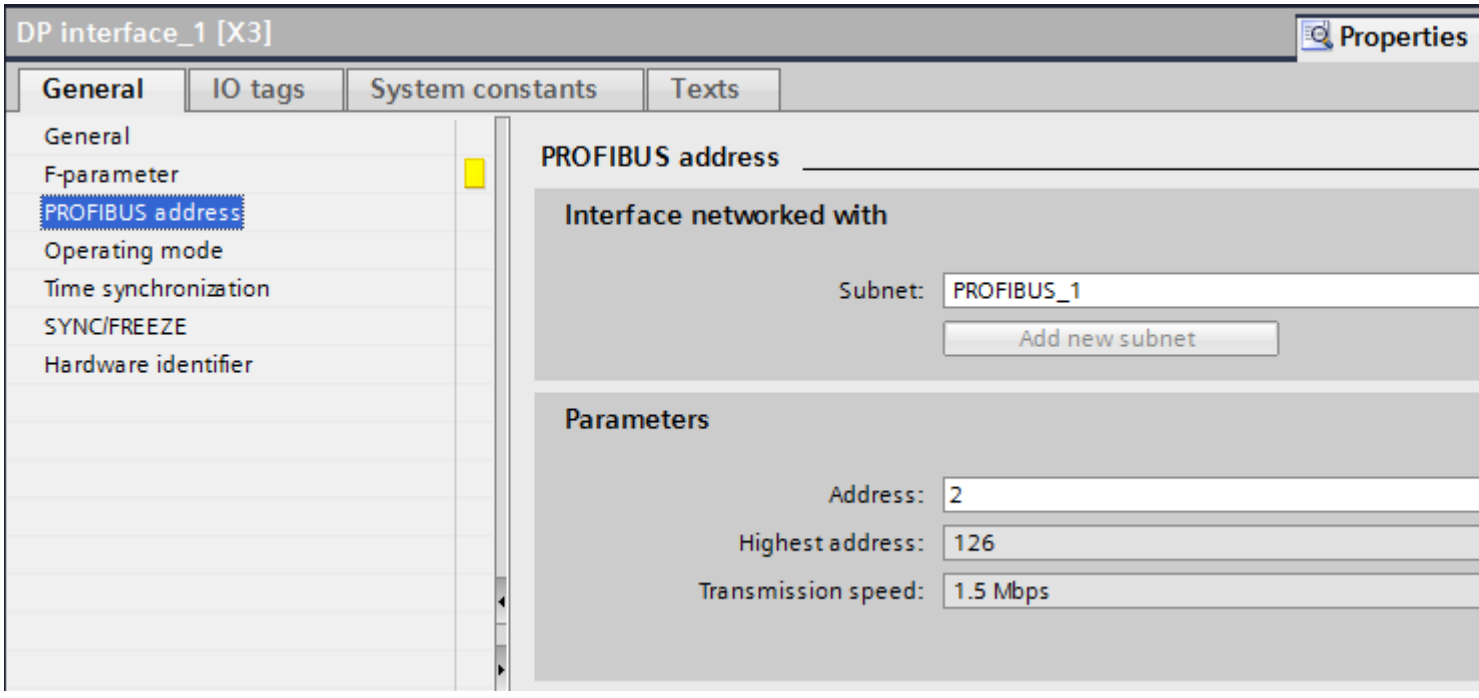
PROFIBUS parameters for the PLC (Basic Panels)

PROFIBUS parameters for the PLC

An overview of the configured parameters can be found in the properties for the PLC.

Displaying and changing PROFIBUS parameters of the PLC

1. Click the PLC in the "Devices & Networks" editor.
2. Change the parameters of the PLC in the Inspector window under "Properties > General > General".



"Interface networked with"

In the "Subnet" area, select the subnet of the HMI connection via which the PLC is connected to the network. You use the "Add new subnet" button to create a new subnet.

"Parameters"

- "Interface type"
  Depending on the HMI device type, you have various interfaces to choose from.
- "Address"
  You assign the PROFIBUS address of the HMI device in the "Address" area. The PROFIBUS address must be unique throughout the PROFIBUS network.
- "Highest address"
  The "Highest address" area displays the highest address of the PROFIBUS network.
- "Transmission speed"
  The "Transmission speed" is determined by the slowest device connected to the network. The setting is identical throughout the network.

See also

Communication with SIMATIC S7-1500 Software Controller (Page 275)

Protection of communication (Basic Panels)

Security levels (Basic Panels)

If you want to protect the controller and HMI device communication, you can assign protection levels for the communication.

For a SIMATIC S7-1500 CPU, you can enter multiple passwords and thereby set up different access rights for various user groups.

The passwords are entered in a table, so that exactly one protection level is assigned to each password.

The effect of the password is given in the "Protection" column.

For the SIMATIC S7-1500 controller, several aspects need to be considered when setting protection levels.

For additional information on this, see:

Setting options for the protection (Page 298)

Example

When configuring the controller, you select the "Complete protection" protection level for a standard CPU (i.e., not an F-CPU).

Afterwards, you enter a separate password for every protection level above it in the table.

For users who do not know any of the passwords, the CPU is completely protected. Not even HMI access is possible.

For users who know one of the assigned passwords, the effect depends on the table row in which the password occurs:

- The password in row 1 (no protection) allows access as if the CPU were completely unprotected. Users who know this password have unrestricted access to the CPU.
- The password in row 2 (write protection) allows access as if the CPU were write-protected. Despite knowing the password, users who know this password only have read access to the CPU.
- The password in row 3 (read and write protection) allows access as if the CPU were read-protected and write-protected, so that only HMI access is possible for users who know this password.

See also

Setting options for the protection (Page 298)

Access password for the HMI connection (Page 302)

Setting options for the protection (Basic Panels)

Access levels

The following section describes how to use the various access levels of the S7-1500 CPUs.

S7-1500 CPUs provide various access levels to limit the access to specific functions.

The individual access levels as well as their associated passwords are specified in the object properties of the CPU. You assign parameters for the access level in a table.

Access level	Access			Access permission
	HMI	Read	Write	Password
Full access (no protection)	✓	✓	✓	
Read access	✓	✓		
HMI access	✓			
No access (complete protection)				

The green checkmarks in the columns to the right of the respective access level specify which operations are possible without knowing the password of this access level.

If you want to use the functions of fields that are not selected in the "Access" column, a password has to be entered:

Example:

You set the access level "Read access". You can see from the table that write access is not possible during operation without entering a password.

The table also shows that full access is necessary for the "Write" function.

To use a function requiring write access during operation, the password for full access must therefore be entered.

NOTICE
Configuring an access level does not replace know-how protection Configuring access levels offers a high degree of protection against unauthorized changes to the CPU by restricting download privileges to the CPU. However, blocks on the memory card are not write- or read-protected. Use know-how protection to protect the code of blocks on the memory card.

Default characteristics

The default access level is "Full access (no protection)". Every user can read and change the hardware configuration and the blocks. A password is not set and is also not required for online access.

The access levels in detail

Below you can find an explanation of the existing access levels and which functions are possible at the respective access level.

- Full access (no protection):
  The hardware configuration and the blocks can be read and changed by all users.
- Read access for F-blocks (F-CPUs only):
  F-blocks of the safety program cannot be changed without authorization by the password of this access level or a higher level.
  Further information is available in the programming and operating manual *SIMATIC Safety - Configuring and Programming.*
- 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, as is changing the operating state (RUN/STOP).
  You cannot load blocks or a hardware configuration into the CPU. Moreover, no writing test functions and firmware updates are possible.

- HMI access:
  Only HMI access and access to diagnostics data is possible. Tags can be read and written via a HMI device.
  At this access level, you can neither load blocks and the hardware configuration into the CPU nor load blocks and the hardware configuration from the CPU into the programming device.
  In addition, the following is **not** possible: Writing test functions, changing the operating state (RUN/STOP) and firmware updates.
- No access (complete protection):
  Only identification data can be read, via "Accessible devices", for example.
  Neither read nor 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).
  Legitimation with a configured password provides you with access in accordance with the associated protection level.

Behavior of functions at different access levels

The table below describes which online functions are possible in the various protection levels.

Function	Full access	Read access	HMI access	No access
Identification of the device, via "Accessible devices", for example	Yes	Yes	Yes	Yes
HMI diagnostics view	Yes	Yes	Yes	No
Monitoring tags (M, I, Q, DB content) via HMI device	Yes	Yes	Yes	No
Modifying tags (M, I, Q, DB content) via HMI device	Yes	Yes	Yes	No
Diagnostics display (for example, device information, connection display, alarm display, diagnostic buffer)	Yes	Yes	Yes	No
Reading cycle time statistics (Online & Diagnostics)	Yes	Yes	Yes	No
Reading information from the hardware configuration (Online & Diagnostics)	Yes	Yes	Yes	No
Reading time-of-day	Yes	Yes	Yes	No
Executing online functions within the hardware configuration (Online & Diagnostics)	Yes	Yes	Yes	No
Acknowledging alarms	Yes	Yes	Yes	No
Receiving alarms	Yes	Yes	Yes	No
Enabling/disabling alarms	Yes	Yes	No	No
Reading tags via test function (STEP 7, tag table or watch table)	Yes	Yes	No	No

Requesting operating state change online (RUN/STOP/ warm restart)	Yes	Yes	No	No
Downloading data blocks, code blocks, hardware configuration to PG/PC	Yes	Yes	No	No
Set time-of-day	Yes	Yes	No	No
Deleting data blocks, code blocks, hardware configuration in the CPU	Yes	No	No	No
Downloading individual data blocks, code blocks, hardware configurations to the CPU	Yes	No	No	No
Loading PLC program to the device and resetting	Yes	No	No	No
Firmware update of CPUs or I/O modules	Yes	No	No	No
Modifying tags via test function (STEP 7, watch table)	Yes	No	No	No
Reading tags in the program status	Yes	No	No	No
Online editing of blocks	Yes	No	No	No
Modifying outputs in STOP mode	Yes	No	No	No

Behavior of a password-protected module during operation

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

Before an online function is executed, the necessary permission is checked and, if necessary, the user is prompted to enter a password.

Example: The module was configured with read access and you want to execute the "Modify tags" function. This requires write access to a test function; therefore, the assigned password must be entered to execute the function.

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. If the online connection is restored after an interruption, you do not have to enter the access data again. To manually cancel access authorization, click "Online > Delete access rights".

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

Access to a password-protected S7-1500 CPU can be restricted locally on the display. The restriction is only in effect when the operating mode switch is set to RUN.

See also

Security levels (Page 297)

Access password for the HMI connection (Basic Panels)

Introduction

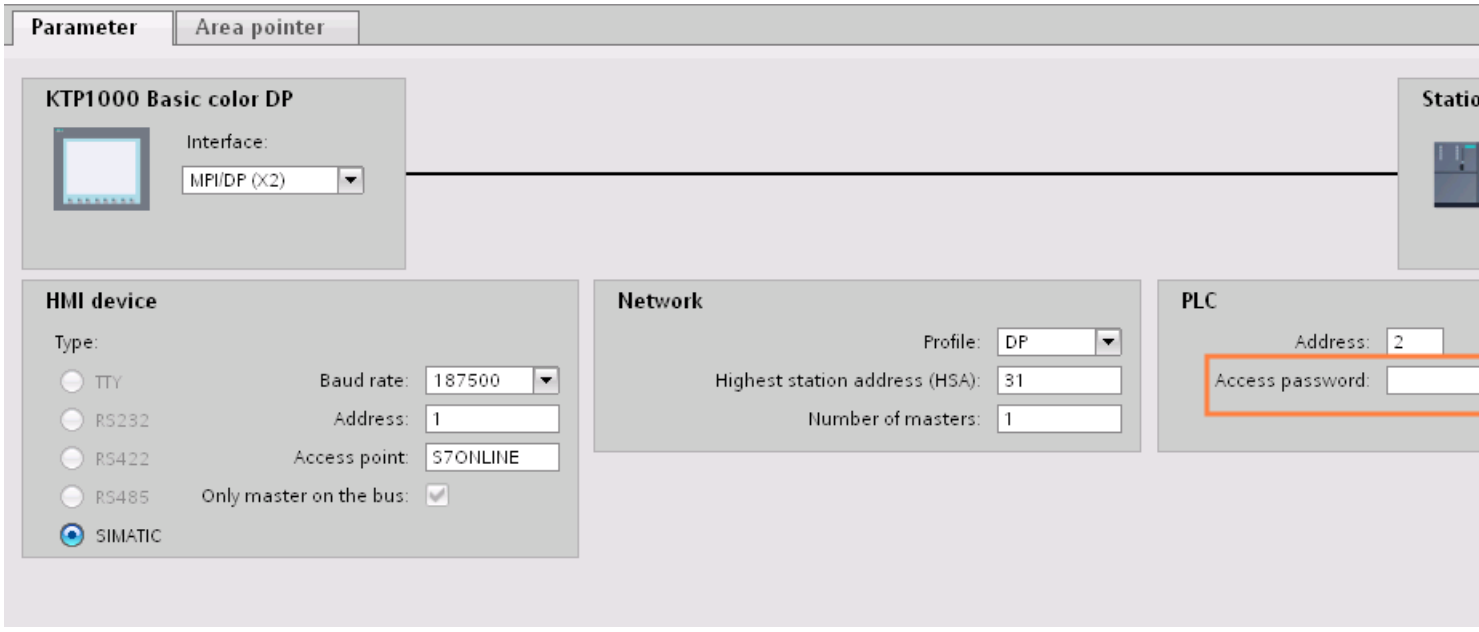
You secure access to a PLC by assigning a password.

You assign the password when you configure the password.

A password is mandatory as of protection level "Complete protection". Communication cannot be established with the PLC if you enter no password or an incorrect password.

Assigning password

You enter the password for the PLC in the "Connections" editor.



See also

Security levels (Page 297)

1.9.4 Data exchange (Basic Panels)

1.9.4.1 Data exchange using area pointers (Basic Panels)

General information on area pointers (Basic Panels)

Introduction

You use an area pointer to access a data area in the PLC. During communication, the PLC and the HMI device alternately access these data areas for read and write operations.

The PLC and the HMI device trigger defined interactions based on the evaluation of stored data.

Configuration of area pointers

Before you use an area pointer, you enable it under "Connections > Area pointer". You then assign the area pointer parameters.

You can find more detailed information on configuring area pointers in:

Configuration of area pointers (Page 66)

Restrictions

You can only configure the following data types for communication with SIMATIC S7 1500 for data exchange using area pointers:

- UInt and array of UInt
- Word and array of Word
- Int and array of Int
- "Array[0..15] of Bool" for area pointer "Coordination"
- Date_And_Time
- DTL and LDT

See also

Configuration of area pointers (Page 66)

"Screen number" area pointer (Page 304)

"Date/time" area pointer (Page 305)

"Date/time PLC" area pointer (Page 306)

"Coordination" area pointer (Page 308)

"Project ID" area pointer (Page 309)

"Job mailbox" area pointer (Page 311)

"Data record" area pointer (Page 313)

"Screen number" area pointer (Basic Panels)

Function

The HMI devices store information about the screen called up on the HMI device in the "Screen number" area pointer.

This allows the transfer of the current screen contents from the HMI device to the PLC. The PLC can trigger specific reactions such as the call of a different screen.

Use

Configure and enable the area pointer in "Communication > Connections" before you put it into use. You can create only **one** instance of the "Screen number" area pointer and only on **one** PLC.

The screen number is always transferred to the PLC when a new screen is activated or when the focus within a screen changes from one screen object to another.

Structure

The area pointer is a data area in the memory of the PLC with a fixed length of 5 words.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st word	Current screen type															
2nd word	Current screen number															
3rd word	Reserved															
4th word	Current field number															
5th word	Reserved															

- Current screen type
  "1" for root screen or
  "4" for permanent area
- Current screen number
  1 to 32767
- Current field number
  1 to 32767

See also

General information on area pointers (Page 303)

"Date/time" area pointer (Page 305)

"Date/time PLC" area pointer (Page 306)

"Coordination" area pointer (Page 308)

"Project ID" area pointer (Page 309)

"Job mailbox" area pointer (Page 311)

"Date/time" area pointer (Basic Panels)

Function

This area pointer is used to transfer the date and time from the HMI device to the PLC.

The PLC writes control job "41" or "40" to the job mailbox.

When it evaluates the control job, the HMI device writes its current date and the time in the data area configured in the "Date/time" area pointer. All definitions are coded in BCD format.

Note

You cannot use the "Date/Time PLC" area pointer if you have configured the "Date/Time" area pointer.

The "Date/Time" area pointer when used in a project which contains multiple connections must be enabled for each configured connection.

"DTL" data type

If you configure the "Date/Time" area pointer, use the DTL data type.

A tag of the "DTL" data type has a length of 12 bytes and saves information on date and time in a predefined structure.

The "DTL" data type has the following structure:

Byte	Component	Data type	Value range
0	Year	UINT	1970 to 2554
1			
2	Month	USINT	0 to 12
3	Day	USINT	1 to 31
4	Day of the week	USINT	1(Sunday) to 7(Saturday) The day of the week is not considered in the value entry.
5	Hour	USINT	0 to 23
6	Minute	USINT	0 to 59
7	Second	USINT	0 to 59
8	Nanoseconds	UDINT	0 to 999 999 999
9			
10			
11			

The "DTL" data type supports time information down to the nanosecond range. Because panels only support time information down to the millisecond, you may encounter the following restriction when using the area pointers:

For the transmission of time information from a panel to the controller, the smallest unit of time is 1 millisecond. The value range from microseconds to nanoseconds of the "DTL" data type is filled with zeros.

Using data types

The data types "Date_And_Time, DTL" and "LDT" can only be used with the "Date/time" and "Date/time PLC" area pointers.

The data format of the "Date/time" area pointer depends on job mailbox 40/41.

If there are no control tags linked to the area pointer, or a control tag is linked with the data type "Array[0..5] of UInt/Word/Int", the following applies:

The configuration of the "Date/time" area pointer is only used for job mailbox 41.

If job mailbox 40 is used, the data format "DATE_AND_TIME (BCD-encoded)" is used (shown in the next section).

If the "Date/time" and "Date/time PLC" area pointers are linked to a control tag with the data type "DATE_AND_TIME", "DTL" or "LDT", the associated data format is used in the corresponding area pointer.

See also

General information on area pointers (Page 303)

"Screen number" area pointer (Page 304)

"Date/time PLC" area pointer (Page 306)

"Job mailbox" area pointer (Page 311)

"Project ID" area pointer (Page 309)

"Coordination" area pointer (Page 308)

"Date/time PLC" area pointer (Basic Panels)

Function

This area pointer is used to transfer the date and time from the PLC to the HMI device. Use this area pointer if the PLC is the time master.

The PLC loads the data area of the area pointer. All definitions are coded in BCD format.

The HMI device reads the data cyclically within the configured acquisition cycle and synchronizes itself.

Note

Set an acquisition cycle of sufficient length for the Date/time area pointer in order to avoid any negative impact on HMI device performance.
Recommended: Acquisition cycle of 1 minute, if the process allows this.

"Date/Time PLC" is a global area pointer and may be configured only once per project.

Note

You cannot use the "Date/Time" area pointer if you have configured the "Date/Time PLC" area pointer.

"DTL" data type

If you configure the "Date/Time" area pointer, use the DTL data type.

A tag of the "DTL" data type has a length of 12 bytes and saves information on date and time in a predefined structure.

The "DTL" data type has the following structure:

Byte	Component	Data type	Value range
0	Year	UINT	1970 to 2554
1			
2	Month	USINT	0 to 12
3	Day	USINT	1 to 31
4	Day of the week	USINT	1(Sunday) to 7(Saturday) The day of the week is not considered in the value entry.
5	Hour	USINT	0 to 23
6	Minute	USINT	0 to 59
7	Second	USINT	0 to 59
8	Nanoseconds	UDINT	0 to 999 999 999
9			
10			
11			

The "DTL" data type supports time information down to the nanosecond range. Because panels only support time information down to the millisecond, you may encounter the following restriction when using the area pointers:

For the transmission of time information from a panel to the controller, the smallest unit of time is 1 millisecond. The value range from microseconds to nanoseconds of the "DTL" data type is filled with zeros.

Using data types

The data types "Date_And_Time, DTL" and "LDT" can only be used with the "Date/time" and "Date/time PLC" area pointers.

The data format of the "Date/time" area pointer depends on job mailbox 40/41.

If there are no control tags linked to the area pointer, or a control tag is linked with the data type "Array[0..5] of UInt/Word/Int", the following applies:

The configuration of the "Date/time" area pointer is only used for job mailbox 41.

If job mailbox 40 is used, the data format "DATE_AND_TIME (BCD-encoded)" is used (shown in the next section).

If the "Date/time" and "Date/time PLC" area pointers are linked to a control tag with the data type "DATE_AND_TIME", "DTL" or "LDT", the associated data format is used in the corresponding area pointer.

See also

General information on area pointers (Page 303)

"Screen number" area pointer (Page 304)

"Date/time" area pointer (Page 305)

"Coordination" area pointer (Page 308)

"Project ID" area pointer (Page 309)

"Job mailbox" area pointer (Page 311)

"Coordination" area pointer (Basic Panels)

Function

The "Coordination" area pointer is used to implement the following functions:

- Detecting the startup of the HMI device in the control program
- Detecting the current operating mode of the HMI device in the control program
- Detecting whether the HMI device is ready to communicate in the control program

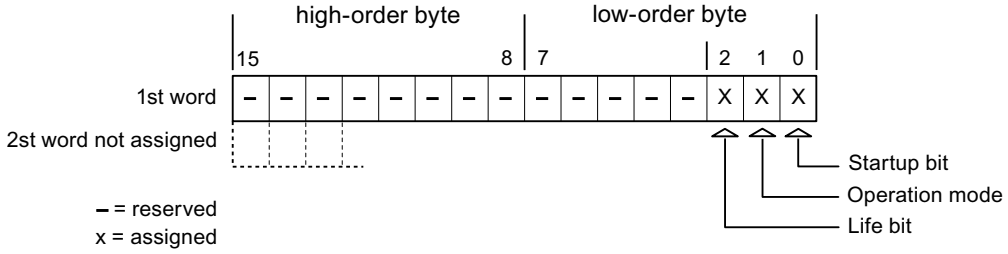
By default, the "Coordination" area pointer has the length of one word and cannot be changed.

Usage

Note

The HMI device always writes the entire coordination area when updating the area pointer. The control program can for this reason not make changes to the coordination area.

Assignment of the bits in the "Coordination" area pointer



Startup bit

The startup bit is set briefly to "0" by the HMI device during startup. It sets the bit permanently to "1" when startup is completed.

Operating mode

The operating mode bit is set to 1 as soon as the user switches the HMI device offline. The status of the operating mode bit is "0" during normal operation of the HMI device. You can determine the current operating mode of the HMI device by reading this bit in the control program.

Life bit

The HMI device inverts the life bit at intervals of approximately one second. You can check whether or not there is still a connection to the HMI device by querying this bit in the control program.

See also

General information on area pointers (Page 303)

"Screen number" area pointer (Page 304)

"Date/time PLC" area pointer (Page 306)

"Project ID" area pointer (Page 309)

"Job mailbox" area pointer (Page 311)

"Date/time" area pointer (Page 305)

"Project ID" area pointer (Basic Panels)

Function

When Runtime starts, a check can be carried out as to whether the HMI device is connected to the correct PLC. This check is important when operating with several HMI devices.

For this, the HMI device compares a value stored on the PLC with the value specified in configuration. This ensures compatibility of the configuration data with the PLC program. If there is no concordance, a system event is given on the HMI device and Runtime is stopped.

Use

Note

HMI connections cannot be switched "online".

The HMI connection in which the "Project ID" area pointer is used must be switched "online".

To use this area pointer, set up the following during the configuration:

- Define the version of the configuration. Values between 1 and 255 are possible.
  You enter the version in the editor "Runtime settings > General" in the "Identification" area.
- Data address of the value for the version that is stored in the PLC:
  You enter the data address in the editor "Communication > Connections" under "Address".

Connection failure

A connection failure to a device on which the "project ID" area pointer is configured results in all the other connections of the device being switched to "offline".

This behavior has the following requirements:

- You have configured several connections in a project.
- You are using the "project ID" area pointer in at least one connection.

Causes which may set connections "offline":

- The PLC is not available.
- The connection has been switched offline in the engineering system.

See also

General information on area pointers (Page 303)

"Screen number" area pointer (Page 304)

"Date/time PLC" area pointer (Page 306)

"Coordination" area pointer (Page 308)

"Job mailbox" area pointer (Page 311)

"Date/time" area pointer (Page 305)

"Job mailbox" area pointer (Basic Panels)

Function

The PLC can use the job mailbox to transfer jobs to the HMI device to trigger corresponding actions on the HMI device. These functions include, for example:

- Display screen
- Set date and time

Data structure

The first word of the job mailbox contains the job number. Depending on the job mailbox, up to three parameters can be transferred.

Word	Most significant byte	Least significant byte
n+0	0	Job number
n+1	Parameter 1	
n+2	Parameter 2	
n+3	Parameter 3	

The HMI device evaluates the job mailbox if the first word of this job is not equal to zero. This means that the parameters must be entered in the job mailbox first, followed by the job number.

When the HMI device accepts the job mailbox, the first word is set to 0 again. The execution of the job mailbox is generally not completed at this point in time.

Job mailboxes

All job mailboxes and their parameters are listed below. The "No." column contains the job number of the job mailbox. Job mailboxes can only be triggered by the PLC when the HMI device is online.

No.	Function	
14	**Set time (BCD-coded)**	
	Parameter 1	Left byte: - Right byte: hours (0-23)
	Parameter 2	Left byte: minutes (0-59) Right byte: seconds (0-59)
	Parameter 3	-
15	**Setting the date (BCD coded)** [3) 4)]	
	Parameter 1	Left byte: - Right byte: weekday (1-7: Sunday-Saturday)
	Parameter 2	Left byte: day (1-31) Right byte: month (1-12)
	Parameter 3	Left byte: year
23	**User logon**	

No.	Function	
14	**Set time (BCD-coded)**	
	Logs the user on with the name "PLC user" at the HMI device with the group number transferred in Parameter 1. The logon is possible only when the transferred group number exists in the project.	
	Parameter 1	Group number 1 to 255
	Parameter 2, 3	-
24	**User logoff**	
	Logs off the current user. (The function corresponds to the "logoff" system function)	
	Parameter 1, 2, 3	-
40	**Transfer date/time to PLC**	
	(in the S7 format DATE_AND_TIME) An interval of at least 5 seconds must be maintained between two successive jobs to prevent overload of the HMI device.	
	Parameter 1, 2, 3	-
41	**Transfer date/time to PLC**	
	(In OP/MP format) An interval of at least 5 seconds must be maintained between successive jobs to prevent overload of the HMI device.	
	Parameter 1, 2, 3	-
46	**Update tag**	
	Causes the HMI device to read the current value of the tags from the PLC whose update ID matches the value transferred in Parameter 1. (Function corresponds to the "UpdateTag" system function.)	
	Parameter 1	1 - 100
49	**Delete alarm buffer**	
	Deletes all analog alarms and discrete alarms of the "Warnings" class from the alarm buffer.	
	Parameter 1, 2, 3	-
50	**Delete alarm buffer**	
	Deletes all analog alarms and discrete alarms of the "Errors" class from the alarm buffer.	
	Parameter 1, 2, 3	-
51	**Screen selection** [2]	
	Parameter 1	Screen number
	Parameter 2	-
	Parameter 3	Field number
69	**Reading data record from PLC** [1]	
	Parameter 1	Recipe number (1-999)
	Parameter 2	Data record number (1-65535)
	Parameter 3	0: Do not overwrite existing data record 1: Overwrite existing data record
70	**Writing data record to PLC** [1]	
	Parameter 1	Recipe number (1-999)
	Parameter 2	Data record number (1-65535)
	Parameter 3	-

1)	Only for devices supporting recipes.
2)	OP 73, OP 77A and TP 177A HMI devices also execute the "Screen selection" job mailbox if the on-screen keyboard is active.
3)	The weekday is ignored on HMI device KTP 600 BASIC PN.
4)	The weekday is ignored when you configure the "Date/Time PLC" area pointer.

See also

General information on area pointers (Page 303)

"Screen number" area pointer (Page 304)

"Date/time PLC" area pointer (Page 306)

"Coordination" area pointer (Page 308)

"Project ID" area pointer (Page 309)

"Date/time" area pointer (Page 305)

"Data record" area pointer (Basic Panels)

"Data record" area pointer (Basic Panels)

Function

When data records are transferred between the HMI device and PLC, both partners access common communications areas on the PLC.

Data transfer types

There are two ways of transferring data records between the HMI device and PLC:

- Transfer without synchronization
- Transfer with synchronization via the data mailbox

Data records are always transferred directly, which means that the tag values are read straight from an address or written straight to an address configured for this tag without being redirected via an interim memory.

Initiating the transfer of data records

There are three ways of triggering the transfer:

- Operator input in the recipe view
- Job mailboxes
  The transfer of data records can also be triggered by the PLC.
- Triggering by configured functions

If the transfer of data records is triggered by a configured function or by a job mailbox, the recipe view on the HMI device remains operable. The data records are transferred in the background.

Simultaneous processing of several transfer requests is, however, not possible. In this case, the HMI device rejects the other transfer requests with a system alarm.

See also

General information on area pointers (Page 303)

Transfer without synchronization (Page 314)

Transfer with synchronization (Page 315)

Transfer started by operator in the recipe view (Page 316)

Transfer with job mailbox (Page 317)

Sequence of transfer triggered by a configured function (Page 319)

Possible causes of error when transferring data records (Page 320)

Transfer without synchronization (Basic Panels)

If you select asynchronous transfer of data records between the HMI device and PLC, there is no coordination over the common data areas. It is therefore unnecessary to set up a data area during configuration.

Asynchronous data record transfer can be a useful alternative, for example when:

- The system is capable of excluding the risk of uncontrolled overwriting of data by the communication peer.
- The PLC does not require information about the recipe number and data record number.
- The transfer of data records is triggered by the operator of the HMI device.

Reading values

When a read job is triggered, the values are read from the PLC addresses and transferred to the HMI device.

- Triggering by the operator in the recipe view:
  The values are downloaded to the HMI device. You can then, for example, process, edit, or save these values in the HMI device.
- Triggering by a function or job mailbox:
  The values are saved immediately to the data volume.

Writing values

When a write job is triggered, the values are written to the PLC addresses.

- Triggering by the operator in the recipe view:
  The current values are written to the PLC.
- Triggering by a function or job mailbox:
  The current values are written to the PLC from the data medium.

See also

"Data record" area pointer (Page 313)

Transfer with synchronization (Basic Panels)

If you select synchronous transfer, both communication partners set status bits in the common data area. You can use this mechanism to prevent uncontrolled overwriting of data in either direction in your control program.

Application

Synchronous data record transfer can be a useful solution, for example, when:

- The PLC is the "active partner" in the transfer of data records.
- The PLC evaluates the information about the recipe number and data record number.
- The transfer of data records is triggered by means of a Job mailbox.

Requirements

In order to synchronize transfer of data records between the HMI device and the PLC, the following requirements must be met during configuration:

- An area pointer has been set up: "Communication > Connections" editor in "Area pointer".
- The PLC with which the HMI device synchronizes transfer of data records is specified in the recipe:
  "Recipes" editor in the inspector window the option "Coordinated transfer of data records" under "General > Synchronization > Settings"

Structure of the data area

The data area has a fixed length of 5 words. Structure of the data area:

	15		**0**
1. Word	Current recipe number (1 - 999)		
2. Word	Current data record number (0 - 65535)		
3. Word	Reserved		
4. Word	Status (0, 2, 4, 12)		
5. Word	Reserved		

- Status
  The status word (word 4) can adopt the following values:

Value		Meaning
Decimal	Binary	
0	0000 0000	Transfer permitted, data mailbox free
2	0000 0010	Transferring
4	0000 0100	Transfer completed without error
12	0000 1100	Transfer completed with error

See also

"Data record" area pointer (Page 313)

Transfer started by operator in the recipe view (Basic Panels)

Reading from the PLC started by the operator in the recipe view

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe number to be read and the status "Transferring" in the data record and sets the data record number to 0.	Abort with system event.
3	The HMI device reads the values from the PLC and displays them in the recipe view. If the recipes have synchronized tags, the values from the PLC are also written to the tags.	
4	The HMI device sets the status "Transfer completed."	
5	The control program must reset the status word to zero in order to enable further transfers.	

Writing to the PLC started by the operator in the recipe view

Step	Action	
	Check: Status word = 0?	
1	Yes	No
	The HMI device enters the recipe and data record number to be written and the status "Transferring" in the data record.	Abort with system event.
2	The HMI device writes the current values to the PLC. If the recipes have synchronized tags, the changed values are synchronized between the recipe view and tags and then written to the PLC.	

Step	Action	
3	The HMI device sets the status "Transfer completed."	
4	If required, the control program can now evaluate the transferred data.	
5	The control program must reset the status word to zero in order to enable further transfers.	

Note

The status word may only be set by the HMI device. The PLC may only reset the status word to zero.

Note

The PLC may only evaluate the recipe and data record numbers when data inconsistency is detected if one of the conditions outlined below has been met:

- The data mailbox status is set to "Transfer completed".
- The data mailbox status is set to "Transfer completed with error".

See also

"Data record" area pointer (Page 313)

Transfer with job mailbox (Basic Panels)

The transfer of data records between the HMI device and the PLC can be initiated by either one of these stations.

The two job mailboxs No. 69 and No. 70 are available for this type of transfer.

No. 69: Read data record from PLC ("PLC → DAT")

Job mailbox no. 69 transfers data records from the PLC to the HMI device. The job mailbox is structured as follows:

	Left byte (LB)	Right byte (RB)
Word 1	0	69
Word 2	Recipe number (1-999)	
Word 3	Data record number (1 to 65535)	
Word 4	Do not overwrite existing data record: 0 Overwrite existing data record: 1	

No. 70: Write data record to PLC ("DAT → PLC")

Job mailbox No. 70 transfers data records from the HMI device to the PLC. The job mailbox is structured as follows:

	Left byte (LB)	Right byte (RB)
Word 1	0	70
Word 2	Recipe number (1-999)	
Word 3	Data record number (1-65,535)	
Word 4	—	

Sequence when reading from the PLC with job mailbox "PLC → DAT" (no. 69)

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the job and the status "Transferring" in the data record.	Abort without return message.
3	The HMI device reads the values from the PLC and saves these to the data record defined in the job mailbox.	
4	• If "Overwrite" was selected in the job, an existing data record is overwritten without any prompt for confirmation. The HMI device sets the status "Transfer completed." • If "Do not overwrite" was selected in the job, and the data record already exists, the HMI device aborts the job and enters 0000 1100 in the status word of the data mailbox.	
5	The control program must reset the status word to zero in order to enable further transfers.	

Sequence of writing to the PLC with job mailbox "DAT → PLC" (no. 70)

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the job and the status "Transferring" in the data mailbox.	Abort without return message.
3	The HMI device fetches the values of the data record specified in the function from the data medium and writes the values to the PLC.	
4	The HMI device sets the status "Transfer completed."	
5	The control program can now evaluate the transferred data. The control program must reset the status word to zero in order to enable further transfers.	

See also

"Data record" area pointer (Page 313)

Sequence of transfer triggered by a configured function (Basic Panels)

Reading from the PLC using a configured function

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the function and the status "Transferring" in the data mailbox.	Abort with system event.
3	The HMI device reads the values from the PLC and stores them in the data record specified in the function.	
4	• If "Yes" was selected for the "Overwrite" function, an existing data record is overwritten without any prompt for confirmation. The HMI device sets the status "Transfer completed." • If "No" was selected for the "Overwrite" function and the data record already exists, the HMI device aborts the job and enters 0000 1100 in the status word of the data mailbox.	
5	The control program must reset the status word to zero in order to enable further transfers.	

Writing to the PLC by means of configured function

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the function and the status "Transferring" in the data mailbox.	Abort with system event.
3	The HMI device fetches the values of the data record specified in the function from the data medium and transfers the values to the PLC.	
4	The HMI device sets the status "Transfer completed."	
5	The control program can now evaluate the transferred data. The control program must reset the status word to zero in order to enable further transfers.	

See also

"Data record" area pointer (Page 313)

Possible causes of error when transferring data records (Basic Panels)

Possible causes of error

The section below shows possible error causes which lead to the cancellation of data record transfer:

- Tag address not set up on the PLC
- Overwriting data records not possible
- Recipe number does not exist
- Data record number does not exist

Note

The status word may only be set by the HMI device. The PLC may only reset the status word to zero.

Note

The PLC may only evaluate the recipe and data record numbers when data inconsistency is detected if one of the conditions outlined below has been met:

- The data mailbox status is set to "Transfer completed".
- The data mailbox status is set to "Transfer completed with error".

Reaction to an aborted transfer due to errors

If the transfer of data records is aborted due to errors, the HMI device reacts as follows:

- Triggering by the operator in the recipe view
  Information in the status bar of the recipe view and output of system alarms
- Triggered by function
  Output of system alarms
- Triggering by job mailbox
  No return message on the HMI device

You can nonetheless evaluate the status of the transfer by querying the status word in the data record.

See also

"Data record" area pointer (Page 313)

1.9.4.2 Trends (Basic Panels)

Trends

A trend is the graphical representation of one or more values from the PLC. The value is read out time-triggered for Basic Panels.

For additional information see:

AUTOHOTSPOT

Time-triggered trends

The HMI device reads in the trend values cyclically at an interval specified in the configuration.

Time-triggered trends are suitable for continuous curves, such as the operating temperature of a motor.

1.9.4.3 Alarms (Basic Panels)

Configuring alarms (Basic Panels)

Configure alarms

Several steps are needed to configure alarms, such as operational messages, error alarms, and acknowledgement.

- Step 1: Create tags
- Step 2: Configure alarms
- Step 3: Configure acknowledgment

You can find additional information in the section:

AUTOHOTSPOT

Distinctive features when configuring alarms

If you are configuring connections of HMI devices to PLCs of other manufacturers, note the following distinctive features when configuring:

- Data types of the tags
- Addressing of tags
- How the bit positions are counted

Data types

For connections with a SIMATIC communication driver, the following data types are supported:

PLC	Permitted data types	
	Discrete alarms	Analog alarms
SIMATIC S7 PLCs	WORD, INT	BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, TIMER

How the bit positions are counted

For connections with a SIMATIC communication driver, the following counting method applies:

How the bit positions are counted	Byte 0								Byte 1							
	Most significant byte								Least significant byte							
In SIMATIC S7 PLCs	7							0	7							0
In WinCC you configure:	15							8	7							0

See also

Acknowledgment of alarms (Page 322)

Communication with SIMATIC S7-1500 Software Controller (Page 275)

Acknowledgment of alarms (Basic Panels)

Procedure

Create suitable tags on the PLC to acknowledge an error alarm. You assign these tags to an alarm in the "Bit messages" editor. You make the assignment in "Properties > Acknowledgment".

Distinction in terms of acknowledgment:

- Acknowledgment by the PLC
- Acknowledgment on the HMI device

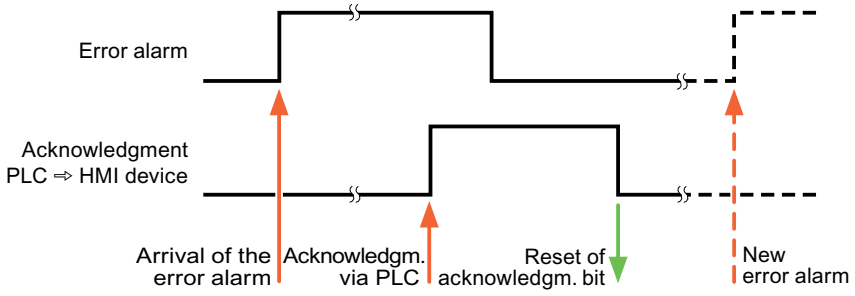
Acknowledgment by the PLC

In "PLC acknowledgment tag", configure the tag or array tag and the bit number that the HMI device uses to identify a PLC acknowledgment.

A bit set in the tag triggers acknowledgment of the assigned error alarm bit at the HMI device. This tag bit returns a function similar to acknowledgment on the HMI device which is triggered by pressing the "ACK" button, for example.

The acknowledgment bit must be located in the same tag as the bit for the error alarm.

Reset the acknowledgment bit before setting the bit in the alarm area again. The figure below shows the pulse diagram.



Acknowledgment on the HMI device

In the "HMI acknowledgment tag" area, configure the tag or array tag as well as the bit number that the HMI device writes to the PLC after acknowledgment. Make sure when you use an array tag that it is not longer than 6 words.

To always create a signal change when setting an assigned acknowledgment bit of a discrete alarm that must be acknowledged, the HMI device will reset the acknowledgment bit assigned to the alarm as soon as it detects an alarm subject to acknowledgment and write the acknowledgment tag in the PLC. There will be a certain delay between detecting the message and writing the acknowledgment tag in the PLC because the HMI device has to process the operations.

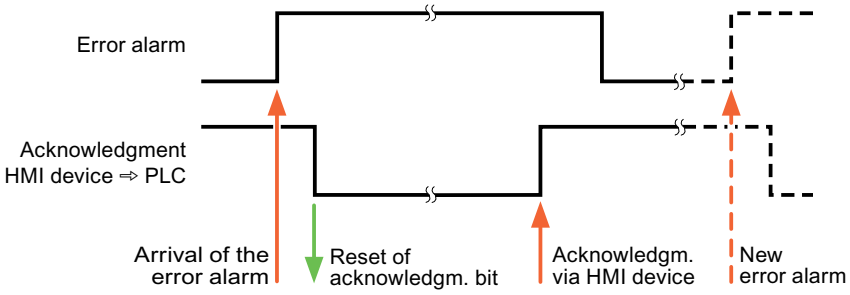
If a discrete alarm subject to acknowledgment is acknowledged by the HMI device, then the corresponding bit in the assigned acknowledgment tag will be set. The entire acknowledgment tag is then written to the PLC by the HMI device. This allows the PLC to recognize that a certain alarm message has been acknowledged at the HMI device.

Note

All alarm bits acknowledged since the last Runtime start will remain in the acknowledgment tag until a new incoming of the respective discrete alarms is detected.

This area should only be read by the PLC because the entire section of the HMI device will be overwritten once the next acknowledgment tag is written.

The figure below shows the pulse diagram.



See also

Configuring alarms (Page 321)

1.9.4.4 LED mapping (Basic Panels)

Function

Keyboard devices have LEDs in the function keys. These LEDs can be activated from the PLC. Thus, it is possible, for example, to signal to the user which key he should press in a given situation by lighting up an LED.

Note

The LED function cannot be configured for Basic Panels.

Requirements

In order to activate an LED, an LED tag or an array tag must be set up in the PLC and specified as an LED tag during configuration.

LED assignment

The assignment of the individual LEDs to the bits in the LED tags is specified when the function keys are configured. In this process, the "LED tag" and the assigned "Bit" are specified for each function key in the "General" group of the properties window.

The "Bit" bit number designates the first of two consecutive bits that control the following LED statuses.

Bit n+ 1	Bit n	LED function	
		All Mobile Panels, all Operator Panels, all Multi Panels, all Comfort Panels	Panel PCs
0	0	Off	Off
0	1	Rapid flashing	Flashing
1	0	Slow flashing	Flashing
1	1	On permanently	On permanently

1.9.5 Performance features of communication (Basic Panels)

1.9.5.1 Device dependencies of the SIMATIC S7-1500 Software Controller (Basic Panels)

Device dependency

If you use devices from an earlier version of the TIA Portal with TIA Portal V14, it may not be possible to configure connections to certain HMI devices.

Basic Panels V11.0

HMI devices	SIMATIC S7-1500 Software Controller
KP300 Basic	No
KP400 Basic	No
KTP400 Basic PN	No
KTP600 Basic DP	No
KTP600 Basic PN	No
KTP1000 Basic DP	No
KTP1000 Basic PN	No
TP1500 Basic PN	No

Basic Panels V12.0

HMI devices	SIMATIC S7-1500 Software Controller
KP300 Basic	No
KP400 Basic	No
KTP400 Basic PN	No
KTP600 Basic DP	No
KTP600 Basic PN	No
KTP1000 Basic DP	No
KTP1000 Basic PN	No
TP1500 Basic PN	No

Basic Panels V13.0

HMI devices	SIMATIC S7-1500 Software Controller
KTP400 Basic	No
KTP700 Basic	No
KTP900 Basic	No
KTP1200 Basic	No

Basic Panels V13.0.1

HMI devices	SIMATIC S7-1500 Software Controller
KTP400 Basic PN	Yes
KTP700 Basic PN	Yes
KTP700 Basic DP	Yes
KTP900 Basic PN	Yes
KTP1200 Basic PN	Yes
KTP1200 Basic DP	Yes

Basic Panels V14.0

HMI devices	SIMATIC S7-1500 Software Controller
KTP400 Basic PN	Yes
KTP700 Basic PN	Yes
KTP700 Basic DP	Yes
KTP900 Basic PN	Yes
KTP1200 Basic PN	Yes
KTP1200 Basic DP	Yes

1.9.6 Configuring time synchronization (Basic Panels)

1.9.6.1 Time synchronization (Basic Panels)

Introduction

To have the same time of day throughout the plant, you can synchronize the time on various plant components using time synchronization. WinCC time synchronization is operated as a master-slave system.

One system component must be a clock for all components of a plant to work with identical time. The component functioning as the clock is referred to as the time master. The components that receive the time are time slaves.

Properties of time synchronization

- The HMI device can define the time as master or can accept the time of the PLC as slave.
- In "master mode", the time is synchronized at each connection setup.
- In "slave mode", the time is synchronized at each connection setup and than at cyclic intervals of 10 minutes.
- The first time synchronization is performed on the HMI device immediately after the start of runtime.
- Time synchronization is only performed on the HMI device during operation of runtime.

See also

Time synchronization restrictions (Page 327)

Configuring time synchronization for integrated connections (Page 328)

Configuring time synchronization for non-integrated connections (Page 329)

1.9.6.2 Time synchronization restrictions (Basic Panels)

Approved HMI devices

You can configure the time synchronization between a SIMATIC S7-1200 or SIMATIC S7-1500 and an HMI device with the following HMI devices:

Device	Operating system
Basic Panels	-
TP177 4"	Windows CE 5.0
Multi Panel 177	Windows CE 5.0
Multi Panel 277	Windows CE 5.0
Multi Panel 377	Windows CE 5.0
Mobile Panel 277	Windows CE 5.0
Mobile 277 IWLAN V2	Windows CE 5.0
Comfort Panels	Windows CE 6.0
PC systems with WinCC RT Advanced	Microsoft Windows

Configuration restrictions

- If an HMI device has several connections to SIMATIC S7-1200 or SIMATIC S7-1500, you can only configure one connection as "slave".
- If you have enabled time synchronization for the HMI device as "slave", you can no longer use the global area pointer "Date/time PLC".
- An HMI device can only request the time from a PLC with "Complete protection" security type configuration if the correct "Access password" is configured.
  Configure the "Access password" for communication with a PLC with "Complete protection" security type in the "Connections" editor of the HMI device.
  This "Access password" must match the password configured on the PLC. The PLC password is assigned in the PLC properties at: "General > Security"
- Basic Panels can only be configured as "Slave".
- If you use Basic Panels for the configuration, it is not possible to use time synchronization via NTP and the "Date/time PLC" area pointer simultaneously.
- Time synchronization with SIMATIC S7-1200 (V1.0) controllers is not possible.
- Time synchronization between the HMI device TP177 4" and SIMATIC S7-1200 (V4.0) controllers is not possible.
- Time synchronization between the HMI device TP177 4" and SIMATIC S7-1500 is not possible.

See also

Time synchronization (Page 326)

Configuring time synchronization for integrated connections (Page 328)

Configuring time synchronization for non-integrated connections (Page 329)

1.9.6.3 Configuring time synchronization for integrated connections

Introduction

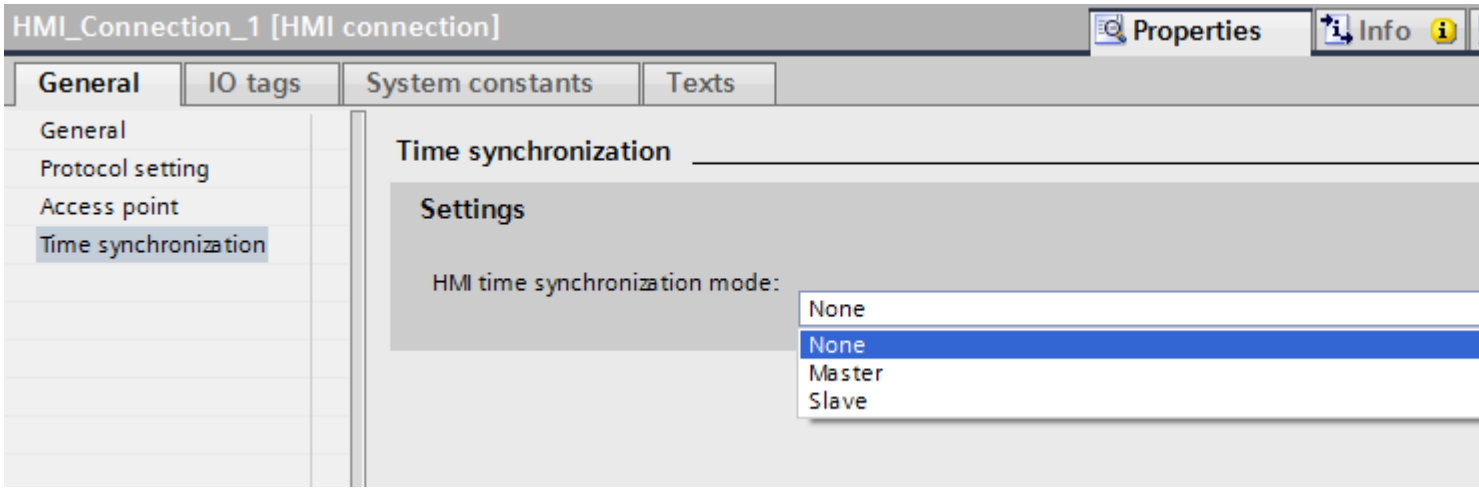
You configure time synchronization for an integrated connection in the "Devices & Networks" editor.

Requirements

- An HMI connection between an HMI device and a SIMATIC S7 1200 or SIMATIC S7 1500 has been configured.
- The HMI device must support the "time synchronization" function.
- The "Devices & Networks" editor is open.

Procedure

1. Click the line of the HMI connection in the "Devices & networks" editor.
2. Select the following in the inspector window under "General > Time synchronization > Settings":
   - None: No time synchronization is used.
   - Master: The HMI device sets the time.
   - Slave: The PLC sets the time.



See also

Time synchronization (Page 326)

Time synchronization restrictions (Page 327)

Configuring time synchronization for non-integrated connections (Page 329)

1.9.6.4 Configuring time synchronization for non-integrated connections (Basic Panels)

Introduction

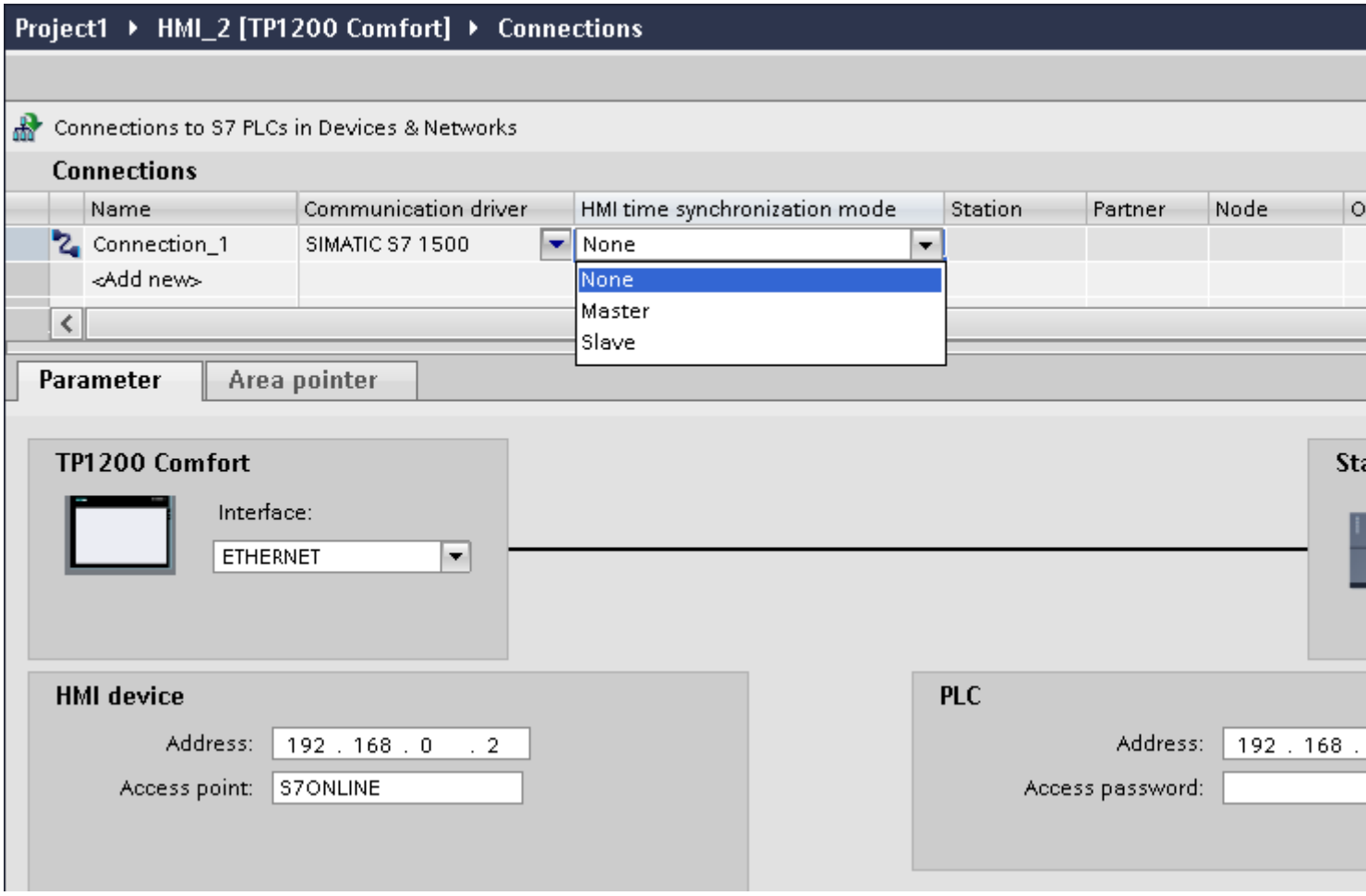
You configure time synchronization for a non-integrated connection in the "Connections" editor.

Requirements

- An HMI device which supports the "time synchronization" function has been created.
- "Connections" editor is open.

Procedure

1. Double-click "<Add>".
2. In the "Communication drivers" column, select the "SIMATIC S7 1500" PLC.
3. Select the following in the "HMI time synchronization mode" column:
   - None: No time synchronization is used.
   - Master: The HMI device sets the time.
   - Slave: The PLC sets the time.



See also

Time synchronization (Page 326)

Time synchronization restrictions (Page 327)

Configuring time synchronization for integrated connections (Page 328)

1.10 Communicating with SIMATIC ET 200 CPU (Basic Panels)

1.10.1 Communication with SIMATIC ET 200 CPU (Basic Panels)

Introduction

This section describes the communication between an HMI device and the SIMATIC ET 200 CPU controller.

You can configure the following communication channels for the SIMATIC ET 200 CPU:

- PROFINET
- PROFIBUS

HMI connection for communication

Configure connections between the HMI device and a SIMATIC ET 200 CPU in the "Devices & Networks" editor. If you have configured a HMI device with a serial port, you must configure a PROFIBUS-capable communication module to the ET 200 CPU.

1.10.2 Communication via PROFINET (Basic Panels)

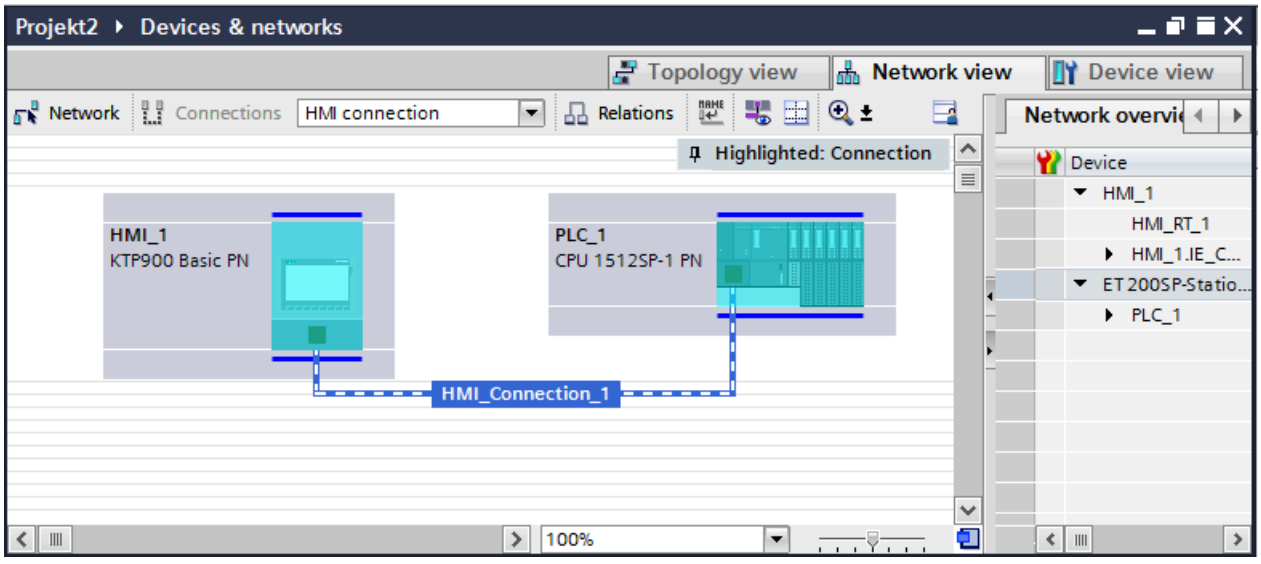
1.10.2.1 Communication via PROFINET (Basic Panels)

HMI connections via PROFINET

If you have inserted an HMI device and a SIMATIC ET 200 CPU into the project, you interconnect the two PROFINET interfaces in the "Devices & Networks" editor.

You can also connect multiple HMI devices to a single SIMATIC ET 200 CPU and multiple SIMATIC ET 200 CPU to a single HMI device.

The maximum number of communication partners that you can connect to an HMI device is dependent on the HMI device used.



Additional information is available in the documentation for the respective HMI device.

HMI connection in the "Devices & Networks" editor

You configure the HMI connection between the PLC and the HMI device via PROFINET in the "Devices & Networks" editor.

Connection in the "Connections" editor

Alternatively, you configure the connection between the PLC and HMI device via PROFINET in the "Connections" editor of the HMI device.

See also

Communication with SIMATIC ET 200 CPU (Page 331)

1.10.2.2 Configuring an HMI connection via PROFINET (Basic Panels)

Introduction

You configure an HMI connection between HMI devices and a SIMATIC ET 200 CPU via PROFINET or Ethernet in the "Devices & Networks" editor.

⚠ 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.

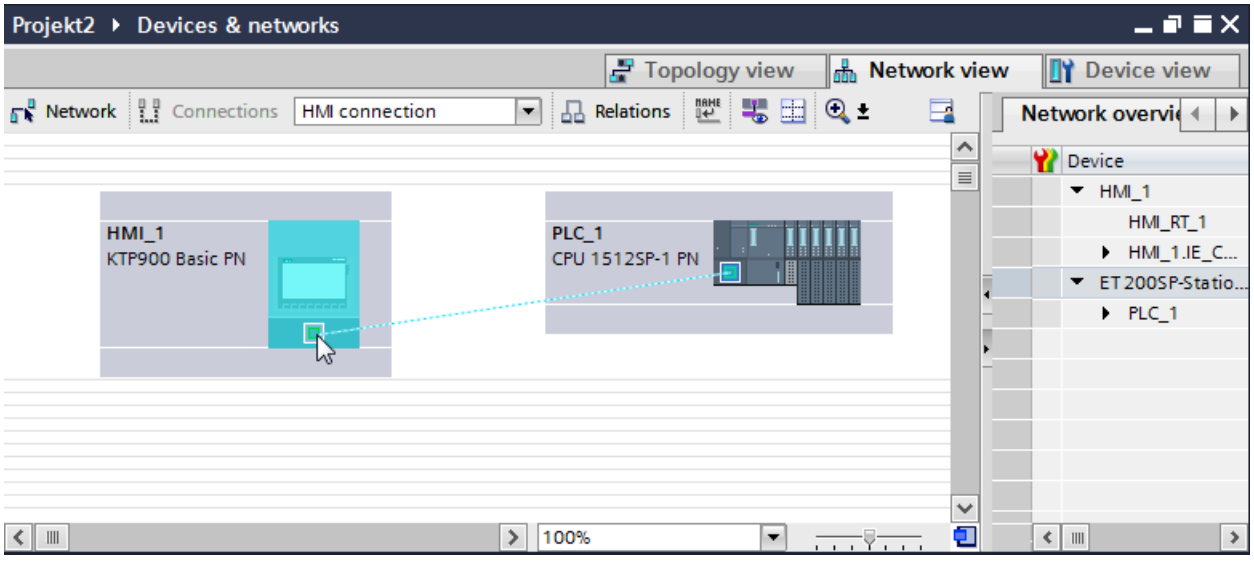
Requirements

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

- SIMATIC ET 200 CPU
- HMI device with PROFINET or Ethernet interface

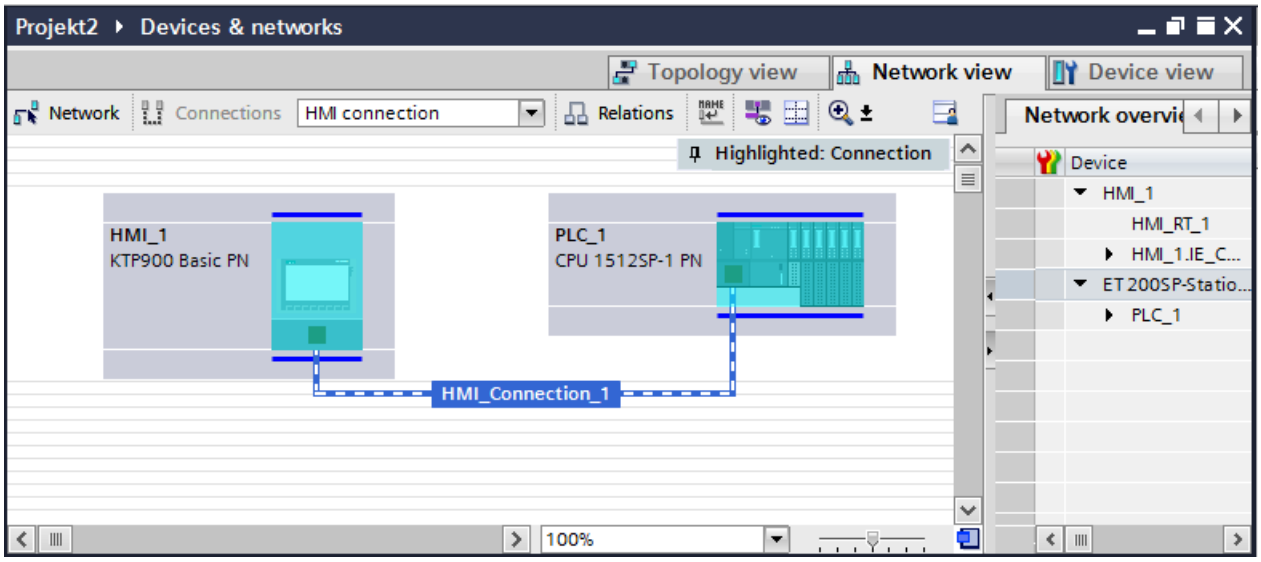
Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed graphically in the network view.
2. Click the "Connections" button and select "HMI connection" for the connection type.
   The devices available for connection are highlighted in color.
3. Click the PROFINET interface of the PLC and use a drag-and-drop operation to draw a connection to the PROFINET or Ethernet interface of the HMI device.



4. Click the connecting line.

5. Click "Highlight HMI connection" and select the HMI connection.



The connection is displayed graphically in the Inspector window.

6. Click the communication partners in the "Network view" and change the PROFINET parameters in the Inspector window according to the requirements of your project. For additional information on this, see section "PROFINET parameters (Page 335)".

Note

The created HMI connection is also shown in the tabular area of the editor on 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 a SIMATIC ET 200 CPU.

The IP address and subnet mask connection parameters are configured.

See also

Communication with SIMATIC ET 200 CPU (Page 331)

PROFINET parameters (Page 335)

1.10.2.3 PROFINET parameters (Basic Panels)

PROFINET parameters for the HMI connection (Basic Panels)

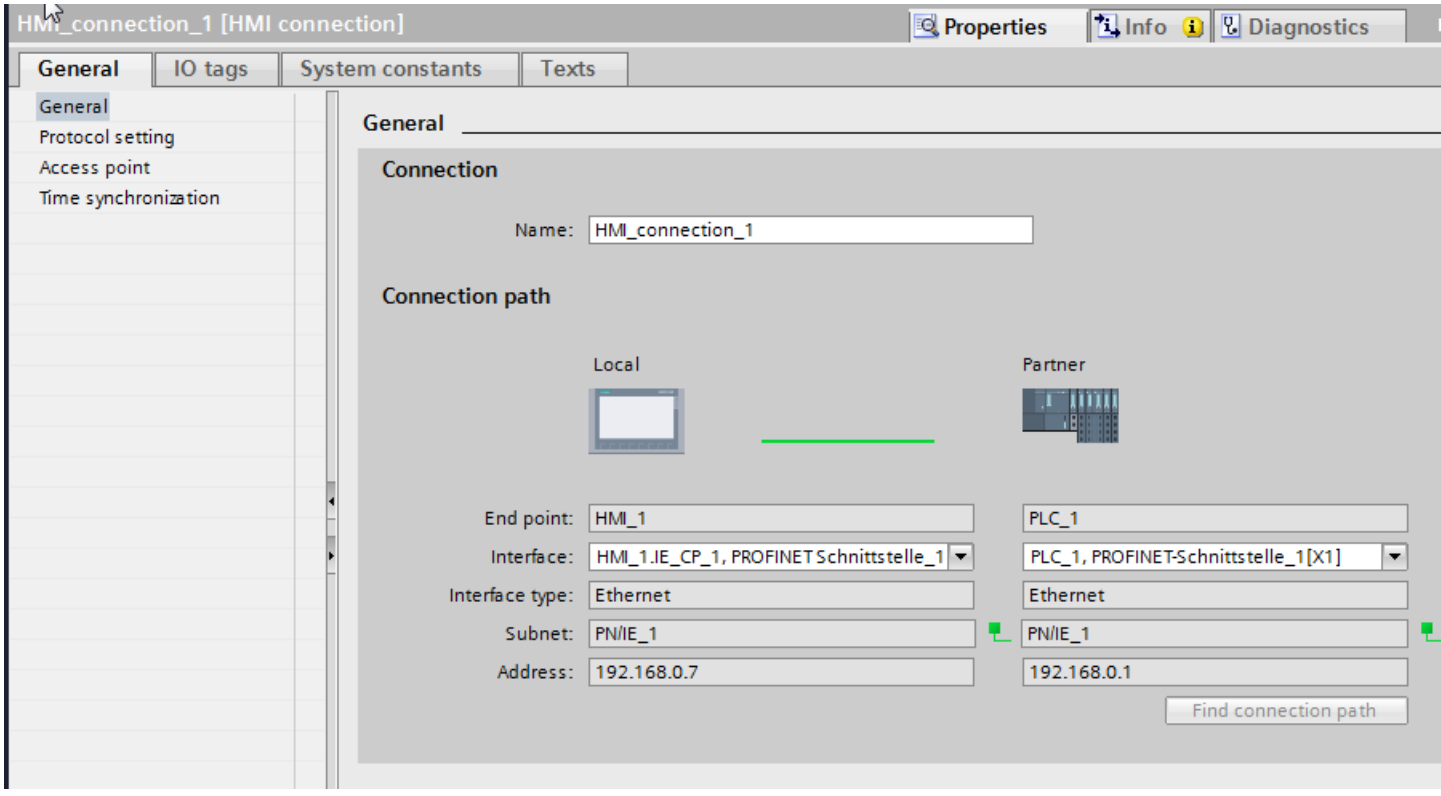
PROFINET parameters for the HMI connection

An overview of the configured HMI connection parameters can be found in the properties for the HMI connection.

Only limited changes are possible in this Inspector window.

Displaying and editing HMI connection parameters

1. Click the HMI connection in the "Devices & Networks" editor.
2. Change the parameters of the HMI connection in the Inspector window under "Properties > General > General".



Connection

The "Connection" area displays the HMI connection created for communication between the devices.

You can edit the name of the HMI connection in this area.

"Connection path"

The communication partners of the selected HMI connection and the associated PROFINET parameters are displayed in the "Connection path" area. Some of the areas displayed cannot be edited in this dialog.

- "End point"
  Displays the device name. This area cannot be edited.
- "Interface"
  Displays the selected interface of the device. You can choose between several interfaces, depending on the device.
- "Interface type"
  Displays the selected interface type. This area cannot be edited.
- "Subnet"
  Displays the selected subnet. This area cannot be edited.
- "Address"
  Displays the selected IP address of the device. This area cannot be edited.
- "Find connection path" button
  Enables the subsequent specification of connections.

See also

Communication via PROFINET (Page 331)

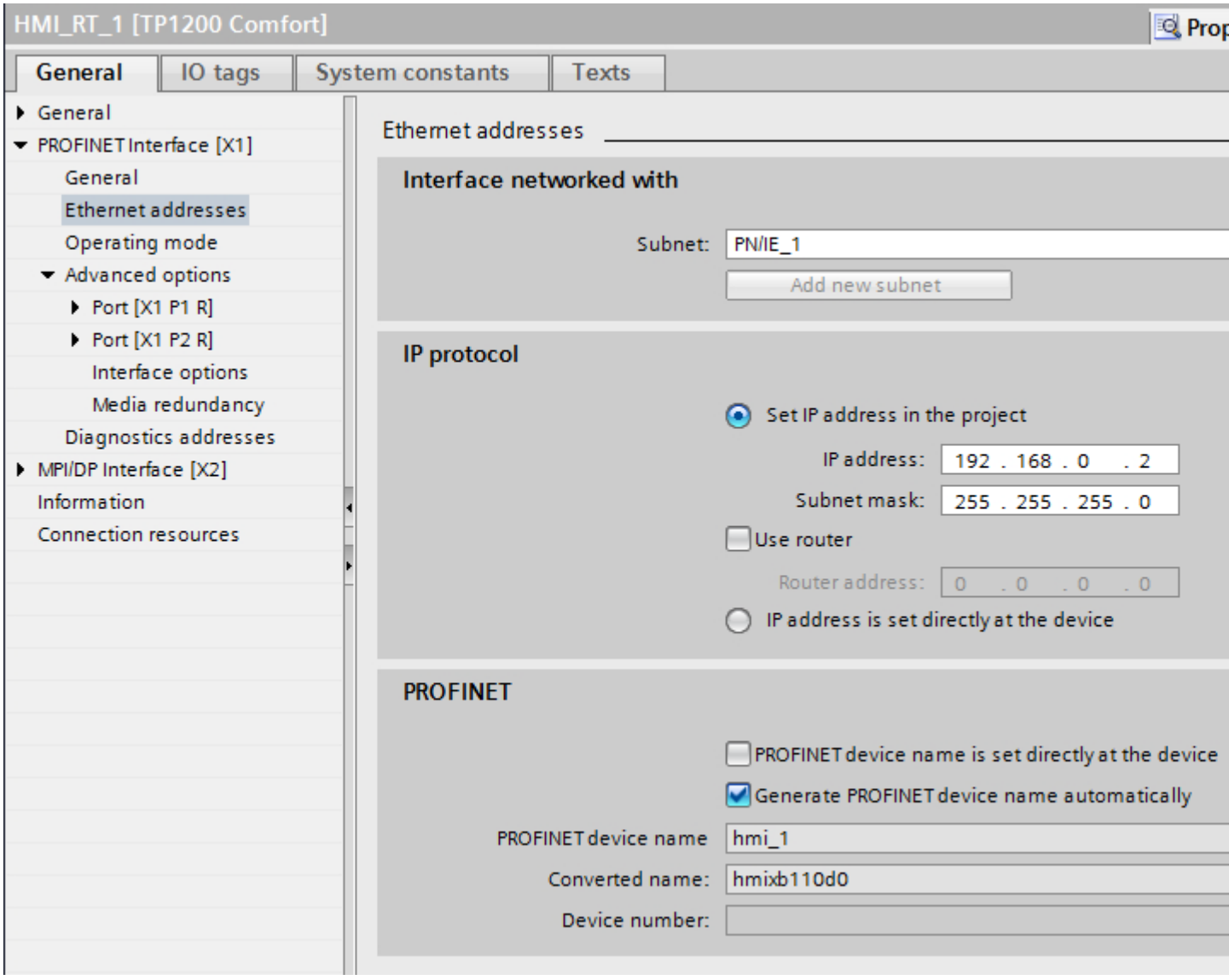
PROFINET parameters for the HMI device (Basic Panels)

PROFINET parameters for the HMI device

An overview of the configured HMI device parameters can be found in the properties for the HMI device.

Displaying and changing PROFINET parameters of the HMI device

1. Click the HMI device in the "Devices & Networks" editor.
2. Change the parameters of the HMI device in the Inspector window under "Properties > General > General".



"Interface networked with"

In the "Interface networked with" area, select the subnet of the HMI connection via which the HMI device is connected to the network. You use the "Add new subnet" button to create a new subnet.

"IP protocol"

- "Set IP address in the project"
  When you transfer the WinCC project to the HMI device, this IP address is set up directly in the HMI device.

  Note

  The device is automatically restarted in the case of HMI devices with the Windows CE 3.0 operating system.

  HMI devices with Windows CE 3.0:
  - OP 77B
  - TP 177B color PN/DP
  - TP 177B mono DP
  - OP 177B color PN/DP
  - OP 177B mono DP
  - Mobile Panel 177 PN
  - Mobile Panel 177 DP
  - TP 277 6"
  - OP 277 6"

- "Subnet mask"
  You assign data of the subnet mask in the "Subnet mask" area.
- "Use IP router"
  If you are using an IP router, select "Use IP router" and enter the router address in the "Router address" field.
- "Set IP address using a different method"
  If the function "Set IP address using a different method" is activated, the IP address is not taken from the project. You have to enter the IP address directly in the Control Panel of the HMI device.

See also

Communication via PROFINET (Page 331)

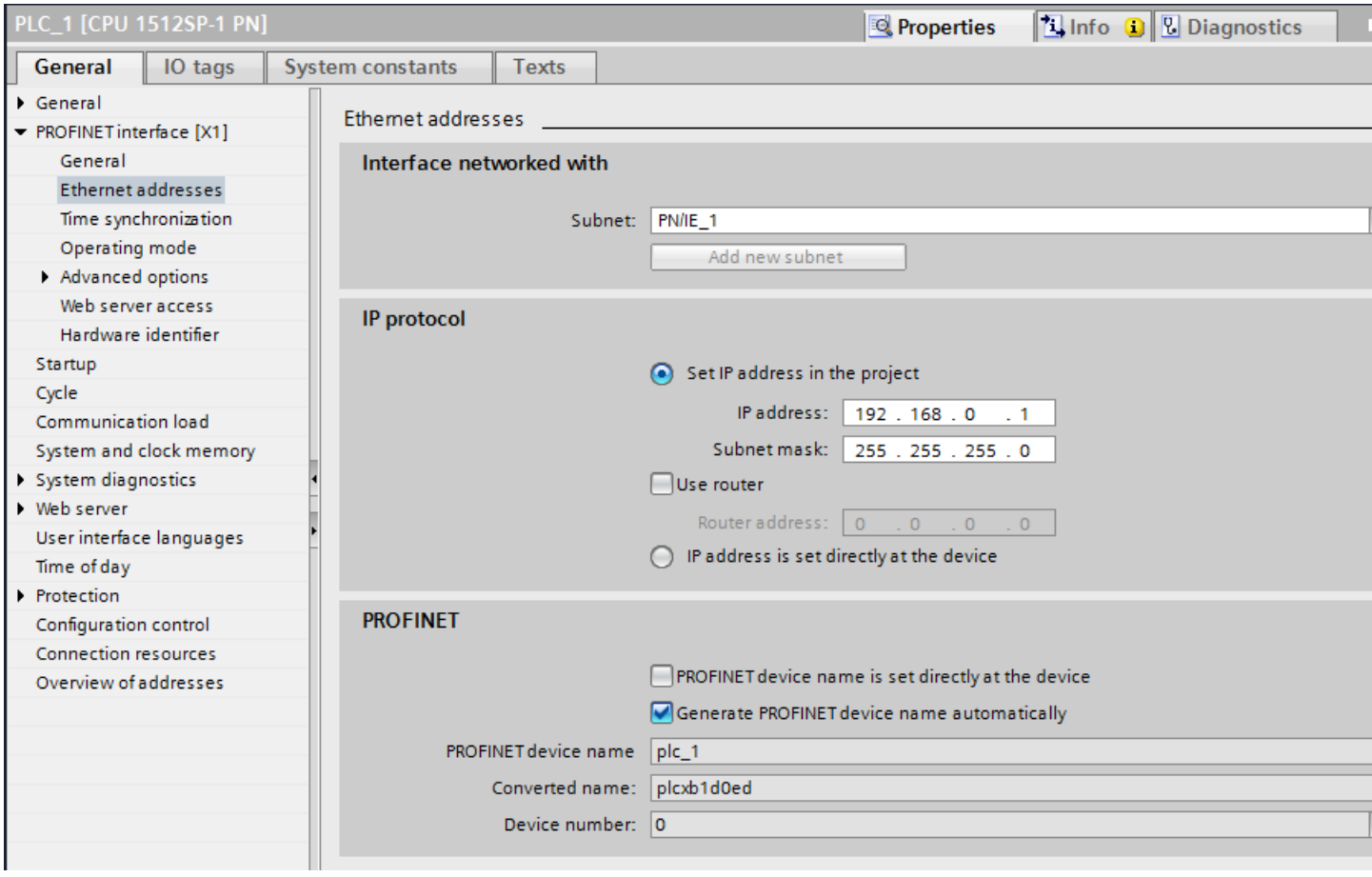
PROFINET parameters for the PLC (Basic Panels)

PROFINET parameters for the PLC

An overview of the configured parameters can be found in the properties for the PLC.

Displaying and changing PROFINET parameters of the PLC

1. Click the PLC in the "Devices & Networks" editor.
2. Change the parameters of the PLC in the Inspector window under "Properties > General > General".



"Interface networked with"

In the "Subnet" area, select the subnet of the HMI connection via which the PLC is connected to the network. You use the "Add new subnet" button to create a new subnet.

"IP protocol"

- "Interface type"
  Depending on the HMI device type, you have various interfaces to choose from.
- "IP address"
  You assign the IP address of the HMI device in the "IP address" area.
- "Subnet mask"
  You assign data of the subnet mask in the "Subnet mask" area.
  If you are using an IP router, select "Use IP router" and enter the router address in the field.

See also

Communication via PROFINET (Page 331)

Configuring Industrial Ethernet (Basic Panels)

Rules for the network configuration

The Ethernet interfaces of the devices have a default IP address that you can change.

IP address

The IP parameters are visible if the communication-capable devices support the TCP/IP protocol.

The IP address consists of 4 decimal figures in the range of 0 to 255. The decimal figures are separated from one another by a dot.

Example: 140.80.0.2

The IP address consists of the following:

- The address of the (sub) net
- The address of the node (generally also called host or network node)

Subnet mask

The subnet mask splits these two addresses. It determines which part of the IP address addresses the network and which part of the IP address addresses the node.

The set bits of the subnet mask determine the network part of the IP address.

Example:

Subnet mask: 255.255.0.0 = 11111111.11111111.00000000.00000000

In the example given for the above IP address, the subnet mask shown here has the following meaning:

The first 2 bytes of the IP address identify the subnet - i.e. 140.80. The last two bytes address the node, thus 0.2.

It is generally true that:

- The network address results from AND linking the IP address and subnet mask.
- The node address results from AND NOT linking the IP address and subnet mask.

Relation between IP address and default subnet mask

An agreement exists relating to the assignment of IP address ranges and so-called "Default subnet masks". The first decimal number (from the left) in the IP address determines the structure of the default subnet mask. It determines the number of "1" values (binary) as follows:

IP address (decimal)	IP address (binary)	Address class	Default subnet mask
0 to 126	0xxxxxxx.xxxxxxxx....	A	255.0.0.0
128 to 191	10xxxxxx.xxxxxxxx...	B	255.255.0.0
192 to 223	110xxxxx.xxxxxxxx...	C	255.255.255.0

Note

Range of values for the first decimal point

A value of between 224 and 255 is also possible for the first decimal number of the IP address (address class D etc). This is, however, not recommended because there is no address check for these values.

Masking other subnets

You can use the subnet mask to add further structures and form "private" subnets for a subnet that is assigned one of the address classes A, B or C. This is done by setting other lower points of the subnet mask to "1". For each bit set to "1", the number of "private" networks doubles and the number of nodes they contain is halved. Externally, the network functions like an individual network as it did previously.

Example:

You have a subnet of address class B (e.g. IP address 129.80.xxx.xxx) and change the default subnet mask as follows:

Masks	Decimal	Binary
Default subnet mask	255.255.0.0	11111111.11111111.00000000.00000000
Subnet mask	255.255.128.0	11111111.11111111.10000000.00000000

Result:

All nodes with addresses between 129.80.001.xxx and 129.80.127.xxx are on one subnet, all nodes with addresses between 129.80.128.xxx and 129.80.255.xxx are on another subnet.

Router

The job of the routers is to connect the subnets. If an IP datagram is to be sent to another network, it first has to be conveyed to a router. To make this possible, in this case you have to enter the address of the router for each node in the subnet.

The IP address of a node in the subnet and the address of the router may only differ at the points at which there is a "0" in the subnet mask.

See also

Communication via PROFINET (Page 331)

1.10.2.4 Setting port options (Basic Panels)

Setting the port options (Basic Panels)

Changing connection settings for the PROFINET IO port

You can change the network settings for the PROFINET IO port as required. By default, the settings are made automatically. In normal situations, this guarantees problem-free communication.

Possible settings for transmission rate / duplex

Depending on the selected device, you can make the following settings for "Transmission rate / duplex":

- Automatic setting
  Recommended default setting of the port. The transmission settings are automatically "negotiated" with the peer port. The "Enable autonegotiation" option is also enabled as a default, in other words, you can use cross cables or patch cables for the connection.
- TP/ITP at x Mbps full duplex (half duplex)
  Setting of the transmission rate and the full duplex/half duplex mode. The effectiveness depends on the "Enable autonegotiation" setting:
  - Autonegotiation enabled
    You can use both cross cable and patch cable.
  - Autonegotiation disabled
    Make sure that you use the correct cable (patch cable or cross cable)! The port is also monitored with this setting.
- Deactivated
  Depending on the module type, the drop down list box can contain the "- Disabled -" option. This option, for example, allows you to prevent access to an unused port for security reasons. With this setting, diagnostic events are not generated.

"Monitor" option

This option enables or disables port diagnostics. Examples of port diagnostics: The link status is monitored, in other words, the diagnostics are generated during link-down and the system reserve is monitored in the case of fiber optic ports.

Option "Enable autonegotiation "

The autonegotiation setting can only be changed if a concrete medium (for example, TP 100 Mbps full duplex) is selected. Whether or not a concrete medium can be set depends on the properties of the module.

If autonegotiation is disabled, this causes the port to be permanently specified, as for example, is necessary for a prioritized startup of the IO device.

You must make sure the partner port has the same settings because with this option the operating parameters of the connected network are not detected and the data transmission rate and transmission mode can accordingly not be optimally set.

Note

When a local port is connected, STEP 7 makes the setting for the partner port if the partner port supports the setting. If the partner port does not accept the setting, an error message is generated.

See also

Wiring rules for disabled autonegotiation (Page 158)

Boundaries at the port (Page 159)

Wiring rules for disabled autonegotiation (Basic Panels)

Requirements

You have made the following settings for the port in question, for example, to accelerate the startup time of the IO device:

- Fixed transmission rate
- Autonegotiation incl. autocrossing disabled

The time for negotiating the transmission rate during startup has been saved.

If you have disabled autonegotiation, you must observe the wiring rules.

Wiring rules for disabled autonegotiation

PROFINET devices have the following two types of ports:

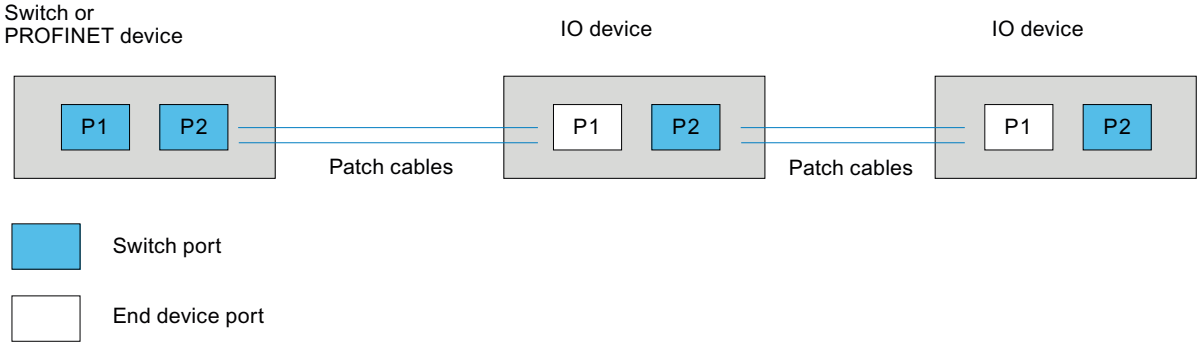
Type of port	PROFINET devices	Note
Switch port with crossed pin assignment	For IO devices: Port 2 For S7 CPUs with 2 ports: Ports 1 and 2	Crossed pin assignment means that the pin assignment for the ports for sending and receiving between the respective PROFINET devices is exchanged internally.
End device port with uncrossed pin assignment	For IO devices: Port 1 For S7 CPUs with one port: Port 1	-

Validity of the wiring rules

The cabling rules described in the following paragraph apply exclusively for the situation in which you have specified a fixed port setting.

Rules for cabling

You can connect several IO devices in line using a single cable type (patch cable). To do this, you connect port 2 of the IO device (distributed I/O) with port 1 of the next IO device. The following graphic gives an example with two IO devices.



See also

Setting the port options (Page 157)

Boundaries at the port (Page 159)

Boundaries at the port (Basic Panels)

Requirements

The respective device must support boundaries settings in order to work with boundaries. If the device for PROFINET does not support boundaries settings, the corresponding parameters are disabled.

Enable boundaries

"Boundaries" are limits for transmission of certain Ethernet frames. The following boundaries can be set at a port:

- "End of discovery of accessible devices"
  No forwarding of DCP frames to identify accessible devices. Devices downstream from this port cannot be reached by the project tree under "Accessible devices". Devices downstream from this port cannot be reached by the CPU.
- "End of topology discovery"
  LLDP frames (Link Layer Discovery Protocol) are not forwarded for topology detection.
- "End of sync domain"
  No forwarding of sync frames transmitted to synchronize nodes within a sync domain.
  If you operate, for example, a PROFINET device with more than two ports in a ring, you should prevent the sync frame from being fed into the ring by setting a sync boundary (at the ports not inside the ring).
  Additional example: If you want to use several sync domains, configure a sync domain boundary for the port connected to a PROFINET device from the other sync domain.

Restrictions

The following restrictions must be observed:

- The individual check boxes can only be used if the port supports the function in question.
- If a partner port has been determined for the port, the following check boxes cannot be used:
  - "End of discovery of accessible devices"
  - "End of topology discovery"

See also

Setting the port options (Page 157)

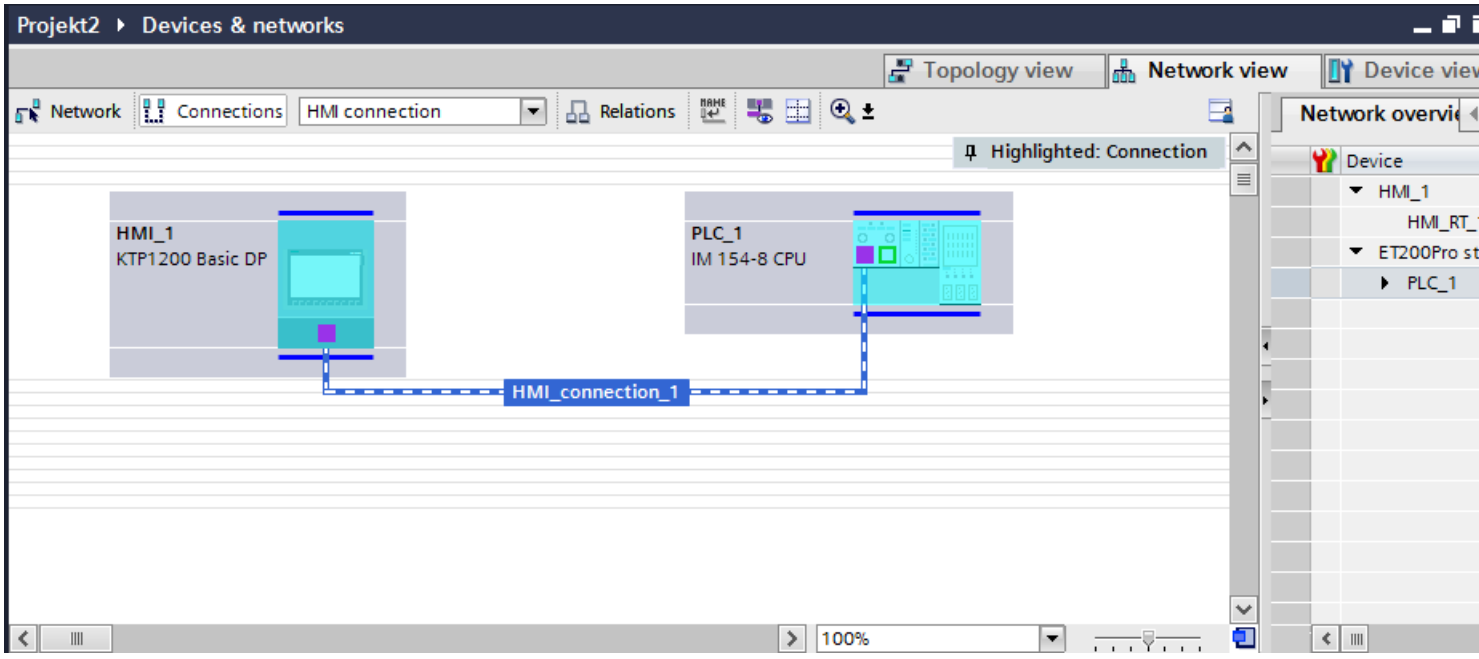
Wiring rules for disabled autonegotiation (Page 158)

1.10.3 Communication via PROFIBUS (Basic Panels)

1.10.3.1 Communication via PROFIBUS (Basic Panels)

HMI connections via PROFIBUS

If you want to connect a SIMATIC ET 200 CPU to a HMI device via PROFIBUS, you must configure a PROFIBUS-capable communication module to a slot of the controller first.



HMI connection in the "Devices & Networks" editor

You configure the HMI connection between the PLC and the HMI device via PROFIBUS in the "Devices & Networks" editor.

Connection in the "Connections" editor

Alternatively, you configure the connection between the PLC and HMI device in the "Connections" editor of the HMI device.

See also

Communication with SIMATIC ET 200 CPU (Page 331)

Configuring an HMI connection via PROFIBUS (Page 347)

1.10.3.2 Configuring an HMI connection via PROFIBUS (Basic Panels)

Introduction

You configure an HMI connection between HMI devices and a SIMATIC ET 200 CPU via PROFIBUS in the "Devices & Networks" editor.

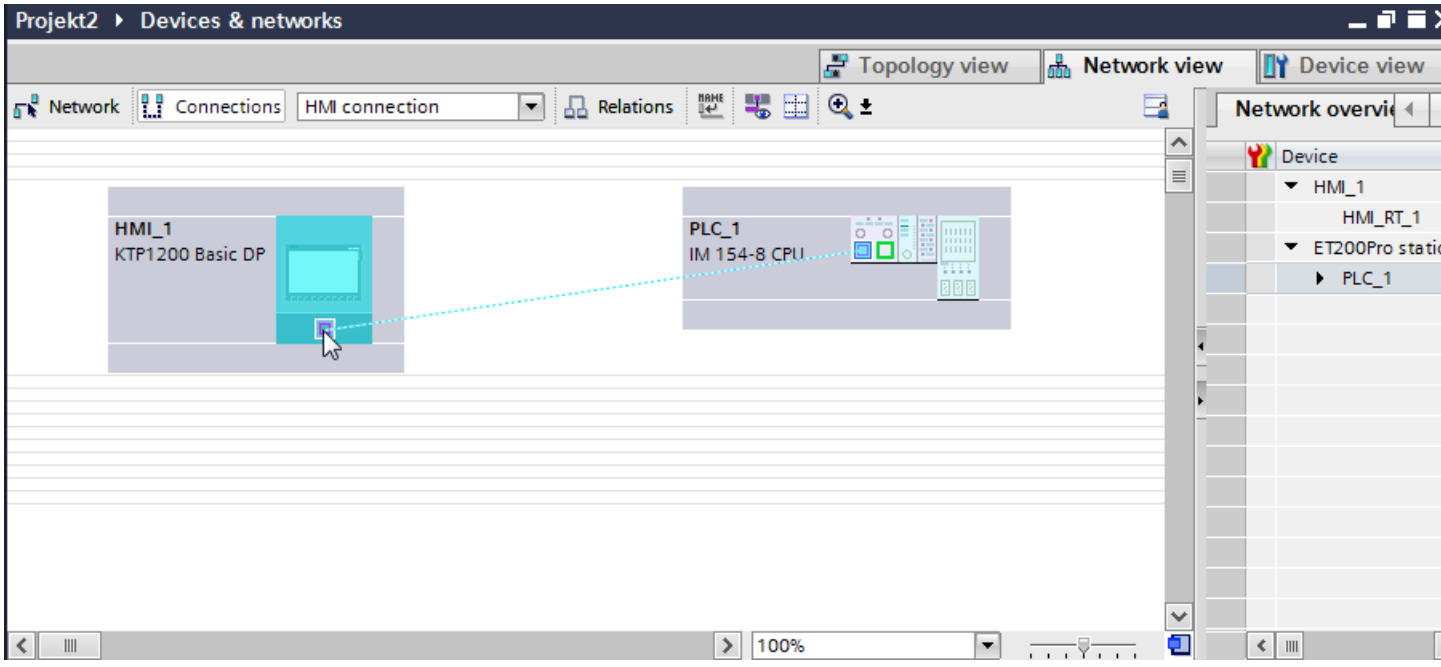
Requirements

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

- HMI device with MPI/DP interface
- SIMATIC ET 200 CPU with DP interface

Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed graphically in the network view.
2. Click the "Connections" button.
   The devices available for connection are highlighted in color.
3. Click the HMI device interface.
4. Select the "PROFIBUS" interface type in the Inspector window under "Properties > General > PROFIBUS address/ MPI address > Parameters".
5. Click the interface of the CPU and use a drag-and-drop operation to draw a connection to the HMI device.



6. Click the name of the connection.
   The connection is displayed graphically in the Inspector window.
7. Click "Highlight HMI connection" and select the HMI connection.
8. Click the communication partners in the "Network view" and change the PROFIBUS parameters in the Inspector window according to the requirements of your project.
   For additional information on this, see section "PROFIBUS parameters (Page 348)".

Note

The created HMI connection is also shown in the tabular area of the editor on 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 an HMI connection between an HMI device and a SIMATIC ET 200 CPU via PROFIBUS.

See also

Communication with SIMATIC ET 200 CPU (Page 331)

Communication via PROFIBUS (Page 346)

PROFIBUS parameters (Page 348)

1.10.3.3 PROFIBUS parameters (Basic Panels)

PROFIBUS parameters for the HMI connection (Basic Panels)

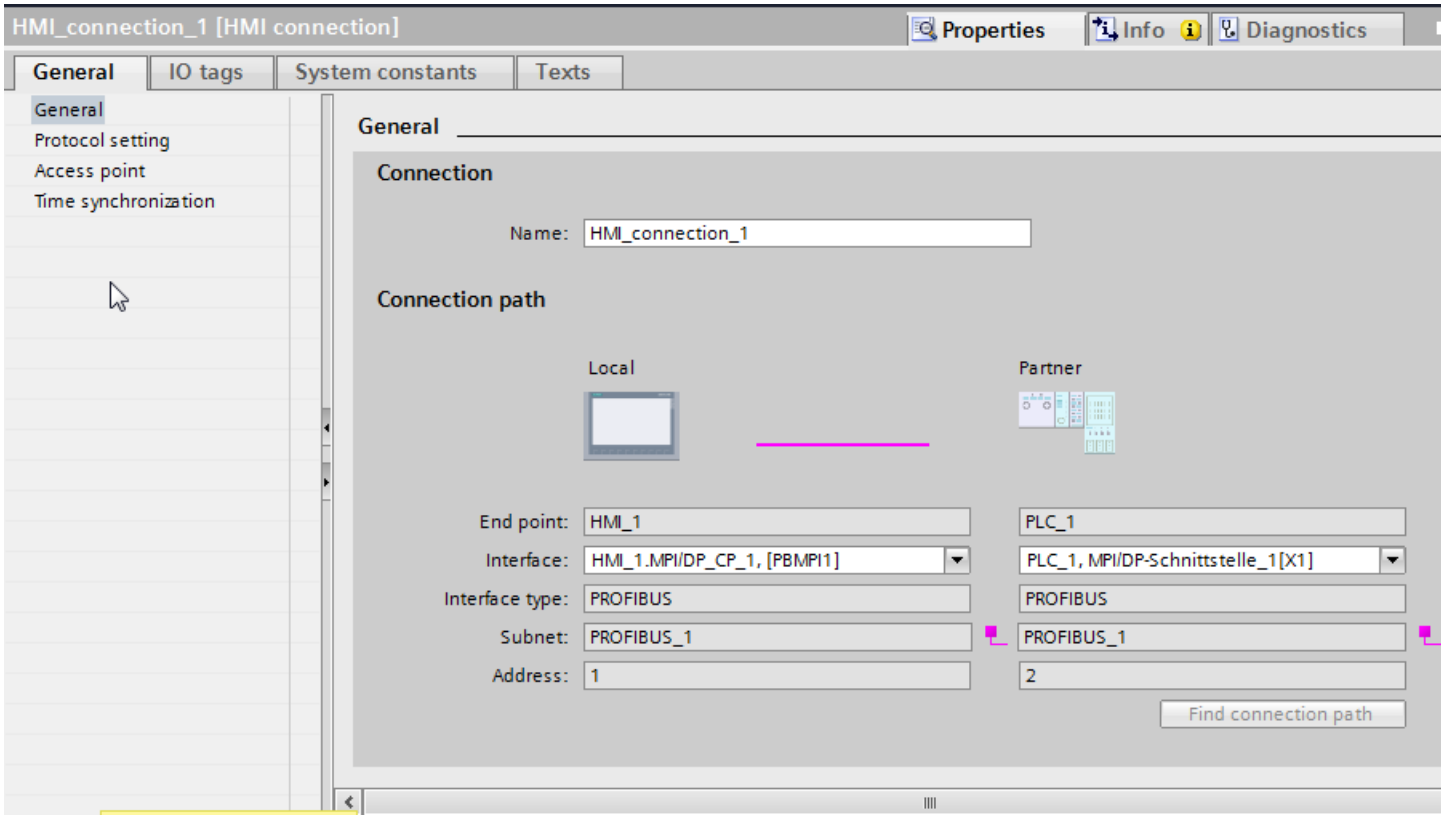
PROFIBUS parameters for the HMI connection

An overview of the configured HMI connection parameters can be found in the properties for the HMI connection.

Only limited changes are possible in this Inspector window.

Displaying and editing HMI connection parameters

1. Click the HMI connection in the "Devices & Networks" editor.
2. Change the parameters of the HMI connection in the Inspector window under "Properties > General > General".



"Connection"

The "Connection" area displays the HMI connection created for communication between the devices.

You can edit the name of the HMI connection in this area.

"Connection path"

The communication partners of the selected HMI connection and the associated PROFIBUS parameters are displayed in the "Connection path" area. Some of the areas displayed cannot be edited in this dialog.

- "End point"
  Displays the device name. This area cannot be edited.
- "Interface"
  Displays the selected interface of the device. You can choose between several interfaces, depending on the device.
- "Interface type"

Displays the selected interface type. This area cannot be edited.

- "Subnet"
  Displays the selected subnet. This area cannot be edited.
- "Address"
  Displays the PROFIBUS address of the device. This area cannot be edited.
- "Find connection path" button
  Enables the subsequent specification of connections.

See also

Communication via PROFIBUS (Page 346)

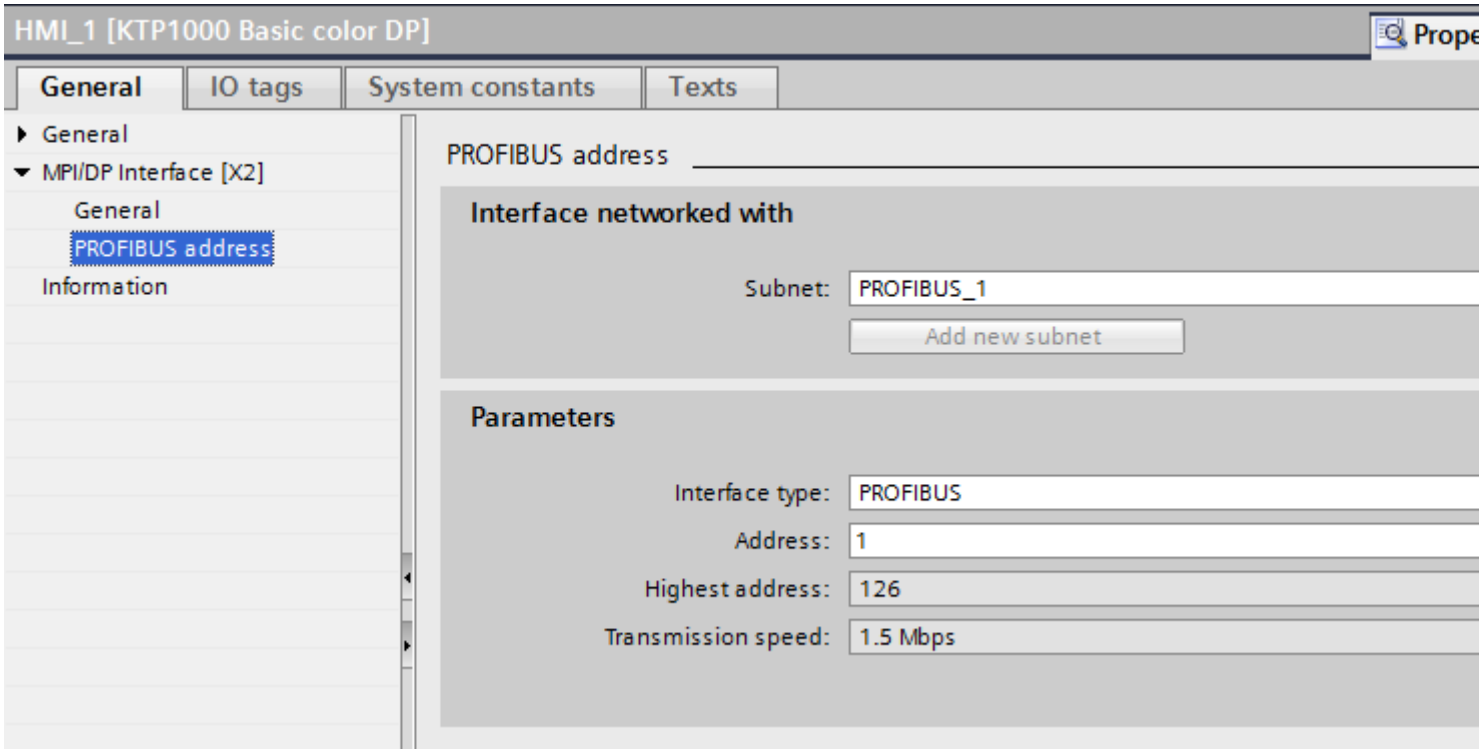
PROFIBUS parameters for the HMI device (Basic Panels)

PROFIBUS parameters for the HMI device

An overview of the configured HMI device parameters can be found in the properties for the HMI device.

Displaying and changing PROFIBUS parameters of the HMI device

1. Click the HMI device in the "Devices & Networks" editor.
2. Change the parameters of the HMI device in the Inspector window under "Properties > General > General".



"Interface networked with"

In the "Interface networked with" area, select the subnet of the HMI connection via which the HMI device is connected to the network. You use the "Add new subnet" button to create a new subnet.

"Parameters"

- "Interface type"
  Depending on the HMI device type, you have various interfaces to choose from.
- "Address"
  You assign the PROFIBUS address of the HMI device in the "Address" area. The PROFIBUS address must be unique throughout the PROFIBUS network.
- "Highest address"
  The "Highest address" area displays the highest address of the PROFIBUS network.
- "Transmission speed"
  The "Transmission speed" is determined by the slowest device connected to the network. The setting is identical throughout the network.

See also

Communication via PROFIBUS (Page 346)

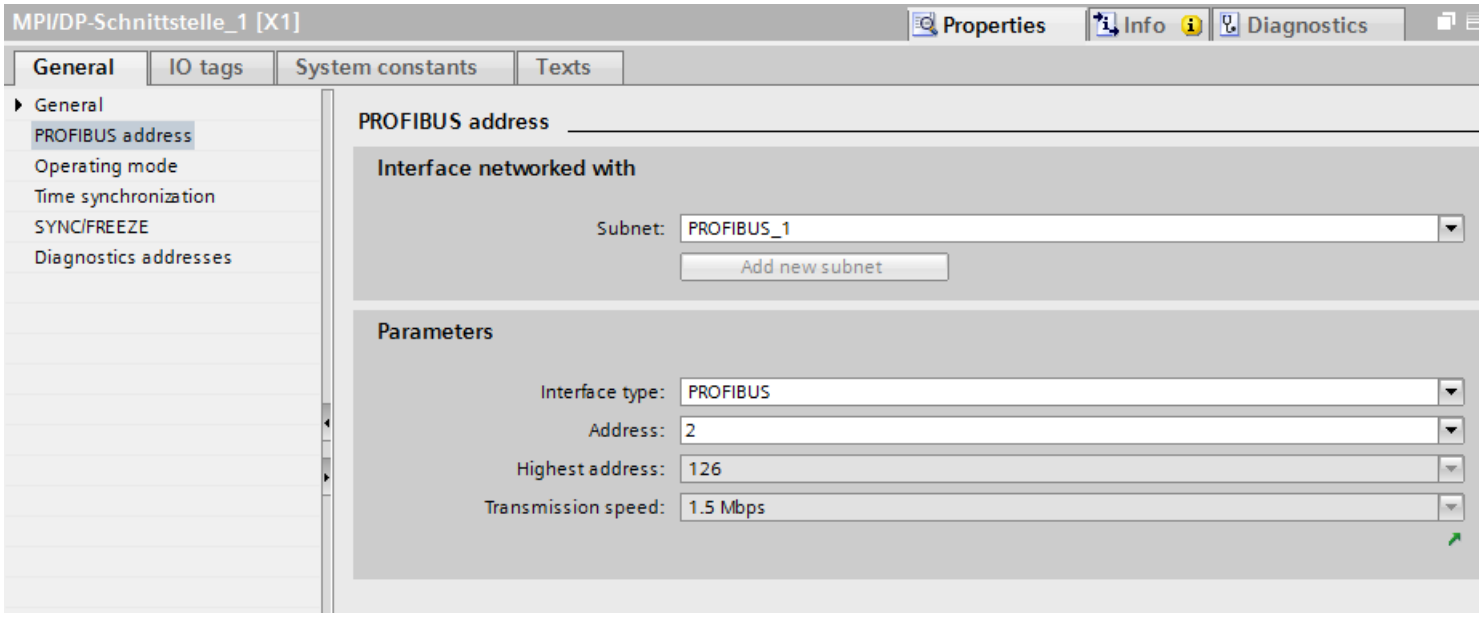
PROFIBUS parameters for the PLC (Basic Panels)

PROFIBUS parameters for the PLC

An overview of the configured parameters can be found in the properties for the PLC.

Displaying and changing PROFIBUS parameters of the PLC

1. Click the PLC in the "Devices & Networks" editor.
2. Change the parameters of the PLC in the Inspector window under "Properties > General > General".



"Interface networked with"

In the "Subnet" area, select the subnet of the HMI connection via which the PLC is connected to the network. You use the "Add new subnet" button to create a new subnet.

"Parameters"

- "Interface type"
  Depending on the HMI device type, you have various interfaces to choose from.
- "Address"
  You assign the PROFIBUS address of the HMI device in the "Address" area. The PROFIBUS address must be unique throughout the PROFIBUS network.
- "Highest address"
  The "Highest address" area displays the highest address of the PROFIBUS network.
- "Transmission speed"
  The "Transmission speed" is determined by the slowest device connected to the network. The setting is identical throughout the network.

See also

Communication via PROFIBUS (Page 346)

1.10.4 Data exchange (Basic Panels)

1.10.4.1 Data exchange using area pointers (Basic Panels)

General information on area pointers (Basic Panels)

Introduction

You use an area pointer to access a data area in the PLC. During communication, the PLC and the HMI device alternately access these data areas for read and write operations.

The PLC and the HMI device trigger defined interactions based on the evaluation of stored data.

Configuration of area pointers

Before you use an area pointer, you enable it under "Connections > Area pointer". You then assign the area pointer parameters.

You can find more detailed information on configuring area pointers in:

Configuring area pointers (Page 66)

See also

Configuring area pointers (Page 66)

"Data mailbox" area pointer (Page 363)

Area pointer "Date/time" (Page 353)

"Date/time PLC" area pointer (Page 355)

Area pointer "Coordination" (Page 356)

Area pointer "Screen number" (Page 358)

Area pointer "Project ID" (Page 359)

Area pointer "Job mailbox" (Page 360)

Area pointer "Date/time" (Basic Panels)

Function

This area pointer is used to transfer the date and time from the HMI device to the PLC.

The PLC writes control job "41" to the job mailbox.

When it evaluates the control job, the HMI device writes its current date and the time in the data area configured in the "Date/time" area pointer.

Note

You cannot use the "Date/Time PLC" area pointer if you have configured the "Date/Time" area pointer.

The "Date/Time" area pointer when used in a project which contains multiple connections must be enabled for each configured connection.

"DTL" data type

If you configure the "Date/Time" area pointer, use the DTL data type.

A tag of the "DTL" data type has a length of 12 bytes and saves information on date and time in a predefined structure.

The "DTL" data type has the following structure:

Byte	Component	Data type	Value range
0	Year	UINT	1970 to 2554
1			
2	Month	USINT	0 to 12
3	Day	USINT	1 to 31
4	Day of the week	USINT	1(Sunday) to 7(Saturday) The day of the week is not considered in the value entry.
5	Hour	USINT	0 to 23
6	Minute	USINT	0 to 59
7	Second	USINT	0 to 59
8	Nanoseconds	UDINT	0 to 999 999 999
9			
10			
11			

The "DTL" data type supports time information down to the nanosecond range. Because panels only support time information down to the millisecond, you may encounter the following restriction when using the area pointers:

For the transmission of time information from a panel to the controller, the smallest unit of time is 1 millisecond. The value range from microseconds to nanoseconds of the "DTL" data type is filled with zeros.

See also

General information on area pointers (Page 353)

"Date/time PLC" area pointer (Page 355)

Area pointer "Coordination" (Page 356)

Area pointer "Screen number" (Page 358)

Area pointer "Project ID" (Page 359)

Area pointer "Job mailbox" (Page 360)

"Date/time PLC" area pointer (Basic Panels)

Function

This area pointer is used to transfer the date and time from the PLC to the HMI device. Use this area pointer if the PLC is the time master.

The PLC loads the data area of the area pointer.

The HMI device reads the data cyclically within the configured acquisition cycle and synchronizes itself.

Note

Set an acquisition cycle of sufficient length for the date/time area pointer PLC to avoid any negative impact on HMI device performance.
Recommended: Acquisition cycle of 1 minute, if the process allows this.

"Date/Time PLC" is a global area pointer and may be configured only once per project.

Note

You cannot use the "Date/Time" area pointer if you have configured the "Date/Time PLC" area pointer.

"DTL" data type

If you configure the "Date/Time PLC" area pointer, use the DTL data type.

A tag of the "DTL" data type has a length of 12 bytes and saves information on date and time in a predefined structure.

The "DTL" data type has the following structure:

Byte	Component	Data type	Value range
0	Year	UINT	1970 to 2554
1			
2	Month	USINT	0 to 12
3	Day	USINT	1 to 31
4	Day of the week	USINT	1(Sunday) to 7(Saturday) The day of the week is not considered in the value entry.
5	Hour	USINT	0 to 23
6	Minute	USINT	0 to 59
7	Second	USINT	0 to 59

Byte	Component	Data type	Value range
8	Nanoseconds	UDINT	0 to 999 999 999
9			
10			
11			

The HMI devices do not support the use of nanoseconds. Values in the nanosecond range will be ignored during processing in Runtime.

The "DTL" data type supports time information down to the nanosecond range. Because panels only support time information down to the millisecond, you may encounter the following restriction when using the area pointers:

For the transmission of time information from a PLC to a panel, the range from microseconds to nanoseconds is ignored. The time information is processed on the panel down to milliseconds.

See also

General information on area pointers (Page 353)

Area pointer "Date/time" (Page 353)

Area pointer "Coordination" (Page 356)

Area pointer "Screen number" (Page 358)

Area pointer "Project ID" (Page 359)

Area pointer "Job mailbox" (Page 360)

Area pointer "Coordination" (Basic Panels)

Function

The "Coordination" area pointer is used to implement the following functions:

- Detecting the startup of the HMI device in the control program
- Detecting the current operating mode of the HMI device in the control program
- Detecting whether the HMI device is ready to communicate in the control program

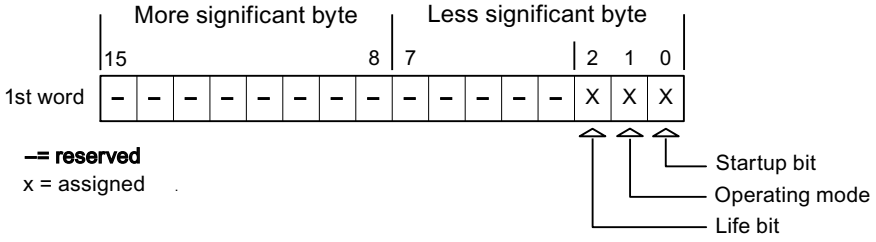
By default, the "Coordination" area pointer has the length of one word and cannot be changed.

Usage

Note

The HMI device always writes the entire coordination area when updating the area pointer. The control program can for this reason not make changes to the coordination area.

Assignment of the bits in the "Coordination" area pointer



Startup bit

The startup bit is set briefly to "0" by the HMI device during startup. It sets the bit permanently to "1" when startup is completed.

Operating mode

The operating mode bit is set to 1 as soon as the user switches the HMI device offline. The status of the operating mode bit is "0" during normal operation of the HMI device. You can determine the current operating mode of the HMI device by reading this bit in the control program.

Life bit

The HMI device inverts the life bit at intervals of approximately one second. You can check whether or not there is still a connection to the HMI device by querying this bit in the control program.

Processing in the PLC

For a simpler evaluation in the PLC program, use a Bool array for this area pointer when using the SIMATIC S7 1200 communication driver. You will have to map the complete 16-bit word of the area pointer. Configure a tag of the data type "Array [0 .. 15] of bool" for this purpose.

Permitted data types

You can use the following data types when you configure the "Coordination" area pointer.

- Word
- UInt
- Bool

See also

General information on area pointers (Page 353)

Area pointer "Date/time" (Page 353)

"Date/time PLC" area pointer (Page 355)

Area pointer "Screen number" (Page 358)
Area pointer "Project ID" (Page 359)
Area pointer "Job mailbox" (Page 360)

Area pointer "Screen number" (Basic Panels)

Function

The HMI devices store information about the screen called up on the HMI device in the "Screen number" area pointer.

This allows the transfer of the current screen contents from the HMI device to the PLC. Certain reactions can be triggered in the PLC, such as the call of a different screen.

Use

Before the "Screen number" area pointer can be used, it must be set up and activated by selecting "Communication ▸ Area pointer". You can create only **one** instance of the "Screen number" area pointer and only on **one** PLC.

The screen number is always transferred to the PLC when a new screen is activated or when the focus within a screen changes from one screen object to another.

Structure

The area pointer is a data area in the memory of the PLC with a fixed length of 5 words.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st word	Current screen type															
2nd word	Current screen number															
3rd word	Reserved															
4th word	Current field number															
5th word	Reserved															

- Current screen type
  "1" for root screen or
  "4" for permanent area
- Current screen number
  1 to 32767
- Current field number
  1 to 32767

Note

Device dependency

Permanent areas are not available on Basic Panels.

Permitted data types

You can use the following data types when you configure the "Screen number" area pointer.

- Word
- UInt

See also

General information on area pointers (Page 353)

Area pointer "Date/time" (Page 353)

"Date/time PLC" area pointer (Page 355)

Area pointer "Coordination" (Page 356)

Area pointer "Project ID" (Page 359)

Area pointer "Job mailbox" (Page 360)

Area pointer "Project ID" (Basic Panels)

Function

When Runtime starts, a check can be carried out as to whether the HMI device is connected to the correct PLC. This check is important when operating with several HMI devices.

For this, the HMI device compares a value stored on the PLC with the value specified in configuration. This ensures compatibility of the configuration data with the PLC program.

A missing compatibility results in a corresponding alarm and Runtime will not be started.

Use

Note

HMI connections cannot be switched "online".

The HMI connection in which the "Project ID" area pointer is used must be switched "online".

To use this area pointer, set up the following during the configuration:

- Define the version of the configuration. Values between 1 and 255 are possible.
  You enter the version in the editor "Runtime settings > General" in the "Identification" area.
- This is where you select the PLC tag or the tag array that you have configured as the data area for the area pointer.

Connection failure

A connection failure to a device on which the "project ID" area pointer is configured results in all the other connections of the device being switched to "offline".

This behavior has the following requirements:

- You have configured several connections in a project.
- You are using the "project ID" area pointer in at least one connection.

Causes which may set connections "offline":

- The PLC is not available.
- The connection has been switched offline in the engineering system.

Permitted data types

You can use the following data types when you configure the "Project ID" area pointer.

- Word
- UInt

See also

General information on area pointers (Page 353)
Area pointer "Date/time" (Page 353)
"Date/time PLC" area pointer (Page 355)
Area pointer "Coordination" (Page 356)
Area pointer "Screen number" (Page 358)
Area pointer "Job mailbox" (Page 360)

Area pointer "Job mailbox" (Basic Panels)

Function

The PLC can use the job mailbox to transfer jobs to the HMI device to trigger corresponding actions on the HMI device. These functions include, for example:

- Display screen
- Set date and time

Data structure

The first word of the job mailbox contains the job number. Depending on the job mailbox, up to three parameters can be transferred.

Word	Most significant byte	Least significant byte
n+0	0	Job number
n+1	Parameter 1	
n+2	Parameter 2	
n+3	Parameter 3	

The HMI device evaluates the job mailbox if the first word of this job is not equal to zero. This means that the parameters must be entered in the job mailbox first, followed by the job number.

When the HMI device accepts the job mailbox, the first word is set to 0 again. The execution of the job mailbox is generally not completed at this point in time.

Job mailboxes

All job mailboxes and their parameters are listed below. The "No." column contains the job number of the job mailbox. Job mailboxes can only be triggered by the PLC when the HMI device is online.

No.	Function	
14	**Set time (BCD-coded)**	
	Parameter 1	Left byte: - Right byte: hours (0-23)
	Parameter 2	Left byte: minutes (0-59) Right byte: seconds (0-59)
	Parameter 3	-
15	**Set date (BCD-coded)**	
	Parameter 1	Left byte: - Right byte: weekday (1-7: Sunday-Saturday)
	Parameter 2	Left byte: day (1-31) Right byte: month (1-12)
	Parameter 3	Left byte: year
23	**User logon**	
	Logs the user on with the name "PLC user" at the HMI device with the group number transferred in Parameter 1. The logon is possible only when the transferred group number exists in the project.	
	Parameter 1	Group number 1 to 255
	Parameter 2, 3	-
24	**User logoff**	
	Logs off the current user. (The function corresponds to the "logoff" system function)	
	Parameter 1, 2, 3	-
40	**Transfer date/time to PLC**	
	An interval of at least 5 seconds must be maintained between two successive jobs to prevent overload of the HMI device.	
	Parameter 1, 2, 3	-
41	**Transfer date/time to PLC**	
	An interval of at least 5 seconds must be maintained between two successive jobs to prevent overload of the HMI device.	
	Parameter 1, 2, 3	-
46	**Update tag**	
	Causes the HMI device to read the current value of the tags from the PLC whose update ID matches the value transferred in Parameter 1. (Function corresponds to the "UpdateTag" system function.)	

No.	Function	
14	**Set time (BCD-coded)**	
	Parameter 1	1 - 100
49	**Clear event buffer**	
	Deletes all analog alarms and discrete alarms of the "Warnings" class from the alarm buffer.	
	Parameter 1, 2, 3	-
50	**Clear error alarm buffer**	
	Deletes all analog alarms and discrete alarms of the "Errors" class from the alarm buffer.	
	Parameter 1, 2, 3	-
51	**Display selection**	
	Parameter 1	Screen number
	Parameter 2	-
	Parameter 3	Field number
69	**Reading data record from PLC** [1]	
	Parameter 1	Recipe number (1-999)
	Parameter 2	Data record number (1-65535)
	Parameter 3	0: Do not overwrite existing data record 1: Overwrite existing data record
70	**Writing data record to PLC** [1]	
	Parameter 1	Recipe number (1-999)
	Parameter 2	Data record number (1-65535)
	Parameter 3	-

[1]	Only devices supporting recipes
[2]	OP 73, OP 77A and TP 177A HMI devices also execute the "Screen selection" job mailbox if the on-screen keyboard is active.
[3]	The weekday is ignored on HMI device KTP 600 BASIC PN.

Permitted data types

You can use the following data types when you configure the "Screen number" area pointer:

- Word
- UInt

See also

General information on area pointers (Page 353)

Area pointer "Date/time" (Page 353)

"Date/time PLC" area pointer (Page 355)

Area pointer "Coordination" (Page 356)

Area pointer "Screen number" (Page 358)

Area pointer "Project ID" (Page 359)

"Data record" area pointer (Basic Panels)

"Data mailbox" area pointer (Basic Panels)

Function

When data records are transferred between the HMI device and PLC, both partners access common communications areas on the PLC.

Data transfer types

There are two ways of transferring data records between the HMI device and PLC:

- Transfer without synchronization
- Transfer with synchronization over the data mailbox

Data records are always transferred directly. That is, the tag values are read from an address or written to an address configured for this tag directly, without redirecting the values by means of interim memory.

Initiating the transfer of data records

There are three ways of triggering the transfer:

- Operator input in the recipe view
- Job mailboxes
  The transfer of data records can also be triggered by the PLC.
- Triggering by configured functions

If the transfer of data records is triggered by a job mailbox, the data in the recipe view will be updated as well. Avoid operating the recipe view while job mailboxes for transfer of data records are being triggered. If you have already started editing a data record and a job mailbox is triggered for transfer of data records, then this job mailbox will be rejected.

Permitted data types

You can use the following data types when you configure the "Data record" area pointer.

- Word
- UInt

See also

General information on area pointers (Page 353)

Transfer without synchronization (Page 364)

Transfer with synchronization (Page 365)

Sequence of the transfer triggered by a job mailbox (Page 367)

Sequence of the transfer when triggered by a configured function (Page 369)

Possible causes of error when transferring data records (Page 370)

Sequence of a transfer started by the operator in the recipe display (Page 366)

Transfer without synchronization (Basic Panels)

If you select asynchronous transfer of data records between the HMI device and PLC, there is no coordination over the common data areas. It is therefore unnecessary to set up a data area during configuration.

Asynchronous data record transfer can be a useful alternative, for example, when:

- The system is capable of excluding the risk of uncontrolled overwriting of data by the communication peer.
- The PLC does not require information about the recipe number and data record number.
- The transfer of data records is triggered by the operator of the HMI device.

Reading values

When a read job is triggered, the values are read from the PLC addresses and transferred to the HMI device.

- Triggering by the operator in the recipe view:
  The values are downloaded to the HMI device. You can then process, edit, or save these values, for example.
- Triggering by a function or job mailbox:
  The values are saved immediately to the data volume.

Writing values

When a write job is triggered, the values are written to the PLC addresses.

- Triggering by the operator in the recipe view:
  The current values are written to the PLC.
- Triggering by a function or job mailbox:
  The current values are written to the PLC from the data medium.

See also

"Data mailbox" area pointer (Page 363)

Sequence of a transfer started by the operator in the recipe display (Page 366)

Transfer with synchronization (Page 365)

Sequence of the transfer triggered by a job mailbox (Page 367)

Sequence of the transfer when triggered by a configured function (Page 369)

Possible causes of error when transferring data records (Page 370)

Transfer with synchronization (Basic Panels)

If you select synchronous transfer, both communication partners set status bits in the common data area. Use this mechanism to prevent uncontrolled overwriting of data in either direction of your control program.

Application

Synchronous data record transfer can be a useful solution, for example, when:

- The PLC is the "active partner" in the transfer of data records.
- The PLC evaluates the information about the recipe number and data record number.
- The transfer of data records is triggered by means of a Job mailbox.

Requirements

In order to synchronize transfer of data records between the HMI device and the PLC, the following requirements must be met during configuration:

- An area pointer has been set up: "Connections ▸ Area pointer" editor.
- The PLC with which the HMI device synchronizes transfer of data records is specified in the recipe:
  "Recipes" editor in the inspector window the option "Coordinated transfer of data records" under "General > Synchronization > Settings"

Structure of the data area

The data area has a fixed length of 5 words. Structure of the data area:

	15		**0**
1. Word	Current recipe number (1 - 999)		
2. Word	Current data record number (0 - 65535)		
3. Word	Reserved		
4. Word	Status (0, 2, 4, 12)		
5. Word	Reserved		

- Status
  The status word (word 4) can adopt the following values:

Value		Meaning
Decimal	Binary	
0	0000 0000	Transfer permitted, data record free
2	0000 0010	Transfer is busy
4	0000 0100	Transfer completed without error
12	0000 1100	Transfer completed with error

See also

"Data mailbox" area pointer (Page 363)
Transfer without synchronization (Page 364)
Sequence of a transfer started by the operator in the recipe display (Page 366)
Sequence of the transfer triggered by a job mailbox (Page 367)
Sequence of the transfer when triggered by a configured function (Page 369)
Possible causes of error when transferring data records (Page 370)

Sequence of a transfer started by the operator in the recipe display (Basic Panels)

Reading from the PLC started by the operator in the recipe view

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe number to be read and the status "Transferring" in the data mailbox and sets the data record number to 0.	Abort with system alarm.
3	The HMI device reads the values from the PLC and displays them in the recipe view. If the recipes have synchronized tags, the values from the PLC are also written to the tags.	
4	The HMI device sets the status "Transfer completed."	
5	The control program must reset the status word to zero in order to enable further transfers.	

Writing to the PLC started by the operator in the recipe view

Step	Action	
	Check: Status word = 0?	
1	Yes	No
	The HMI device enters the recipe and data record number to be written and the status "Transferring" in the data mailbox.	Abort with system alarm.

Step	Action	
2	The HMI device writes the current values to the PLC. If the recipes have synchronized tags, the changed values are synchronized between the recipe view and tags and then written to the PLC.	
3	The HMI device sets the status "Transfer completed."	
4	If required, the control program can now evaluate the transferred data.	
5	The control program must reset the status word to zero in order to enable further transfers.	

Note

The status word may only be set by the HMI device. The PLC may only reset the status word to zero.

Note

The PLC may only evaluate the recipe and data record numbers when data inconsistency is detected if one of the conditions outlined below has been met:

- The data mailbox status is set to "Transfer completed".
- The data mailbox status is set to "Transfer completed with error".

See also

Transfer without synchronization (Page 364)

"Data mailbox" area pointer (Page 363)

Transfer with synchronization (Page 365)

Possible causes of error when transferring data records (Page 370)

Sequence of the transfer when triggered by a configured function (Page 369)

Sequence of the transfer triggered by a job mailbox (Page 367)

Sequence of the transfer triggered by a job mailbox (Basic Panels)

The transfer of data records between the HMI device and the PLC can be initiated by either one of these stations.

The two job mailboxes No. 69 and No. 70 are available for this type of transfer.

No. 69: Read data record from PLC ("PLC → DAT")

Job mailbox no. 69 transfers data mailboxes from the PLC to the HMI device. The job mailbox is structured as follows:

	Most significant byte	Least significant byte
Word 1	0	69
Word 2	Recipe number (1-999)	

	Most significant byte	Least significant byte
Word 3	Data record number (1-65,535)	
Word 4	Do not overwrite existing data record: 0 Overwrite existing data record: 1	

No. 70: Write data record to PLC ("DAT → PLC")

Job mailbox no. 70 transfers data mailboxes from the HMI device to the PLC. The job mailbox is structured as follows:

	Most significant byte	Least significant byte
Word 1	0	70
Word 2	Recipe number (1-999)	
Word 3	Data record number (1-65,535)	
Word 4	—	

Sequence when reading from the PLC with job mailbox "PLC → DAT" (no. 69)

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the job and the status "Transferring" in the data mailbox.	Abort without return message.
3	The HMI device reads the values and stores the values in the data record specified in the job mailbox.	
4	• If "Overwrite" was selected in the job, an existing data record is overwritten without any prompt for confirmation. The HMI device sets the status "Transfer completed". • If "Do not overwrite" was selected in the job, and the data record already exists, the HMI device aborts the job and enters 0000 1100 in the status word of the data mailbox.	
5	The control program must reset the status word to zero in order to enable further transfers.	

Sequence writing to the PLC with job mailbox "DAT → PLC" (no. 70)

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the job and the status "Transferring" in the data mailbox.	Abort without return message.
3	The HMI device fetches the values of the data record specified in the job from the data medium and writes the values to the PLC.	

Step	Action	
4	The HMI device sets the status "Transfer completed."	
5	The PLC program can now evaluate the transferred data. To allow further transfers, the PLC program must set the status word to 0 again.	

See also

"Data mailbox" area pointer (Page 363)
Transfer without synchronization (Page 364)
Transfer with synchronization (Page 365)
Sequence of a transfer started by the operator in the recipe display (Page 366)
Sequence of the transfer when triggered by a configured function (Page 369)
Possible causes of error when transferring data records (Page 370)

Sequence of the transfer when triggered by a configured function (Basic Panels)

Reading from the PLC using a configured function

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the function and the status "Transferring" in the data mailbox.	Abort with system alarm.
3	The HMI device reads the values from the PLC and stores them in the data record specified in the function.	
4	• If "Yes" was selected for the "Overwrite" function, an existing data record is overwritten without any prompt for confirmation. The HMI device sets the status "Transfer completed." • If "No" was selected for the "Overwrite" function and the data record already exists, the HMI device aborts the job and enters 0000 1100 in the status word of the data mailbox.	
5	The control program must reset the status word to zero in order to enable further transfers.	

Writing to the PLC by means of configured function

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the function and the status "Transferring" in the data mailbox.	Abort with system alarm.

Step	Action	
3	The HMI device fetches the values of the data record specified in the function from the data medium and transfers the values to the PLC.	
4	The HMI device sets the status "Transfer completed."	
5	The control program can now evaluate the transferred data. The control program must reset the status word to zero in order to enable further transfers.	

See also

"Data mailbox" area pointer (Page 363)

Transfer without synchronization (Page 364)

Transfer with synchronization (Page 365)

Sequence of a transfer started by the operator in the recipe display (Page 366)

Sequence of the transfer triggered by a job mailbox (Page 367)

Possible causes of error when transferring data records (Page 370)

Possible causes of error when transferring data records (Basic Panels)

Possible causes of error

The section below shows possible error causes which lead to the cancellation of data record transfer:

- Tag address not set up on the PLC
- Overwriting data records not possible
- Recipe number does not exist
- Data record number does not exist

Note

The status word may only be set by the HMI device. The PLC may only reset the status word to zero.

Note

The PLC may only evaluate the recipe and data record numbers when data inconsistency is detected if one of the conditions outlined below has been met:

- The data mailbox status is set to "Transfer completed".
- The data mailbox status is set to "Transfer completed with error".

Reaction to an aborted transfer due to errors

If the transfer of data records is aborted due to errors, the HMI device reacts as follows:

- Triggering by the operator in the recipe view
  Information in the status bar of the recipe view and output of system alarms
- Triggered by function
  Output of system alarms
- Triggering by job mailbox
  No return message on the HMI device

You can nonetheless evaluate the status of the transfer by querying the status word in the data mailbox.

Note

Availability for specific devices

Notes in the status bar of the recipe view are not available in Basic Panels.

See also

"Data mailbox" area pointer (Page 363)

Transfer without synchronization (Page 364)

Transfer with synchronization (Page 365)

Sequence of a transfer started by the operator in the recipe display (Page 366)

Sequence of the transfer triggered by a job mailbox (Page 367)

Sequence of the transfer when triggered by a configured function (Page 369)

1.10.4.2 Trends (Basic Panels)

Trends

A trend is the graphical representation of one or more values from the PLC. The value is read out time-triggered for Basic Panels.

For additional information see:

AUTOHOTSPOT

Time-triggered trends

The HMI device reads in the trend values cyclically at an interval specified in the configuration.

Time-triggered trends are suitable for continuous curves, such as the operating temperature of a motor.

1.10.4.3 Alarms (Basic Panels)

Configuring alarms (Basic Panels)

Configure alarms

Several steps are needed to configure alarms, such as operational messages, error alarms, and acknowledgement.

- Step 1: Create tags
- Step 2: Configure alarms
- Step 3: Configure acknowledgment

You can find additional information in the section:

AUTOHOTSPOT

Distinctive features when configuring alarms

If you are configuring connections of HMI devices to PLCs of other manufacturers, note the following distinctive features when configuring:

- Data types of the tags
- Addressing of tags
- How the bit positions are counted

Data types

For connections with a SIMATIC communication driver, the following data types are supported:

PLC	Permitted data types	
	Discrete alarms	Analog alarms
SIMATIC S7 PLCs	WORD, INT	BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, TIMER

How the bit positions are counted

For connections with a SIMATIC communication driver, the following counting method applies:

How the bit positions are counted	Byte 0								Byte 1							
	Most significant byte								Least significant byte							
In SIMATIC S7 PLCs	7							0	7							0
In WinCC you configure:	15							8	7							0

See also

Acknowledgment of alarms (Page 373)

Acknowledgment of alarms (Basic Panels)

Procedure

Create suitable tags on the PLC to acknowledge an error alarm. You assign these tags to an alarm in the "Bit messages" editor. You make the assignment in "Properties > Acknowledgment".

Distinction in terms of acknowledgment:

- Acknowledgment by the PLC
- Acknowledgment on the HMI device

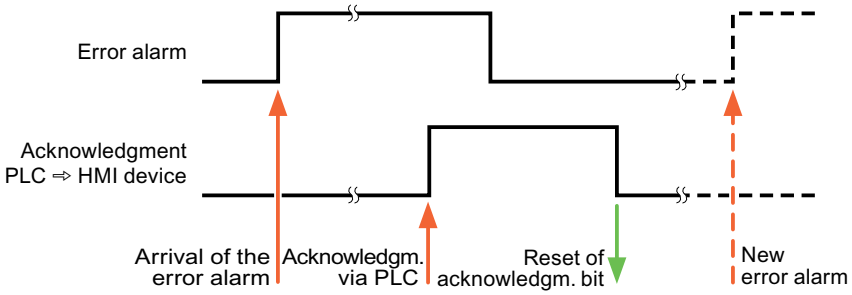
Acknowledgment by the PLC

In "PLC acknowledgment tag", configure the tag or array tag and the bit number that the HMI device uses to identify a PLC acknowledgment.

A bit set in the tag triggers acknowledgment of the assigned error alarm bit at the HMI device. This tag bit returns a function similar to acknowledgment on the HMI device which is triggered by pressing the "ACK" button, for example.

The acknowledgment bit must be located in the same tag as the bit for the error alarm.

Reset the acknowledgment bit before setting the bit in the alarm area again. The figure below shows the pulse diagram.



Acknowledgment on the HMI device

In the "HMI acknowledgment tag" area, configure the tag or array tag as well as the bit number that the HMI device writes to the PLC after acknowledgment. Make sure when you use an array tag that it is not longer than 6 words.

To always create a signal change when setting an assigned acknowledgment bit of a discrete alarm that must be acknowledged, the HMI device will reset the acknowledgment bit assigned to the alarm as soon as it detects an alarm subject to acknowledgment and write the acknowledgment tag in the PLC. There will be a certain delay between detecting the message

and writing the acknowledgment tag in the PLC because the HMI device has to process the operations.

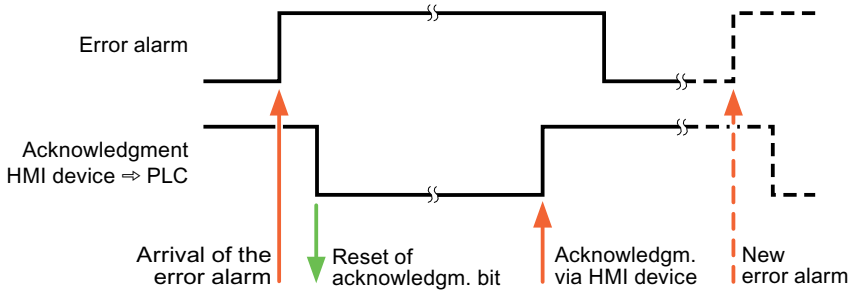
If a discrete alarm subject to acknowledgment is acknowledged by the HMI device, then the corresponding bit in the assigned acknowledgment tag will be set. The entire acknowledgment tag is then written to the PLC by the HMI device. This allows the PLC to recognize that a certain alarm message has been acknowledged at the HMI device.

Note

All alarm bits acknowledged since the last Runtime start will remain in the acknowledgment tag until a new incoming of the respective discrete alarms is detected.

This area should only be read by the PLC because the entire section of the HMI device will be overwritten once the next acknowledgment tag is written.

The figure below shows the pulse diagram.



See also

Configuring alarms (Page 372)

1.10.4.4 LED mapping (Basic Panels)

Function

Keyboard devices have LEDs in the function keys. These LEDs can be activated from the PLC. Thus, it is possible, for example, to signal to the user which key he should press in a given situation by lighting up an LED.

Note

The LED function cannot be configured for Basic Panels.

Requirements

In order to activate an LED, an LED tag or an array tag must be set up in the PLC and specified as an LED tag during configuration.

LED assignment

The assignment of the individual LEDs to the bits in the LED tags is specified when the function keys are configured. In this process, the "LED tag" and the assigned "Bit" are specified for each function key in the "General" group of the properties window.

The "Bit" bit number designates the first of two consecutive bits that control the following LED statuses.

Bit n+ 1	Bit n	LED function	
		All Mobile Panels, all Operator Panels, all Multi Panels, all Comfort Panels	Panel PCs
0	0	Off	Off
0	1	Rapid flashing	Flashing
1	0	Slow flashing	Flashing
1	1	On permanently	On permanently

1.10.5 Performance features of communication (Basic Panels)

1.10.5.1 Device dependencies of SIMATIC ET 200 CPU (Basic Panels)

Communication with SIMATIC ET 200 CPU controller

If you use devices from an earlier version of the TIA Portal with TIA Portal V14, it may not be possible to configure integrated connections to certain HMI devices.

Basic Panels V11.0

HMI devices	SIMATIC ET 200 CPU
KP300 Basic	Yes
KP400 Basic	Yes
KTP400 Basic PN	Yes
KTP600 Basic DP	Yes
KTP600 Basic PN	Yes
KTP1000 Basic DP	Yes
KTP1000 Basic PN	Yes
TP1500 Basic PN	Yes

Basic Panels V12.0

HMI devices	SIMATIC ET 200 CPU
KP300 Basic	Yes
KP400 Basic	Yes
KTP400 Basic PN	Yes
KTP600 Basic DP	Yes
KTP600 Basic PN	Yes
KTP1000 Basic DP	Yes
KTP1000 Basic PN	Yes
TP1500 Basic PN	Yes

Basic Panels V13.0

HMI devices	SIMATIC ET 200 CPU
KTP400 Basic PN	Yes
KTP700 Basic PN / DP	Yes
KTP900 Basic PN	Yes
KTP1200 Basic PN / DP	Yes

Basic Panels V13.0.1

HMI devices	SIMATIC ET 200 CPU
KTP400 Basic PN	Yes
KTP700 Basic PN / DP	Yes
KTP900 Basic PN	Yes
KTP1200 Basic PN / DP	Yes

Basic Panels V14.0

HMI devices	SIMATIC ET 200 CPU
KTP400 Basic PN	Yes
KTP700 Basic PN	Yes
KTP700 Basic DP	Yes
KTP900 Basic PN	Yes
KTP1200 Basic PN	Yes
KTP1200 Basic DP	Yes

Panels V11.0

HMI devices	SIMATIC ET 200 CPU
OP 73	Yes
OP 77A	Yes
OP 77B	Yes
TP 177A	Yes
TP 177A Portrait	Yes
TP 177B 4"	Yes
TP 177B 6" mono	Yes
TP 177B 6"	Yes
OP 177B 6" mono	Yes
OP 177B 6"	Yes
TP 277 6"	Yes
OP 277 6"	Yes

Multi Panels V11.0

HMI devices	SIMATIC ET 200 CPU
MP 177 6" Touch	Yes
MP 277 8" Key	Yes
MP 277 10" Key	Yes
MP 277 10" Touch	Yes
MP 377 12" Key	Yes
MP 377 12" Touch	Yes
MP 377 15" Touch	Yes
MP 377 19" Touch	Yes

Multi Panels V12.0

HMI devices	SIMATIC ET 200 CPU
MP 177 6" Touch	Yes
MP 277 8" Key	Yes
MP 277 10" Key	Yes
MP 277 10" Touch	Yes
MP 377 12" Key	Yes
MP 377 12" Touch	Yes
MP 377 15" Touch	Yes
MP 377 19" Touch	Yes

Mobile Panels V11.0

HMI devices	SIMATIC ET 200 CPU
Mobile Panel 177 6" DP	Yes
Mobile Panel 177 6" PN	Yes
Mobile Panel 277 8"	Yes
Mobile Panel 277 8" IWLAN V2	Yes
Mobile Panel 277F 8" IWLAN V2	Yes
Mobile Panel 277F 8" IWLAN V2 (RFID tag)	Yes
Mobile Panel 277 10"	Yes

Mobile Panels V12.0

HMI devices	SIMATIC ET 200 CPU
Mobile Panel 177 6" DP	Yes
Mobile Panel 177 6" PN	Yes
Mobile Panel 277 8"	Yes
Mobile Panel 277 8" IWLAN V2	Yes
Mobile Panel 277F 8" IWLAN V2	Yes
Mobile Panel 277F 8" IWLAN V2 (RFID tag)	Yes
Mobile Panel 277 10"	Yes

Mobile Panels V13.0.1

HMI devices	SIMATIC ET 200 CPU
KTP700 Mobile	Yes
KTP700F Mobile	Yes
KTP900 Mobile	Yes
KTP900F Mobile	Yes

Mobile Panels V14.0

HMI devices	SIMATIC ET 200 CPU
KTP700 Mobile	Yes
KTP700F Mobile	Yes
KTP900 Mobile	Yes
KTP900F Mobile	Yes
KTP400F Mobile	Yes

Comfort Panels V11.0

HMI devices	SIMATIC ET 200 CPU
KP400 Comfort	Yes
KTP400 Comfort	Yes
KTP400 Comfort Portrait	Yes
KP700 Comfort	Yes
TP700 Comfort	Yes
TP700 Comfort Portrait	Yes
KP900 Comfort	Yes
TP900 Comfort	Yes
TP900 Comfort Portrait	Yes
KP1200 Comfort	Yes
TP1200 Comfort	Yes
TP1200 Comfort Portrait	Yes
KP1500 Comfort	Yes
TP1500 Comfort	Yes
TP1500 Comfort Portrait	Yes
TP1900 Comfort	Yes
TP1900 Comfort Portrait	Yes
TP2200 Comfort	Yes
TP2200 Comfort Portrait	Yes

Comfort Panels V12.0

HMI devices	SIMATIC ET 200 CPU
KP400 Comfort	Yes
KTP400 Comfort	Yes
KTP400 Comfort Portrait	Yes
KP700 Comfort	Yes
TP700 Comfort	Yes
TP700 Comfort Portrait	Yes
KP900 Comfort	Yes
TP900 Comfort	Yes
TP900 Comfort Portrait	Yes
KP1200 Comfort	Yes
TP1200 Comfort	Yes
TP1200 Comfort Portrait	Yes
KP1500 Comfort	Yes
TP1500 Comfort	Yes
TP1500 Comfort Portrait	Yes
TP1900 Comfort	Yes

HMI devices	SIMATIC ET 200 CPU
TP1900 Comfort Portrait	Yes
TP2200 Comfort	Yes
TP2200 Comfort Portrait	Yes

Comfort Panels V13.0

HMI devices	SIMATIC ET 200 CPU
KP400 Comfort	Yes
KTP400 Comfort	Yes
KTP400 Comfort Portrait	Yes
KP700 Comfort	Yes
TP700 Comfort	Yes
TP700 Comfort Portrait	Yes
KP900 Comfort	Yes
TP900 Comfort	Yes
TP900 Comfort Portrait	Yes
KP1200 Comfort	Yes
TP1200 Comfort	Yes
TP1200 Comfort Portrait	Yes
KP1500 Comfort	Yes
TP1500 Comfort	Yes
TP1500 Comfort Portrait	Yes
TP1900 Comfort	Yes
TP1900 Comfort Portrait	Yes
TP2200 Comfort	Yes
TP2200 Comfort Portrait	Yes

Comfort Panels V13.0.1

HMI devices	SIMATIC ET 200 CPU
KP400 Comfort	Yes
KTP400 Comfort	Yes
KTP400 Comfort Portrait	Yes
KP700 Comfort	Yes
TP700 Comfort	Yes
TP700 Comfort Portrait	Yes
KP900 Comfort	Yes
TP900 Comfort	Yes
TP900 Comfort Portrait	Yes
KP1200 Comfort	Yes
TP1200 Comfort	Yes

HMI devices	SIMATIC ET 200 CPU
TP1200 Comfort Portrait	Yes
KP1500 Comfort	Yes
TP1500 Comfort	Yes
TP1500 Comfort Portrait	Yes
TP1900 Comfort	Yes
TP1900 Comfort Portrait	Yes
TP2200 Comfort	Yes
TP2200 Comfort Portrait	Yes

Comfort Panels V14.0

HMI devices	SIMATIC ET 200 CPU
KP400 Comfort	Yes
KTP400 Comfort	Yes
KTP400 Comfort Portrait	Yes
KP700 Comfort	Yes
TP700 Comfort	Yes
TP700 Comfort Portrait	Yes
TP700 Comfort Outdoor	Yes
TP700 Comfort Outdoor Portrait	Yes
KP900 Comfort	Yes
TP900 Comfort	Yes
TP900 Comfort Portrait	Yes
KP1200 Comfort	Yes
TP1200 Comfort	Yes
TP1200 Comfort Portrait	Yes
KP1500 Comfort	Yes
TP1500 Comfort	Yes
TP1500 Comfort Portrait	Yes
TP1500 Comfort Outdoor	Yes
TP1500 Comfort Outdoor Portrait	Yes
TP1900 Comfort	Yes
TP1900 Comfort Portrait	Yes
TP2200 Comfort	Yes
TP2200 Comfort Portrait	Yes

Runtime V11.0

HMI devices	SIMATIC ET 200 CPU
WinCC RT Advanced	Yes
WinCC RT Professional	Yes

Runtime V12.0

HMI devices	SIMATIC ET 200 CPU
WinCC RT Advanced	Yes
WinCC RT Professional	Yes

Runtime V13.0

HMI devices	SIMATIC ET 200 CPU
WinCC RT Advanced	Yes
WinCC RT Professional	Yes

Runtime V13.0.1

HMI devices	SIMATIC ET 200 CPU
WinCC RT Advanced	Yes
WinCC RT Professional	Yes

Runtime V14.0

HMI devices	SIMATIC ET 200 CPU
WinCC RT Advanced	Yes
WinCC RT Professional	Yes

See also

Communication with SIMATIC ET 200 CPU (Page 331)
Valid data types for SIMATIC ET 200 CPU (Page 382)

1.10.5.2 Valid data types for SIMATIC ET 200 CPU (Basic Panels)

Valid data types for connections with SIMATIC ET 200 CPU

The table lists the data types that can be used when configuring tags and area pointers.

Data type	Length
BOOL	1 bit
SINT	1 byte
INT	2 bytes
DINT	4 bytes
USINT	1 byte
UINT	2 bytes

Data type	Length	
UDINT	4 bytes	
REAL	4 bytes	
LREAL	8 bytes	
TIME	4 bytes	
DATE	2 bytes	
TIME_OF_DAY, TOD	4 bytes	
STRING	(2+n) bytes, n = 0 to 254	
WSTRING	(4+2*n) bytes, n = 0 to 254	Basic Panels
	(4+2*n) bytes, n = 0 to 4094	Panels, RT Advanced
	(4+2*n) bytes, n = 0 to 65534	RT Professional
CHAR	1 byte	
WCHAR	2 bytes	RT Professional
Array	(Element number * data type length) bytes [1]	
BYTE	1 byte	
WORD	2 bytes	
DWORD	4 bytes	
Date_And_Time	8 bytes	
DTL	8 bytes	
LDT	8 bytes	

[1] Example of "Length of an array": The length is 400 bytes (100 * 4) for 100 elements of the data type REAL

See also

Communication with SIMATIC ET 200 CPU (Page 331)
Device dependencies of SIMATIC ET 200 CPU (Page 375)

1.10.6 Creating connections in the "Connections" editor (Basic Panels)

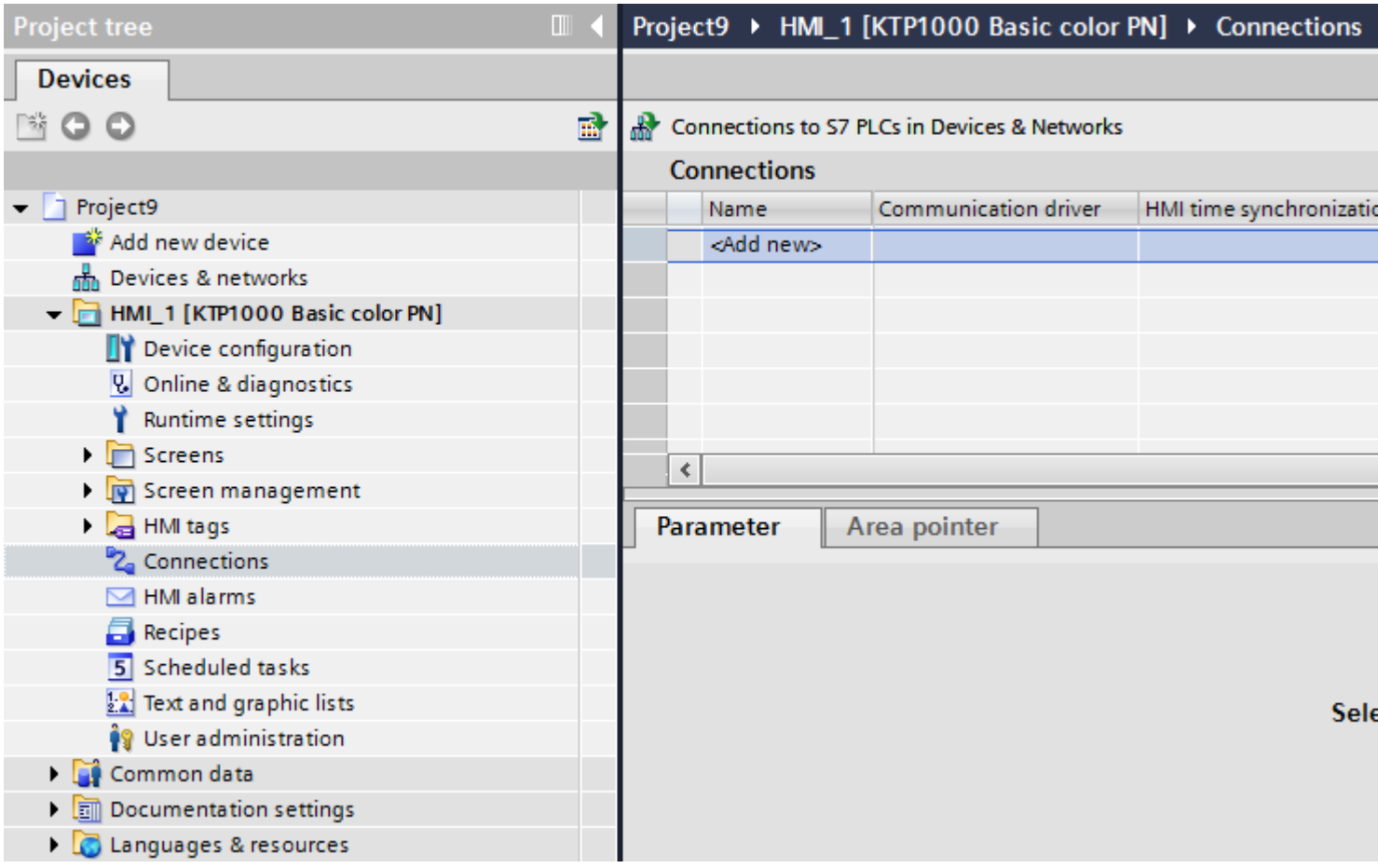
1.10.6.1 Creating a PROFINET connection (Basic Panels)

Requirements

- A project is open.
- An HMI device with a PROFINET interface has been created.

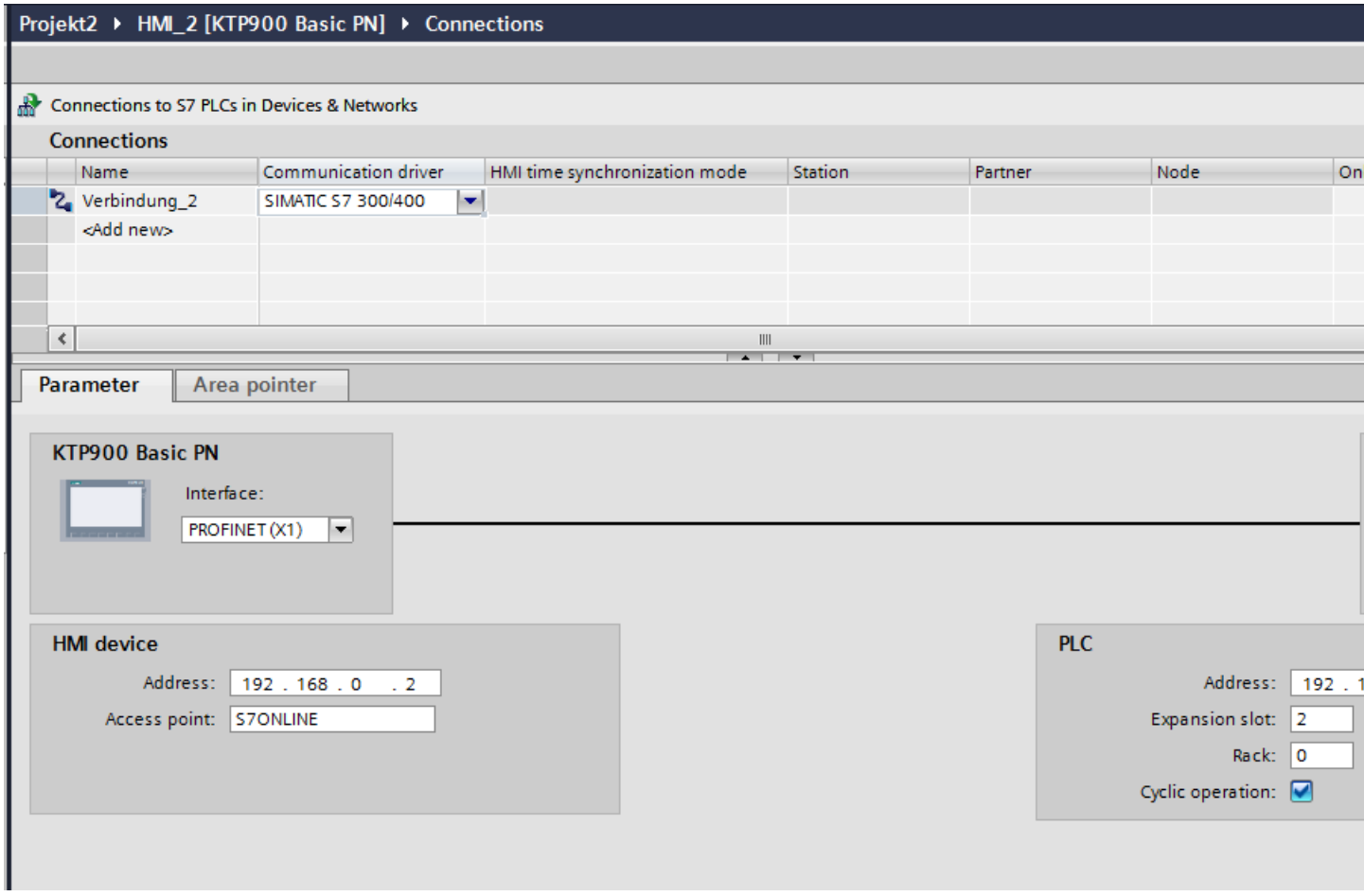
Procedure

1. Open the "Connections" editor of the HMI device.
2. Double-click "<Add>".



3. In the "Communication drivers" column, select the "SIMATIC S7 300/400" driver.
4. Click the name of the connection.

5. Select a PROFINET interface of the HMI device in the Inspector window under "Parameters > Interface".



6. Set the IP addresses of the communication partners in the Inspector window:
   - HMI device: "Parameters > HMI device > Address"
   - PLC: "Parameters > PLC > Address"

See also

Communication via PROFINET (Page 331)

Communication with SIMATIC ET 200 CPU (Page 331)

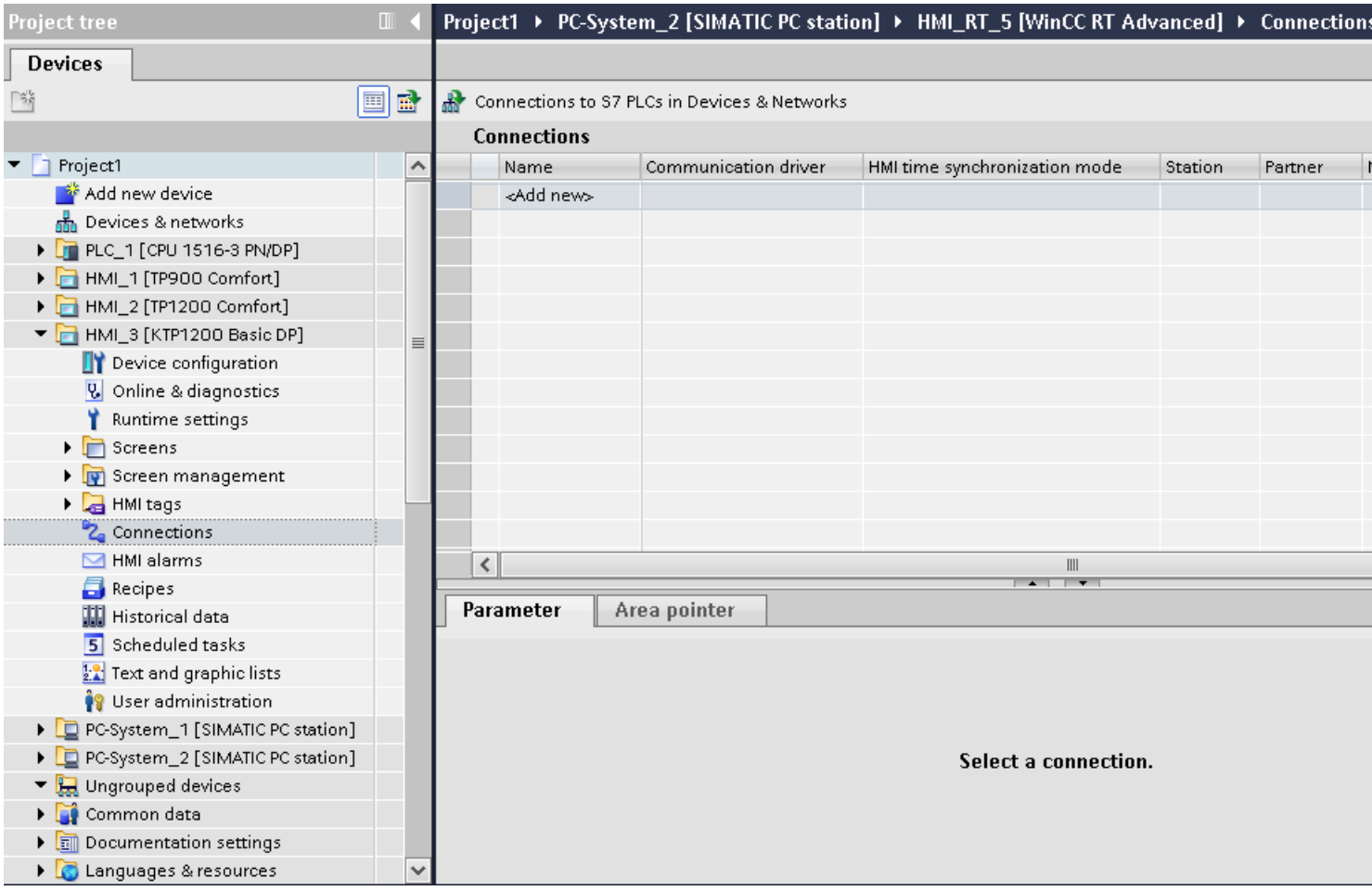
1.10.6.2 Creating a PROFIBUS DP connection (Basic Panels)

Requirements

- A project is open.
- An HMI device with a PROFIBUS interface has been created.

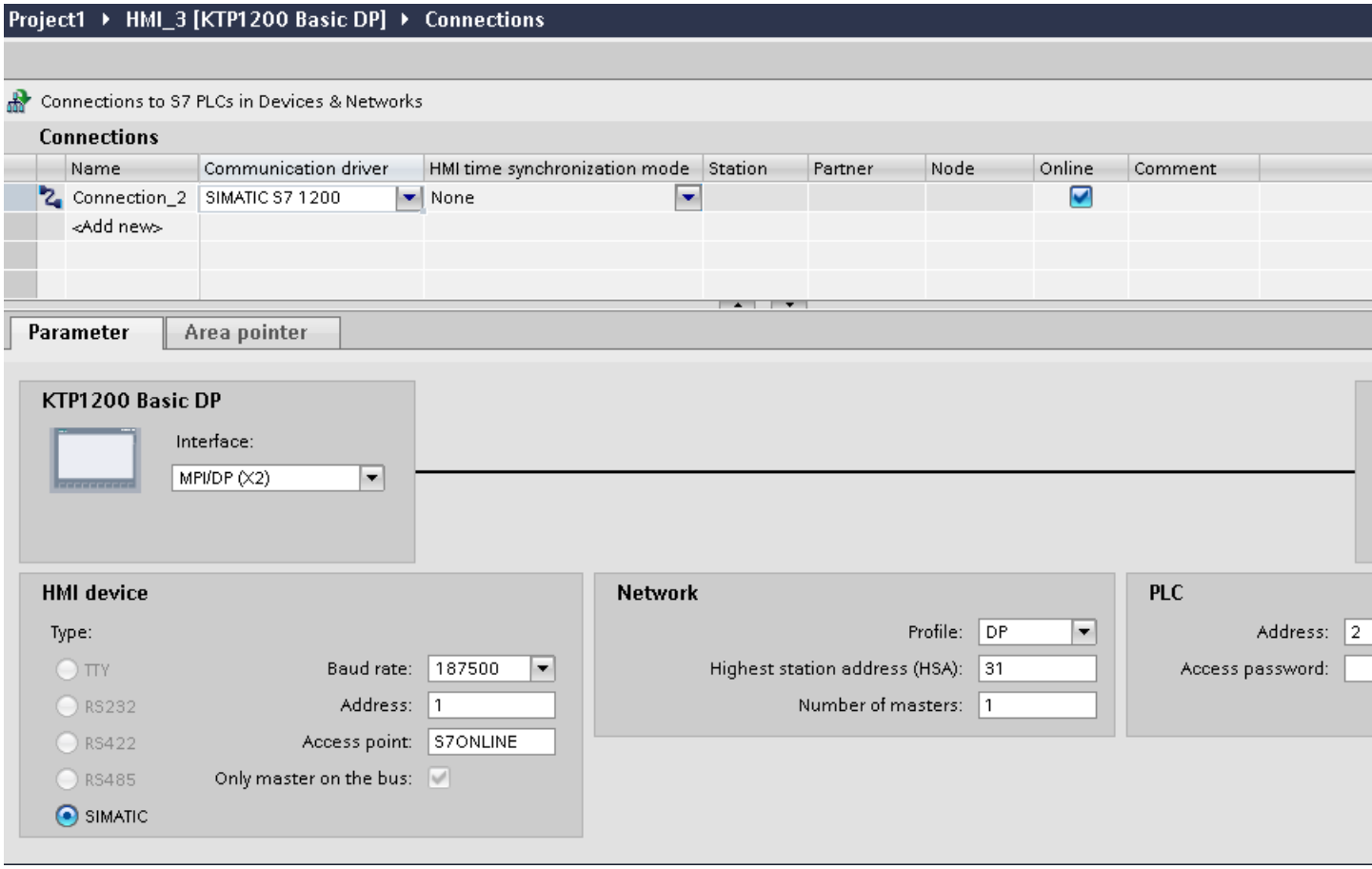
Procedure

1. Open the "Connections" editor of the HMI device.
2. Double-click "<Add>".



3. In the "Communication drivers" column, select the "SIMATIC S7 1200" driver.
4. Click the name of the connection.
5. Select the "MPI/DP" interface in the Inspector window under "Parameters".

6. Select the "DP" profile in the Inspector window under "Parameters > Network".



7. Set the addresses of the communication partners in the Inspector window:
   - HMI device: "Parameters > HMI device > Address"
   - PLC: "Parameters > PLC > Address"

See also

Communication with SIMATIC ET 200 CPU (Page 331)
Communication via PROFIBUS (Page 346)

1.10.6.3 Connection parameters (Basic Panels)

PROFIBUS parameters (Basic Panels)

Parameters for the HMI device

You assign the parameters for the HMI device in the network once under "HMI device". The change applies to each communication partner.

- "Type"
  Specifies the physical connection used.
- "Interface"
  For "Interface", you select the HMI device interface via which the HMI device is connected to the PROFIBUS network.
- "Baud rate"
  For "Baud rate", you set the transmission speed of the data in the network. The baud rate is determined by the slowest HMI device connected to the network. The setting must be identical throughout the network.

  Note

  If you set a baud rate of 1.5 Mbaud for OP 73 or OP 77A, the highest station address must be less than or equal to 63.

- "Address"
  You set the PROFIBUS address of the HMI device under "Address". The PROFIBUS address must be unique in the PROFIBUS network.
- "Only master on bus"
  Disables an additional safety feature against bus faults when the HMI device is connected to the network. A passive station (slave) can only send data if it is requested to do so by an active station (master).
  In S7-200, you must set an HMI device as the master.
- "Access point"
  The access point defines a logical device name through which the communication partner can be reached.

Parameters for the network

Under "Network", you set the parameters for the PROFIBUS network to which the HMI device is linked.

- "Profile"
  For "Profile", you select the network profile that is used in the network. In "Profile", set "DP", "Universal", or "Standard". The setting must be identical throughout the network.
- "Highest address"
  For "Highest station address", set the highest station address. The highest station address must be greater than or equal to the highest actual PROFIBUS address. The setting must be identical throughout the network.

  Note

  If you set a baud rate of 1.5 Mbaud in the OP 73 or the OP 77A, the highest station address must be less than or equal 63.

- "Number of masters"
  For "Number of masters", set the number of masters in the PROFIBUS network. This information is necessary to correctly calculate the bus parameters.

Parameters for the PLC

Under "PLC", you address the S7 module with which the HMI device will exchange data. Assign a name for the connection for each communication partner.

- "Address"
  For "Address", set the PROFIBUS address of the S7 module (CPU, FM, or CP) to which the HMI device is connected.
- "Access password"
  Enter a password in the "Access password" field. This password must match the one you saved to the PLC.

  Note

  You only need a password if you have set "Complete protection" at the PLC.

  No connection is set up to the PLC if the "Complete protection" security level is stored on the PLC and you do not enter a password.

See also

Creating a PROFIBUS DP connection (Page 385)

Ethernet parameters (Basic Panels)

Parameters for the HMI device

You set the parameters for the HMI device in the network under "HMI device".. The changes are not transferred automatically to the HMI device. You must change the settings in the Control Panel of the HMI device.

- "Interface"
  If you are directly connected to the HMI device during configuration, you can set up the IP address of the HMI device in WinCC.

  Note

  The IP address in the Control Panel will be overwritten upon subsequent loading if you have already set up the IP address in the HMI device control panel.

  The IP address already set up in the Control Panel will be retained upon subsequent loading if you activate "Set IP address using a different method".

  The IP address is transferred to the HMI device during project transfer.
  To set up the IP address of the HMI device:

  – Click the HMI device.
  – Open the "Device configuration" editor.
  – Click the Ethernet interface.
  – Assign the IP address in the inspector window under:
    "General > PROFINET interface > Ethernet addresses"
- "Address"
  You assign the IP address of the HMI device in the "Address" area.
  When you transfer the WinCC project to the HMI device, this IP address is set up directly in the HMI device.
- "Access point"
  Specifies the access point for the PG/PC interface that can be used to reach the communication partner.

Parameters for the PLC

Under "PLC", you address the S7 module with which the HMI device will exchange data. Assign a name for the connection for each communication partner.

- "Address"
  Under "Address", set the IP address of the S7 module to which the HMI device is connected.
- "Access password"
  Enter a password in the "Access password" field. This password must match the one you saved to the PLC.

Note

You only need a password if you have set "Complete protection" at the PLC.

No connection is set up to the PLC if the "Complete protection" security level is stored on the PLC and you do not enter a password.

See also

Creating a PROFINET connection (Page 383)

1.10.7 Configuring time synchronization (Basic Panels)

1.10.7.1 Time synchronization (Basic Panels)

Introduction

To have the same time of day throughout the plant, you can synchronize the time on various plant components using time synchronization. WinCC time synchronization is operated as a master-slave system.

One system component must be a clock for all components of a plant to work with identical time. The component functioning as the clock is referred to as the time master. The components that receive the time are time slaves.

Properties of time synchronization

- The HMI device can define the time as master or can accept the time of the PLC as slave.
- In "master mode", the time is synchronized at each connection setup.
- In "slave mode", the time is synchronized at each connection setup and than at cyclic intervals of 10 minutes.
- The first time synchronization is performed on the HMI device immediately after the start of runtime.
- Time synchronization is only performed on the HMI device during operation of runtime.

See also

Time synchronization restrictions (Page 392)
Configuring time synchronization for integrated connections (Page 393)
Configuring time synchronization for non-integrated connections (Page 394)

1.10.7.2 Time synchronization restrictions (Basic Panels)

Approved HMI devices

You can configure the time synchronization between a SIMATIC S7-1200 or SIMATIC S7-1500 and an HMI device with the following HMI devices:

Device	Operating system
Basic Panels	-
TP177 4"	Windows CE 5.0
Multi Panel 177	Windows CE 5.0
Multi Panel 277	Windows CE 5.0
Multi Panel 377	Windows CE 5.0
Mobile Panel 277	Windows CE 5.0
Mobile 277 IWLAN V2	Windows CE 5.0
Comfort Panels	Windows CE 6.0
PC systems with WinCC RT Advanced	Microsoft Windows

Configuration restrictions

- If an HMI device has several connections to SIMATIC S7-1200 or SIMATIC S7-1500, you can only configure one connection as "slave".
- If you have enabled time synchronization for the HMI device as "slave", you can no longer use the global area pointer "Date/time PLC".
- An HMI device can only request the time from a PLC with "Complete protection" security type configuration if the correct "Access password" is configured.
  Configure the "Access password" for communication with a PLC with "Complete protection" security type in the "Connections" editor of the HMI device.
  This "Access password" must match the password configured on the PLC. The PLC password is assigned in the PLC properties at: "General > Security"
- Basic Panels can only be configured as "Slave".
- If you use Basic Panels for the configuration, it is not possible to use time synchronization via NTP and the "Date/time PLC" area pointer simultaneously.
- Time synchronization with SIMATIC S7-1200 (V1.0) controllers is not possible.
- Time synchronization between the HMI device TP177 4" and SIMATIC S7-1200 (V4.0) controllers is not possible.
- Time synchronization between the HMI device TP177 4" and SIMATIC S7-1500 is not possible.

See also

Time synchronization (Page 391)

Configuring time synchronization for integrated connections (Page 393)

Configuring time synchronization for non-integrated connections (Page 394)

1.10.7.3 Configuring time synchronization for integrated connections

Introduction

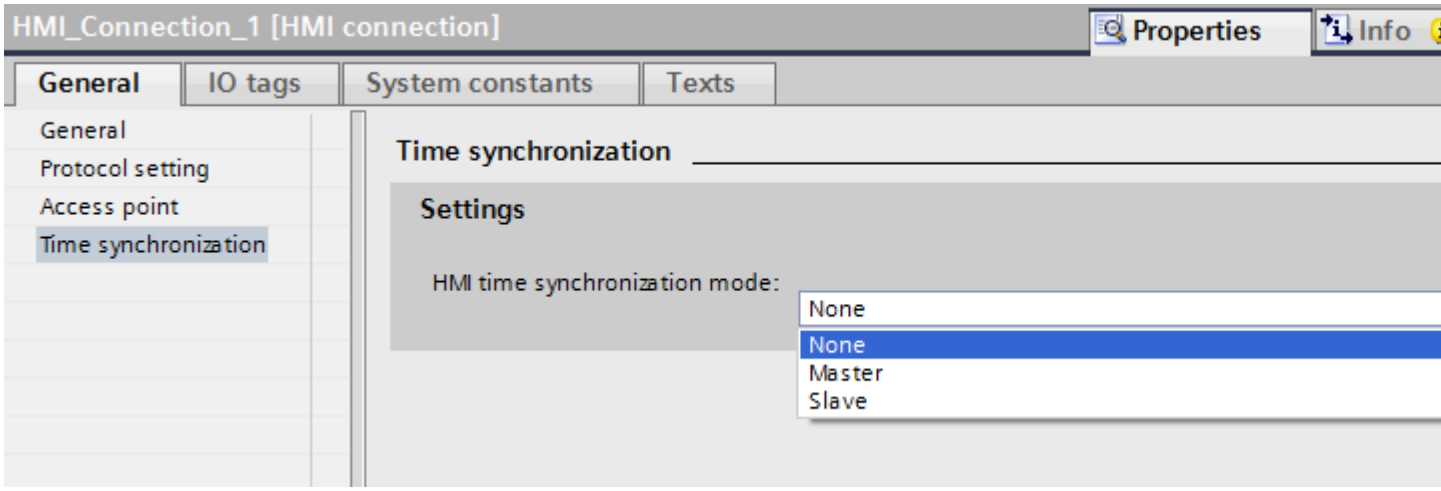
You configure time synchronization for an integrated connection in the "Devices & Networks" editor.

Requirements

- An HMI connection between an HMI device and a SIMATIC S7 1200 or SIMATIC S7 1500 has been configured.
- The HMI device must support the "time synchronization" function.
- The "Devices & Networks" editor is open.

Procedure

1. Click the line of the HMI connection in the "Devices & networks" editor.
2. Select the following in the inspector window under "General > Time synchronization > Settings":
   - None: No time synchronization is used.
   - Master: The HMI device sets the time.
   - Slave: The PLC sets the time.



See also

Time synchronization (Page 391)

Time synchronization restrictions (Page 392)

Configuring time synchronization for non-integrated connections (Page 394)

1.10.7.4 Configuring time synchronization for non-integrated connections (Basic Panels)

Introduction

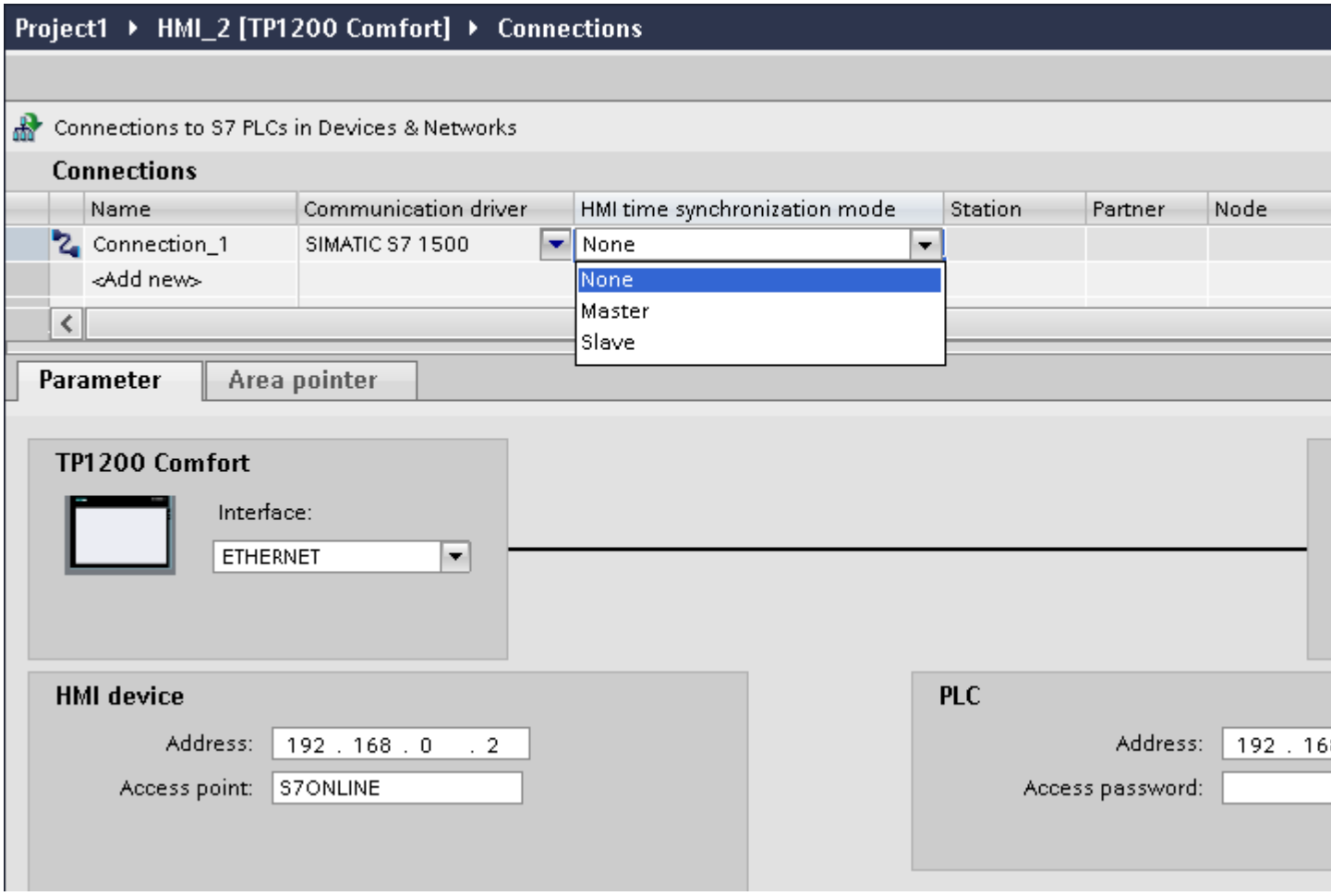
You configure time synchronization for a non-integrated connection in the "Connections" editor.

Requirements

- An HMI device which supports the "time synchronization" function has been created.
- "Connections" editor is open.

Procedure

1. Double-click "<Add>".
2. In the "Communication drivers" column, select the "SIMATIC S7 1500" PLC.
3. Select the following in the "HMI time synchronization mode" column:
   - None: No time synchronization is used.
   - Master: The HMI device sets the time.
   - Slave: The PLC sets the time.



See also

Time synchronization (Page 391)

Time synchronization restrictions (Page 392)

Configuring time synchronization for integrated connections (Page 393)

1.11 Communicating with SIMATIC S7 200 (Basic Panels)

1.11.1 Communication with SIMATIC S7 200 (Basic Panels)

Introduction

This section describes the communication between an HMI device and the SIMATIC S7 200 PLC.

You can configure the following communication channels for the SIMATIC S7 200 PLC:

- PROFINET and Ethernet
- PROFIBUS
- MPI
- PPI

HMI connection for communication

You configure connections between the HMI device and a SIMATIC S7 200 in the "Connections" editor of the HMI device.

See also

Creating a connection to SIMATIC S7 200 (Page 396)

1.11.2 Creating a connection to SIMATIC S7 200 (Basic Panels)

Introduction

You configure a connection to the SIMATIC S7 200 PLC in the "Connections" editor of the HMI device. The interfaces are named differently depending on the HMI device.

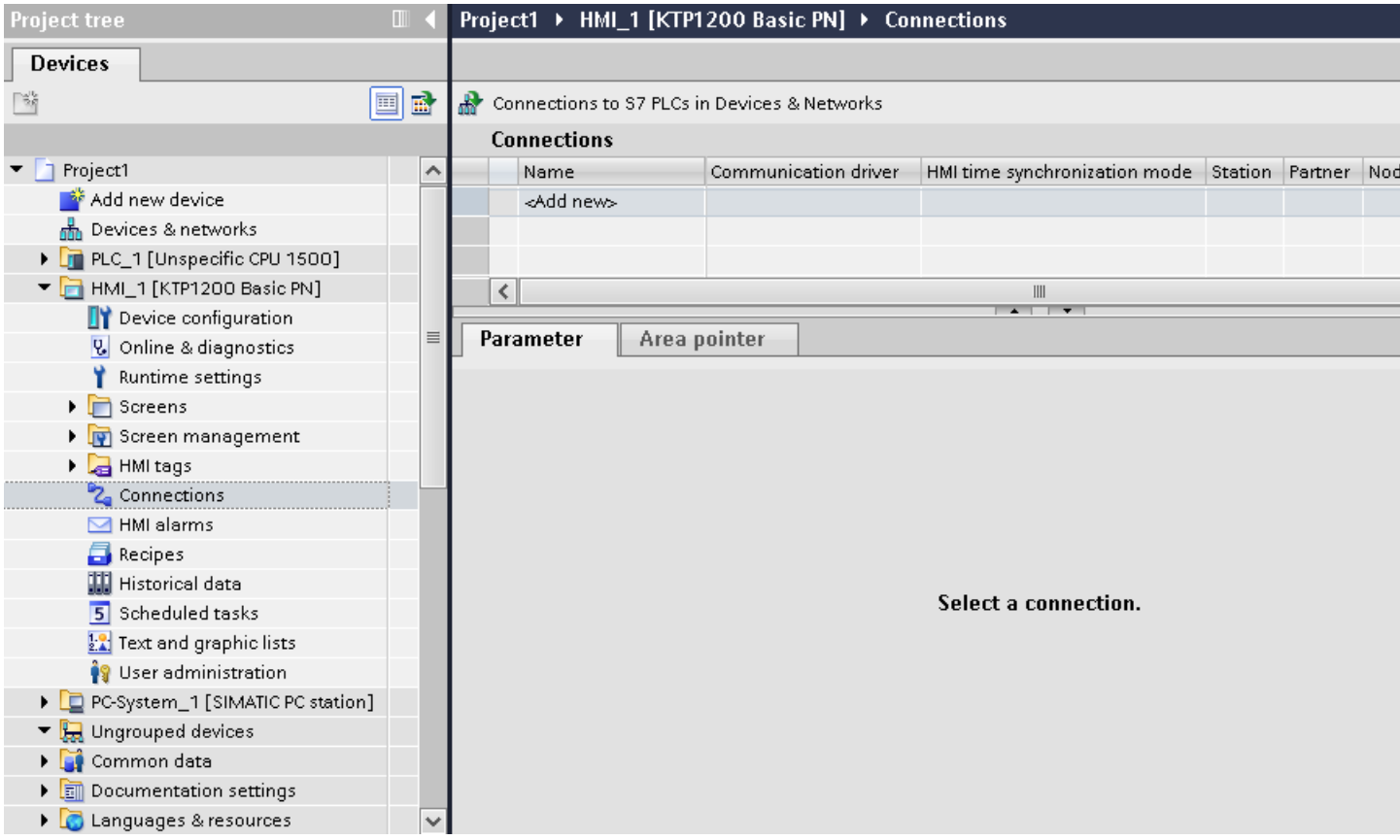
Requirements

- A project is open.
- An HMI device has been created.

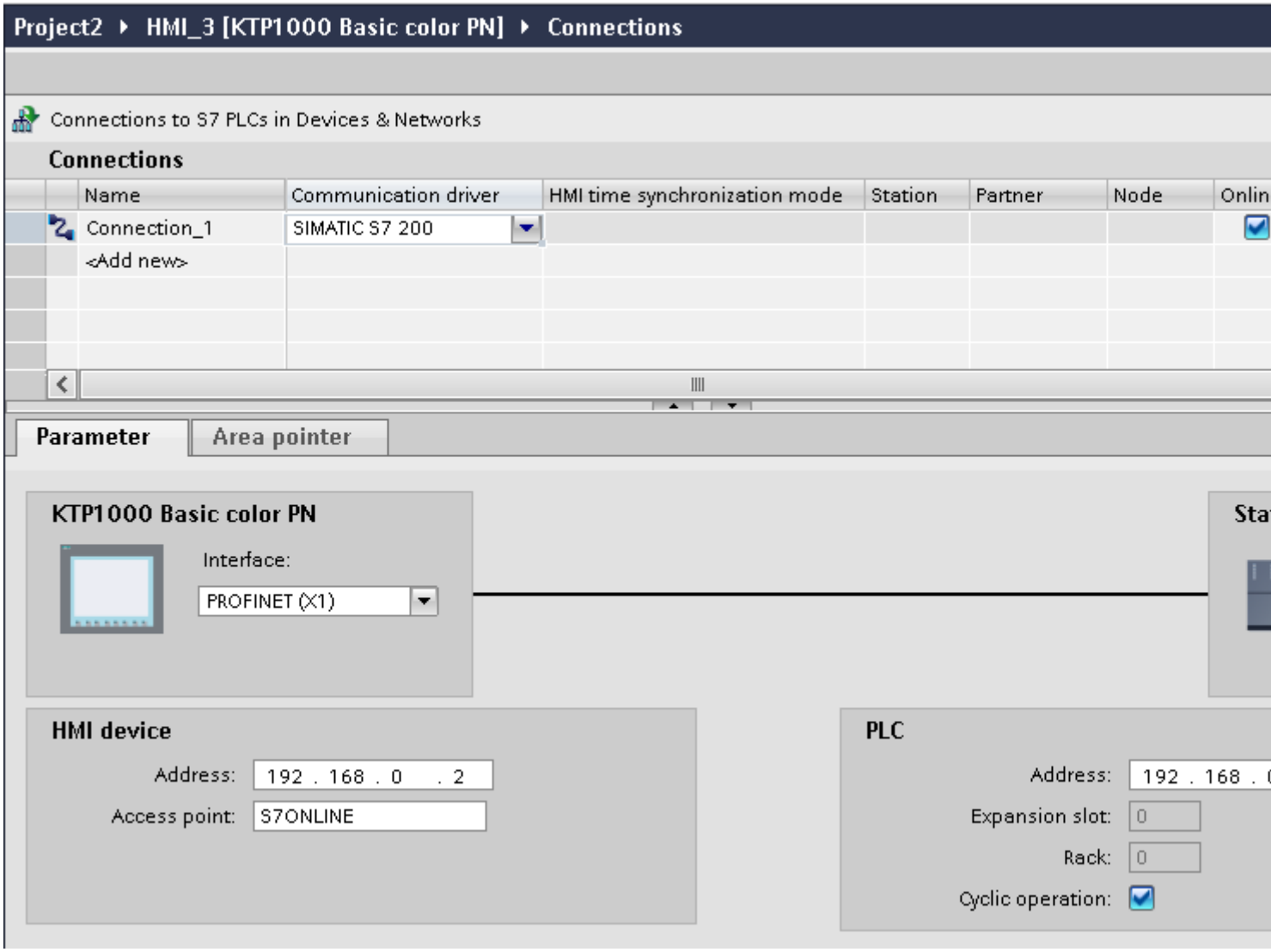
Procedure

1. Double-click the HMI device under "Devices" in the project tree.
2. Double-click the "Connections" item.

3. Double-click "<Add>" in the "Connections" editor.



4. In the "Communication drivers" column, select the "SIMATIC S7 200" driver.
5. Select all necessary connection parameters for the interface in the Inspector window under "Parameters".



See the "Parameters for the connection (Page 399)" section for more detailed information.

See also

Communication with SIMATIC S7 200 (Page 396)

Parameters for the connection (Page 399)

1.11.3 Parameters for the connection (Basic Panels)

1.11.3.1 Cyclic operation (Basic Panels)

Handling the "Cyclic operation" selection

If "Cyclic operation" is enabled, the HMI device sends a message frame to the CPU at the beginning of communication indicating that certain tags are required on a recurring basis.

The CPU then always transmits the data at the same cyclic interval. This saves the HMI device from having to output new requests for the data.

If cyclic operation is disabled, the HMI device sends a request whenever information is required.

Additional properties:

- Cyclic operation reduces data transmission load at the HMI device. The PLC resources are used to relieve load on the HMI device.
- The PLC only supports a certain number of cyclic services. The HMI device handles the operation if the PLC cannot provide any further resources for cyclic services.
- The HMI device generates the cycle if the PLC does not support the cyclic mode.
- Screen tags are not integrated in cyclic operation.
- Cyclic mode is only set up at the restart of Runtime.
- The HMI device transfers several jobs to the PLC if cyclic mode is enabled, depending on the PLC.
- The HMI device only transfers one job to the PLC if cyclic mode is disabled.

See also

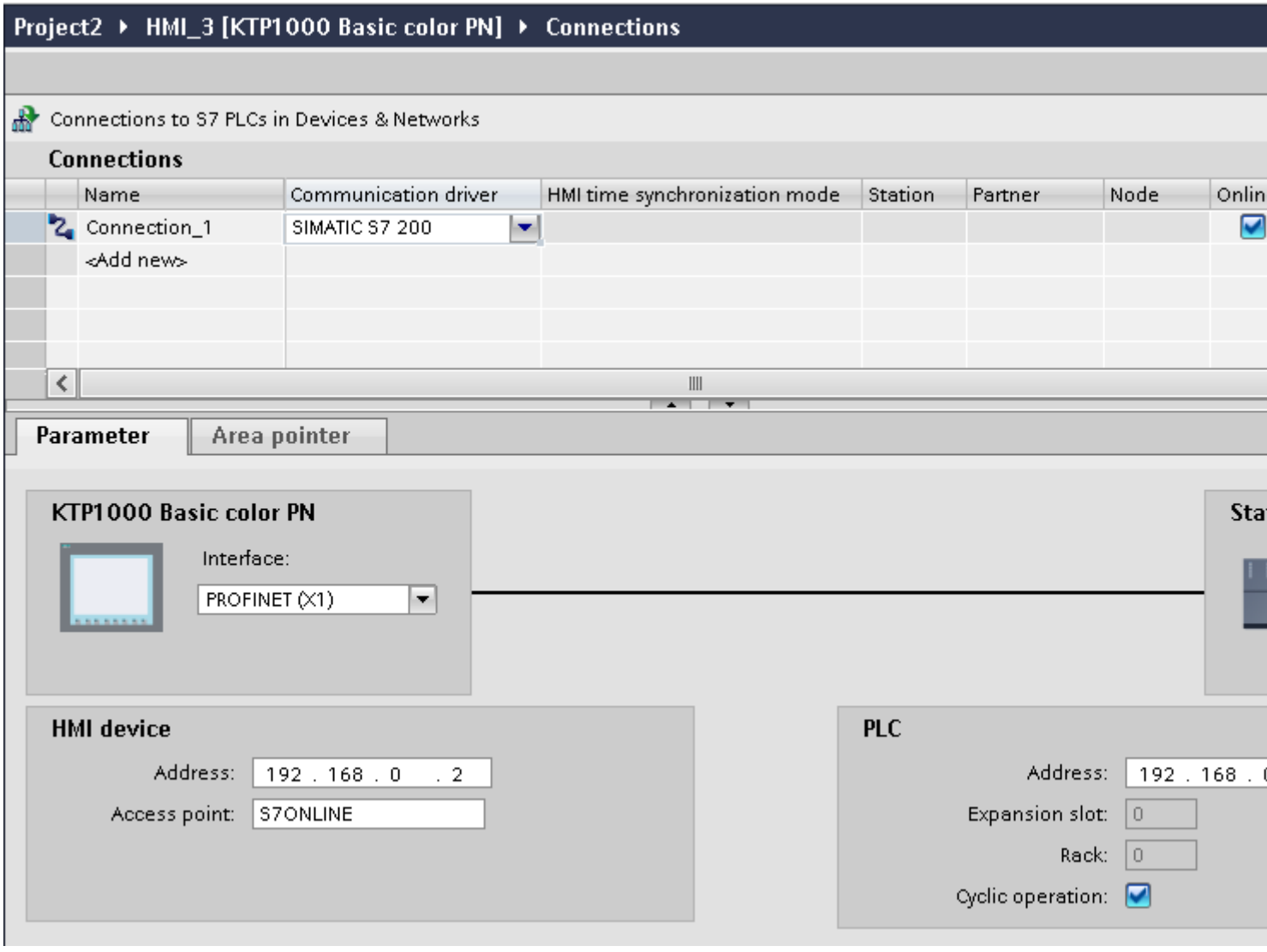
Communication with SIMATIC S7 200 (Page 396)

1.11.3.2 Parameters for the connection (SIMATIC S7 200) (Basic Panels)

Parameters to be set

To set the connection parameters, such as addresses and profiles, click the connection that you have created in the "Connections" editor.

The communication partners are displayed schematically in the Inspector window under "Parameters". The "HMI device" and "PLC" areas are available for assigning parameters according to the interface used.



See also

Communication with SIMATIC S7 200 (Page 396)

1.11.3.3 Ethernet parameters (Basic Panels)

Parameters for the HMI device

You set the parameters for the HMI device in the network under "HMI device".. The changes are not transferred automatically to the HMI device. You must change the settings in the Control Panel of the HMI device.

- "Interface"
  If you are directly connected to the HMI device during configuration, you can set up the IP address of the HMI device in WinCC.

  Note

  The IP address in the Control Panel will be overwritten upon subsequent loading if you have already set up the IP address in the HMI device control panel.

  The IP address already set up in the Control Panel will be retained upon subsequent loading if you activate "Set IP address using a different method".

  The IP address is transferred to the HMI device during project transfer.
  To set up the IP address of the HMI device:

  – Click the HMI device.
  – Open the "Device configuration" editor.
  – Click the Ethernet interface.
  – Assign the IP address in the inspector window under:
    "General > PROFINET interface > Ethernet addresses"

- "Address"
  You assign the IP address of the HMI device in the "Address" area.
  When you transfer the WinCC project to the HMI device, this IP address is set up directly in the HMI device.

- "Access point"
  The access point defines a logical device name through which the communication partner can be reached.

Parameters for the PLC

Under "PLC", you address the S7 module with which the HMI device will exchange data. Assign a name for the connection for each communication partner.

- "Address"
  Under "Address", set the IP address of the S7 module to which the HMI device is connected.

- "Expansion slot"
  Defines the number of the expansion slot of the CPU to be addressed.

- "Rack"
  Defines the rack number of the CPU to be addressed.
- "Cyclic operation"

  Note

  The setting "Cyclic operation" cannot be configured for the SIMATIC S7 1200 PLC.

  When cyclic operation is enabled, the PLC optimizes the data exchange between the HMI device and the PLC. This increases system performance.
  Disable cyclic mode if you are operating several HMI devices in parallel.

See also

Communication with SIMATIC S7 200 (Page 396)

1.11.3.4 PROFIBUS parameters (Basic Panels)

Parameters for the HMI device

You assign the parameters for the HMI device in the network once under "HMI device". The change applies to each communication partner.

- "Type"
  Specifies the physical connection used.
- "Interface"
  For "Interface", you select the HMI device interface via which the HMI device is connected to the PROFIBUS network.
- "Baud rate"
  For "Baud rate", you set the transmission speed of the data in the network. The baud rate is determined by the slowest HMI device connected to the network. The setting must be identical throughout the network.

  Note

  If you set a baud rate of 1.5 Mbaud for OP 73 or OP 77A, the highest station address must be less than or equal to 63.

- "Address"
  You set the PROFIBUS address of the HMI device under "Address". The PROFIBUS address must be unique in the PROFIBUS network.
- "Sole master on bus"
  Disables an additional safety feature against bus faults when the HMI device is connected to the network. A passive station (slave) can only send data if it is requested to do so by an active station (master).
  In S7 200, you must set an HMI device as the master.
- "Access point"
  The access point defines a logical device name through which the communication partner can be reached.

Parameters for the network

Under "Network", you set the parameters for the PROFIBUS network to which the HMI device is linked.

- "Profile"
  For "Profile", you select the network profile that is used in the network. In "Profile", set "DP", "Universal", or "Standard". The setting must be identical throughout the network.
- "Highest address"
  For "Highest station address", set the highest station address. The highest station address must be greater than or equal to the highest actual PROFIBUS address. The setting must be identical throughout the network.

  Note

  If you set a baud rate of 1.5 Mbaud in the OP 73 or the OP 77A, the highest station address must be less than or equal 63.

- "Number of masters"
  For "Number of masters", set the number of masters in the PROFIBUS network. This information is necessary to correctly calculate the bus parameters.

Parameters for the PLC

Under "PLC", you address the S7 module with which the HMI device will exchange data. Assign a name for the connection for each communication partner.

- "Address"
  For "Address", set the PROFIBUS address of the S7 module (CPU, FM, or CP) to which the HMI device is connected.
- "Cyclic operation"

  Note

  The setting "Cyclic operation" cannot be configured for the SIMATIC S7 1200 PLC.

  When cyclic operation is enabled, the PLC optimizes the data exchange between the HMI device and the PLC. This increases system performance. Disable cyclic mode if you are operating several HMI devices in parallel. This setting is not required for SIMATIC S7-200 PLCs.

See also

Communication with SIMATIC S7 200 (Page 396)

1.11.3.5 MPI parameters (Basic Panels)

Parameters for the HMI device

You assign the parameters for the HMI device in the network once under "HMI device". The change applies to each communication partner.

- "Type"
  Specifies the physical connection used.
- "Interface"
  For "Interface", you select the HMI device interface via which the HMI device is connected to the MPI network.
- "Baud rate"
  For "Baud rate", you set the transmission speed of the data in the network. The baud rate is determined by the slowest HMI device connected to the network. The setting must be identical throughout the network.

  Note

  If you set a baud rate of 1.5 Mbaud for OP 73 or OP 77A, the highest station address must be less than or equal to 63.

- "Address"
  For "Address", you set the MPI address of the HMI device. The MPI address must be unique throughout the MPI network.
- "Sole master on bus"
  Disables an additional safety feature against bus faults when the HMI device is connected to the network. A passive station (slave) can only send data if it is requested to do so by an active station (master). If you have connected only slaves to the HMI device, you must therefore disable the "Sole master on bus" safety feature.
  In S7 200, you must set an HMI device as the master.

Parameters for the network

Under "Network", you set the parameters for the MPI network to which the HMI device is linked.

- "Profile"
  For "Profile", you select the network profile that is used in the network. In "Profile", set "MPI". The setting must be identical throughout the network.
- "Highest address"
  For "Highest station address", set the highest station address. The highest station address must be greater than or equal to the highest actual MPI address. The setting must be identical throughout the network.
- "Number of masters"
  This setting is not required for MPI.

Parameters for the PLC

Under "PLC", you address the S7 module with which the HMI device will exchange data. Assign a name for the connection for each communication partner.

- "Address"
  For "Address", set the MPI address of the S7 module (CPU, FM, or CP) to which the HMI device is connected.
- "Cyclic mode"
  When cyclic mode is enabled, the PLC optimizes the data exchange between the HMI device and the PLC. This improves system performance. Disable cyclic mode if you are operating several HMI devices in parallel. This setting is not required for SIMATIC S7-200.

See also

Communication with SIMATIC S7 200 (Page 396)

1.11.3.6 PPI parameters (Basic Panels)

Parameters for the HMI device

You assign the parameters for the HMI device in the network once under "HMI device". The change applies to each communication partner.

- "Type"
  Specifies the physical connection used.
- "Interface"
  For "Interface", you select the HMI device interface via which the HMI device is connected to the PP network.
- "Baud rate"
  For "Baud rate", you set the transmission speed of the data in the network. The baud rate is determined by the slowest HMI device connected to the network. The setting must be identical throughout the network.

  Note

  If you set a baud rate of 1.5 Mbaud for OP 73 or OP 77A, the highest station address must be less than or equal to 63.

- "Address"
  For "Address", you set the PPI address of the HMI device. The PPI address must be unique throughout the PPI network.

- "Access point"
  For "Access point", you set the access point via which the communication partner is reached.
- "Sole master on bus"
  Disables an additional safety feature against bus faults when the HMI device is connected to the network. A passive station (slave) can only send data if it is requested to do so by an active station (master). If you have connected only slaves to the HMI device, you must therefore disable the "Sole master on bus" safety feature.
  In S7 200, you must set an HMI device as the master.

Parameters for the network

Under "Network", you set the parameters for the network to which the HMI device is linked.

- "Profile"
  For "Profile", you select the network profile that is used in the network. In "Profile", set "PPI". The setting must be identical throughout the network.
- "Highest address"
  For "Highest station address", set the highest station address. The highest station address must be greater than or equal to the highest actual MPI address. The setting must be identical throughout the network.
- "Number of masters"
  Set the number of the master on the network to "1".

Parameters for the PLC

Under "PLC", you address the S7 module with which the HMI device will exchange data. Assign a name for the connection for each communication partner.

- "Address"
  For "Address", set the PPI address of the S7 module to which the HMI device is connected.
- "Cyclic operation"
  This parameter is not required for communication via PPI.

See also

Communication with SIMATIC S7 200 (Page 396)

1.11.4 Data exchange (Basic Panels)

1.11.4.1 Data exchange using area pointers (Basic Panels)

General information on area pointers (Basic Panels)

Introduction

You use an area pointer to access a data area in the PLC. During communication, the PLC and the HMI device alternately access these data areas for read and write operations.

The PLC and the HMI device trigger defined interactions based on the evaluation of stored data.

Configuration of area pointers

Before you use an area pointer, you enable it under "Connections > Area pointer". You then assign the area pointer parameters.

You can find more detailed information on configuring area pointers in:

Configuring area pointers (Page 66)

See also

Configuring area pointers (Page 66)

"Screen number" area pointer (Page 407)

"Date/time" area pointer (Page 408)

"Date/time PLC" area pointer (Page 409)

"Coordination" area pointer (Page 411)

"Project ID" area pointer (Page 412)

"Job mailbox" area pointer (Page 413)

"Data record" area pointer (Page 415)

"Screen number" area pointer (Basic Panels)

Function

The HMI devices store information about the screen called up on the HMI device in the "Screen number" area pointer.

This allows the transfer of the current screen contents from the HMI device to the PLC. The PLC can trigger specific reactions such as the call of a different screen.

Use

Configure and enable the area pointer in "Communication > Connections" before you put it into use. You can create only **one** instance of the "Screen number" area pointer and only on **one** PLC.

The screen number is always transferred to the PLC when a new screen is activated or when the focus within a screen changes from one screen object to another.

Structure

The area pointer is a data area in the memory of the PLC with a fixed length of 5 words.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st word	Current screen type															
2nd word	Current screen number															
3rd word	Reserved															
4th word	Current field number															
5th word	Reserved															

- Current screen type
  "1" for root screen or
  "4" for permanent area
- Current screen number
  1 to 32767
- Current field number
  1 to 32767

See also

General information on area pointers (Page 407)

"Date/time" area pointer (Basic Panels)

Function

This area pointer is used to transfer the date and time from the HMI device to the PLC.

The PLC writes control job "41" to the job mailbox.

When it evaluates the control job, the HMI device writes its current date and the time in the data area configured in the "Date/time" area pointer. All definitions are coded in BCD format.

The "Date/Time" area pointer when used in a project which contains multiple connections must be enabled for each configured connection.

Note

You cannot use the "Date/Time PLC" area pointer if you have configured the "Date/Time" area pointer.

Note

Symbolic addressing is not possible if you are using the "Date/Time" area pointer.

The "Date/Time" area pointer when used in a project which contains multiple connections must be enabled for each configured connection.

The date/time data area has the following structure:

Data word	Most significant byte								Least significant byte								
	7							0	7							0	
n+0	Reserved								Hour (0 to 23)								Time
n+1	Minute (0 to 59)								Second (0 to 59)								
n+2	Reserved								Reserved								
n+3	Reserved								Weekday (1 to 7, 1=Sunday)								Date
n+4	Day (1 to 31)								Month (1 to 12)								
n+5	Year (80 to 99/0 to 29)								Reserved								

Note

When making entries in the "Year" data area, you should note that values 80 to 99 result in years 1980 through 1999, while the values 0 to 29 result in the years 2000 through 2029.

See also

General information on area pointers (Page 407)

"Date/time PLC" area pointer (Basic Panels)

Function

This area pointer is used to transfer the date and time from the PLC to the HMI device. Use this area pointer if the PLC is the time master.

The PLC loads the data area of the area pointer. All definitions are coded in BCD format.

The HMI device reads the data cyclically within the configured acquisition cycle and synchronizes itself.

Note

Set an acquisition cycle of sufficient length for the Date/time area pointer in order to avoid any negative impact on HMI device performance.
Recommended: Acquisition cycle of 1 minute, if the process allows this.

"Date/Time PLC" is a global area pointer and may be configured only once per project.

Note

You cannot use the "Date/Time" area pointer if you have configured the "Date/Time PLC" area pointer.

The date/time data area has the following structure:

DATE_AND_TIME format (in BCD code)

Data word	Most significant byte			Least significant byte		
	7	. . . . .	0	7	. . . . .	0
n+0	Year (80 to 99/0 to 29)			Month (1 to 12)		
n+1	Day (1 to 31)			Hour (0 to 23)		
n+2	Minute (0 to 59)			Second (0 to 59)		
n+3	Reserved			Reserved		Weekday (1 to 7, 1=Sunday)
n+4 [1)]	Reserved			Reserved		
n+5 [1)]	Reserved			Reserved		

1) The two data words must exist in the data area to ensure that the data format matches WinCC flexible and to avoid reading false information.

Note

When making entries in the "Year" data area, you should note that values 80 to 99 result in years 1980 through 1999, while the values 0 to 29 result in the years 2000 through 2029.

See also

General information on area pointers (Page 407)

"Coordination" area pointer (Basic Panels)

Function

The "Coordination" area pointer is used to implement the following functions:

- Detecting the startup of the HMI device in the control program
- Detecting the current operating mode of the HMI device in the control program
- Detecting whether the HMI device is ready to communicate in the control program

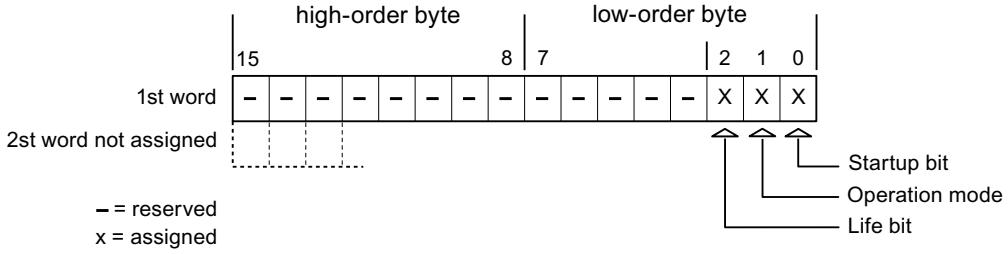
By default, the "Coordination" area pointer has the length of one word and cannot be changed.

Usage

Note

The HMI device always writes the entire coordination area when updating the area pointer. The control program can for this reason not make changes to the coordination area.

Assignment of the bits in the "Coordination" area pointer



Startup bit

The startup bit is set briefly to "0" by the HMI device during startup. It sets the bit permanently to "1" when startup is completed.

Operating mode

The operating mode bit is set to 1 as soon as the user switches the HMI device offline. The status of the operating mode bit is "0" during normal operation of the HMI device. You can determine the current operating mode of the HMI device by reading this bit in the control program.

Life bit

The HMI device inverts the life bit at intervals of approximately one second. You can check whether or not there is still a connection to the HMI device by querying this bit in the control program.

See also

General information on area pointers (Page 407)

"Project ID" area pointer (Basic Panels)

Function

When Runtime starts, a check can be carried out as to whether the HMI device is connected to the correct PLC. This check is important when operating with several HMI devices.

For this, the HMI device compares a value stored on the PLC with the value specified in configuration. This ensures compatibility of the configuration data with the PLC program. If there is no concordance, a system event is given on the HMI device and Runtime is stopped.

Use

Note

HMI connections cannot be switched "online".

The HMI connection in which the "Project ID" area pointer is used must be switched "online".

To use this area pointer, set up the following during the configuration:

- Define the version of the configuration. Values between 1 and 255 are possible.
  You enter the version in the editor "Runtime settings > General" in the "Identification" area.
- Data address of the value for the version that is stored in the PLC:
  You enter the data address in the editor "Communication > Connections" under "Address".

Connection failure

A connection failure to a device on which the "Project ID" area pointer is configured results in all the other connections of the device being switched to "offline".

This behavior has the following requirements:

- You have configured several connections in a project.
- You are using the "Project ID" area pointer in at least one connection.

Causes which may set connections "offline":

- The PLC is not available.
- The connection has been switched offline in the Engineering System.

See also

General information on area pointers (Page 407)

"Job mailbox" area pointer (Basic Panels)

Function

The PLC can use the job mailbox to transfer jobs to the HMI device to trigger corresponding actions on the HMI device. These functions include, for example:

- Display screen
- Set date and time

Data structure

The first word of the job mailbox contains the job number. Depending on the job mailbox, up to three parameters can be transferred.

Word	Most significant byte	Least significant byte
n+0	0	Job number
n+1	Parameter 1	
n+2	Parameter 2	
n+3	Parameter 3	

The HMI device evaluates the job mailbox if the first word of this job is not equal to zero. This means that the parameters must be entered in the job mailbox first, followed by the job number.

When the HMI device accepts the job mailbox, the first word is set to 0 again. The execution of the job mailbox is generally not completed at this point in time.

Job mailboxes

All job mailboxes and their parameters are listed below. The "No." column contains the job number of the job mailbox. Job mailboxes can only be triggered by the PLC when the HMI device is online.

No.	Function	
14	**Set time (BCD-coded)**	
	Parameter 1	Left byte: - Right byte: hours (0-23)
	Parameter 2	Left byte: minutes (0-59) Right byte: seconds (0-59)
	Parameter 3	-
15	**Setting the date (BCD coded)** [3]	
	Parameter 1	Left byte: - Right byte: Weekday (1-7: Sunday-Saturday)
	Parameter 2	Left byte: day (1-31) Right byte: month (1-12)
	Parameter 3	Left byte: year
23	**User logon**	

No.	Function	
14	**Set time (BCD-coded)**	
	Logs the user on with the name "PLC user" at the HMI device with the group number transferred in Parameter 1. The logon is possible only when the transferred group number exists in the project.	
	Parameter 1	Group number 1 to 255
	Parameter 2, 3	-
24	**User logoff**	
	Logs off the current user. (The function corresponds to the "logoff" system function)	
	Parameter 1, 2, 3	-
40	**Transfer date/time to PLC**	
	(In the S7 format DATE_AND_TIME) An interval of at least 5 seconds must be maintained between two successive jobs to prevent overload of the HMI device.	
	Parameter 1, 2, 3	-
41	**Transfer date/time to PLC**	
	(In OP/MP format) An interval of at least 5 seconds must be maintained between successive jobs to prevent overload of the HMI device.	
	Parameter 1, 2, 3	-
46	**Update tag**	
	Causes the HMI device to read the current value of the tags from the PLC whose update ID matches the value transferred in Parameter 1. (Function corresponds to the "UpdateTag" system function.)	
	Parameter 1	1 - 100
49	**Delete alarm buffer**	
	Deletes all analog alarms and discrete alarms of the "Warnings" class from the alarm buffer.	
	Parameter 1, 2, 3	-
50	**Delete alarm buffer**	
	Deletes all analog alarms and discrete alarms of the "Errors" class from the alarm buffer.	
	Parameter 1, 2, 3	-
51	**Screen selection** [2]	
	Parameter 1	Screen number
	Parameter 2	-
	Parameter 3	Field number
69	**Reading data record from PLC** [1]	
	Parameter 1	Recipe number (1-999)
	Parameter 2	Data record number (1-65535)
	Parameter 3	0: Do not overwrite existing data record 1: Overwrite existing data record
70	**Writing data record to PLC** [1]	
	Parameter 1	Recipe number (1-999)
	Parameter 2	Data record number (1-65535)
	Parameter 3	-

1)	Only devices supporting recipes
2)	OP 73, OP 77A and TP 177A HMI devices also execute the "Screen selection" job mailbox if the on-screen keyboard is active.
3)	The weekday is ignored on HMI device KTP 600 BASIC PN.

See also

General information on area pointers (Page 407)

"Data record" area pointer (Basic Panels)

"Data record" area pointer (Basic Panels)

Function

When data records are transferred between the HMI device and PLC, both partners access common communications areas on the PLC.

Data transfer types

There are two ways of transferring data records between the HMI device and PLC:

- Transfer without synchronization
- Transfer with synchronization over the data record

Data records are always transferred directly. That is, the tag values are read from an address or written to an address configured for this tag directly, without redirecting the values by means of interim memory.

Initiating the transfer of data records

There are three ways of triggering the transfer:

- Operator input in the recipe view
- Job mailboxes
  The transfer of data records can also be triggered by the PLC.
- Triggering by configured functions

If the transfer of data records is triggered by a configured function or by a job mailbox, the recipe view on the HMI device remains operable. The data records are transferred in the background.

Simultaneous processing of several transfer requests is, however, not possible. In this case, the HMI device rejects the other transfer requests with a system event.

See also

General information on area pointers (Page 407)
Sequence of a transfer started by the operator in the recipe view (Page 416)
Sequence of the transfer triggered by a job mailbox (Page 417)
Sequence of the transfer when triggered by a configured function (Page 418)
Possible causes of error when transferring data records (Page 419)
Transfer without synchronization (Page 420)
Transfer with synchronization (Page 421)

Sequence of a transfer started by the operator in the recipe view (Basic Panels)

Reading from the PLC started by the operator in the recipe view

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe number to be read and the status "Transferring" in the data record and sets the data record number to 0.	Abort with system event.
3	The HMI device reads the values from the PLC and displays them in the recipe view. If the recipes have synchronized tags, the values from the PLC are also written to the tags.	
4	The HMI device sets the status "Transfer completed."	
5	The control program must reset the status word to zero in order to enable further transfers.	

Writing to the PLC started by the operator in the recipe view

Step	Action	
	Check: Status word = 0?	
1	Yes	No
	The HMI device enters the recipe and data record number to be written and the status "Transferring" in the data mailbox.	Abort with system event.
2	The HMI device writes the current values to the PLC. If the recipes have synchronized tags, the changed values are synchronized in the recipe view and tags and then written to the PLC.	
3	The HMI device sets the status "Transfer completed."	
4	If required, the control program can now evaluate the transferred data.	
5	The control program must reset the status word to zero in order to enable further transfers.	

Note

The status word may only be set by the HMI device. The PLC may only reset the status word to zero.

Note

The PLC may only evaluate the recipe and data record numbers when data inconsistency is detected if one of the conditions outlined below has been met:

- The data mailbox status is set to "Transfer completed".
- The data mailbox status is set to "Transfer completed with error".

See also

"Data record" area pointer (Page 415)

Sequence of the transfer triggered by a job mailbox (Basic Panels)

The transfer of data records between the HMI device and the PLC can be initiated by the HMI device or by the PLC.

The two job mailboxes No. 69 and No. 70 are available for this type of transfer.

No. 69: Read data record from PLC ("PLC → DAT")

Job mailbox no. 69 transfers data records from the PLC to the HMI device. The job mailbox is structured as follows:

	Left byte (LB)	Right byte (RB)
Word 1	0	69
Word 2	Recipe number (1-999)	
Word 3	Data record number (1-65,535)	
Word 4	Do not overwrite existing data record: 0 Overwrite existing data record: 1	

No. 70: Write data record to PLC ("DAT → PLC")

Job mailbox No. 70 transfers data records from the HMI device to the PLC. The job mailbox is structured as follows:

	Left byte (LB)	Right byte (RB)
Word 1	0	70
Word 2	Recipe number (1-999)	
Word 3	Data record number (1-65,535)	
Word 4	—	

Sequence when reading from the PLC with job mailbox "PLC → DAT" (no. 69)

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the job and the status "Transferring" in the data mailbox.	Abort without return message.
3	The HMI device reads the values from the PLC and saves these to the data record defined in the job mailbox.	
4	• If "Overwrite" was selected in the job, an existing data record is overwritten without any prompt for confirmation. The HMI device sets the status "Transfer completed." • If "Do not overwrite" was selected in the job, and the data record already exists, the HMI device aborts the job and enters 0000 1100 in the status word of the data mailbox.	
5	The control program must reset the status word to zero in order to enable further transfers.	

Sequence of writing to the PLC using job mailbox "DAT → PLC" (no. 70)

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the job and the status "Transferring" in the data mailbox.	Abort without return message.
3	The HMI device fetches the values of the data record specified in the job from the data medium and writes the values to the PLC.	
4	The HMI device sets the status "Transfer completed."	
5	The control program can now evaluate the transferred data. The control program must reset the status word to zero in order to enable further transfers.	

See also

"Data record" area pointer (Page 415)

Sequence of the transfer when triggered by a configured function (Basic Panels)

Reading from the PLC using a configured function

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the function and the status "Transferring" in the data mailbox.	Abort with system event.

Step	Action	
3	The HMI device reads the values from the PLC and stores them in the data record specified in the function.	
4	• If "Yes" was selected for the "Overwrite" function, an existing data record is overwritten without any prompt for confirmation. The HMI device sets the status "Transfer completed." • If "No" was selected for the "Overwrite" function and the data record already exists, the HMI device aborts the job and enters 0000 1100 in the status word of the data mailbox.	
5	The control program must reset the status word to zero in order to enable further transfers.	

Writing to the PLC by means of configured function

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the function and the status "Transferring" in the data mailbox.	Abort with system event.
3	The HMI device fetches the values of the data record specified in the function from the data medium and transfers the values to the PLC.	
4	The HMI device sets the status "Transfer completed."	
5	The control program can now evaluate the transferred data. The control program must reset the status word to zero in order to enable further transfers.	

See also

"Data record" area pointer (Page 415)

Possible causes of error when transferring data records (Basic Panels)

Possible causes of faults

The section below shows possible causes of errors which lead to a data record transfer being terminated with errors:

- Tag address not set up on the PLC
- Overwriting data records not possible

- Recipe number does not exist
- Data record number does not exist

Note

The status word may only be set by the HMI device. The PLC may only reset the status word to zero.

Note

The PLC may only evaluate the recipe and data record numbers when data inconsistency is detected if one of the conditions outlined below has been met:

- The data mailbox status is set to "Transfer completed".
- The data mailbox status is set to "Transfer completed with error".

Reaction to an aborted transfer due to errors

If the transfer of data records is aborted due to errors, the HMI device reacts as follows:

- Triggering by the operator in the recipe view
  Information in the status bar of the recipe view and output of system events
- Triggered by function
  Output of system events
- Triggering by job mailbox
  No return message on the HMI device

You can nonetheless evaluate the status of the transfer by querying the status word in the data mailbox.

See also

"Data record" area pointer (Page 415)

Transfer without synchronization (Basic Panels)

If you select asynchronous transfer of data records between the HMI device and PLC, there is no coordination over the common data areas. It is therefore unnecessary to set up a data area during configuration.

Asynchronous data record transfer can be a useful alternative, for example, when:

- The system is capable of excluding the risk of uncontrolled overwriting of data by the communication peer.
- The PLC does not require information about the recipe number and data record number.
- The transfer of data records is triggered by the operator of the HMI device.

Reading values

When a read job is triggered, the values are read from the PLC addresses and transferred to the HMI device.

- Triggering by the operator in the recipe view:
  The values are downloaded to the HMI device. You can then process, edit, or save these values, for example.
- Triggering by a function or job mailbox:
  The values are saved immediately to the data volume.

Writing values

When a write job is triggered, the values are written to the PLC addresses.

- Triggering by the operator in the recipe view:
  The current values are written to the PLC.
- Triggering by a function or job mailbox:
  The current values are written to the PLC from the data medium.

See also

"Data record" area pointer (Page 415)

Transfer with synchronization (Basic Panels)

If you select synchronous transfer, both communication partners set status bits in the common data area. You can use this mechanism to prevent uncontrolled overwriting of data in either direction in your control program.

Application

Synchronous data record transfer can be a useful solution, for example, when:

- The PLC is the "active partner" in the transfer of data records.
- The PLC evaluates the information about the recipe number and data record number.
- The transfer of data records is triggered by means of a Job mailbox.

Requirements

In order to synchronize transfer of data records between the HMI device and the PLC, the following requirements must be met during configuration:

- An area pointer has been set up: "Communication > Connections" editor in "Area pointer".
- The PLC with which the HMI device synchronizes transfer of data records is specified in the recipe:
  "Recipes" editor in the Inspector window the "Coordinated transfer of data records" option under "General > Synchronization > Settings".

Structure of the data area

The data area has a fixed length of 5 words. Structure of the data area:

	15		0
1. Word	Current recipe number (1 - 999)		
2. Word	Current data record number (0 - 65535)		
3. Word	Reserved		
4. Word	Status (0, 2, 4, 12)		
5. Word	Reserved		

- Status
  The status word (word 4) can adopt the following values:

Value		Meaning
Decimal	Binary	
0	0000 0000	Transfer permitted, data record free
2	0000 0010	Transferring
4	0000 0100	Transfer completed without error
12	0000 1100	Transfer completed with error

See also

"Data record" area pointer (Page 415)

1.11.4.2 Trends (Basic Panels)

Trends

A trend is the graphical representation of one or more values from the PLC. The value is read out time-triggered for Basic Panels.

For additional information see:

AUTOHOTSPOT

Time-triggered trends

The HMI device reads in the trend values cyclically at an interval specified in the configuration.

Time-triggered trends are suitable for continuous curves, such as the operating temperature of a motor.

1.11.4.3 Alarms (Basic Panels)

Configuring alarms (Basic Panels)

Configure alarms

Several steps are needed to configure alarms, such as operational messages, error alarms, and acknowledgement.

- Step 1: Create tags
- Step 2: Configure alarms
- Step 3: Configure acknowledgment

You can find additional information in the section:

AUTOHOTSPOT

Distinctive features when configuring alarms

If you are configuring connections of HMI devices to PLCs of other manufacturers, note the following distinctive features when configuring:

- Data types of the tags
- Addressing of tags
- How the bit positions are counted

Data types

For connections with a SIMATIC communication driver, the following data types are supported:

PLC	Permitted data types	
	Discrete alarms	Analog alarms
SIMATIC S7 PLCs	WORD, INT	BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, TIMER

How the bit positions are counted

For connections with a SIMATIC communication driver, the following counting method applies:

How the bit positions are counted	Byte 0								Byte 1							
	Most significant byte								Least significant byte							
In SIMATIC S7 PLCs	7							0	7							0
In WinCC you configure:	15							8	7							0

See also

Acknowledgment of alarms (Page 424)

Acknowledgment of alarms (Basic Panels)

Procedure

Create suitable tags on the PLC to acknowledge an error alarm. You assign these tags to an alarm in the "Bit messages" editor. You make the assignment in "Properties > Acknowledgment".

Distinction in terms of acknowledgment:

- Acknowledgment by the PLC
- Acknowledgment on the HMI device

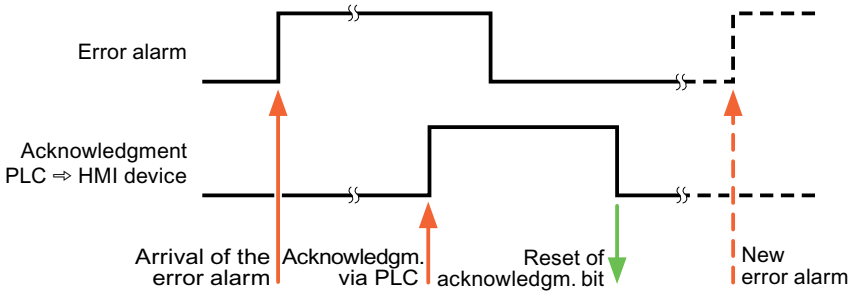
Acknowledgment by the PLC

In "PLC acknowledgment tag", configure the tag or array tag and the bit number that the HMI device uses to identify a PLC acknowledgment.

A bit set in the tag triggers acknowledgment of the assigned error alarm bit at the HMI device. This tag bit returns a function similar to acknowledgment on the HMI device which is triggered by pressing the "ACK" button, for example.

The acknowledgment bit must be located in the same tag as the bit for the error alarm.

Reset the acknowledgment bit before setting the bit in the alarm area again. The figure below shows the pulse diagram.



Acknowledgment on the HMI device

In the "HMI acknowledgment tag" area, configure the tag or array tag as well as the bit number that the HMI device writes to the PLC after acknowledgment. Make sure when you use an array tag that it is not longer than 6 words.

To always create a signal change when setting an assigned acknowledgment bit of a discrete alarm that must be acknowledged, the HMI device will reset the acknowledgment bit assigned to the alarm as soon as it detects an alarm subject to acknowledgment and write the acknowledgment tag in the PLC. There will be a certain delay between detecting the message

and writing the acknowledgment tag in the PLC because the HMI device has to process the operations.

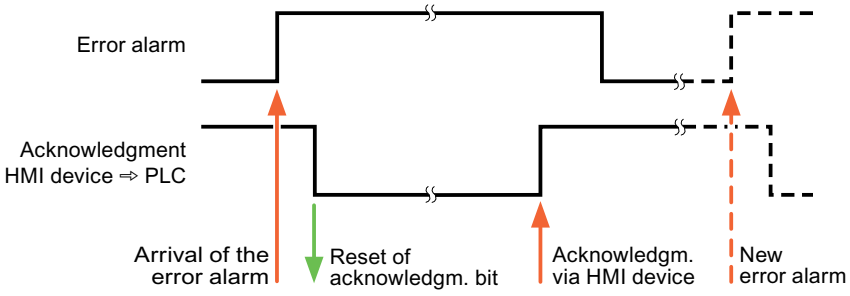
If a discrete alarm subject to acknowledgment is acknowledged by the HMI device, then the corresponding bit in the assigned acknowledgment tag will be set. The entire acknowledgment tag is then written to the PLC by the HMI device. This allows the PLC to recognize that a certain alarm message has been acknowledged at the HMI device.

Note

All alarm bits acknowledged since the last Runtime start will remain in the acknowledgment tag until a new incoming of the respective discrete alarms is detected.

This area should only be read by the PLC because the entire section of the HMI device will be overwritten once the next acknowledgment tag is written.

The figure below shows the pulse diagram.



See also

Configuring alarms (Page 423)

1.11.5 Performance features of communication (Basic Panels)

1.11.5.1 Device dependency S7 200 (Basic Panels)

Communication with the SIMATIC S7-200 controller

If you use devices from an earlier version of the TIA Portal with TIA Portal V14, it may not be possible to configure integrated connections to certain HMI devices.

Basic Panels V11.0

HMI devices	SIMATIC S7-200
KP300 Basic	Yes
KP400 Basic	Yes
KTP400 Basic PN	Yes
KTP600 Basic DP	Yes
KTP600 Basic PN	Yes
KTP1000 Basic DP	Yes
KTP1000 Basic PN	Yes
TP1500 Basic PN	Yes

Basic Panels V12.0

HMI devices	SIMATIC S7-200
KP300 Basic	Yes
KP400 Basic	Yes
KTP400 Basic PN	Yes
KTP600 Basic DP	Yes
KTP600 Basic PN	Yes
KTP1000 Basic DP	Yes
KTP1000 Basic PN	Yes
TP1500 Basic PN	Yes

Basic Panels V13.0

HMI devices	SIMATIC S7-200
KTP400 Basic PN	Yes
KTP700 Basic PN	Yes
KTP700 Basic DP	Yes
KTP900 Basic PN	Yes
KTP1200 Basic PN	Yes
KTP1200 Basic DP	Yes

Basic Panels V13.0.1

HMI devices	SIMATIC S7-200
KTP400 Basic PN	Yes
KTP700 Basic PN	Yes
KTP700 Basic DP	Yes
KTP900 Basic PN	Yes

HMI devices	SIMATIC S7-200
KTP1200 Basic PN	Yes
KTP1200 Basic DP	Yes

Basic Panels V14.0

HMI devices	SIMATIC S7-200
KTP400 Basic PN	Yes
KTP700 Basic PN	Yes
KTP700 Basic DP	Yes
KTP900 Basic PN	Yes
KTP1200 Basic PN	Yes
KTP1200 Basic DP	Yes

See also

Communication with SIMATIC S7 200 (Page 396)
Permitted data types for SIMATIC S7 200 (Page 427)

1.11.5.2 Permitted data types for SIMATIC S7 200 (Basic Panels)

Permitted data types for connections with SIMATIC S7 200

The table lists the data types that can be used when configuring tags and area pointers.

Data type	Length
Bool	1 bit
Byte	1 byte
Char	1 byte
Array	(Element number * data type length) bytes [1]
Word	2 bytes
Int	2 bytes
DWord	4 bytes
DInt	4 bytes
Real	4 bytes
StringChar	(n)bytes, n=0 to 254 bytes
Timer	2 bytes

[1] Example of "Length of an array": The length is 400 bytes (100 * 4) for 100 elements of the data type REAL

Note

Disconnection with a PPI network

If you are using arrays in the configuration, an array size of approximately 1000 bytes may cause an interruption of the connection.

Use smaller arrays in your configuration.

See also

Communication with SIMATIC S7 200 (Page 396)

Device dependency S7 200 (Page 425)

1.12 Communicating with SIMATIC LOGO! (Basic Panels)

1.12.1 Communication with SIMATIC LOGO! (Basic Panels)

Introduction

This section describes the communication between an HMI device and the SIMATIC LOGO! controller.

You can configure the following communication channels for the SIMATIC LOGO! controller:

- PROFINET
- Ethernet

HMI connection for communication

You configure connections between the HMI device and SIMATIC LOGO! in the "Connections" editor of the HMI device.

Data exchange

Data exchange with the SIMATIC LOGO! control system is possible by means of tags.

Data cannot be exchanged using area pointers.

See also

Creating a connection to SIMATIC LOGO! (Page 429)

1.12.2 Creating a connection to SIMATIC LOGO! (Basic Panels)

Introduction

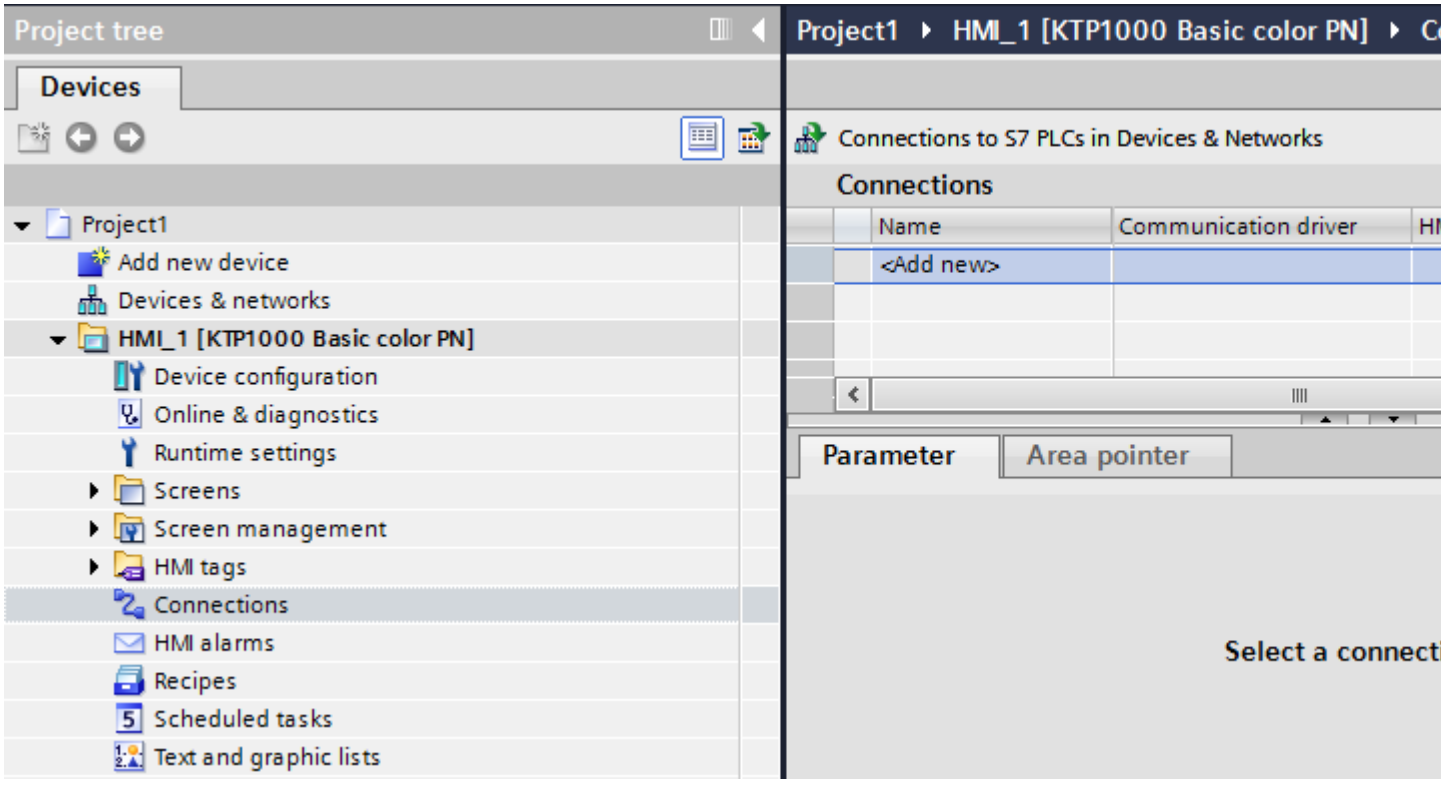
You configure a connection to the SIMATIC LOGO! controller in the "Connections" editor of the HMI device. The interfaces are named differently depending on the HMI device.

Requirements

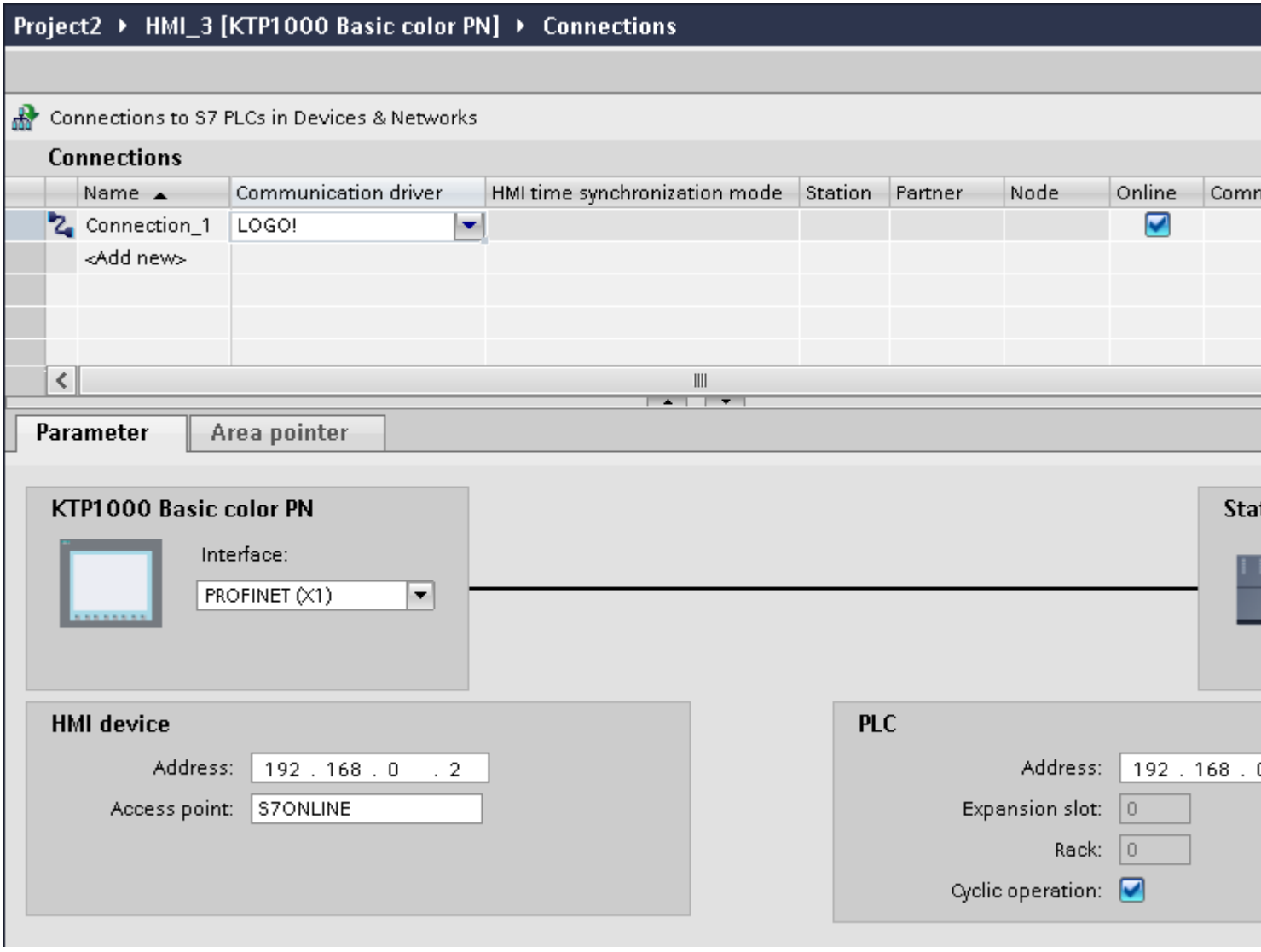
- A project is open.
- An HMI device has been created.

Procedure

1. Double-click the HMI device under "Devices" in the project tree.
2. Double-click the "Connections" item.
3. Double-click "<Add>" in the "Connections" editor.



4. In the "Communication drivers" column, select the "SIMATIC LOGO!" driver.
5. Select all necessary connection parameters for the interface in the Inspector window under "Parameters".



See the "AUTOHOTSPOT" section for more detailed information.

See also

Communication with SIMATIC LOGO! (Page 428)

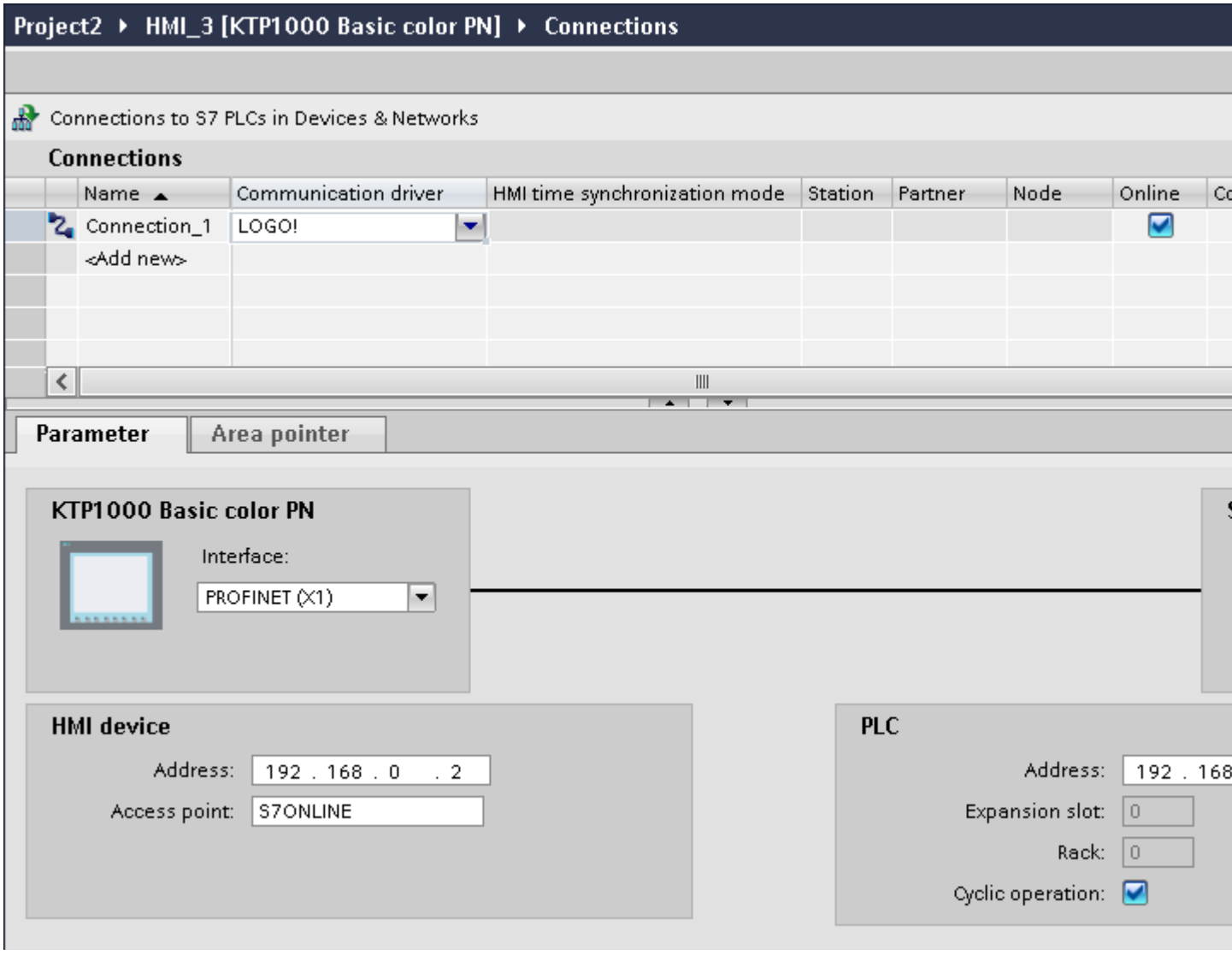
1.12.3 Connection parameters (Basic Panels)

1.12.3.1 Connection parameters (Basic Panels)

Parameters to be set

To set the connection parameters, such as addresses and profiles, click the connection that you have created in the "Connections" editor.

The communication partners are displayed schematically in the Inspector window under "Parameters". The "HMI device" and "PLC" areas are available for assigning parameters according to the interface used.



See also

Communication with SIMATIC LOGO! (Page 428)

1.12.3.2 Ethernet parameters (Basic Panels)

Parameters for the HMI device

You set the parameters for the HMI device in the network under "HMI device".. The changes are not transferred automatically to the HMI device. You must change the settings in the Control Panel of the HMI device.

- "Interface"
  If you are directly connected to the HMI device during configuration, you can set up the IP address of the HMI device in WinCC.

  Note

  The IP address in the Control Panel will be overwritten upon subsequent loading if you have already set up the IP address in the HMI device control panel.

  The IP address already set up in the Control Panel will be retained upon subsequent loading if you activate "Set IP address using a different method".

  The IP address is transferred to the HMI device during project transfer.
  To set up the IP address of the HMI device:

  – Click the HMI device.
  – Open the "Device configuration" editor.
  – Click the Ethernet interface.
  – Assign the IP address in the inspector window under:
    "General > PROFINET interface > Ethernet addresses"
- "Address"
  You assign the IP address of the HMI device in the "Address" area.
  When you transfer the WinCC project to the HMI device, this IP address is set up directly in the HMI device.
- "Access point"
  The access point defines a logical device name through which the communication partner can be reached.

Parameters for the PLC

Under "PLC", you address the S7 module with which the HMI device will exchange data. Assign a name for the connection for each communication partner.

- "Address"
  Under "Address", set the IP address of the S7 module to which the HMI device is connected.
- "Expansion slot"
  Defines the number of the expansion slot of the CPU to be addressed.

- "Rack"
  Defines the rack number of the CPU to be addressed.
- "Cyclic operation"

  Note

  The setting "Cyclic operation" cannot be configured for the SIMATIC S7 1200 PLC.

  When cyclic operation is enabled, the PLC optimizes the data exchange between the HMI device and the PLC. This increases system performance.
  Disable cyclic mode if you are operating several HMI devices in parallel.

See also

Communication with SIMATIC LOGO! (Page 428)

1.12.3.3 Cyclic operation (Basic Panels)

Handling the "Cyclic operation" selection

If "Cyclic operation" is enabled, the HMI device sends a message frame to the CPU at the beginning of communication indicating that certain tags are required on a recurring basis.

The CPU then always transmits the data at the same cyclic interval. This saves the HMI device from having to output new requests for the data.

If cyclic operation is disabled, the HMI device sends a request whenever information is required.

Additional properties:

- Cyclic operation reduces data transmission load at the HMI device. The PLC resources are used to relieve load on the HMI device.
- The PLC only supports a certain number of cyclic services. The HMI device handles the operation if the PLC cannot provide any further resources for cyclic services.
- The HMI device generates the cycle if the PLC does not support the cyclic mode.
- Screen tags are not integrated in cyclic operation.
- Cyclic mode is only set up at the restart of Runtime.
- The HMI device transfers several jobs to the PLC if cyclic mode is enabled, depending on the PLC.
- The HMI device only transfers one job to the PLC if cyclic mode is disabled.

See also

Communication with SIMATIC LOGO! (Page 428)

1.12.4 Data exchange (Basic Panels)

1.12.4.1 Trends (Basic Panels)

Trends

A trend is the graphical representation of one or more values from the PLC. The value is read out time-triggered for Basic Panels.

For additional information see:

AUTOHOTSPOT

Time-triggered trends

The HMI device reads in the trend values cyclically at an interval specified in the configuration.

Time-triggered trends are suitable for continuous curves, such as the operating temperature of a motor.

1.12.4.2 Alarms (Basic Panels)

Configuring alarms (Basic Panels)

Configure alarms

Several steps are needed to configure alarms, such as operational messages, error alarms, and acknowledgement.

- Step 1: Create tags
- Step 2: Configure alarms
- Step 3: Configure acknowledgment

You can find additional information in the section:

AUTOHOTSPOT

Distinctive features when configuring alarms

If you are configuring connections of HMI devices to PLCs of other manufacturers, note the following distinctive features when configuring:

- Data types of the tags
- Addressing of tags
- How the bit positions are counted

Data types

For connections with a SIMATIC communication driver, the following data types are supported:

PLC	Permitted data types	
	Discrete alarms	Analog alarms
SIMATIC S7 PLCs	WORD, INT	BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, TIMER

How the bit positions are counted

For connections with a SIMATIC communication driver, the following counting method applies:

How the bit positions are counted	Byte 0								Byte 1							
	Most significant byte								Least significant byte							
In SIMATIC S7 PLCs	7							0	7							0
In WinCC you configure:	15							8	7							0

See also

Acknowledgment of alarms (Page 435)

Acknowledgment of alarms (Basic Panels)

Procedure

Create suitable tags on the PLC to acknowledge an error alarm. You assign these tags to an alarm in the "Bit messages" editor. You make the assignment in "Properties > Acknowledgment".

Distinction in terms of acknowledgment:

- Acknowledgment by the PLC
- Acknowledgment on the HMI device

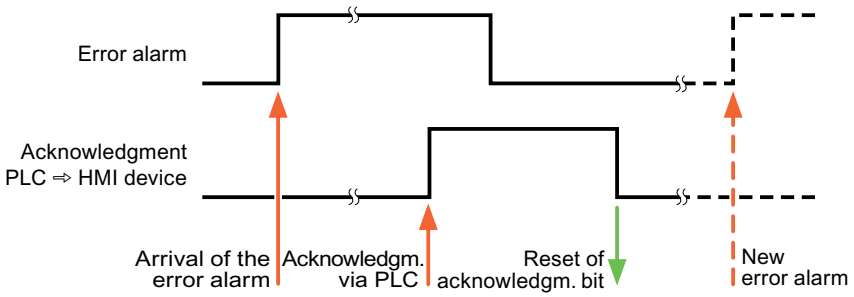
Acknowledgment by the PLC

In "PLC acknowledgment tag", configure the tag or array tag and the bit number that the HMI device uses to identify a PLC acknowledgment.

A bit set in the tag triggers acknowledgment of the assigned error alarm bit at the HMI device. This tag bit returns a function similar to acknowledgment on the HMI device which is triggered by pressing the "ACK" button, for example.

The acknowledgment bit must be located in the same tag as the bit for the error alarm.

Reset the acknowledgment bit before setting the bit in the alarm area again. The figure below shows the pulse diagram.



Acknowledgment on the HMI device

In the "HMI acknowledgment tag" area, configure the tag or array tag as well as the bit number that the HMI device writes to the PLC after acknowledgment. Make sure when you use an array tag that it is not longer than 6 words.

To always create a signal change when setting an assigned acknowledgment bit of a discrete alarm that must be acknowledged, the HMI device will reset the acknowledgment bit assigned to the alarm as soon as it detects an alarm subject to acknowledgment and write the acknowledgment tag in the PLC. There will be a certain delay between detecting the message and writing the acknowledgment tag in the PLC because the HMI device has to process the operations.

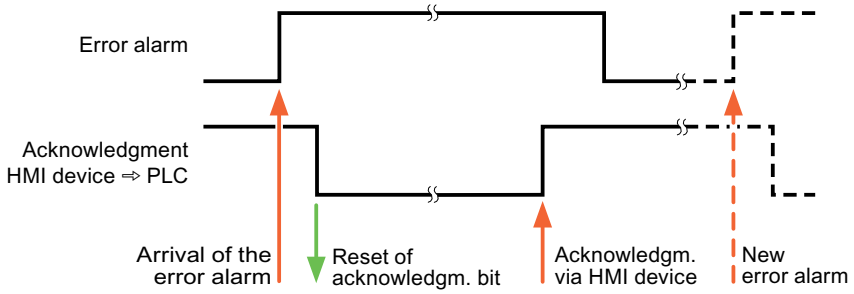
If a discrete alarm subject to acknowledgment is acknowledged by the HMI device, then the corresponding bit in the assigned acknowledgment tag will be set. The entire acknowledgment tag is then written to the PLC by the HMI device. This allows the PLC to recognize that a certain alarm message has been acknowledged at the HMI device.

Note

All alarm bits acknowledged since the last Runtime start will remain in the acknowledgment tag until a new incoming of the respective discrete alarms is detected.

This area should only be read by the PLC because the entire section of the HMI device will be overwritten once the next acknowledgment tag is written.

The figure below shows the pulse diagram.



See also

Configuring alarms (Page 434)

1.12.5 Performance features of communication (Basic Panels)

1.12.5.1 SIMATIC LOGO! device dependency (Basic Panels)

Communication with the SIMATIC LOGO! controller

If you use devices from an earlier version of the TIA Portal with TIA Portal V14, it may not be possible to configure integrated connections to certain HMI devices.

Basic Panels V11.0

HMI devices	SIMATIC LOGO!
KP300 Basic	No
KP400 Basic	No
KTP400 Basic PN	No
KTP600 Basic DP	No
KTP600 Basic PN	No
KTP1000 Basic DP	No
KTP1000 Basic PN	No
TP1500 Basic PN	No

Basic Panels V12.0

HMI devices	SIMATIC LOGO!
KP300 Basic	Yes
KP400 Basic	Yes
KTP400 Basic PN	Yes
KTP600 Basic DP	No
KTP600 Basic PN	Yes
KTP1000 Basic DP	No
KTP1000 Basic PN	Yes
TP1500 Basic PN	Yes

Basic Panels V13.0

HMI devices	SIMATIC LOGO!
KTP400 Basic PN	Yes
KTP700 Basic PN	Yes
KTP700 Basic DP	No
KTP900 Basic PN	Yes
KTP1200 Basic PN	Yes
KTP1200 Basic DP	No

Basic Panels V13.0.1

HMI devices	SIMATIC LOGO!
KTP400 Basic PN	Yes
KTP700 Basic PN	Yes
KTP700 Basic DP	No
KTP900 Basic PN	Yes
KTP1200 Basic PN	Yes
KTP1200 Basic DP	No

Basic Panels V14.0

HMI devices	SIMATIC LOGO!
KTP400 Basic PN	Yes
KTP700 Basic PN	Yes
KTP700 Basic DP	No
KTP900 Basic PN	Yes
KTP1200 Basic PN	Yes
KTP1200 Basic DP	No

See also

Communication with SIMATIC LOGO! (Page 428)

1.12.5.2 Valid data types for SIMATIC LOGO! (Basic Panels)

Valid data types for connections with SIMATIC LOGO!

Data type	Length
Bool	1 bit
Byte	1 byte
Int	2 bytes

Data type	Length
DInt	4 bytes
Word	2 bytes
DWord	4 bytes
Array	(Element number * data type length) bytes [1]

[1] Example of "Length of an array": The length is 200 bytes (100 * 2) for 100 elements of the data type Int

See also

Communication with SIMATIC LOGO! (Page 428)

1.13 Communication with other PLCs (Basic Panels)

1.13.1 Communication with other PLCs (Basic Panels)

Introduction

Communication with other PLCs is communication with PLCs that are not in the SIMATIC family.

These PLCs have proprietary protocols for data exchange. The protocols are configured as communication drivers in WinCC.

Communication drivers

The following communication drivers are supported in WinCC and are already installed:

- Allen-Bradley
  - Allen-Bradley EtherNet/IP
  - Allen-Bradley DF1
- Mitsubishi
  - Mitsubishi MC TCP/IP
  - Mitsubishi FX
- Modicon Modbus
  - Modicon Modbus TCP/IP
  - Modicon Modbus RTU
- Omron
  - Omron Host Link

Note

When serial communication drivers are used, the MPI transfer channel must not be active in "Transfer settings" in the Control Panel of the HMI device.

Communication drivers in WinCC RT Professional

The following communication drivers are supported for RT Professional:

- Allen-Bradley
  - Allen-Bradley EtherNet/IP
- Mitsubishi
  - Mitsubishi MC TCP/IP
- Modicon Modbus
  - Modicon Modbus TCP

Connections between HMI devices and other PLCs

You configure the connections between HMI devices and other PLCs in the "Connections" editor of the HMI device. These connections are non-integrated connections.

See also

Distinctive features when configuring (Page 440)

1.13.2 Distinctive features when configuring (Basic Panels)

Distinctive features for data exchange

Distinctive features apply when configuring connections to other PLCs, compared to configuring integrated connections.

Note the following distinctive features when configuring:

- Addressing of tags
- Permitted data types
- Distinctive features when configuring area pointers
- Distinctive features when configuring alarms
- Distinctive features when configuring trends

For more detailed information on distinctive features when configuring, refer to Section "Data exchange" of the respective communication driver.

Special considerations for specifying the acquisition modes for tags

If you select the acquisition modes "Cyclic in operation" or "On demand" for updating of tag values in Runtime, the status of the connection is updated even if no tags are contained in the screen.

You are informed about the status of the connection via the corresponding alarms, even if the acquisition cycle of more than one second was configured.

If the connection to the PLC was interrupted, the alarm 220007 informs you about this.

See also

Communication with other PLCs (Page 439)

1.13.3 Communication drivers (Basic Panels)

1.13.3.1 Allen-Bradley (Basic Panels)

Allen-Bradley communication drivers (Basic Panels)

Introduction

This section describes the communication between an HMI device and PLCs that use Allen-Bradley communication drivers.

The following communication drivers are supported:

- Allen-Bradley EtherNet/IP
- Allen-Bradley DF1

Data exchange

Data is exchanged by means of tags or area pointers.

- Tags
  The PLC and the HMI device use process values for data exchange. You create tags in the configuration that point to addresses in the PLC. The HMI device reads the value from the defined address, and then displays it. The operator may also enter values on the HMI device, which are then written to the address in the PLC.
- Area pointers
  Area pointers are used to exchange specific data and are only set up when these data are used.

See also

Communication with other PLCs (Page 439)

Allen-Bradley EtherNet/IP (Basic Panels)

Configuring a connection via Allen-Bradley EtherNet/IP (Basic Panels)

Introduction

You configure a connection to a PLC with an Allen-Bradley EtherNet/IP communication driver in the "Connections" editor of the HMI device.

The Ethernet interfaces are named differently depending on the HMI device.

Example: PROFINET interface corresponds to the Ethernet interface

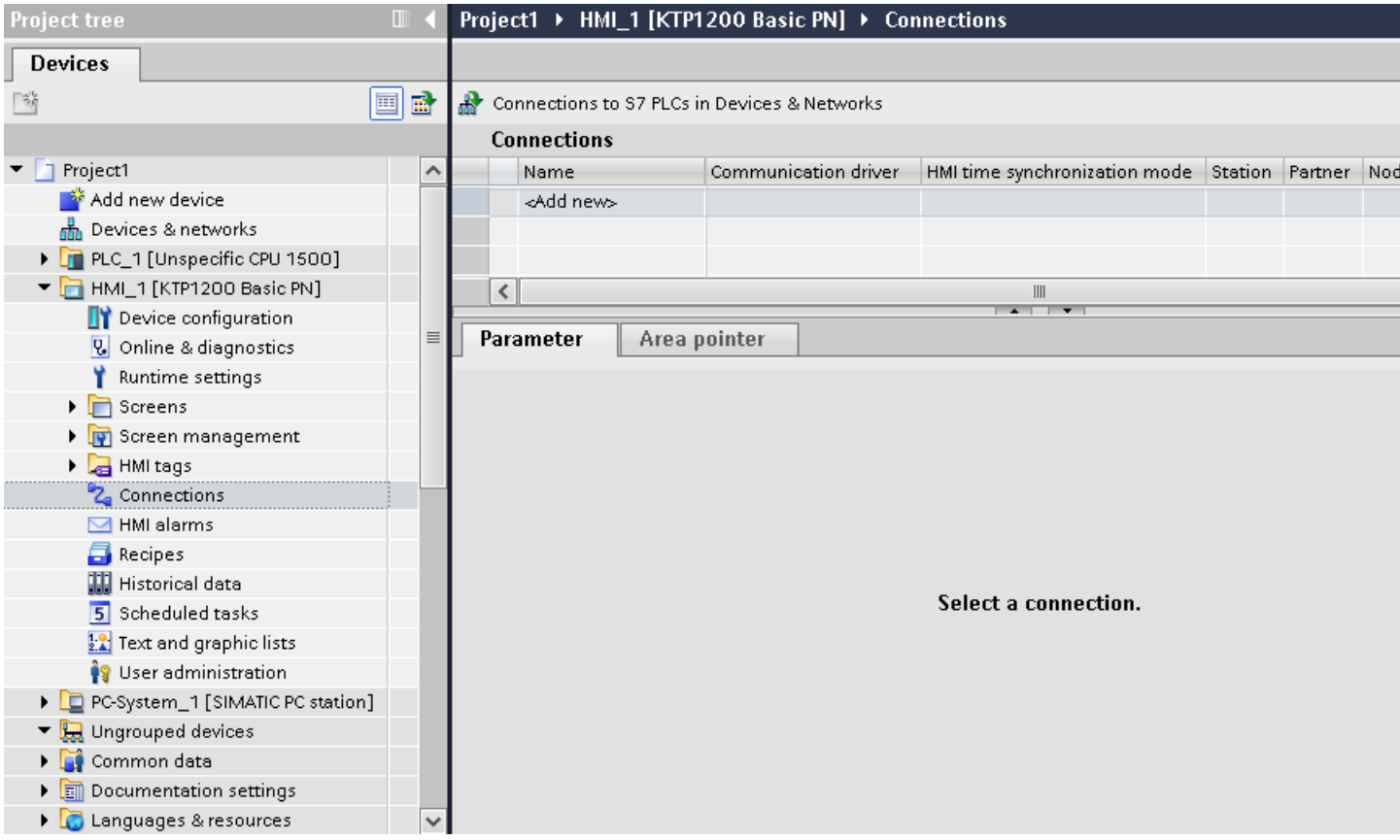
Requirements

- A project is open.
- An HMI device has been created.

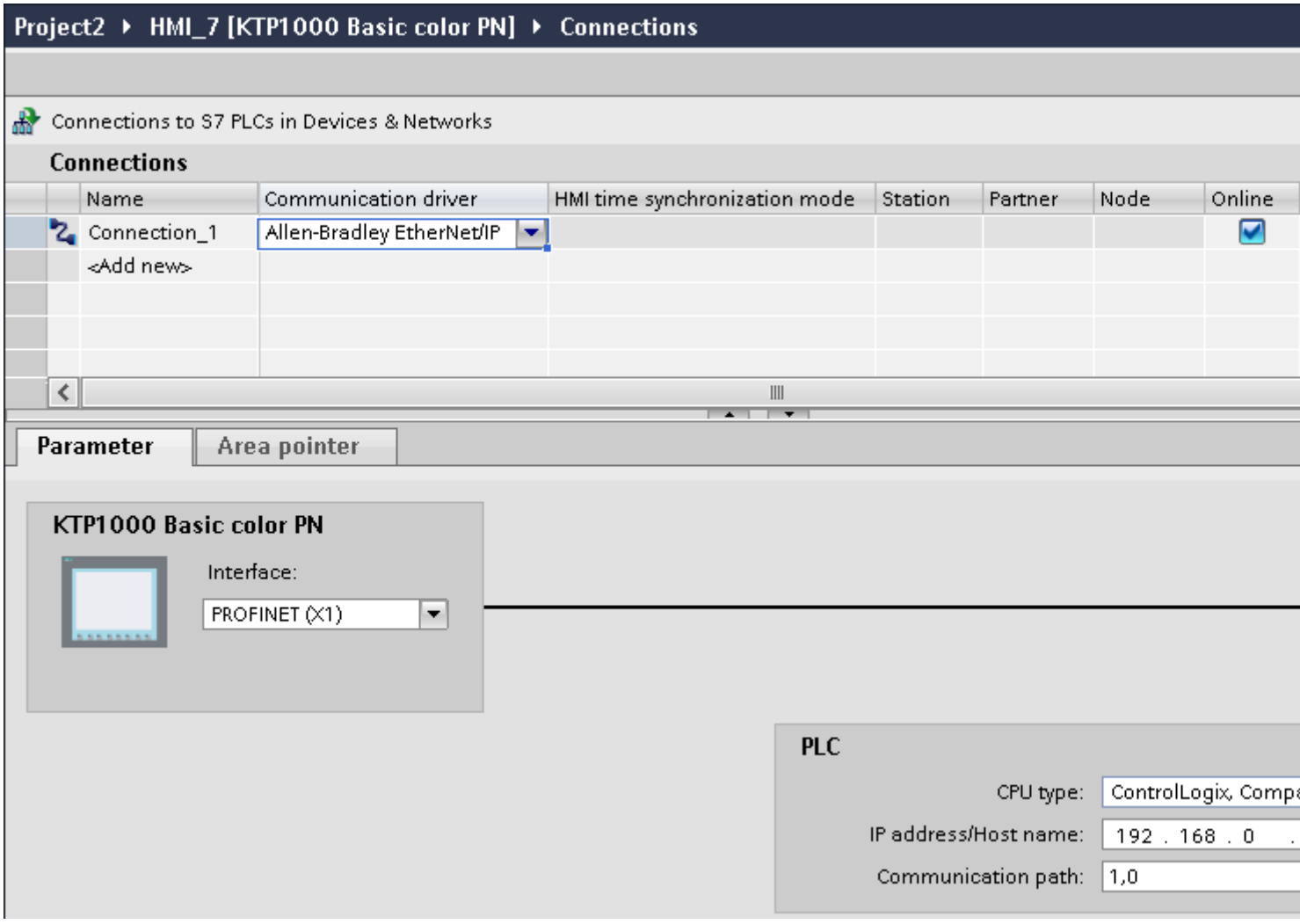
Procedure

1. Double-click the HMI device under "Devices" in the project tree.
2. Double-click the "Connections" item.

3. Double-click "<Add>" in the "Connections" editor.



4. In the "Communication drivers" column, select the "Allen-Bradley EtherNet/IP" driver.



5. Select all necessary connection parameters for the interface in the Inspector window under "Parameters".

See also

Allen-Bradley communication drivers (Page 441)

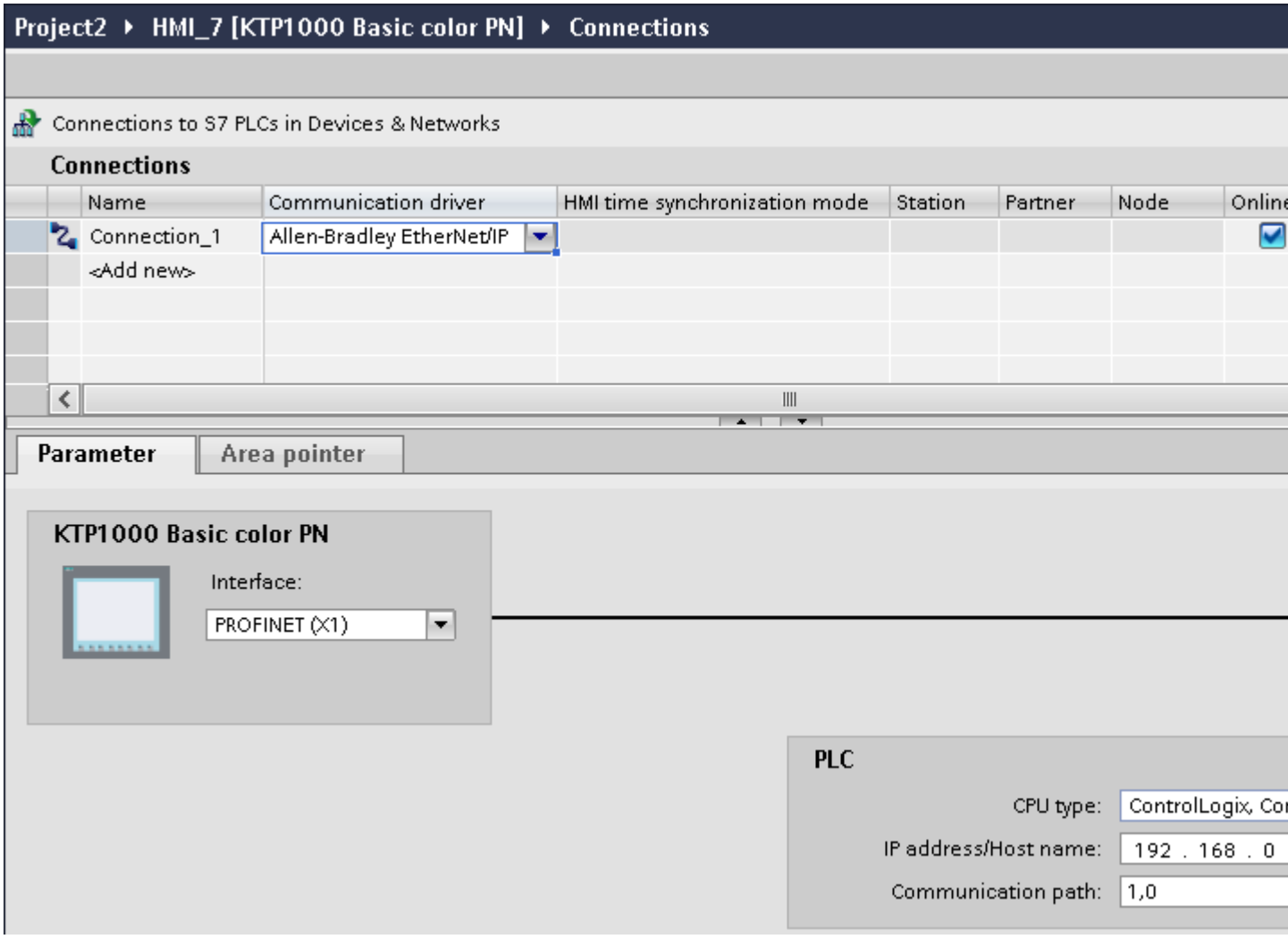
Parameters for the connection (Allen-Bradley EtherNet/IP) (Page 444)

Parameters for the connection (Allen-Bradley EtherNet/IP) (Basic Panels)

Parameters to be set

To assign the connection parameters, such as addresses and profiles, click the connection that you have created in the "Connections" editor.

The communication partners are displayed schematically in the Inspector window under "Parameters". The "PLC" area is available for assigning parameters according to the interface used.



Parameters for the HMI device

You can select only one interface for the HMI device in the Inspector window under "Parameters". Depending on the HMI device, there are several interfaces available.

If you are directly connected to the HMI device during configuration, you can set up the IP address of the HMI device in WinCC. The IP address is transferred to the HMI device upon subsequent loading.

Note

The IP address in the control panel will be overwritten upon subsequent loading if you have already set up the IP address in the HMI device control panel.

The IP address already set up in the control panel will be retained upon subsequent loading if you activate "Set IP address using a different method".

To set up the IP address of the HMI device:

1. Click on the HMI device.
2. Open the "Device configuration" editor.
3. Click the Ethernet interface.
4. Assign the IP address in the inspector window under:
   "General > PROFINET interface > Ethernet addresses"

Parameters for the PLC

- CPU type
  For "CPU type", set the CPU type of the PLC used.
- IP address
  Set the IP address or host name of the Ethernet/IP module of the PLC. Only the IP address can be used on a Basic Panel.
- Communication path
  Set the CIP path from the Ethernet module to the PLC. This establishes a logical connection between the Ethernet module and PLC, even if both devices are located in different CIP networks.
  For additional information on this see section: AUTOHOTSPOT

See also

Allen-Bradley communication drivers (Page 441)

Configuring a connection via Allen-Bradley EtherNet/IP (Page 442)

Connecting HMI device to PLC (Basic Panels)

Connections via Allen-Bradley EtherNet/IP (Basic Panels)

Connection

The HMI device can be connected to the Allen-Bradley PLC using the following components:

- Existing Ethernet network that also contains the PLCs
- Cross-over Ethernet cable connected directly to the Ethernet interface of the CPU or the communication module

The connection of the HMI device to an Allen-Bradley PLC is limited primarily to the physical connection of the HMI device. Special blocks for the connection are not required in the PLC.

See also

Configuring a connection via Allen-Bradley EtherNet/IP (Page 442)

Communication types (Basic Panels)

Approved communication types with Allen-Bradley EtherNet/IP

The following communication types are system-tested and approved:

- Point-to-point connection to the approved PLCs
- Multipoint connection from a HMI device (Allen-Bradley Ethernet/IP-Client) with up to 4 PLCs with the respectively approved PLCs. CPU types can be mixed.

Connection

Connection with the following PLCs is approved with Allen-Bradley EtherNet/IP:

- CPU type: "ControlLogix, CompactLogix"
  - ControlLogix
    556x(1756-L6x) with Ethernet module 1756-ENBT
  - Guard Logix-System ControlLogix
    556xS(1756-L6xS) with Ethernet module 1756-ENBT
    556xS(1756-L7x) with Ethernet module 1756-ENBT and 1756-EN2TR
    556xS(1756-L7xS) with Ethernet module 1756-ENBT and 1756-EN2TR
  - CompactLogix
    533xE(1769-L3xE) with Ethernet interface onboard
    532xE(1769-L2xE) with Ethernet interface onboard
    534x (1768-L4x) with Ethernet module 1768-ENBT
    534x(1768-L45S) with Ethernet module 1768-ENBT
    534x(1768-L43S) with Ethernet module 1768-ENBT
- CPU type: "SLC, MicroLogix"
  - MicroLogix 1100 (with Ethernet interface onboard)
  - MicroLogix 1400 (with Ethernet interface onboard)
  - SLC 5/05 (with Ethernet interface onboard)

See also

Allen-Bradley communication drivers (Page 441)

Performance features of communication (Basic Panels)

Permitted data types for Allen-Bradley EtherNet/IP (Basic Panels)

Permitted data types

The table lists the data types that can be used when configuring tags and area pointers.

CPU type: ControlLogix, CompactLogix

Data type	Length
Bool	1 bit
DInt	4 bytes
Int	2 bytes
Real	4 bytes
SInt	1 byte
String	1 to 82 characters
UDInt	4 bytes

Data type	Length
UInt	2 bytes
USInt	1 byte

Permitted data types arrays

Address	Permitted data types
Array	SInt, USInt, Int, UInt, DInt, UDInt, Real
Individual bits from the basic data types of the PLC SInt, USInt, Int, UInt, DInt, UDInt	Bool*

* Any changed value of certain defined bits is written back to the PLC. There is no check to determine whether any other bits have changed. The PLC (or other PLCs) may only read access the value.

CPU type: SLC, MicroLogix

Data type	Operand type	Length
ASCII	A	0 to 80 characters
Bool	N, R, C, T, B, S, I, O	1 bit
DInt	N	4 bytes
Int	N, R, C, T, S	2 bytes
Real	N, F	4 bytes
String	ST	1 to 82 characters
UDInt	N	4 bytes
UInt	N, R, C, T, B, I, O	2 bytes

Permitted data types arrays

Address	Permitted data types
Array	Int, UInt, DInt, UDInt, Real

Distinctive features for connections with Allen-Bradley Ethernet/IP

With the communication driver Allen Bradley Ethernet/IP and the CPU type SLC, MicroLogix, you can only use array tags for discrete alarms and trends.

Note

I/O modules with 8 or 16 ports occupy one data word on the PLC.

I/O modules with 24 or 32 ports occupy two data words.

The HMI device does not output an error message if using non-existent bits.

You should always make sure that I/O modules with 8 or 24 ports only occupy the bits that are actually assigned to a port.

See also

Configuring a connection via Allen-Bradley EtherNet/IP (Page 442)

Supported CPU types for Allen-Bradley EtherNet/IP (Page 450)

Supported CPU types for Allen-Bradley EtherNet/IP (Basic Panels)

CPU types

The following CPU types are supported for configuring the Allen-Bradley EtherNet/IP communication driver.

- CompactLogix
  - 533xE(1769-L3xE) with Ethernet interface onboard
  - 532xE(1769-L2xE) with Ethernet interface onboard
  - 534x(1768-L4x) with Ethernet module 1768-ENBT
  - 534x(1768-L45S) with Ethernet module 1768-ENBT
  - 534x(1768-L43S) with Ethernet module 1768-ENBT
- ControlLogix
  - 556x(1756-L6x) with Ethernet module 1756-ENBT
- GuardLogix
  - 556xS(1756-L6xS) with Ethernet module 1756-ENBT
  - 556xS(1756-L7x) with Ethernet module 1756-ENBT and 1756-EN2TR
  - 556xS(1756-L7xS) with Ethernet module 1756-ENBT and 1756-EN2TR
- MicroLogix
  - MicroLogix 1100/1400 with Ethernet interface onboard
- SLC50x
  - SLC5/05 with Ethernet interface onboard

See also

Configuring a connection via Allen-Bradley EtherNet/IP (Page 442)

Permitted data types for Allen-Bradley EtherNet/IP (Page 448)

Addressing in the C.Logix CPU type (Basic Panels)

Addressing (Basic Panels)

Addressing

A tag is uniquely referenced in WinCC by means of an address in the PLC. The address must correspond with the tag name in the PLC. The tag address is defined by a string with a length of up to 128 characters.

Using characters for addressing

Valid characters for tag addressing:

- Letters (a to z, A to Z)
- Numbers (0 to 9)
- Underscore (_)

The tag address consists of tag name and other character strings used to specify the tag in the PLC.

Tag name properties:

- The tag name may begin but not end with an underscore character.
- Strings with successive underscore and space characters are invalid.
- The address may not exceed a length of 128 characters.

Note

The characters reserved for tag addressing may not be used in program/tag names or at any other address instance.

The reserved characters are listed below:

Reserved character	Function
.	Element delimiter
:	Definition of a program tag
,	Delimiter for addressing multi-dimensional arrays
/	Reserved for bit addressing.
[ ]	Addressing of array elements or arrays

PLC and program tags

The Allen-Bradley EtherNet/IP communication driver supports addressing of PLC tags (global project tags) and/or program tags (global program tags).

A program tag is declared based on the program name in the PLC and actual tag name which are delimited by colon. PLC tags are simply addressed by their name.

Note

Addressing errors

Addressing errors occur when the tag name and data type are inconsistent.

Note that the tag name defined in the address field in WinCC must match the tag name in the PLC. Make sure that the data types of tags in WinCC match the data types in the PLC.

Note

Module-specific tags, e.g. for data on input and output modules, cannot be addressed directly. Instead, use an alias tag in the PLC.

Example: Local:3:O. Data cannot be addressed in WinCC.

If the alias "MyOut" is defined for Local:3:O in the PLC, you can address with WinCC via MyOut.Data.

See also

Addressing syntax (Page 452)

Address multiplexing (Page 455)

Examples for addressing (Page 455)

Examples: Communication path (Page 456)

Addressing syntax (Basic Panels)

Notation of addresses

The tables below define the notation for the individual addressing options for Allen-Bradley EtherNet/IP.

Table 1-2 Access to arrays, basic data types and structure elements

Data types	Type	Address
Basic data types	PLC tag	Tag name
	Program tag	Programname:tagname
Arrays	PLC tag	Array tag
	Program tag	Program name: array tag
Bits	PLC tag	Tagname/bitnumber
	Program tag	Programname:tagname/bitnumber
Structure elements	PLC tag	Structure tag. Structure element
	Program tag	Program name: structure tag. structure element

Note

Bit addressing with the data types Bool, Real and String is not permitted and will cause an addressing fault.

Description of the syntax

Syntax description:

```
 (Programname:)tagname([x(,y)(,z)]){.tagname([x(,y)(,z)])}(/
bitnumber)
```

- The "( )" defines an optional, single instance of an expression.
- The "{ }" defines an optional expression with multiple single instances.

The address string length may not exceed 128 characters.

See also

Addressing (Page 451)

Addressing types (Page 453)

Address multiplexing (Page 455)

Addressing types (Basic Panels)

Arrays

An array is a data structure that includes a number of data of the same type. WinCC only supports one-dimensional arrays.

In the address column of the tag editor, enter the array name possibly by specifying a start element. The length is defined in the Array Elements input box of the tag editor. If array limits in the PLC are exceeded (due to faulty indexing), addressing errors result.

These arrays must be declared in the PLC as controller or program tags.

Two- or three-dimensional arrays in the PLC can only be addressed in WinCC if these can be mapped area-wise onto one-dimensional arrays .

Note

During all read accesses and all write accesses, all array elements of a tag are always read or written, respectively. The contents of an array tag which is interconnected with a PLC are always transferred whenever there is a change. The HMI device and the PLC cannot concurrently write data to the same array tag for this reason. Instead of writing data only to a single element, the program writes the entire array to the PLC.

Array elements

Elements of one-dimensional, two-dimensional and three-dimensional arrays in the PLC are indexed by setting an index and the corresponding notation in the tag editor. Array addressing starts at element "0", with arrays of all basic types being valid for element addressing. Read/ write operations are only carried out at the addressed element, and not for the entire array.

Bits and bit tags

Bit access is allowed to all basic data types with the exception of Bool, Real and String. Bit addressing is also allowed at array/structure elements. The Bool data type is set in WinCC when bits and bit tags in the basic data types are addressed.

Single-digit bit numbers are addressed with "/x" or "/0x" (x = bit number). Bit numbers are defined by up to two digits.

Note

With the "Bool" data type in the data types SInt, Int and DInt, after changing the specified bit the complete tag is then written in the PLC again. In the meantime, no check is made as to whether other bits in the tag have since changed. Therefore, the PLC may have only read access to the specified tag.

Structures

User-defined data types are created by means of structures. These structures group tags of different data types. Structures may consist of basic types, arrays and of other structures. In WinCC, only structure elements are addressed and not entire structures.

Structure elements

Structure elements are addressed by means of the name of the structure and of the required structure element. This addressing is separated by point. In addition to basic data types, the structure elements may represent arrays or other structures. Only one-dimensional arrays may be used as a structure element.

Note

The nesting depth of structures is only limited by the maximum length of 128 characters for the address.

See also

Addressing (Page 451)

Addressing syntax (Page 452)

Address multiplexing (Page 455)

Address multiplexing (Basic Panels)

Address multiplexing

Address multiplexing is possible with the CompactLogix, ControlLogix CPU type.

Address multiplexing requires two tags:

- "Tag_1" of data type "String"; contains a logical address such as "HMI:Robot5.Block5" as value.
  The value may change to a second valid address, for example, "HMI:Robot4.Block3".
- "Tag_2" is a tag in which the "Allen-Bradley EtherNet/IP" communication driver is set up as a connection.
  Enter a valid name of an HMI_tag in square brackets as the address.
  - e.g.: "[Tag_1]"
  - The tag must be of the String data type.
  - The square brackets indicate address multiplexing.
  - The address is derived from the actual value in "Tag_1".

  Note

  You can only multiplex entire Allen-Bradley EtherNet/IP addresses. Multiplexing of address elements is not possible. "HMI:Robot[Tag_1].Block5" is an invalid address.

  You can optionally click the arrow right icon in the "Address" column. Replace the "Constant" with the "Multiplex" entry by clicking the arrow on the left edge of the next address dialog box. Now the tag selection list only returns tags of data type "String".

  You can also configure a function triggered by a "change of value" event for multiplexed tags.

See also

Addressing (Page 451)

Addressing syntax (Page 452)

Addressing types (Page 453)

Examples for addressing (Basic Panels)

Example of a table for addressing

The table below defines the basic variants for addressing PLC tags. Other addressing variants are possible by means of combination.

Type	Type	Address
General	PLC tag	Tag name
	Program tag	Program:tagname

Type	Type	Address
Array	Access to an element of a 2-dimensional array	Arraytag[Dim1,Dim2]
	Element of structure array (1-dimensional)	Arraytag[Dim1].structureelement
	Bit in element basic type array (2-dimensional)	Arraytag[Dim1,Dim2]/Bit
Structure	Array in structure	Structuretag.arraytag
	Bit in the element of an array in the substructure	Structuretag.structure2.arraytag [element]/bit

Note

Program tags are addressed by leading the address with the program name derived from the PLC with colon delimiter.

Example: Programname:arraytag[Dim1,Dim2]

Access to array elements

Type	Address
PLC tag	Arraytag[Dim1]
	Arraytag[Dim1,Dim2]
	Arraytag[Dim1,Dim2,Dim3]
Program tag	Programname:arraytag[Dim1]
	Programname:arraytag[Dim1,Dim2]
	Programname:arraytag[Dim1,Dim2,Dim3]

See also

Addressing (Page 451)

Examples: Communication path (Basic Panels)

Example 1:

Connection with a PLC in the same Allen-Bradley rack.

1,0

Number	Meaning
1	Stands for a backplane connection.
0	Stands for a CPU slot number.

Example 2:

Connection with a PLC in remote Allen-Bradley racks. Two Allen-Bradley racks are networked on Ethernet.

1,2,2,190.130.3.101,1,5

Number	Meaning
1	Backplane connection
2	Stands for the CPU slot number of the second Ethernet module.
2	Stands for an Ethernet connection.
190.130.3.101	IP address of a remote AB rack on the network – in particular the third Ethernet module
1	Backplane connection
5	Slot number of the CPU

See also

Addressing (Page 451)

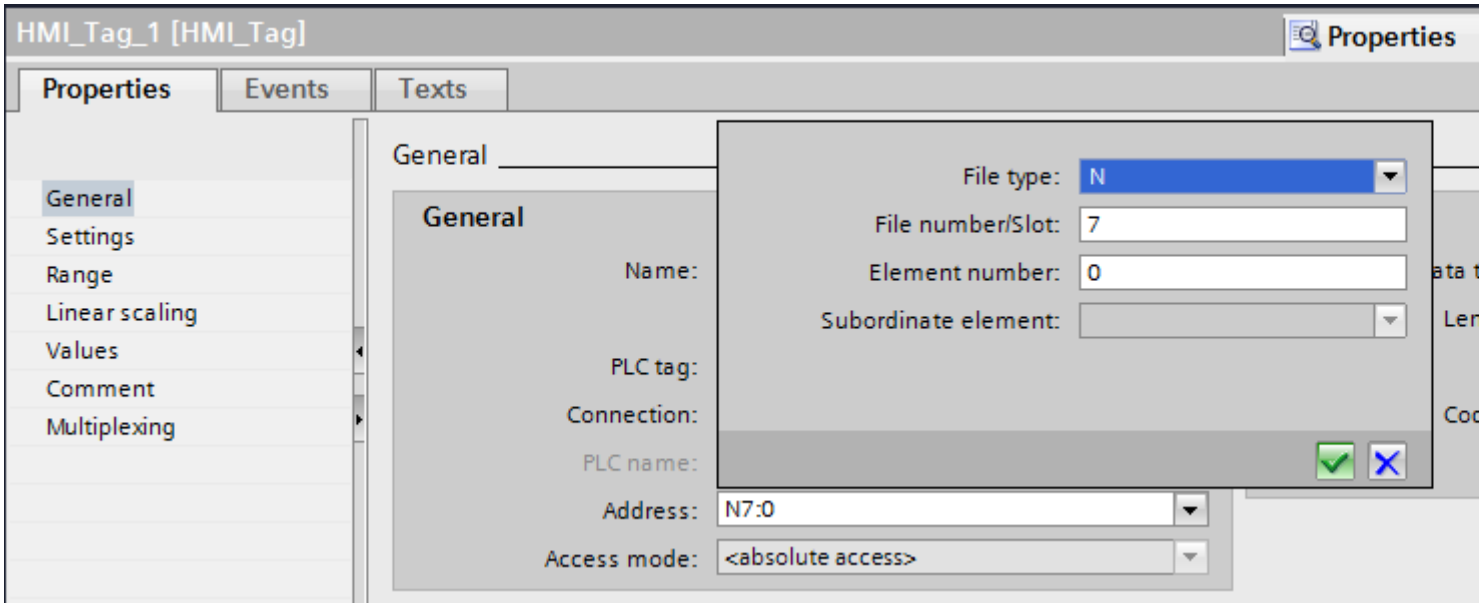
Addressing in the SLC, MicroLogix CPU type (Basic Panels)

Addressing

The addressing in the SLC, MicroLogix CPU type is entered in the following order:

- Operand type
- File number
- Element number

- Subordinate element
- Bit number



The address then appears in the following format without spaces:

- File type file number : Element number . Subordinate element
- e. g. T10:2.ACC

Operand type

You have the following options under operand type:

- I
- O
- S
- B
- C
- T
- R
- F
- N
- ST
- A

File number

Select the number between two limits under file number:

- Low limit
- High limit

The limit values depend on the selected operand type.

Subordinate element

You can select a subordinate element when you have selected one of the following operand types:

- R
- C
- T

See also

Allen-Bradley communication drivers (Page 441)

Commissioning components (Basic Panels)

Transferring a project to the HMI device

1. Switch the HMI device to "transfer mode".
2. Set all necessary transfer parameters.
   - Interface
   - Transfer parameters
   - Target storage location
3. Start the project transfer.
   The project is compiled automatically.
   All compilation and transfer steps are logged to a message window.

Note

If running the CompactLogix PLC with firmware earlier than version 18, you will possibly have to restart the HMI device following the transfer of the PLC program.

You could also terminate the connection before transferring the PLC program and set up the connection again after having completed the transfer of the PLC program.

Interconnecting the PLC with the HMI device

1. Interconnect the PLC with the HMI device using a suitable cable.
2. The message "Connection to PLC .... is established" is output to the HMI device.

Optimizing the configuration (Basic Panels)

Acquisition cycle and update time

The acquisition cycles for the "Area pointers" and of the tags defined in the configuration software are decisive in terms of the update times which can actually be achieved.

The update time is equivalent to the acquisition cycle + transmission time + processing time.

Items to observe when optimizing the update times in configuration data:

- Optimize the maximum and minimum size of the data areas.
- Acquisition cycles which are too short lead to unnecessary load on overall performance. Set the acquisition cycle according to the rate of change of the process values. The rate of temperature changes at a furnace, for example, is significantly slower compared to the speed rate of an electrical drive. A time of approx. 1 second is a benchmark for the acquisition cycle.
- Avoid any gaps when entering the alarm or screen tags in a data area.
- Changes in the PLC can only be detected reliably if these are available at least within the actual acquisition cycle.

Screens

The refresh rate of screens is determined by the type and volume of data to be visualized.

Only configure short acquisition cycles for objects which actually require shorter refresh cycles. This procedure reduces update times.

Trends

The HMI device always updates all bit-triggered trends whose group bit is set in the "Trend transfer area". It resets the bits in the next cycle.

The group bit in the PLC program can only be set again after the HMI device has reset all bits.

Job mailboxes

A high rate and volume of job mailboxes transferred may lead to overload in communication between the HMI device and the PLC.

The HMI device confirms acceptance of the job mailbox by entering the value zero in the first data word of the job mailbox. The HMI device now processes the job for which it requires a certain time slice. It may take the HMI device some time to process a new job mailbox which is transferred in immediate succession to the job mailbox. The next job mailbox is only accepted if sufficient computing resources are available.

Allen-Bradley DF1 (Basic Panels)

Configuring a connection via Allen-Bradley DF1 (Basic Panels)

Introduction

You configure a connection to a PLC with an Allen-Bradley DF1 communication driver in the "Connections" editor of the HMI device.

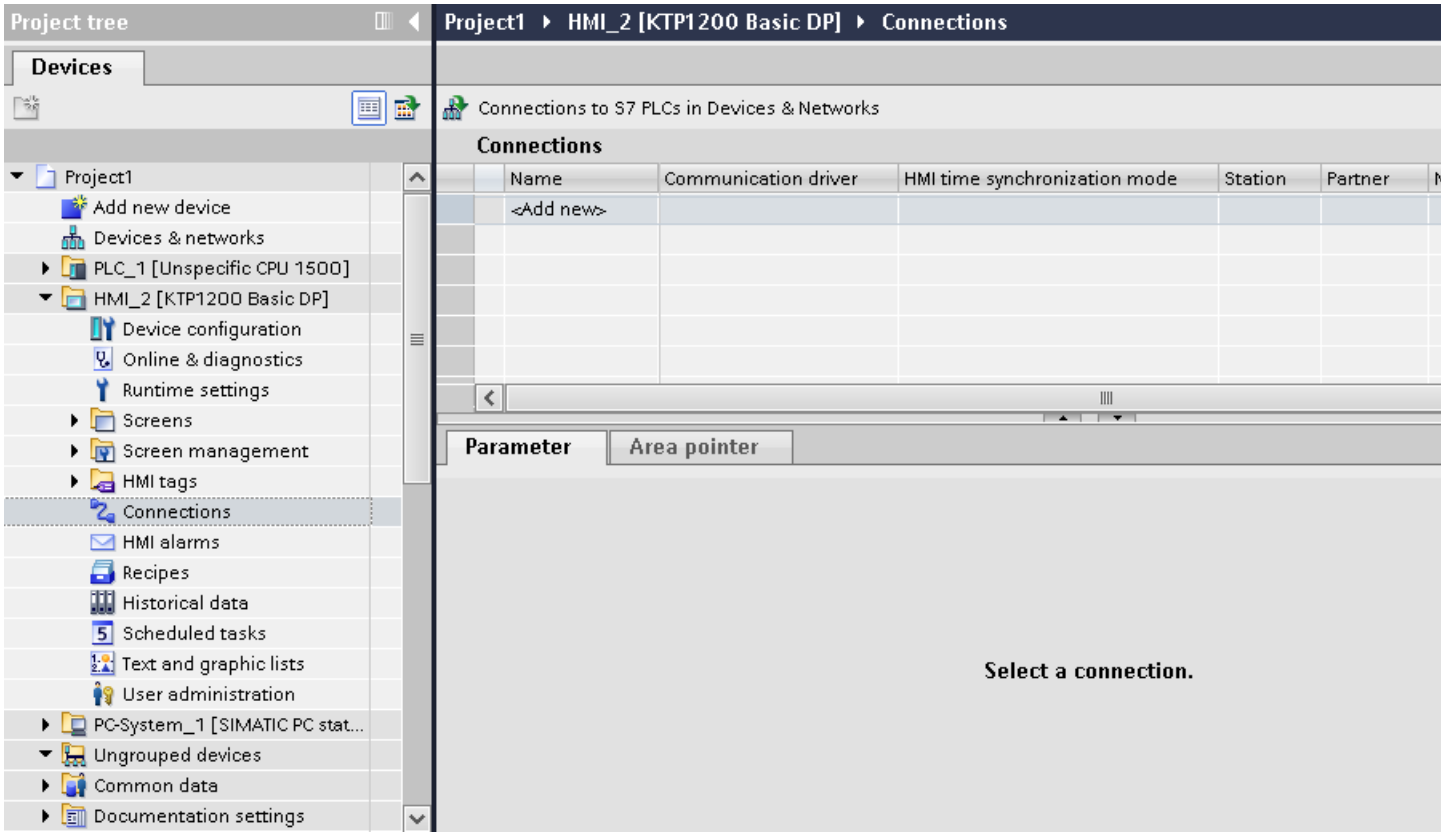
The interfaces are named differently depending on the HMI device.

Requirements

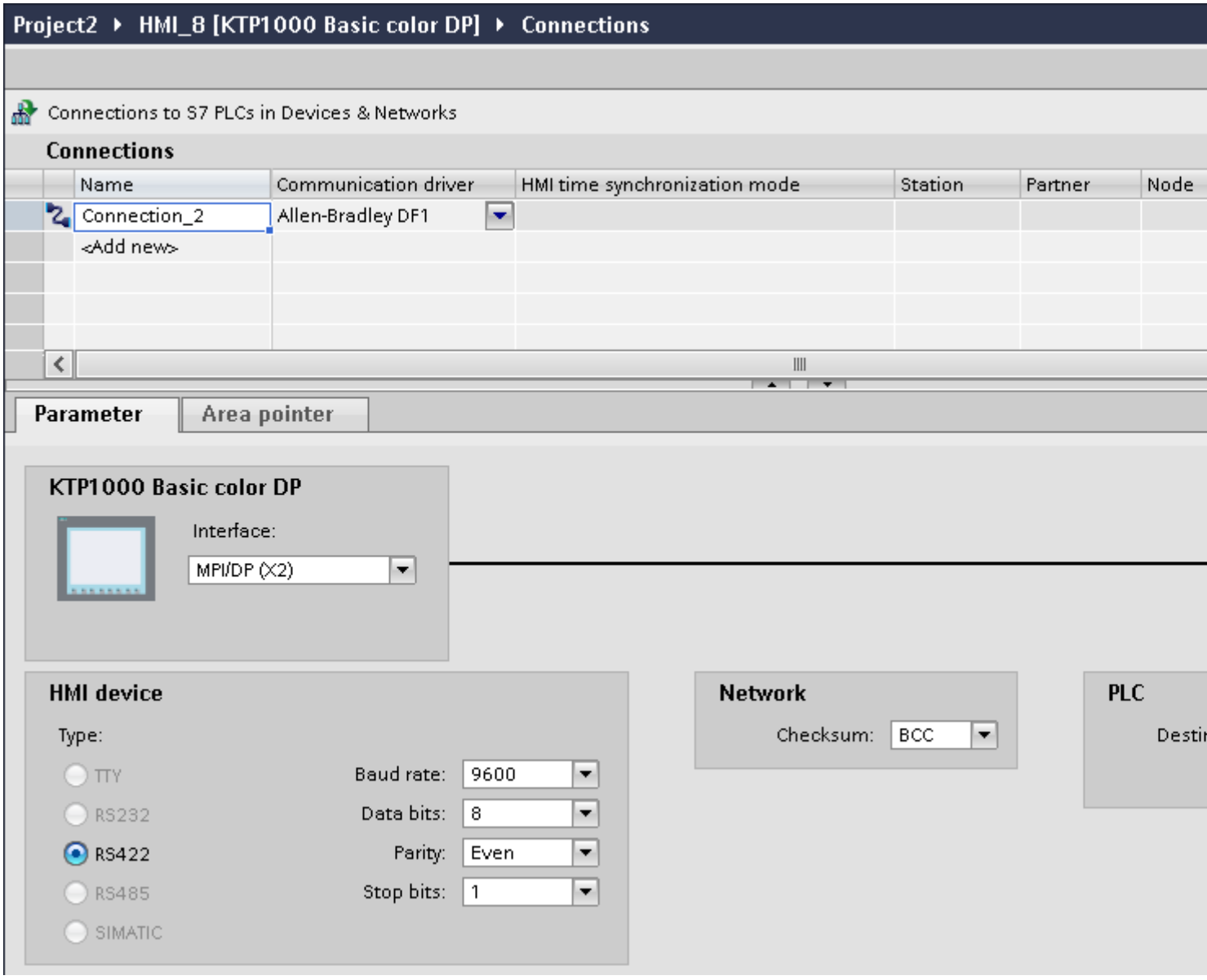
- A project is open.
- An HMI device has been created.

Procedure

1. Double-click the HMI device under "Devices" in the project tree.
2. Double-click the "Connections" item.
3. Double-click "<Add>" in the "Connections" editor.



4. In the "Communication drivers" column, select the "Allen-Bradley DF1" driver.



5. Select all necessary connection parameters for the interface in the Inspector window under "Parameters".

See also

Allen-Bradley communication drivers (Page 441)

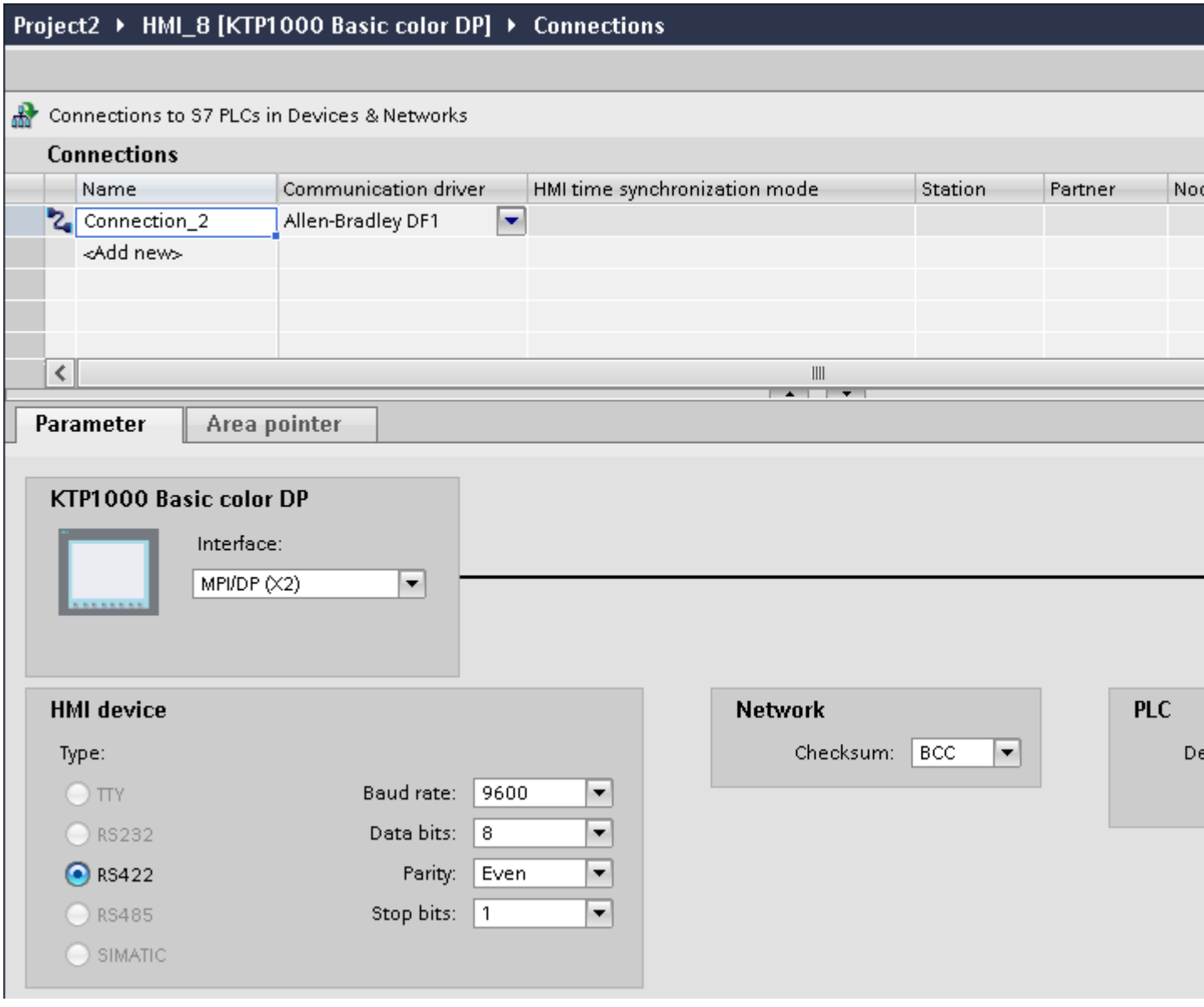
Parameters for the connection (Allen-Bradley DF1) (Page 463)

Parameters for the connection (Allen-Bradley DF1) (Basic Panels)

Parameters to be set

To set the connection parameters, such as addresses and profiles, click the connection that you have created in the "Connections" editor.

The communication partners are displayed schematically in the Inspector window under "Parameters". The "HMI device", "Network", and "PLC" areas are available for assigning parameters according to the interface used.



Parameters for the HMI device

- Interface
  Under "Interface", you select the interface of the HMI device to which the PLC is connected. For additional information, refer to the device manual for the HMI device.
- Type
  Specifies the physical connection used.

  Note

  If you are using the IF1B interface, you must also reconnect the RS-485 receive data and the RTS signal to the rear of the HMI devices via 4 DIP switches.

- Baud rate
  For "Baud rate", select the transmission speed between the HMI device and PLC.
- Data bits
  For "Data bits", you can choose between "7 bits" and "8 bits".
- Parity
  For "Parity", you can choose from "None", "Even", and "Odd".
- Stop bits
  For "Stop bits", you can choose between 1 and 2 bits.

Parameters for the network

- Checksum
  For "Checksum", choose the method for determining the error code: "BCC" or "CRC".

Parameters for the PLC

- Destination address
  For "Destination address", choose the PLC address. If there is a point-to-point DF1 connection, you set the address "0".
- CPU type
  For "CPU type", set the CPU type of the PLC used.

  Note

  Assign the DF1 FULL-DUPLEX driver in the CPU as follows: "NO HANDSHAKING" for "Control Line" and "AUTO-DETECT" for "Embedded Responses".

See also

Allen-Bradley communication drivers (Page 441)

Configuring a connection via Allen-Bradley DF1 (Page 461)

Connecting HMI device to PLC (Basic Panels)

Connections via Allen-Bradley DF1 (Basic Panels)

Connection

The connection is established when you have matched the parameters of the PLC and the HMI device. Special blocks for the connection are not required in the PLC.

Note

Rockwell offers a variety of communication adapters for integrating "DF1 devices" for the DH485, DH, and DH+ networks. Of these connections, the direct connection and the connection via KF2 and KF3 module are approved. None of the other connections have been system-tested by SIEMENS AG and are therefore not approved.

See also

Configuring a connection via Allen-Bradley DF1 (Page 461)

Communication partners for Allen-Bradley DF1 (Basic Panels)

Connectable PLCs

The communication drivers listed below support Allen-Bradley PLCs: :

PLC	DF1 (point-to-point) RS-232	DF1 (point-to-point) RS-422	DF1 (multipoint) over KF2 module to DH+ LAN RS-232/RS-422	DF1 (multipoint) over KF3 module to DH485 LAN RS-232
SLC500	–	–	–	X
SLC501	–	–	–	X
SLC502	–	–	–	X
SLC503	X	–	–	X
SLC504	X	–	X	X
SLC505	X	–	–	X
MicroLogix	X	–	–	X
PLC-5 [1)]	X	X	X	–

[1)] Processors released for PLC-5: PLC-5/11, PLC-5/20, PLC-5/30, PLC-5/40, PLC-5/60, and PLC-5/80.

See also

Configuring a connection via Allen-Bradley DF1 (Page 461)

Communication types (Basic Panels)

PLCs with Allen-Bradley DF1 communication driver

The communication between the HMI device and the following Allen Bradley PLCs is described in this section:

- SLC500
- SLC501
- SLC502
- SLC503
- SLC504
- SLC505
- PLC5
- MicroLogix

In these PLCs the connection is made by the PLC-internal protocols Allen Bradley DF1, Allen Bradley DH485 and Allen Bradley DH+.

The Allen-Bradley DF1 communication driver is used here, the protocol of which is converted into one of the other two PLC-internal protocols in multipoint communication with the communication modules KF2 (Allen Bradley DH+) and KF3(Allen Bradley DH485).

Enabled types of communication with Allen-Bradley DF1

The following communication types are system-tested and enabled:

- HMI (Allen Bradley DF1)
  Point-to-point connection
- HMI (Allen Bradley DF1)
  Via KF2 module to Allen Bradley DH+ (communication with up to 4 PLCs)
- HMI (Allen Bradley DF1)
  Via KF3 module to Allen Bradley DH485 (communication with up to 4 PLCs)

Connectable PLCs

The Allen Bradley DF1 communication driver is available for the following Allen-Bradley PLCs:

PLC	DF1 (point-to-point) RS 232	DF1 (point-to-point) RS 422	DF1 (multipoint) via KF2 module to DH+ LAN RS 232/RS 422	DF1 (multipoint) via KF3 module to DH485 LAN RS 232 [2]
SLC500	–	–	–	X
SLC501	–	–	–	X
SLC502	–	–	–	X
SLC503	X [2]	–	–	X
SLC504	X [2]	–	X	X
SLC505	X [2]	–	–	X
MicroLogix	X [2]	–	–	X
PLC-5 [1]	X	X	X	–

[1] Only the following processors are approved for PLC-5: PLC-5/11, PLC-5/20, PLC-5/30, PLC-5/40, PLC-5/60 und PLC-5/80.

[2] For HMI devices which only have an RS 422/485 interface and the communication partner is an RS 232 interface, the RS 422/232 converter is tested and approved.
Order number: 6AV6 671-8XE00-0AX0

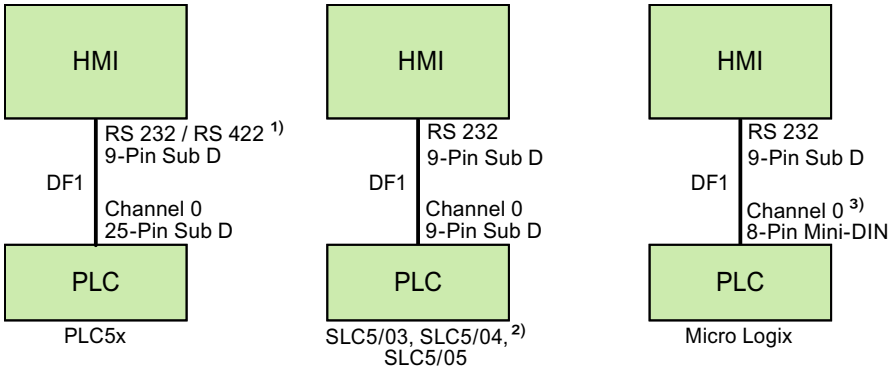
See also

Configuring a connection via Allen-Bradley DF1 (Page 461)

DF1 protocol with multi-point connection (Basic Panels)

Point-to-point connection with DF1 protocol

Only point-to-point connections can be established with the DF1 protocol.



1) Only RS 232 is possible for Panel PC and PC.

2) A point-to-point connection to the SLC500, SLC501, and SLC502 PLCs via DF1 is not possible.

3) For MicroLogix ML1500 LRP, Channel 1 (9-pin Sub D) is also possible.

Connecting cable

HMI panel interface used	For connection to PLC5x	For connection to SLC5/03, SLC5/04, SLC5/05	For connection to MicroLogix
RS 232 9-pin	Allen-Bradley cable 1784-CP10	Allen-Bradley cable 1747-CP3	Allen-Bradley cable 1761-CBL-PM02
RS 422 9-pin	Connecting cable 9-pin Sub D RS 422	—	—

Refer to the relevant device manual to determine which HMI device interface is to be used.

The cable pin assignments can be found in Section "Connecting cables for Allen-Bradley".

See also

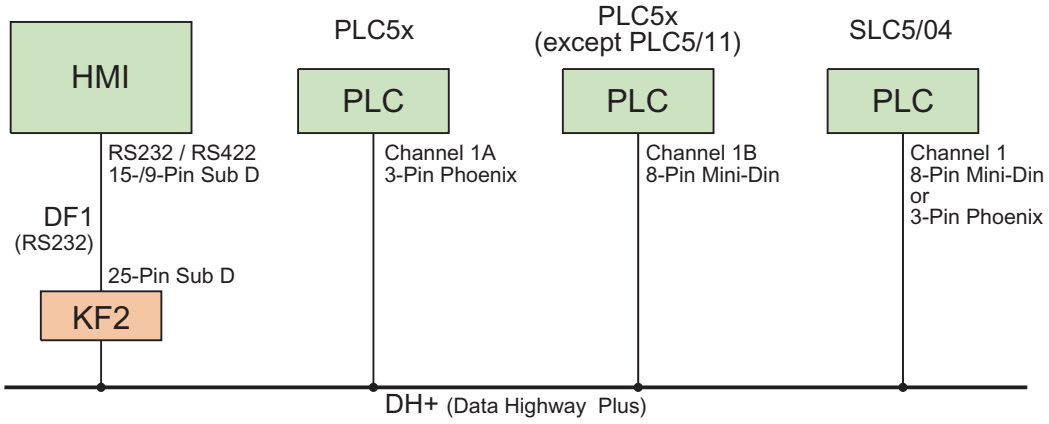
Configuring a connection via Allen-Bradley DF1 (Page 461)

Allen-Bradley communication drivers (Page 441)

DF1 protocol with multi-point connection via KF2 module (Basic Panels)

DF1 protocol with multi-point connection via KF2 module to DH+ LAN

The use of a KF2 protocol interface module enables a connection to be made to PLCs in the DH+ LAN (Data Highway Plus Local Area Network).



Connecting cable

HMI panel interface used	For connection to KF2 interface module
RS 232 9-pin	Allen-Bradley cable 1784-CP10 and 25-pin socket/socket adapter
RS 422 9-pin	9-pin Sub D RS 422 connecting cable and 25-pin female/female adapter

Refer to the Allen-Bradley documentation for the cable connection from the PLCs to the DH+ data bus.

Refer to the relevant device manual to determine which HMI device interface is to be used.

The cable pin assignments can be found in Section "Connecting cables for Allen-Bradley".

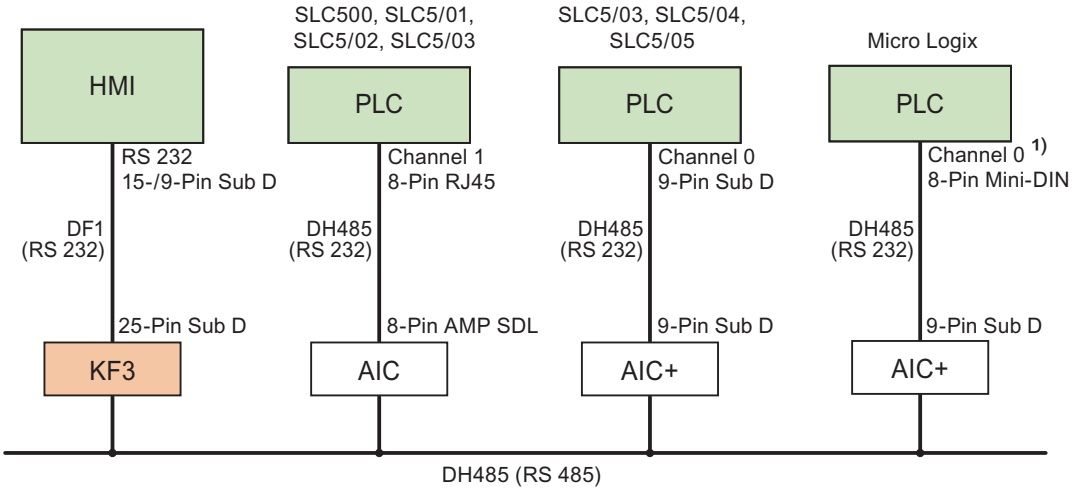
See also

Configuring a connection via Allen-Bradley DF1 (Page 461)

Allen-Bradley communication drivers (Page 441)

DF1 protocol with multi-point connection via KF3 module (Basic Panels)

DF1 protocol with multi-point connection via KF3 module to DH485 LAN



1) For MicroLogix ML1500 LRP, Channel 1 (9-pin Sub D) is also possible.

Connecting cable

HMI panel interface used	For connection to KF3 interface module
RS 232 9-pin	Allen-Bradley cable 1784-CP10 and 25-pin socket/socket adapter

Refer to the relevant device manual to determine which HMI device interface is to be used.

The cable pin assignments can be found in Section "Connecting cables for Allen-Bradley".

See also

Configuring a connection via Allen-Bradley DF1 (Page 461)

Allen-Bradley communication drivers (Page 441)

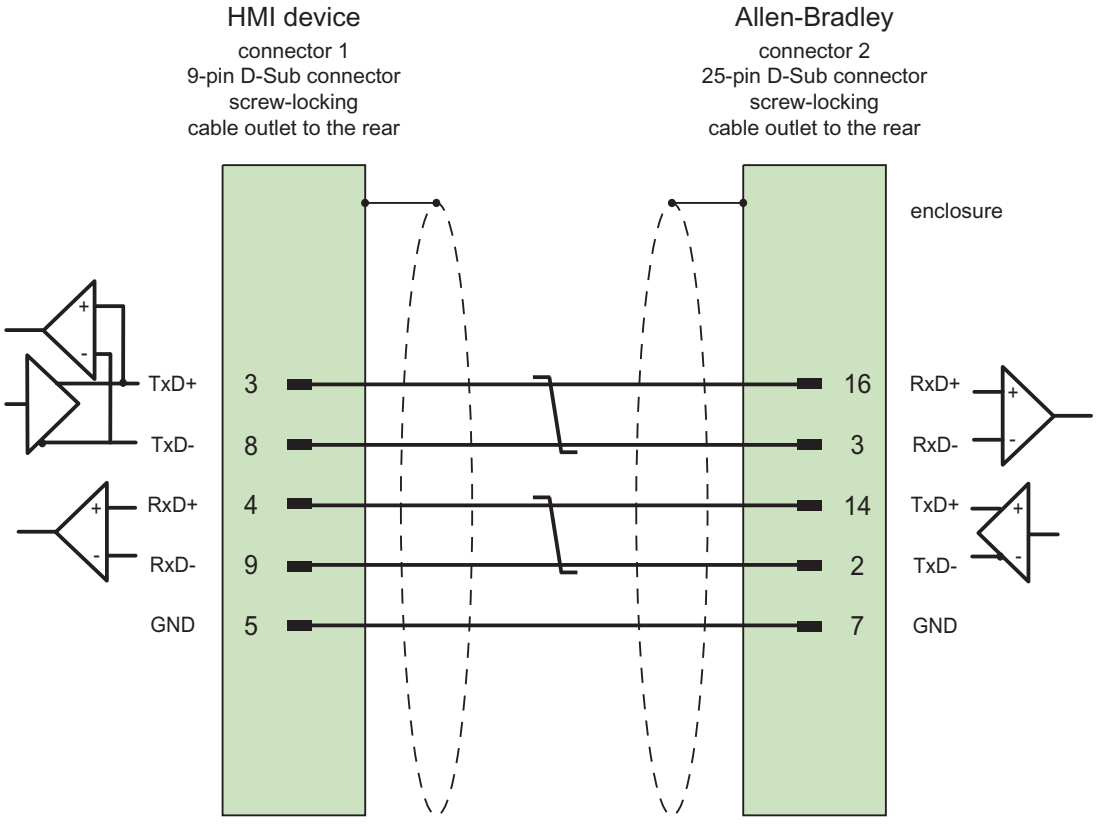
Connecting cables for Allen-Bradley DF1 (Basic Panels)

Connecting cable 9-pin Sub D RS 422 for Allen-Bradley (Basic Panels)

Connecting cable 9-pin Sub D RS 422

For interconnecting the HMI device (RS 422, 9-pin sub D) - PLC5x, KF2, KF3

You require an additional 25-pin, female / female adapter (gender changer) for interconnections with KF2 and KF3.



Shield with large-area contact to housing at both ends, interconnected shield contacts
Cable: 3 x 2 x 0.14 mm^2, shielded,
max. length 60 m

See also

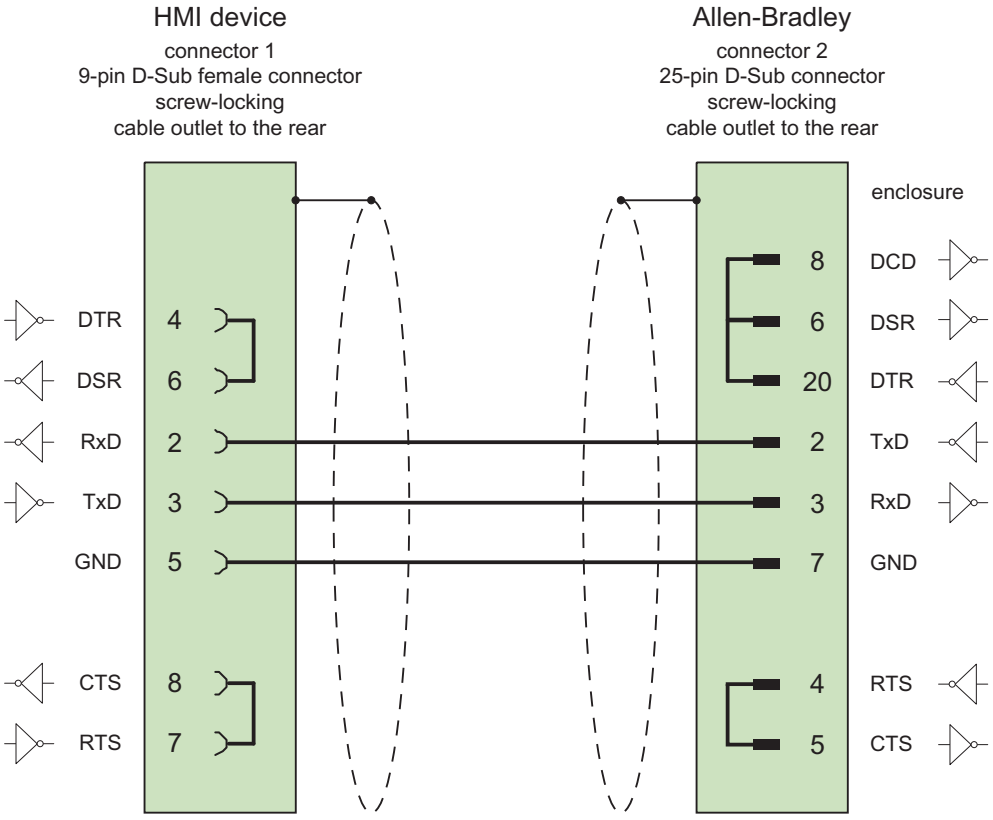
Connections via Allen-Bradley DF1 (Page 465)

Connecting cable 1784-CP10, RS 232, for Allen-Bradley (Basic Panels)

Allen-Bradley cable 1784-CP10

For interconnecting the HMI device (RS 232, 9-pin sub D) - PLC5x, KF2, KF3

You require an additional 25-pin, female / female adapter (gender changer) for interconnections with KF2 and KF3.



Screen connected with housing over large area on both sides

max. length 15 m

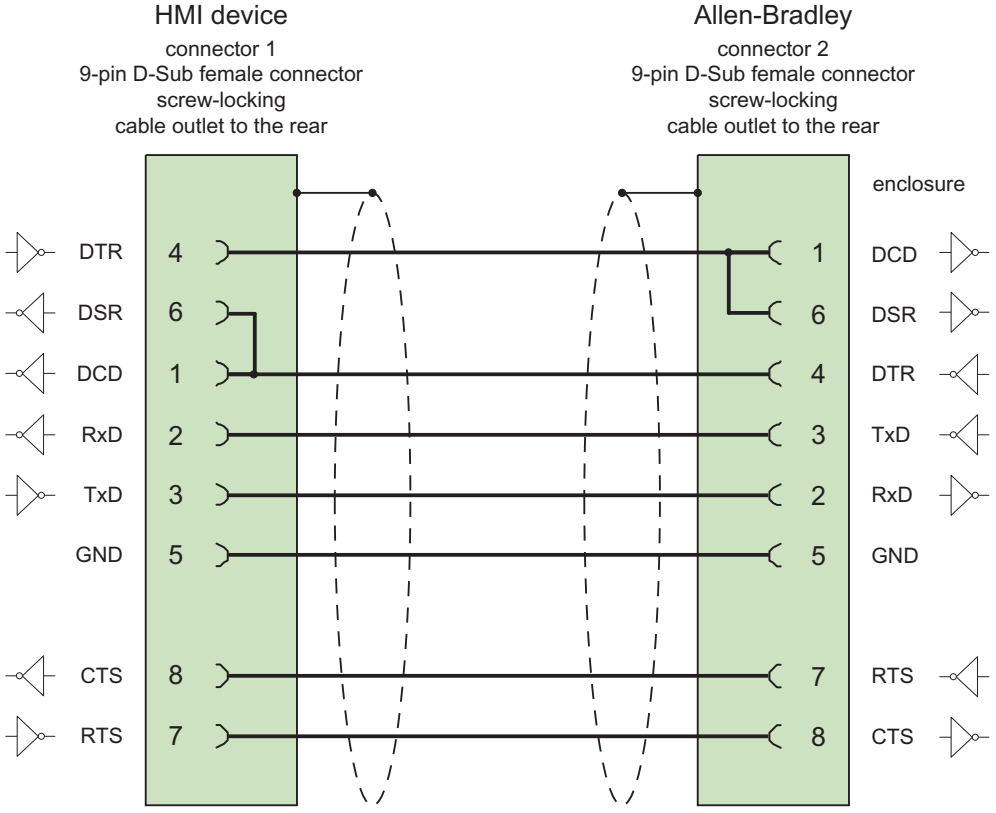
See also

Connections via Allen-Bradley DF1 (Page 465)

Connecting cable 1747-CP3, RS-232, for Allen-Bradley (Basic Panels)

Allen-Bradley cable 1747-CP3

For interconnecting the HMI device (RS 232, 9-pin sub D) - SLC503, SLC504, SLC505 (Channel 0), AIC+



Screen connected with housing over large area on both sides

max. length 3 m

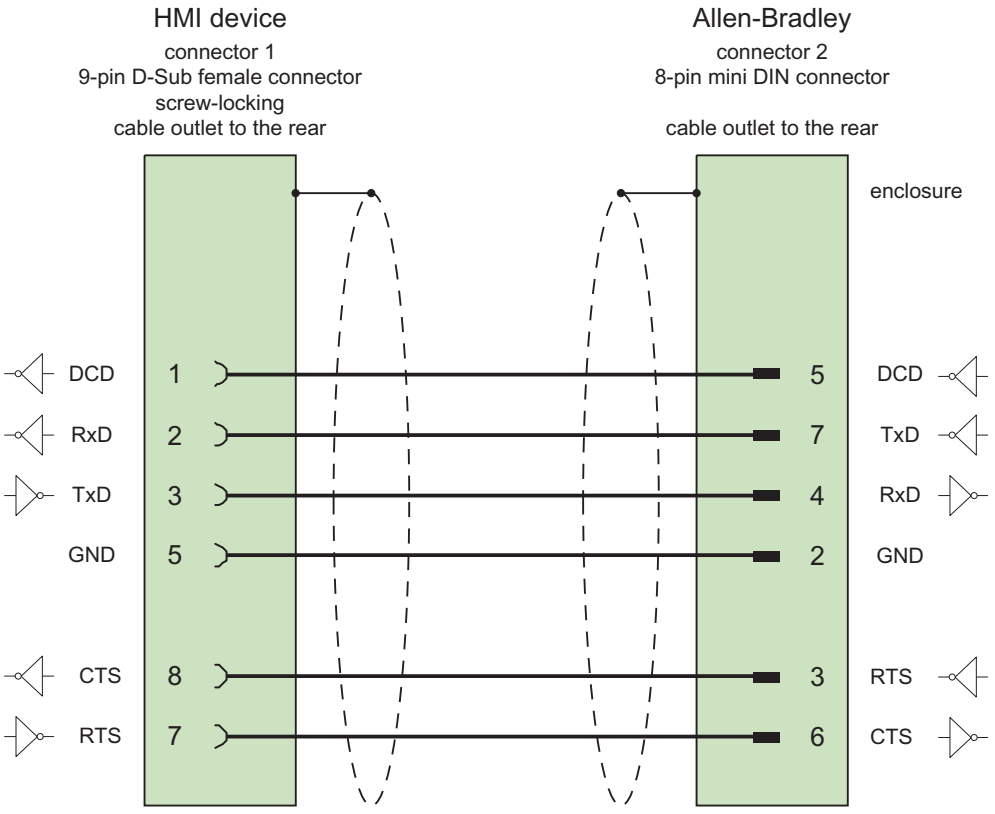
See also

Connections via Allen-Bradley DF1 (Page 465)

Connecting cable 1761-CBL-PM02, RS-232, for Allen-Bradley (Basic Panels)

Allen-Bradley cable 1761-CBL-PM02

For interconnecting the HMI device (RS 232, 9-pin sub D) - Micro Logix, AIC+



Screen connected with housing over large area on both sides

max. length 15 m

See also

Connections via Allen-Bradley DF1 (Page 465)

Performance features of communication (Basic Panels)

Permitted data types for Allen-Bradley DF1 (Basic Panels)

Permitted data types for Allen-Bradley DF1

The table lists the user data types that can be used when configuring tags and area pointers.

Data type	Operand type	Length
ASCII	A [1)]	1 to 80 characters
Bool	N, R, C, T, B, S, I, O	1 bit
Int	N, R, C, T, S	2 bytes
DInt	N	4 bytes

Data type	Operand type	Length
UInt	N, R, C, T, B, I, O	2 bytes
UDInt	N	4 bytes
Real	N, F [1]	4 bytes

[1] Selectable depending on the selected CPU type.

Abbreviations

In WinCC, formats of the data types are abbreviated as follows:

- UNSIGNED INT = UInt
- UNSIGNED LONG = UDInt
- SIGNED INT = Int
- SIGNED LONG = DInt

Distinctive features for connections with Allen-Bradley DF1

With Allen Bradley DF1, array tags may only be used for discrete alarms and trends.

Note

I/O modules with 8 or 16 ports occupy one data word on the PLC.

I/O modules with 24 or 32 ports occupy two data words.

The HMI device does not output an error message if using non-existent bits.

You should always make sure that I/O modules with 8 or 24 ports only occupy the bits that are actually assigned to a port.

See also

Allen-Bradley communication drivers (Page 441)

Supported CPU types for Allen-Bradley DF1 (Basic Panels)

CPU types

The following CPU types are supported for configuring the Allen-Bradley DF1 communication driver.

- SLC
  - SLC500
  - SLC501
  - SLC502
  - SLC503
  - SLC504
  - SLC505
- MicroLogix
  - MicroLogix 1x00
  - MicroLogix 1100 / 1400
- PLC 5
  - PLC-5/11
  - PLC-5/20
  - PLC-5/40
  - PLC-5/60
  - PLC-5/80

See also

Allen-Bradley communication drivers (Page 441)

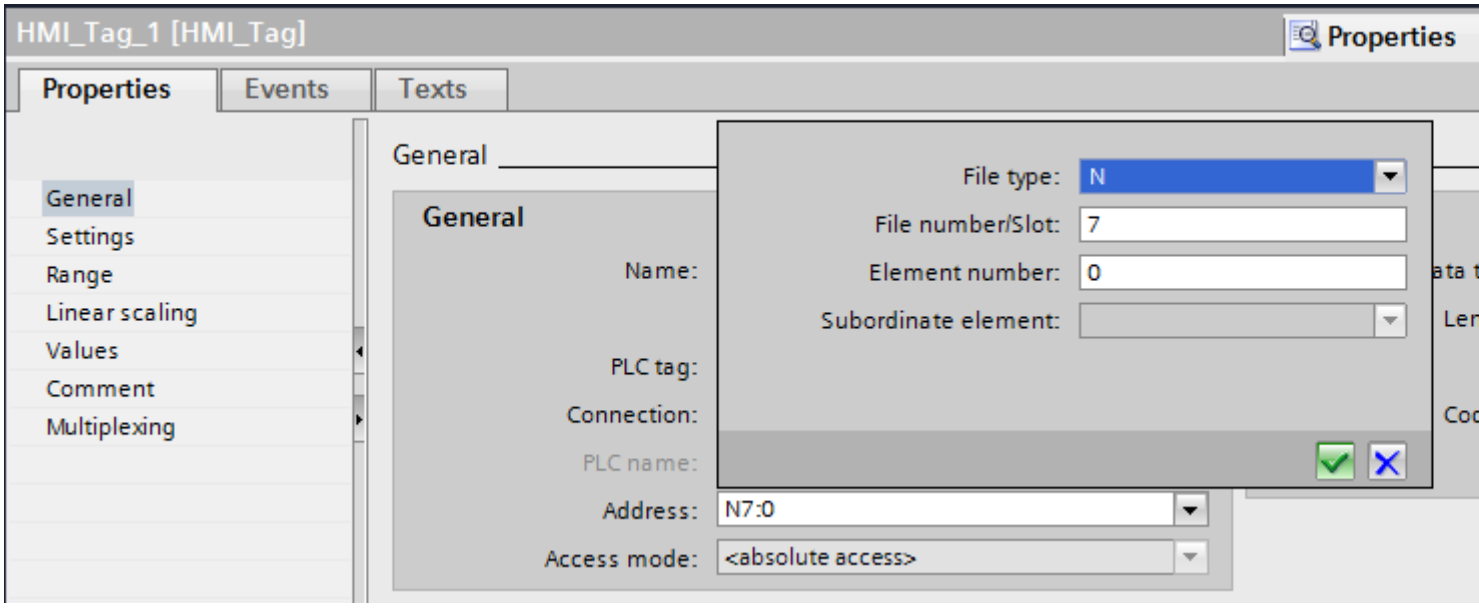
Addressing (Basic Panels)

Addressing

The addressing is entered in the following order in the Allen-Bradley DF1 communication driver:

- Operand type
- File number
- Element number

- Subordinate element
- Bit number



The address then appears in the following format without spaces:

- File type file number : Element number . Subordinate element
- e. g. T8:2.ACC

Operand type

You have the following options under operand type:

- I
- O
- S
- B
- T
- C
- R
- N
- A
- D only for PLC5 CPU type

File number

Select the number between two limits under file number:

- Low limit
- High limit

The limit values depend on the selected file type.

Subordinate element

You can select a subordinate element when you have selected one of the following data types:

- R
- C
- T

See also

Allen-Bradley communication drivers (Page 441)

Address areas for Allen Bradley DF1 (Page 477)

Address areas for Allen Bradley DF1 (Basic Panels)

MicroLogix

Address areas	Data types					
	Bool	Int	UInt	DInt	UDInt	Real
N	N7:0/0 - N255:255/15	N7:0 - N255:255	N7:0 - N255:255	N7:0 - N255:254	N7:0 - N255:254	N7:0 - N255:254
F	--	--	--	--	F8:0 - F255:255	--
R	R6:0.EN - R255:255.ER - R255:255.DN - R255:255.FD - R255:255.IN - R255:255.EU - R255:255.EM - R255:255.UL	R6:0.LEN - R255:255.POS	R6:0.LEN - R255:255.POS	--	--	--
C	C5:0.CU - C255:255.CD - C255:255.DN - C255:255.OV - C255:255.UN	C5:0.PRE - C255:255.ACC	C5:0.PRE - C255:255.ACC	--	--	--

Address areas	Data types					
	Bool	Int	UInt	DInt	UDInt	Real
T	T4:0.DN - T255:255.TT - T255:255.EN	T4:0.PRE - T255:255.ACC	T4:0.PRE - T255:255.ACC	--	--	--
B	B3:0/0 - B255:255/15	--	B3:0 - B255:255	--	--	--
S	S2:0/0 - S2:65/15	S2:0 - S2:65	--	--	--	--
I	I0:0/0 - I38:255/15	--	I0:0 - I38:255	--	--	--
O	O0:0/0 - O38:255/15	--	O0:0/0 - O38:255	--	--	--

SLC500

Address areas	Data types					
	Bool	Int	UInt	DInt	UDInt	Real
N	N7:0/0 - N255:255/15	N7:0 - N255:255	N7:0 - N255:255	N7:0 - N255:254	N7:0 - N255:254	N7:0 - N255:254
R	R6:0.EN - R255:255.ER - R255:255.DN - R255:255.FD - R255:255.IN - R255:255.EU - R255:255.EM - R255:255.UL	R6:0.LEN - R255:255.POS	R6:0.LEN - R255:255.POS	--	--	--
C	C5:0.CU - C255:255.CD - C255:255.DN - C255:255.OV - C255:255.UN	C5:0.PRE - C255:255.ACC	C5:0.PRE - C255:255.ACC	--	--	--
T	T4:0.DN - T255:255.TT - T255:255.EN	T4:0.PRE - T255:255.ACC	T4:0.PRE - T255:255.ACC	--	--	--
B	B3:0/0 - B255:255/15	--	B3:0 - B255:255	--	--	--
S	S2:0/0 - S2:15/15	S2:0 - S2:15	--	--	--	--
I	I0:0/0 - I38:255/15	--	I0:0 - I38:255	--	--	--
O	O0:0/0 - O38:255/15	--	O0:0 - O38:255	--	--	--

SLC501/502

Address areas	Data types					
	Bool	Int	UInt	DInt	UDInt	Real
N	N7:0/0 - N255:255/15	N7:0 - N255:255	N7:0 - N255:255	N7:0 - N255:254	N7:0 - N255:254	N7:0 - N255:254
R	R6:0.EN - R255:255.ER - R255:255.DN - R255:255.FD - R255:255.IN - R255:255.EU - R255:255.EM - R255:255.UL	R6:0.LEN - R255:255.POS	R6:0.LEN - R255:255.POS	--	--	--
C	C5:0.CU - C255:255.CD - C255:255.DN - C255:255.OV - C255:255.UN	C5:0.PRE - C255:255.ACC	C5:0.PRE - C255:255.ACC	--	--	--
T	T4:0.DN - T255:255.TT - T255:255.EN	T4:0.PRE - T255:255.ACC	T4:0.PRE - T255:255.ACC	--	--	--
B	B3:0/0 - B255:255/15	--	B3:0 - B255:255	--	--	--
S	S2:0/0 - S2:32/15	S2:0 - S2:32	--	--	--	--
I	I0:0/0 - I38:255/15	--	I0:0 - I38:255	--	--	--
O	O0:0/0 - O38:255/15	--	O0:0 - O38:255	--	--	--

PLC5

Address areas	Data types						
	Bool	Int	UInt	DInt	UDInt	Real	ASCII
N	N3:0/0 - N999:999/15	N3:0 - N999:999	N3:0 - N999:999	N3:0 - N999:999	N3:0 - N999:998	N3:0 - N999:998	--
F	--	--	--	--	--	F3:0 - F999:999	--
A	--	--	--	--	--	--	A3:0 - A999:999

Address areas	Data types						
	Bool	Int	UInt	DInt	UDInt	Real	ASCII
R	R3:0.EN - R999:999.ER - R999:999.DN - R999:999.FD - R999:999.IN - R999:999.EU - R999:999.EM - R999:999.UL	R3:0.LEN - R999:999.POS	R3:0.LEN - R999:999.POS	--	--	--	--
C	C3:0.CU - C999:999.CD - C999:999.DN - C999:999.OV - C999:999.UN	C3:0.PRE - C999:999.ACC	C3:0.PRE - C999:999.ACC	--	--	--	--
T	T3:0.DN - T999:999.TT - T999:999.EN	T3:0.PRE - T999:999.ACC	T3:0.PRE - T999:999.ACC	--	--	--	--
B	B3:0/0 - B999:999/15	--	B3:0 - B999:999	--	--	--	--
S	S2:0/0 - S2:127/15	S2:0 - S2:127	--	--	--	--	--
I	I1:0/0 - I1:277/17	--	I1:0 - I1:277	--	--	--	--
O	O0:0/0 - O0:277/17	--	O0:0 - O0:277	--	--	--	--
D	D3:0/0 - D999:999/15	D3:0 - D999:999	D3:0 - D999:999	--	D3:0 - D999:998	--	--

See also

Allen-Bradley communication drivers (Page 441)

Addressing (Page 475)

Commissioning components (Basic Panels)

Transferring a project to the HMI device

1. Switch the HMI device to "transfer mode".
2. Set all necessary transfer parameters.
   - Interface
   - Transfer parameters
   - Target storage location
3. Start the project transfer.
   The project is compiled automatically.
   All compilation and transfer steps are logged to a message window.

Interconnecting the PLC with the HMI device

1. Interconnect the PLC with the HMI device using a suitable cable.
2. The message "Connection to PLC .... is established" is output to the HMI device.

Optimizing the configuration (Basic Panels)

Acquisition cycle and update time

The acquisition cycles for the "Area pointers" and of the tags defined in the configuration software are decisive in terms of the update times which can actually be achieved.

The update time is equivalent to the acquisition cycle + transmission time + processing time.

Items to observe when optimizing the update times in configuration data:

- Optimize the maximum and minimum size of the data areas.
- Acquisition cycles which are too short lead to unnecessary load on overall performance. Set the acquisition cycle according to the rate of change of the process values. The rate of temperature changes at a furnace, for example, is significantly slower compared to the speed rate of an electrical drive. A time of approx. 1 second is a benchmark for the acquisition cycle.
- Avoid any gaps when entering the alarm or screen tags in a data area.
- Changes in the PLC can only be detected reliably if these are available at least within the actual acquisition cycle.

Screens

The refresh rate of screens is determined by the type and volume of data to be visualized.

Only configure short acquisition cycles for objects which actually require shorter refresh cycles. This procedure reduces update times.

Trends

The HMI device always updates all bit-triggered trends whose group bit is set in the "Trend transfer area". It resets the bits in the next cycle.

The group bit in the PLC program can only be set again after the HMI device has reset all bits.

Job mailboxes

A high rate and volume of job mailboxes transferred may lead to overload in communication between the HMI device and the PLC.

The HMI device confirms acceptance of the job mailbox by entering the value zero in the first data word of the job mailbox. The HMI device now processes the job for which it requires a certain time slice. It may take the HMI device some time to process a new job mailbox which is transferred in immediate succession to the job mailbox. The next job mailbox is only accepted if sufficient computing resources are available.

Data exchange (Basic Panels)

Area pointers for Allen-Bradley (Basic Panels)

Area pointers for connections using an Allen-Bradley communication driver

You use an area pointer to access a data area in the PLC.

For more detailed information on area pointers and their configuration, refer to Section "Data exchange using area pointers (Page 649)".

Distinctive features for connections via Allen-Bradley EtherNet/IP

You can configure the following area pointers

Area pointer	Allen-Bradley EtherNet/IP	Allen-Bradley DF1
Screen number	Yes	Yes
Date/time	Yes	Yes
Date/time PLC	Yes	Yes
Coordination	Yes	Yes
Project ID	Yes	Yes
Job mailbox	Yes	Yes
Data record	Yes	Yes

Restrictions Allen-Bradley Ethernet/IP

The following restrictions apply for configuring area pointers.

CPU type	Data types	File types
ControlLogix,CompactLogix	Int, UInt	--
SLC, MicroLogix	Int, UInt	N, B

Restrictions Allen-Bradley DF1

The following restrictions apply for configuring area pointers.

CPU type	Data types	File types
MicroLogix	--	N, O, I, B
SLC50x	--	N, O, I, B
PLC5	--	N, O, I, B

See also

Data exchange using area pointers (Page 649)

Trends (Basic Panels)

Trends

A trend is the graphical representation of one or more values from the PLC. The value is read out time-triggered for Basic Panels.

For additional information see:

AUTOHOTSPOT

Time-triggered trends

The HMI device reads in the trend values cyclically at an interval specified in the configuration.

Time-triggered trends are suitable for continuous curves, such as the operating temperature of a motor.

Alarms (Basic Panels)

Configuring alarms (Basic Panels)

Configuring alarms for non-integrated connections

Several steps are necessary to configure alarms such as warnings, error messages and acknowledgement.

- Step 1: Create tags
- Step 2: Configure alarms
- Step 3: Configure acknowledgment

You can find additional information in the section:

AUTOHOTSPOT

Distinctive features when configuring alarms

If you are configuring connections of HMI devices to PLCs of other manufacturers, note the following distinctive features when configuring:

- Data types of the tags
- Addressing of tags
- How the bit positions are counted

Restrictions

Only tags whose "File type" is "N", "O", "I", "S" and "B" are allowed for use as a "trigger tag" for discrete alarms. These tags are only valid for the data types "Int" and "UInt".

Data types

For connections with an Allen-Bradley communication driver, the following data types are supported:

Communication drivers	PLC	Permitted data types	
		Discrete alarms	Analog alarms
Allen-Bradley DF1	SLC500, SLC501, SLC502, SLC503, SLC504, SLC505, PLC5, MicroLogix	Int, UInt	Int, UInt, Long, ULong, Real
Allen-Bradley EtherNet/IP	ControlLogix, Compact-Logix, SLC, Micrologix	Int, UInt	SInt, USInt, Int, UInt, DInt, UDInt, Real

How the bit positions are counted

For connections with an Allen-Bradley communication driver, the following counting method applies:

How the bit positions are counted	Left byte								Right byte							
In Allen-Bradley PLCs	15							8	7							0
In WinCC you configure:	15							8	7							0

Acknowledgment of alarms (Basic Panels)

Procedure

Create suitable tags on the PLC to acknowledge an error alarm. You assign these tags to an alarm in the "Bit messages" editor. You make the assignment in "Properties > Acknowledgment".

Distinction in terms of acknowledgment:

- Acknowledgment by the PLC
- Acknowledgment on the HMI device

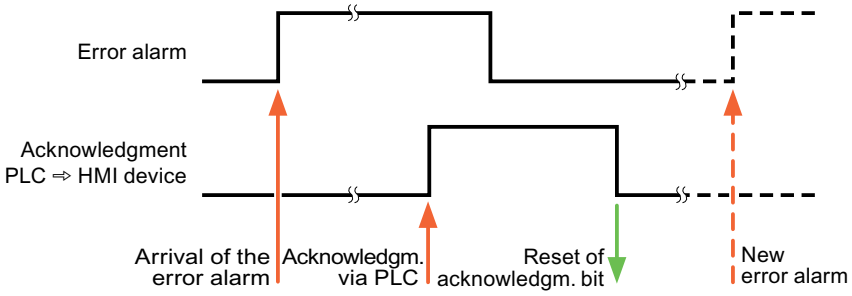
Acknowledgment by the PLC

In "PLC acknowledgment tag", configure the tag or array tag and the bit number that the HMI device uses to identify a PLC acknowledgment.

A bit set in the tag triggers acknowledgment of the assigned error alarm bit at the HMI device. This tag bit returns a function similar to acknowledgment on the HMI device which is triggered by pressing the "ACK" button, for example.

The acknowledgment bit must be located in the same tag as the bit for the error alarm.

Reset the acknowledgment bit before setting the bit in the alarm area again. The figure below shows the pulse diagram.



Acknowledgment on the HMI device

In the "HMI acknowledgment tag" area, configure the tag or array tag as well as the bit number that the HMI device writes to the PLC after acknowledgment. Make sure when you use an array tag that it is not longer than 6 words.

To always create a signal change when setting an assigned acknowledgment bit of a discrete alarm that must be acknowledged, the HMI device will reset the acknowledgment bit assigned to the alarm as soon as it detects an alarm subject to acknowledgment and write the acknowledgment tag in the PLC. There will be a certain delay between detecting the message and writing the acknowledgment tag in the PLC because the HMI device has to process the operations.

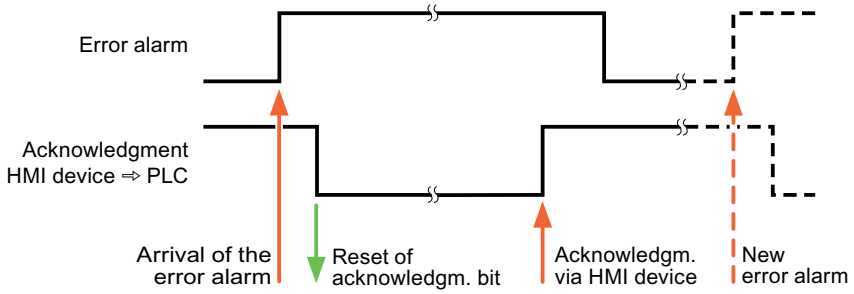
If a discrete alarm subject to acknowledgment is acknowledged by the HMI device, then the corresponding bit in the assigned acknowledgment tag will be set. The entire acknowledgment tag is then written to the PLC by the HMI device. This allows the PLC to recognize that a certain alarm message has been acknowledged at the HMI device.

Note

All alarm bits acknowledged since the last Runtime start will remain in the acknowledgment tag until a new incoming of the respective discrete alarms is detected.

This area should only be read by the PLC because the entire section of the HMI device will be overwritten once the next acknowledgment tag is written.

The figure below shows the pulse diagram.



1.13.3.2 Mitsubishi (Basic Panels)

Mitsubishi communication drivers (Basic Panels)

Introduction

This section describes the communication between an HMI device and PLCs that use Mitsubishi communication drivers.

The following communication drivers are supported:

- Mitsubishi MC TCP/IP
- Mitsubishi FX

Note

Exceeding value ranges with Mitsubishi MC and Mitsubishi FX

With the following data types, the Mitsubishi MC and Mitsubishi FX communication drivers do not check whether the value of a recipe tag exceeds the value range of the PLC tags.

- 4-bit block
- 12-bit block
- 20-bit block
- 24-bit block
- 28-bit block

Data exchange

Data is exchanged by means of tags or area pointers.

- Tags
  The PLC and the HMI device use process values for data exchange. You create tags in the configuration that point to addresses in the PLC. The HMI device reads the value from the defined address, and then displays it. The operator may also enter values on the HMI device, which are then written to the address in the PLC.
- Area pointers
  Area pointers are used to exchange specific data and are only set up when these data are used.

Mitsubishi MC TCP/IP (Basic Panels)

Configuring a connection via Mitsubishi MC TCP/IP (Basic Panels)

Introduction

You configure a connection to a PLC with a Mitsubishi MC TCPI/IP communication driver in the "Connections" editor of the HMI device.

The Ethernet interfaces are named differently depending on the HMI device.

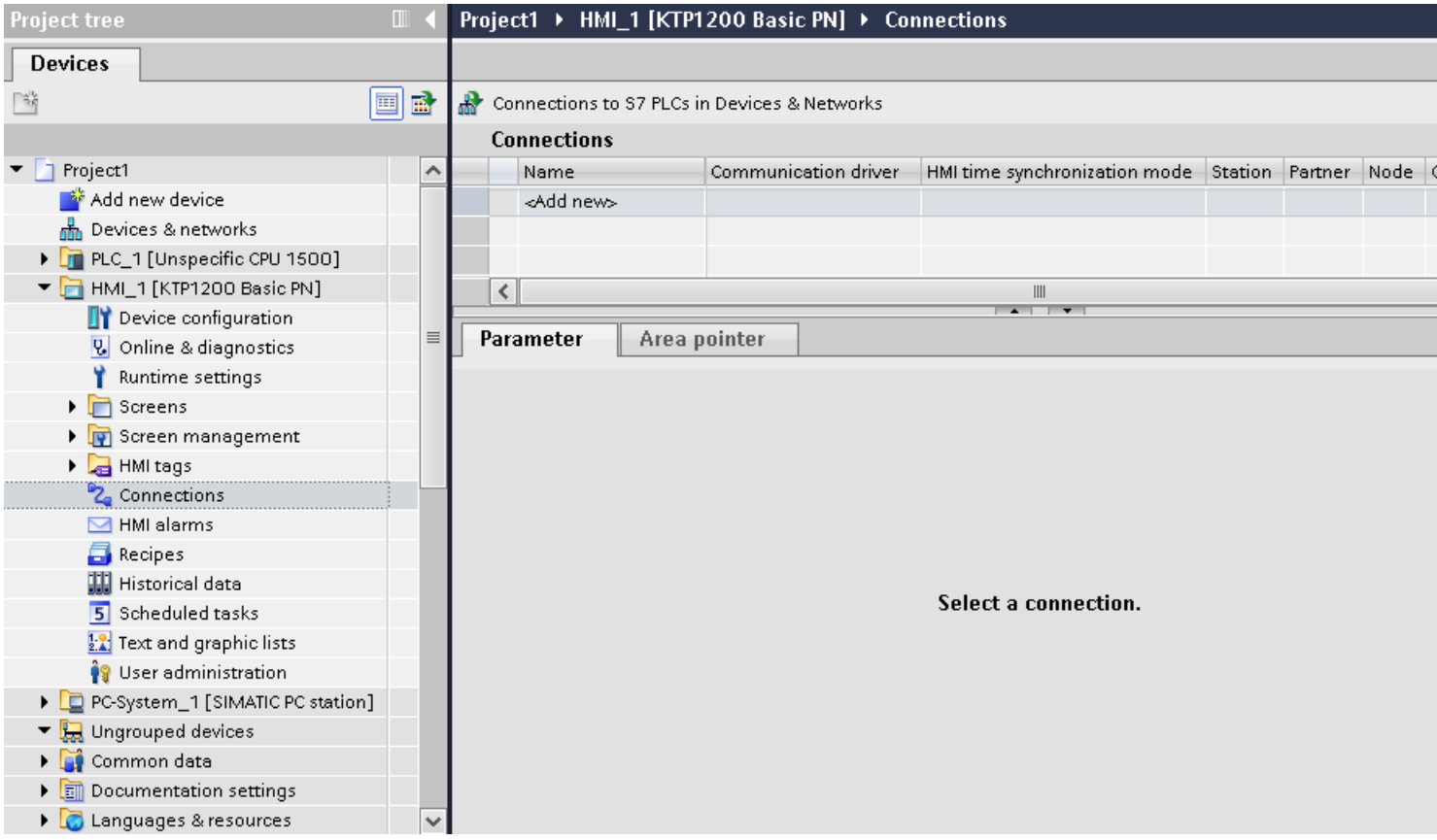
Example: PROFINET interface corresponds to the Ethernet interface

Requirements

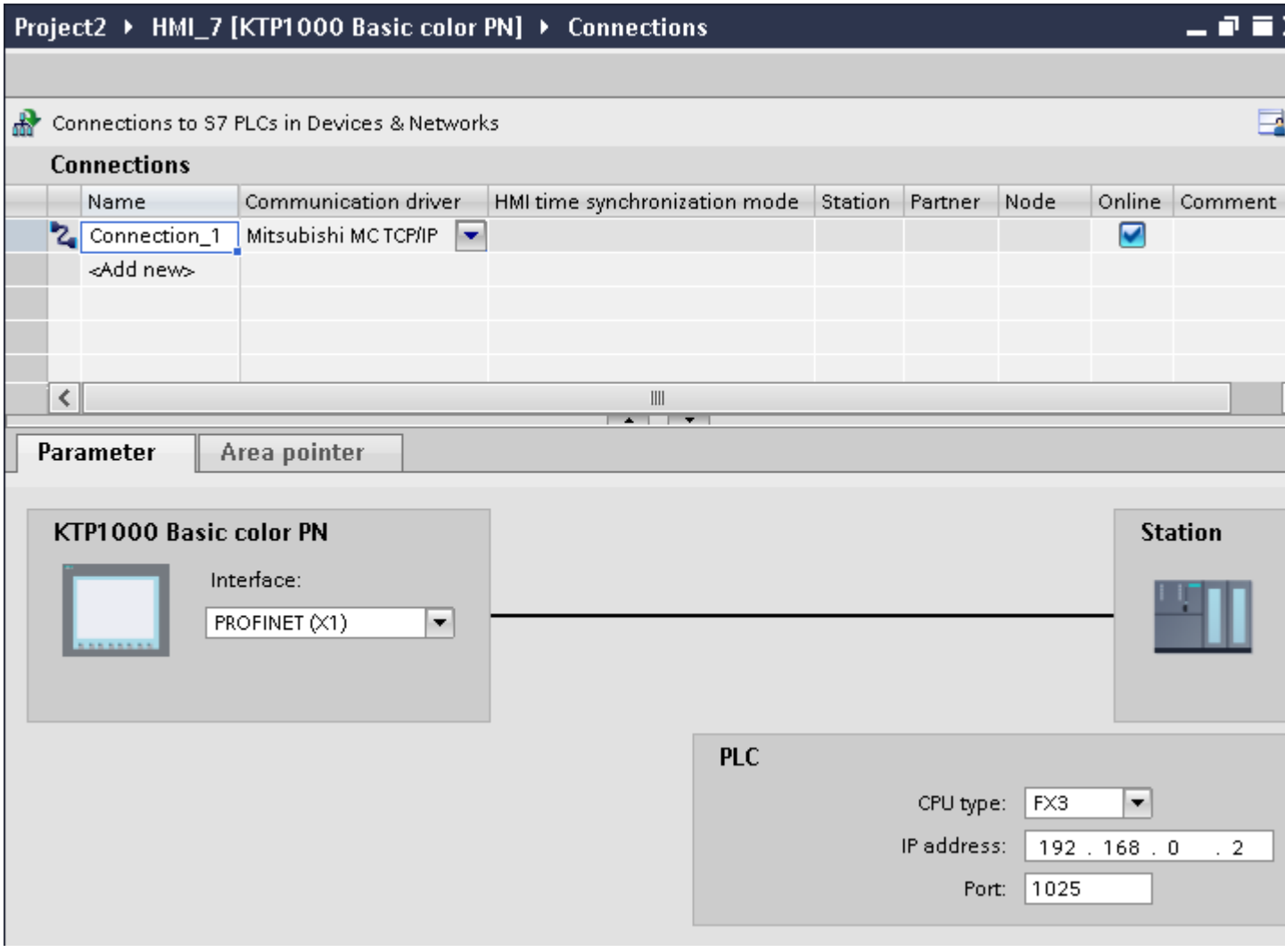
- A project is open.
- An HMI device has been created.

Procedure

1. Double-click the HMI device under "Devices" in the project tree.
2. Double-click the "Connections" item.
3. Double-click "<Add>" in the "Connections" editor.



4. In the "Communication drivers" column, select the "Mitsubishi MC TCPI/IP" driver.



5. Select all necessary connection parameters for the interface in the Inspector window under "Parameters".

See also

Mitsubishi communication drivers (Page 486)

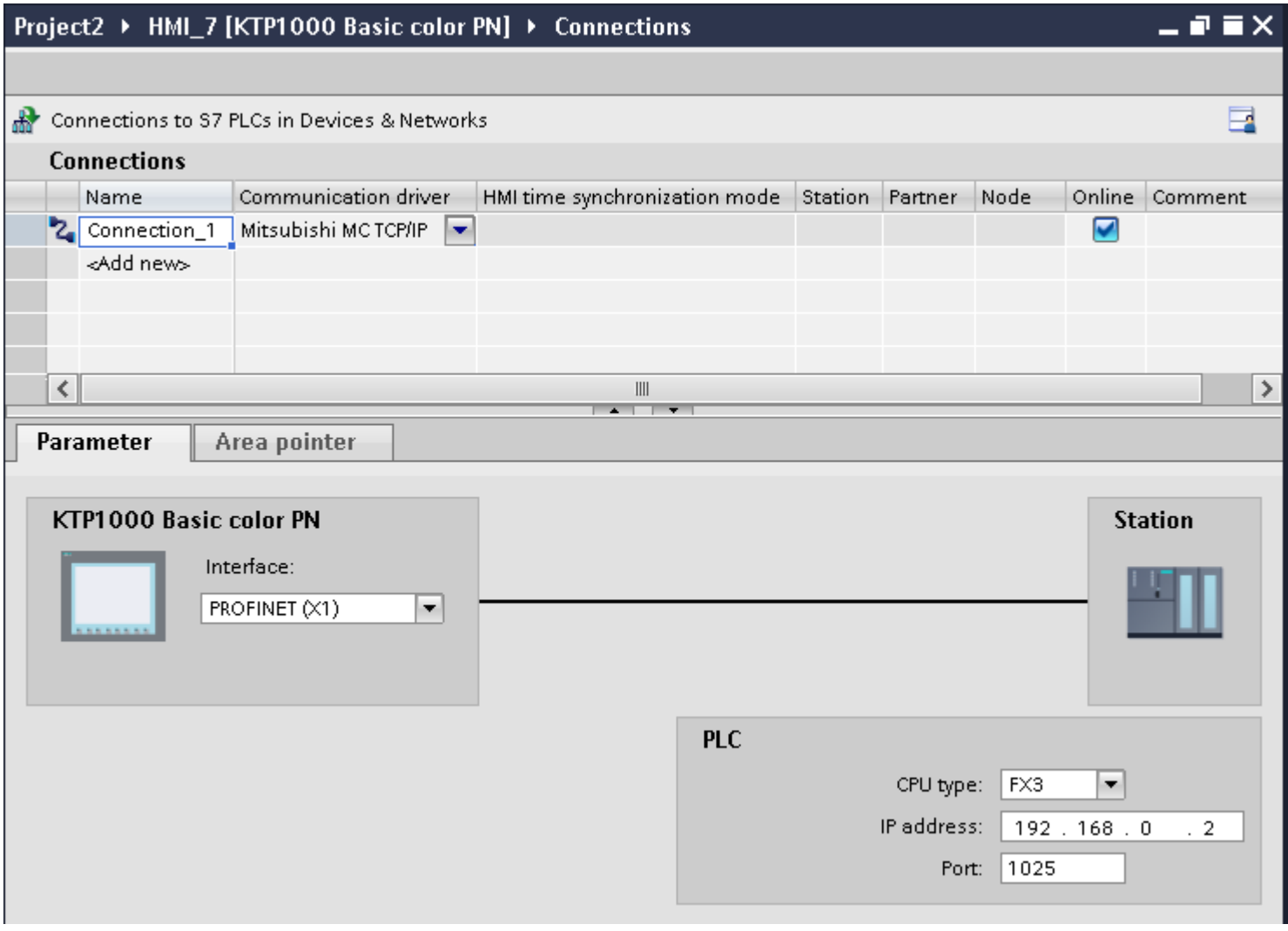
Parameters for the connection (Mitsubishi MC TCP/IP) (Page 489)

Parameters for the connection (Mitsubishi MC TCP/IP) (Basic Panels)

Parameters to be set

To set the connection parameters, such as addresses and profiles, click the connection that you have created in the "Connections" editor.

The communication partners are displayed schematically in the Inspector window under "Parameters". The "HMI device" and "PLC" areas are available for assigning parameters according to the interface used.



Parameters for the HMI device

You can select only one interface for the HMI device in the Inspector window under "Parameters". Depending on the HMI device, there are several interfaces available.

If you are directly connected to the HMI device during configuration, you can set up the IP address of the HMI device in WinCC. The IP address is transferred to the HMI device during project transfer.

Note

The IP address in the control panel will be overwritten upon subsequent loading if you have already set up the IP address in the HMI device control panel.

The IP address already set up in the control panel will be retained upon subsequent loading if you activate "Set IP address using a different method".

To set up the IP address of the HMI device:

1. Click on the HMI device.
2. Open the "Device configuration" editor.
3. Click the Ethernet interface.
4. Assign the IP address in the inspector window under:
   "General > PROFINET interface > Ethernet addresses"

Parameters for the PLC

- CPU type
  For "CPU type", you set the type of PLC to which the HMI device is connected.
  The following settings are possible:
  –FX3
  –Q
  If you select the FX3 CPU type, the Mitsubishi MC protocol "1E" is used and "3E" for the "Q" CPU type.
  The "Binary code" protocol variant is always used.

  Note

  If the CPU type is changed for a configured connection, tags with the following properties must be revised:

  - Operands that do not exist for the new CPU type, such as "W", "B", "F".
  - Inputs and outputs with different addressing (hexadecimal/octal)
  - Addresses greater than the valid address area of the new CPU type

- IP address
  Set the IP address or host name of the Ethernet/IP module of the PLC. Only the IP address can be used on a Basic Panel.
- Port
  Set the port number of the module of the PLC.

See also

Mitsubishi communication drivers (Page 486)

Configuring a connection via Mitsubishi MC TCP/IP (Page 487)

Connecting HMI device to PLC (Basic Panels)

Connections via Mitsubishi MC TCP/IP (Basic Panels)

Connection

The HMI device can be connected to the Mitsubishi PLC using the following components:

- Existing Ethernet network that also contains the PLCs
- Cross-over Ethernet cable connected directly to the Ethernet interface of the CPU or the communication module

The connection of the HMI device to a Mitsubishi PLC is limited primarily to the physical connection of the HMI device. Special blocks for the connection are not required in the PLC.

Connect the HMI device to one or several Q-series and/or FX3 PLCs. Connect the HMI device via the following interfaces:

- Communication interface OnBoard
- Approved communication module suitable for the PLC

Note

Timeout response with TCP/IP (Ethernet)

Due to the use of the TCP/IP protocol, the breakdown of a connection is detected at the earliest after approximately one minute. Communication failure cannot be reliably detected if no tags are requested, for example, no output tags in the current screen.

Configure area pointer coordination for each PLC. This setting ensures that a communication failure is recognized after approximately two minutes, even in the aforementioned scenario.

See also

Configuring a connection via Mitsubishi MC TCP/IP (Page 487)

Communication types (Page 493)

Communication types (Basic Panels)

Approved communication types

- Only applies for Mitsubishi FX(PG protocol):
  The point-to-point connection from a HMI device to an approved Mitsubishi FX–CPU via Mitsubishi FX is system-tested and approved by Siemens AG.
- Only applies for Mitsubishi MC TCP/IP:
  The following communication types are system-tested and approved:
  - Point-to-point connection to the approved PLCs
  - Multipoint connection from a HMI device with up to 4 PLCs with the respectively approved PLCs. CPU types (FX3 and Q) can be mixed.

  Note

  The HMI device is a client and the PLC must operate as a server.

Connectable PLCs

Connections can be implemented for the following Mitsubishi PLCs:

	Mitsubishi FX (PG protocol)	Mitsubishi MC TCP/IP
PLC		
MELSEC FX1n, FX2n	Yes	No
MELSEC FX3U, FX3UC, FX3G with communication module FX3U-ENET	No	Yes
MELSEC System Q • Q-series with the communication module QJ71E71-100 • QnUDEH CPU with Ethernet interface onboard	No	Yes

See also

Configuring a connection via Mitsubishi MC TCP/IP (Page 487)

Connections via Mitsubishi MC TCP/IP (Page 492)

Parameterization of the communications modules (Basic Panels)

FX3 PLCs (Basic Panels)

Procedure

1. Start the FX-Configurator.
2. Select the module.
3. Assign the following settings in the "Operational settings" dialog:
   - Communication data code:
     Binary code
   - Initial timing:
     Always wait for OPEN
   - IP address:
     IP address
   - Send frame setting:
     Ethernet(V2.0)
   - TCP Existence confirmation setting:
     Use the Ping
4. Assign the following settings in the "Open Settings" dialog:
   - Protocol:
     TCP
   - Open system:
     Unpassive
   - Fixed buffer:
     Receive
   - Fixed buffer communication procedure:
     Procedure exist(MC)
   - Pairing open
     Disable
   - Existence confirmation
     No confirm
   - Host station Port No. (DEC)
     Port number

     Note

     The port number chosen in the communication module must match the port number in WinCC. A connection with a port number must be assigned for each connected HMI device.

     You must specify port numbers in decimal values.
5. Confirm the default settings of the other dialog boxes.

The network no. and station no. parameters are not relevant for the connection and can be chosen as required.

See also

Parameters for the connection (Mitsubishi MC TCP/IP) (Page 489)

Q PLCs (Basic Panels)

Procedure

1. Click "Edit network parameters".
2. Select the network type:
   – Ethernet
   The network number and the group / station number are not evaluated and can be freely assigned

3. Assign the following settings in the "Operational settings" dialog:
   - Communication data code:
     Binary code
   - Initial timing:
     Always wait for OPEN
   - IP address:
     IP address
   - Send frame setting:
     Ethernet(V2.0)
   - Enable write operations during RUN
4. Assign the following settings in the "Open settings" dialog:
   - Protocol:
     TCP
   - Open system:
     Unpassive
   - Pairing open
     Disable
   - Existence confirmation
     No confirm
   - Host station Port No. (HEX)
     Port-Nummer

   Note

   The port number chosen in the communication module must match the port number in WinCC. A connection with a port number must be assigned for each connected HMI device.

   You must specify port numbers in hexadecimal values.

See also

Parameters for the connection (Mitsubishi MC TCP/IP) (Page 489)

Internal Ethernet port of the Q0xUDEH CPU (Basic Panels)

Procedure

1. Assign the following settings in the "Internal Ethernet Port" dialog:
   - IP address:
     IP address
   - Communication data code:
     Binary code
   - Enable online changes
2. Assign the following settings in the "Open settings" dialog:
   - Protocol:
     TCP
   - Open system:
     MC-Protocol
   - Host station Port No. (HEX)
     Port number

   Note

   The port number chosen in the communication module must match the port number in WinCC. A connection with a port number must be assigned for each connected HMI device.

See also

Parameters for the connection (Mitsubishi MC TCP/IP) (Page 489)

Performance features of communication (Basic Panels)

Permitted data types for Mitsubishi MC TCPI/IP (Basic Panels)

Permitted data types

The table lists the data types that can be used when configuring tags and area pointers.

Data type	Operand type	Length
4-bit block	M, X, Y, B, F	1 byte
8-bit block	M, X, Y B, F	1 byte
12-bit block	M, X, Y B, F	2 bytes
16-bit block	M, X, Y B, F	2 bytes
20-bit block	M, X, Y B, F	4 bytes
24-bit block	M, X, Y B, F	4 bytes
28-bit block	M, X, Y B, F	4 bytes

Data type	Operand type	Length
32-bit block	M, X, Y B, F	4 bytes
Bool	M, D, X, Y B, F	1-bit
DInt	D, W	4 bytes
DWord	D, C, W	4 bytes
Int	D, W	2 bytes
Real 1)	D, W	4 bytes
String 1)	D	1 to 80 characters
Word	D, T, C, W	2 bytes

1) The "String" and "Real" data types are not available for all CPUs.
2) Operand types B, F and W are only available for CPU type "Q".

Note

Note the following for write accesses:

Tags can only be written if "Enable online changes" or "Enable write operations during RUN" was selected when parameterizing the Mitsubishi communication modules.

For data type "Bool" in operand type "D", the entire word is written back to the PLC following a change to the specified bit. There is no check to determine whether any other bits in the word have changed. As a result, the PLC only has read access to the specified word.

Note

Array elements in I/O fields cannot be used in communication with a Mitsubishi PLC.

See also

Configuring a connection via Mitsubishi MC TCP/IP (Page 487)

Supported CPU types for Mitsubishi MC TCP/IP (Basic Panels)

CPU types

The following CPU types are supported for configuring the Mitsubishi MC TCP/IP communication driver.

- FX3 series
  - FX 3G / FX 3G with communication modul FX3U-ENET
  - FX 3U / FX 3U with communication modul FX3U-ENET
  - FX 3UC / FX 3UC with communication modul FX3U-ENET
- Q series
  - Q-Series with QJ71E71-100 communication module
- iQ series / QnUD
  - QnUDEHCPU with built in ethernet module

See also

Configuring a connection via Mitsubishi MC TCP/IP (Page 487)

Addresses for Mitsubishi MC TCP/IP (Basic Panels)

Address areas for connections via Mitsubishi MC TCP/IP

The address area boundaries differ for the different series; refer to the Mitsubishi Computerlink manuals for this information.

Examples of address area boundaries dependent on the CPU and communication format:

Name	Operand type	Max. address FX3	Max. address Q-Series
Output/Input	Y/X	Octal X/Y 0 - 777	HEX X/Y 0 - 7FF
Bit memory	M	M0 - M3071 and M8000 - M8255	M/L/S 0 - 8191
Data register	D	D0 - 7999 D8000 - D8255	D0 - 8191 D9000 - D9255 becomes SD1000 - SD1255
Counter	C	C0 - 255	C0 - 1023
Timer	T	T0 - 255	T0 - 2047
Link register	W	--	Hex: W0 - FFF
Link flag	B	--	Hex: B0 - FFF
Error flag	F	--	F0 - 2047

See also

Configuring a connection via Mitsubishi MC TCP/IP (Page 487)

Address areas for Mitsubishi MC TCP/IP (Basic Panels)

FX3

Address areas	Data types														
	Bool	Int	Word	DInt	DWord	Real	String	4-bit block	8-bit block	12-bit block	16-bit block	20-bit block	24-bit block	28-bit block	32-bit block
M	M0 - M9999	--	--	--	--	--	--	M0 - M9996	M0 - M9992	M0 - M9988	M0 - M9984	M0 - M9980	M0 - M9976	M0 - M9972	M0 - M9968
D	D0.0 - D9999.15	D0 - D9999	D0 - D9999	D0 - D9999	D0 - D9999	D0 - D9999	D0 - D9999	--	--	--	--	--	--	--	--
T	--	--	T0 - T999	--	--	--	--	--	--	--	--	--	--	--	--
C	--	C0 - C999	C0 - C999	C0 - C998	C0 - C998	--	--	--	--	--	--	--	--	--	--
X	X0 - X777	--	--	--	--	--	--	X0 - X774	X0 - X770	X0 - X764	X0 - X760	X0 - X754	X0 - X750	X0 - X744	X0 - X740
Y	Y0 - Y777	--	--	--	--	--	--	Y0 - Y774	Y0 - Y770	Y0 - Y764	Y0 - Y760	Y0 - Y754	Y0 - Y750	Y0 - Y744	Y0 - Y740

Q

Address areas	Data types														
	Bool	Int	Word	DInt	DWord	Real	String	4-bit block	8-bit block	12-bit block	16-bit block	20-bit block	24-bit block	28-bit block	32-bit block
M	M0 - M9999	--	--	--	--	--	--	M0 - M9996	M0 - M9992	M0 - M9988	M0 - M9984	M0 - M9980	M0 - M9976	M0 - M9972	M0 - M9968
F	F0 - F9999	--	--	--	--	--	--	F0 - F9996	F0 - F9992	F0 - F9988	F0 - F9984	F0 - F9980	F0 - F9976	F0 - F9972	F0 - F9968
B	B0 - BFFFF	--	--	--	--	--	--	B0 - BFFFC	B0 - BFFF8	B0 - BFFF4	B0 - BFFF0	B0 - BFFEC	B0 - BFFE8	B0 - BFFE4	B0 - BFFE0
D	D0.0 - D65534.15	D0 - D65534	D0 - D65534	D0 - D65533	D0 - D65533	D0 - D65533	D0 - D65534	--	--	--	--	--	--	--	--

Address areas	Data types														
	Bool	Int	Word	DInt	DWord	Real	String	4-bit block	8-bit block	12-bit block	16-bit block	20-bit block	24-bit block	28-bit block	32-bit block
T	--	--	T0 - T2047	--	--	--	--	--	--	--	--	--	--	--	--
C		C0 - C2047	C0 - C2047	C0 - C2046	C0 - C2046	--	--	--	--	--	--	--	--	--	--
W	--	W0 - WFFFF	W0 - WFFFF	W0 - WFFFE	W0 - WFFFE	W0 - WFFFE	--	--	--	--	--	--	--	--	--
X	X0 - XFFFF	--	--	--	--	--	--	X0 - XFFFC	X0 - XFFF8	X0 - XFFF4	X0 - XFFF0	X0 - XFFEC	X0 - XFFE8	X0 - XFFE4	X0 - XFFE0
Y	Y0 - YFFFF	--	--	--	--	--	--	Y0 - YFFFC	Y0 - YFFF8	Y0 - YFFF4	Y0 - YFFF0	Y0 - YF-FEC	Y0 - YFFE8	Y0 - YFFE4	Y0 - YFFE0

See also

Configuring a connection via Mitsubishi MC TCP/IP (Page 487)

Commissioning components (Basic Panels)

Transferring a project to the HMI device

1. Switch the HMI device to "transfer mode".
2. Set all necessary transfer parameters.
   - Interface
   - Transfer parameters
   - Target storage location
3. Start the project transfer.
   The project is compiled automatically.
   All compilation and transfer steps are logged to a message window.

Interconnecting the PLC with the HMI device

1. Interconnect the PLC with the HMI device using a suitable cable.
2. The message "Connection to PLC .... is established" is output to the HMI device.

Optimizing the configuration (Basic Panels)

Acquisition cycle and update time

The acquisition cycles for the "Area pointers" and of the tags defined in the configuration software are decisive in terms of the update times which can actually be achieved.

The update time is equivalent to the acquisition cycle + transmission time + processing time.

Items to observe when optimizing the update times in configuration data:

- Optimize the maximum and minimum size of the data areas.
- Acquisition cycles which are too short lead to unnecessary load on overall performance. Set the acquisition cycle according to the rate of change of the process values. The rate of temperature changes at a furnace, for example, is significantly slower compared to the speed rate of an electrical drive. A time of approx. 1 second is a benchmark for the acquisition cycle.
- Avoid any gaps when entering the alarm or screen tags in a data area.
- Changes in the PLC can only be detected reliably if these are available at least within the actual acquisition cycle.

Screens

The refresh rate of screens is determined by the type and volume of data to be visualized.

Only configure short acquisition cycles for objects which actually require shorter refresh cycles. This procedure reduces update times.

Trends

The HMI device always updates all bit-triggered trends whose group bit is set in the "Trend transfer area". It resets the bits in the next cycle.

The group bit in the PLC program can only be set again after the HMI device has reset all bits.

Job mailboxes

A high rate and volume of job mailboxes transferred may lead to overload in communication between the HMI device and the PLC.

The HMI device confirms acceptance of the job mailbox by entering the value zero in the first data word of the job mailbox. The HMI device now processes the job for which it requires a certain time slice. It may take the HMI device some time to process a new job mailbox which is transferred in immediate succession to the job mailbox. The next job mailbox is only accepted if sufficient computing resources are available.

Mitsubishi FX (Basic Panels)

Configuring a connection via Mitsubishi FX (Basic Panels)

Introduction

You configure a connection to a PLC with a Mitsubishi FX communication driver in the "Connections" editor of the HMI device.

The Mitsubishi FX protocol is also referred to as the Mitsubishi PG protocol.

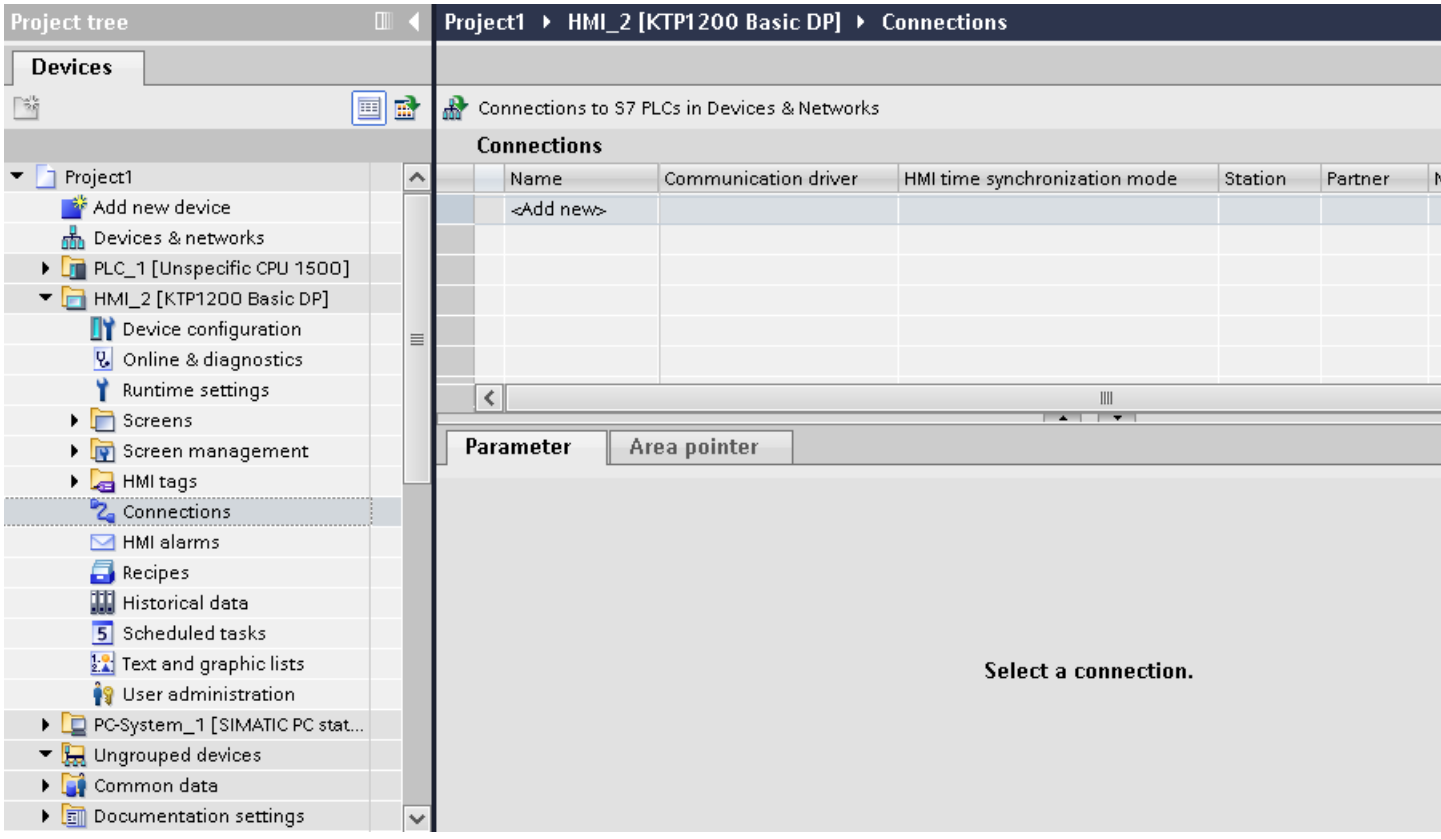
The interfaces are named differently depending on the HMI device.

Requirements

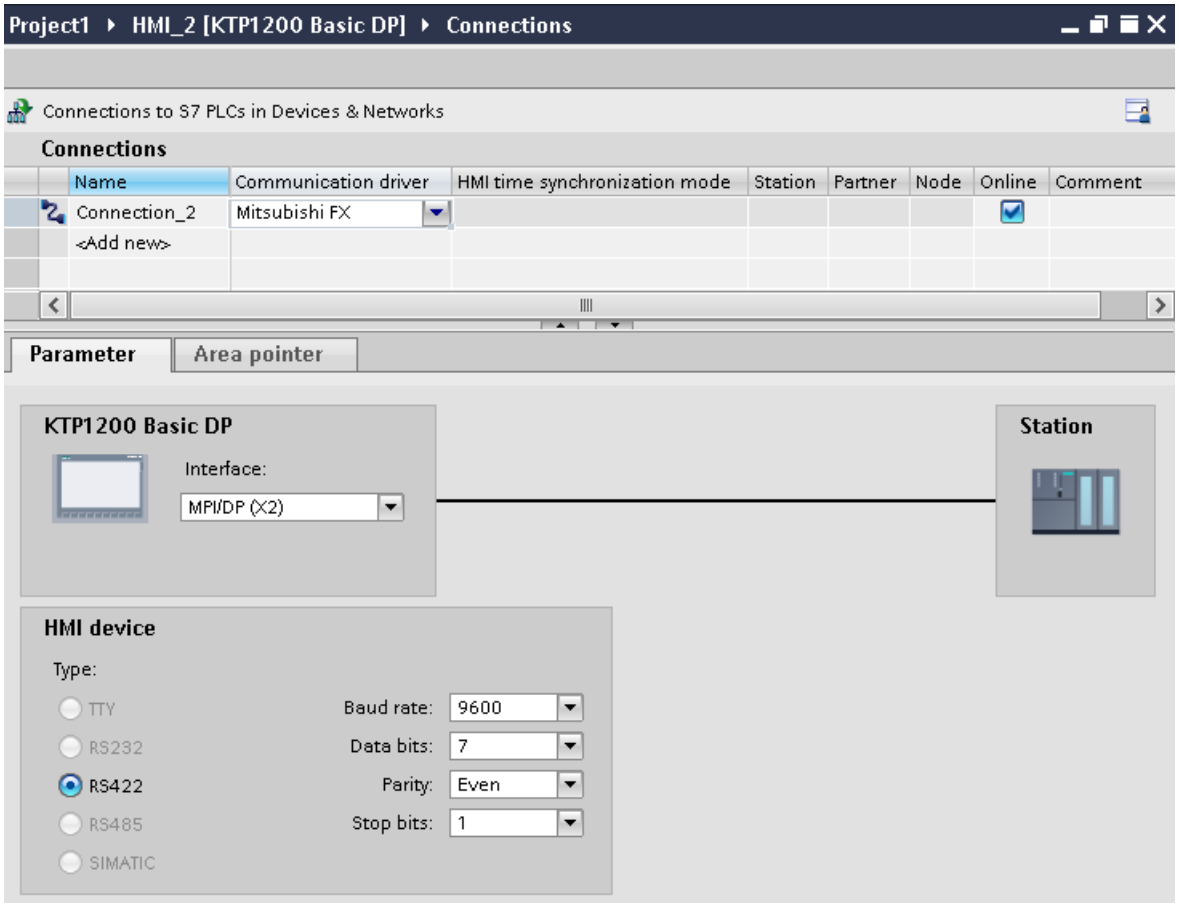
- A project is open.
- An HMI device has been created.

Procedure

1. Double-click the HMI device under "Devices" in the project tree.
2. Double-click the "Connections" item.
3. Double-click "<Add>" in the "Connections" editor.



4. In the "Communication drivers" column, select the "Mitsubishi FX" driver.



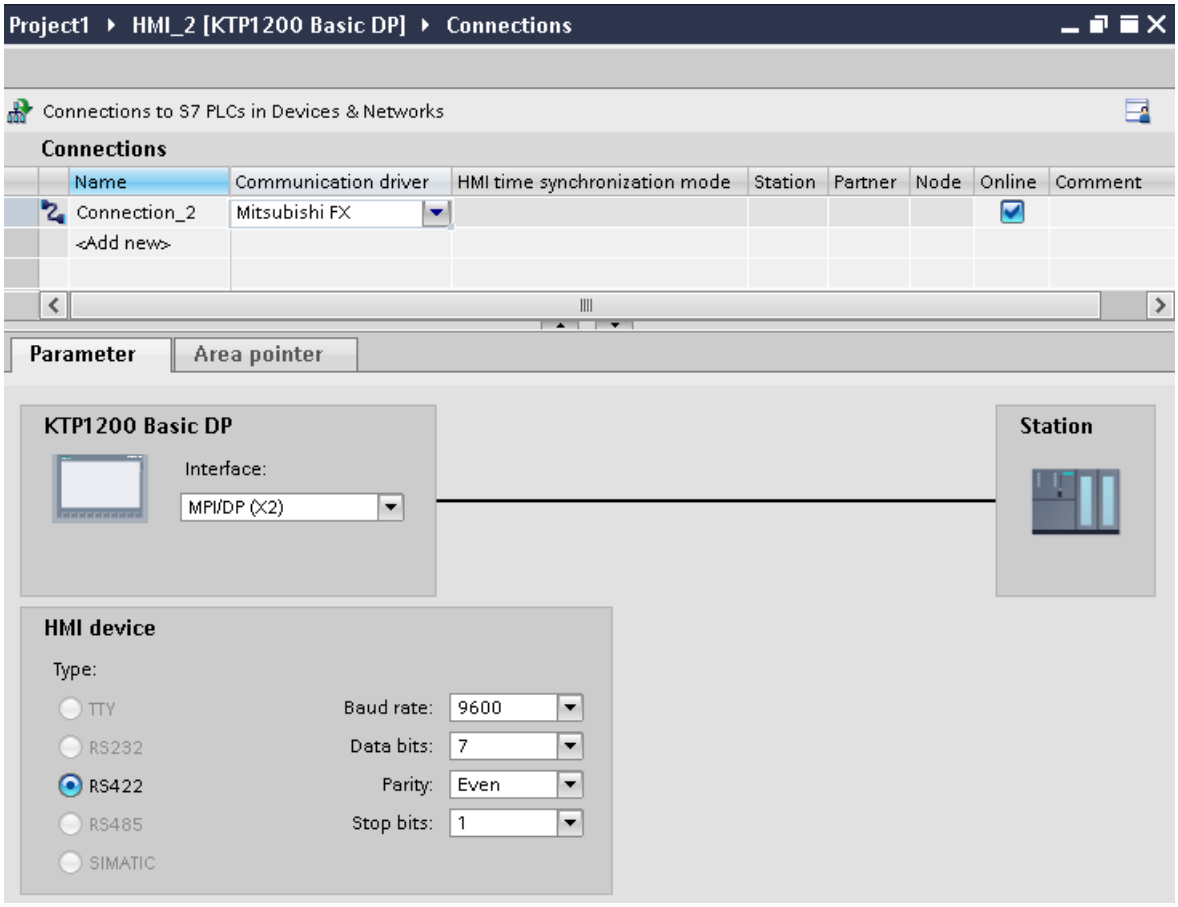
5. Select all necessary connection parameters for the interface in the Inspector window under "Parameters".

Parameters for the connection (Mitsubishi FX) (Basic Panels)

Parameters to be set

To assign the connection parameters, such as addresses and profiles, click the connection that you have created in the "Connections" editor.

The communication partners are displayed schematically in the Inspector window under "Parameters". The "HMI device", "Network", and "PLC" areas are available for assigning parameters according to the interface used.



Parameters for the HMI device

You can select an interface for the HMI device in the Inspector window under "Parameters". Depending on the HMI device, there are several interfaces available.

- "Type"
  Specifies the physical connection used.

  Note

  If you use the IF1B interface, you must switch over the RS422 receive data and the RTS signal additionally by 4 DIL-switches on the back of the HMI device.

Parameters for the PLC

- Baud rate: For "Baud rate", select the transmission speed between the HMI device and PLC. Select the baud rate "9600".
- Data bits: Under "Data bits", select "7 bits".
- Parity: Under "Parity", select "Even".
- Stop bits: Under "Stop bits", select "1 bit".

Connecting HMI device to PLC (Basic Panels)

Communication types (Basic Panels)

Approved communication types

- Only applies for Mitsubishi FX(PG protocol):
  The point-to-point connection from a HMI device to an approved Mitsubishi FX–CPU via Mitsubishi FX (PG protocol := protocol for access to the program and memory elements of the FX series PC CPU version V1.21 and after) is system-tested and approved by Siemens AG.
- Only applies for Mitsubishi MC TCP/IP:
  The following communication types are system-tested and approved:
  - Point-to-point connection to the approved PLCs
  - Multipoint connection from a HMI device with up to 4 PLCs with the respectively approved PLCs. CPU types (FX3 and Q) can be mixed.

  Note

  The HMI device is a client and the PLC must operate as a server.

Connectable PLCs

Connections can be implemented for the following Mitsubishi PLCs:

	Mitsubishi FX (PG protocol)	Mitsubishi MC TCP/IP
PLC		
MELSEC FX1n, FX2n	Yes	No
MELSEC FX3U, FX3UC, FX3G with communication module FX3U-ENET	No	Yes
MELSEC System Q • Q-series with the communication module QJ71E71-100 • QnUDEH CPU with Ethernet interface onboard	No	Yes

Connections via Mitsubishi FX (Basic Panels)

Connection

Connect the HMI device to the programming interface of the CPU (RS 422) (see documentation of the PLC).

The connection between the HMI device and the Mitsubishi PLC is basically restricted to setting the interface parameters. Special blocks for the connection are not required in the PLC.

Connecting cable

The following connecting cables are available to connect the HMI device to the PLC.

Interface to HMI device or adapter	Mitsubishi Electric PLC via FX protocol
	FX1n, Fx2n, Mini DIN, 8–pin
RS 232, 9-pin	Mitsubishi SC-09 [1]
RS 422, 9-pin	Connecting cable RS422-2P

[1] Since the Mitsubishi PLCs communicate via RS 422 as a standard, the Mitsubishi programming cable SC–09 with integrated RS 422/RS 232 adaptor is necessary for connecting a HMI device via RS 232.

Note

Appl. only to R

Cable length is r .32 m.

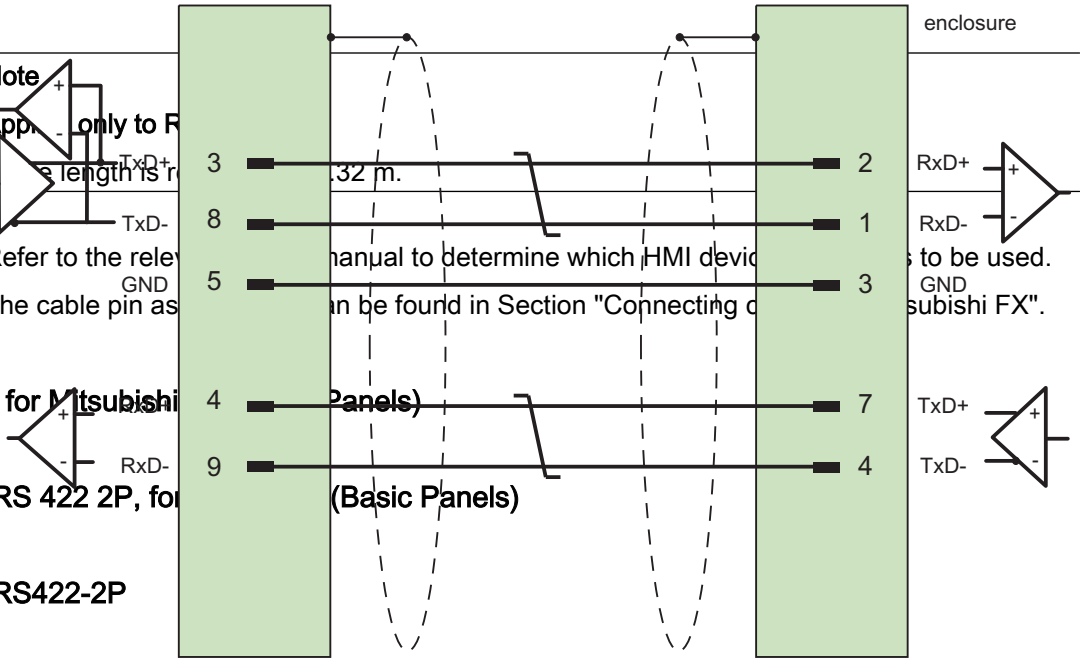
Refer to the relev anual to determine which HMI devic to be used.

The cable pin as an be found in Section "Connecting c subishi FX".

Connecting cables for Mitsubishi Panels)

Connecting cable RS 422 2P, fo (Basic Panels)

Connecting cable RS422-2P



Shield with large-area contact to housing at both ends
Cable: 3 x 2 x 0.14 mm², shielded,
max. length 500 m

Performance features of communication (Basic Panels)

Permitted data types for Mitsubishi FX (Basic Panels)

Permitted data types

The table lists the data types that can be used when configuring tags and area pointers.

Data type	Operand type	Length
4-bit block	M, X, Y	1 byte
8-bit block	M, X, Y	1 byte
12-bit block	M, X, Y	2 bytes
16-bit block	M, X, Y	2 bytes
20-bit block	M, X, Y	4 bytes
24-bit block	M, X, Y	4 bytes
28-bit block	M, X, Y	4 bytes
32-bit block	M, X, Y	4 bytes
Bool	D, M, X, Y	1-bit
DWord	D, C-32 bit	4 bytes
Real	D	4 bytes
String	D	1 to 50 characters
Word	D, T, C-16 bit	2 bytes

Note

Note the following for write accesses:

For data type "Bool" in operand type "D", the entire word is written back to the PLC following a change to the specified bit. There is no check to determine whether any other bits in the word have changed. As a result, the PLC only has read access to the specified word.

Note

Array elements in I/O fields cannot be used in communication with a Mitsubishi PLC.

Supported CPU types for Mitsubishi FX (Basic Panels)

CPU types

The following CPU types are supported for configuring the Mitsubishi FX communication driver.

- FX1 series
  - FX1n
- FX2 series
  - FX2n

Address areas for Mitsubishi FX (Basic Panels)

FX1n and FX2n

Address areas	Data types											
	Bool	Word	DWord	Real	String	4-bit block	8-bit block	12-bit block	16-bit block	20-bit block	24-bit block	28-bit block	32-bit block
M	M0 - M9999					M0 - M9996	M0 - M9992	M0 - M9988	M0 - M9984	M0 - M9980	M0 - M9976	M0 - M9972	M0 - M9968
D	D0.0 - D999.15	D0 - D999	D0 - D998	D0 - D998	D0 - D998								
T		T0 - T255											
C-16-Bit		C-16-Bit 0 - C-16-Bit 199											
C-32-Bit			C-32-Bit 200 - C-32-Bit 255										
X	X0 - X255					X0 - X252	X0 - X248	X0 - X244	X0 - X240	X0 - X236	X0 - X232	X0 - X228	X0 - X224
Y	Y0 - X255					Y0 - Y252	Y0 - Y248	Y0 - Y244	Y0 - Y240	Y0 - Y236	Y0 - Y232	Y0 - Y228	Y0 - Y224

Commissioning components (Basic Panels)

Transferring a project to the HMI device

1. Switch the HMI device to "transfer mode".
2. Set all necessary transfer parameters.
   - Interface
   - Transfer parameters
   - Target storage location
3. Start the project transfer.
   The project is compiled automatically.
   All compilation and transfer steps are logged to a message window.

Interconnecting the PLC with the HMI device

1. Interconnect the PLC with the HMI device using a suitable cable.
2. The message "Connection to PLC .... is established" is output to the HMI device.

Optimizing the configuration (Basic Panels)

Acquisition cycle and update time

The acquisition cycles for the "Area pointers" and of the tags defined in the configuration software are decisive in terms of the update times which can actually be achieved.

The update time is equivalent to the acquisition cycle + transmission time + processing time.

Items to observe when optimizing the update times in configuration data:

- Optimize the maximum and minimum size of the data areas.
- Acquisition cycles which are too short lead to unnecessary load on overall performance. Set the acquisition cycle according to the rate of change of the process values. The rate of temperature changes at a furnace, for example, is significantly slower compared to the speed rate of an electrical drive. A time of approx. 1 second is a benchmark for the acquisition cycle.
- Avoid any gaps when entering the alarm or screen tags in a data area.
- Changes in the PLC can only be detected reliably if these are available at least within the actual acquisition cycle.

Screens

The refresh rate of screens is determined by the type and volume of data to be visualized.

Only configure short acquisition cycles for objects which actually require shorter refresh cycles. This procedure reduces update times.

Trends

The HMI device always updates all bit-triggered trends whose group bit is set in the "Trend transfer area". It resets the bits in the next cycle.

The group bit in the PLC program can only be set again after the HMI device has reset all bits.

Job mailboxes

A high rate and volume of job mailboxes transferred may lead to overload in communication between the HMI device and the PLC.

The HMI device confirms acceptance of the job mailbox by entering the value zero in the first data word of the job mailbox. The HMI device now processes the job for which it requires a certain time slice. It may take the HMI device some time to process a new job mailbox which is transferred in immediate succession to the job mailbox. The next job mailbox is only accepted if sufficient computing resources are available.

Data exchange (Basic Panels)

Area pointers for Mitsubishi (Basic Panels)

Area pointers for connections via Mitsubishi communication drivers

You use an area pointer to access a data area in the PLC.

For more detailed information on area pointers and their configuration, refer to Section "Data exchange using area pointers (Page 649)".

Special considerations for connections via Mitsubishi communication drivers

You can configure the following area pointers

Area pointers	Mitsubishi MC TCP/IP	Mitsubishi FX
Screen number	Yes	Yes
Date/time	Yes	Yes
Date/time PLC	Yes	Yes
Coordination	Yes	Yes
Project ID	Yes	Yes
Job mailbox	Yes	Yes
Data record	Yes	Yes

Restrictions for Mitsubishi FX and MC TCP/IP

The following restrictions apply for configuring area pointers.

CPU type	Data types	Operand type
FX/FX3	Int, Word	D
Q	Int, Word	D

See also

Data exchange using area pointers (Page 649)

Trends (Basic Panels)

Trends

A trend is the graphical representation of one or more values from the PLC. The value is read out time-triggered for Basic Panels.

For additional information see:

AUTOHOTSPOT

Time-triggered trends

The HMI device reads in the trend values cyclically at an interval specified in the configuration.

Time-triggered trends are suitable for continuous curves, such as the operating temperature of a motor.

Alarms (Basic Panels)

Configuring alarms (Basic Panels)

Configuring alarms for non-integrated connections

Several steps are necessary to configure alarms such as warnings, error messages and acknowledgement.

- Step 1: Create tags
- Step 2: Configure alarms
- Step 3: Configure acknowledgment

You can find additional information in the section:

AUTOHOTSPOT

Distinctive features when configuring alarms

If you are configuring connections of HMI devices to PLCs of other manufacturers, note the following distinctive features when configuring:

- Data types of the tags
- Addressing of tags
- How the bit positions are counted

Data types

For connections with a Mitsubishi communication driver, the following data types are supported:

PLC	Permitted data types	
	Discrete alarms	Analog alarms
FX1n, FX2n, FX3 series, Q-Series, iQ-Series	Word, Int [1]	4-bit block, 8-bit block, 12-bit block, 16-bit block, 20-bit block, 24-bit block, 28-bit block, 32-bit block, Word, DWord, Int [1], DInt [1], Real,
[1] Not for Mitsubishi FX communication driver		

How the bit positions are counted

For connections with a Mitsubishi communication driver, the following counting method applies:

How the bit positions are counted	Left byte								Right byte							
In Mitsubishi PLCs	15							8	7							0
In WinCC you configure:	15							8	7							0

Restrictions on alarms

- Mitsubishi MC TCP/IP
  Only tags of operand type "D" and data types "Word" and "Int" are permitted as trigger tags for discrete alarms. You can use array tags (operand type: "D"; data types: "ARRAY [x..y] of Word" or "ARRAY [x..y] of Int") for discrete alarms.
- Mitsubishi FX
  Only tags of operand type "D" and data type "Word" are permitted as trigger tags for discrete alarms. You can use array tags (operand type "D"; data type "ARRAY [x..y] of Word") for discrete alarms."

Acknowledgment of alarms (Basic Panels)

Procedure

Create suitable tags on the PLC to acknowledge an error alarm. You assign these tags to an alarm in the "Bit messages" editor. You make the assignment in "Properties > Acknowledgment".

Distinction in terms of acknowledgment:

- Acknowledgment by the PLC
- Acknowledgment on the HMI device

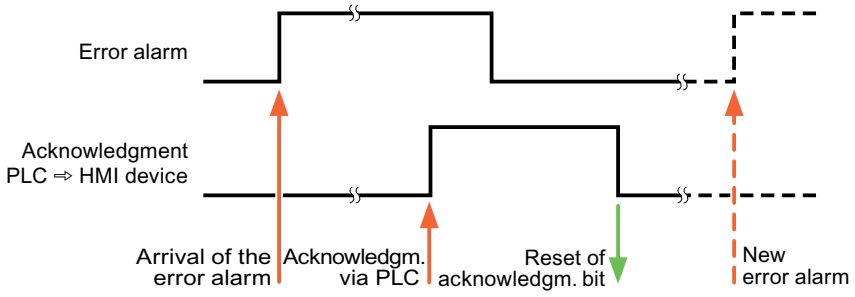
Acknowledgment by the PLC

In "PLC acknowledgment tag", configure the tag or array tag and the bit number that the HMI device uses to identify a PLC acknowledgment.

A bit set in the tag triggers acknowledgment of the assigned error alarm bit at the HMI device. This tag bit returns a function similar to acknowledgment on the HMI device which is triggered by pressing the "ACK" button, for example.

The acknowledgment bit must be located in the same tag as the bit for the error alarm.

Reset the acknowledgment bit before setting the bit in the alarm area again. The figure below shows the pulse diagram.



Acknowledgment on the HMI device

In the "HMI acknowledgment tag" area, configure the tag or array tag as well as the bit number that the HMI device writes to the PLC after acknowledgment. Make sure when you use an array tag that it is not longer than 6 words.

To always create a signal change when setting an assigned acknowledgment bit of a discrete alarm that must be acknowledged, the HMI device will reset the acknowledgment bit assigned to the alarm as soon as it detects an alarm subject to acknowledgment and write the acknowledgment tag in the PLC. There will be a certain delay between detecting the message and writing the acknowledgment tag in the PLC because the HMI device has to process the operations.

If a discrete alarm subject to acknowledgment is acknowledged by the HMI device, then the corresponding bit in the assigned acknowledgment tag will be set. The entire acknowledgment

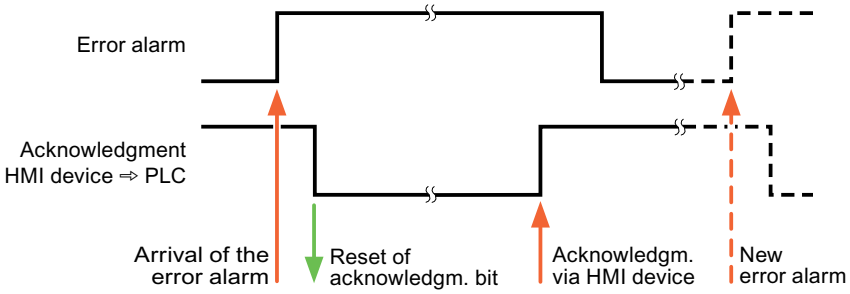
tag is then written to the PLC by the HMI device. This allows the PLC to recognize that a certain alarm message has been acknowledged at the HMI device.

Note

All alarm bits acknowledged since the last Runtime start will remain in the acknowledgment tag until a new incoming of the respective discrete alarms is detected.

This area should only be read by the PLC because the entire section of the HMI device will be overwritten once the next acknowledgment tag is written.

The figure below shows the pulse diagram.



1.13.3.3 Modicon Modbus (Basic Panels)

Modicon Modbus communication drivers (Basic Panels)

Introduction

This section describes the communication between an HMI device and PLCs that use Modicon Modbus communication drivers.

The following communication drivers are supported:

- Modicon Modbus TCP/IP
- Modicon Modbus RTU

Data exchange

Data is exchanged by means of tags or area pointers.

- Tags
  The PLC and the HMI device use process values for data exchange. You create tags in the configuration that point to addresses in the PLC. The HMI device reads the value from the defined address, and then displays it. The operator may also enter values on the HMI device, which are then written to the address in the PLC.
- Area pointers
  Area pointers are used to exchange specific data and are only set up when these data are used.

Modicon Modbus TCP/IP (Basic Panels)

Configuring a connection via Modicon Modbus TCP/IP (Basic Panels)

Introduction

You configure a connection to one of the PLCs with Modicon Modbus TCP/IP communication driver in the "Connections" editor of the HMI device.

The Ethernet interfaces are named differently depending on the HMI device.

Example: PROFINET interface corresponds to the Ethernet interface

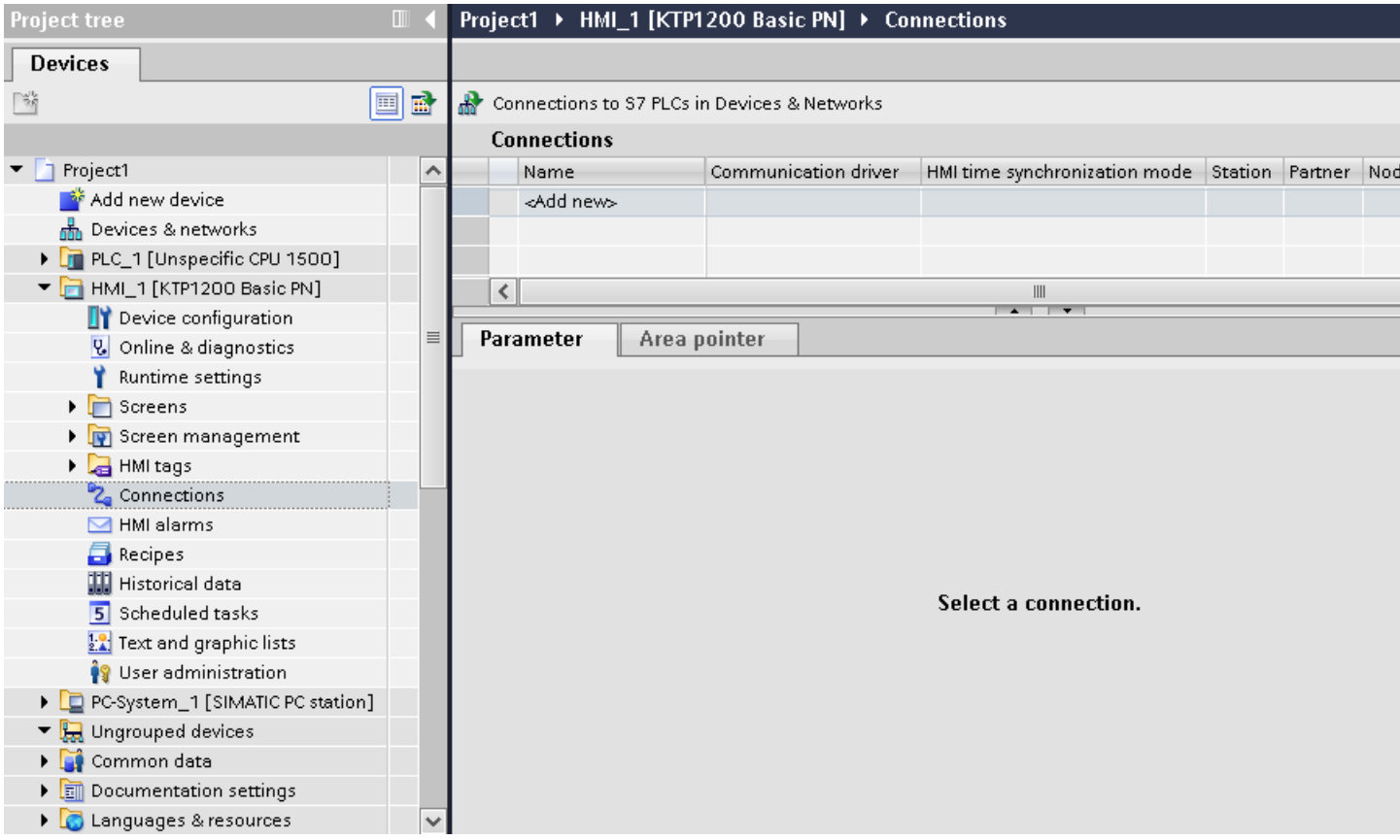
Requirements

- A project is open.
- An HMI device has been created.

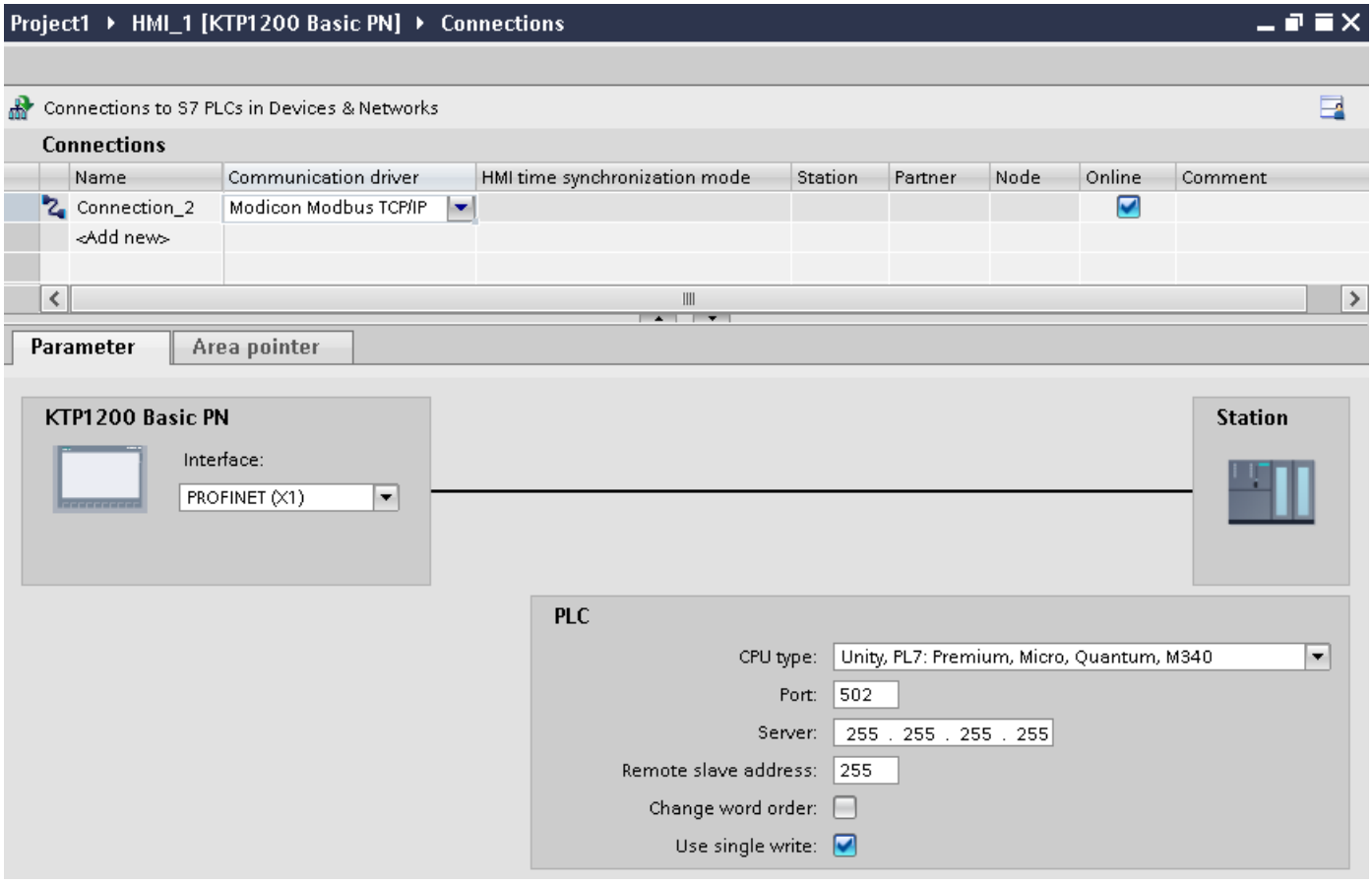
Procedure

1. Double-click the HMI device under "Devices" in the project tree.
2. Double-click the "Connections" item.

3. Double-click "<Add>" in the "Connections" editor.



4. Select the "Modicon Modbus TCP" driver in the "Communication driver" column.



5. Select all necessary connection parameters for the interface in the Inspector window under "Parameters".

See also

Modicon Modbus communication drivers (Page 515)

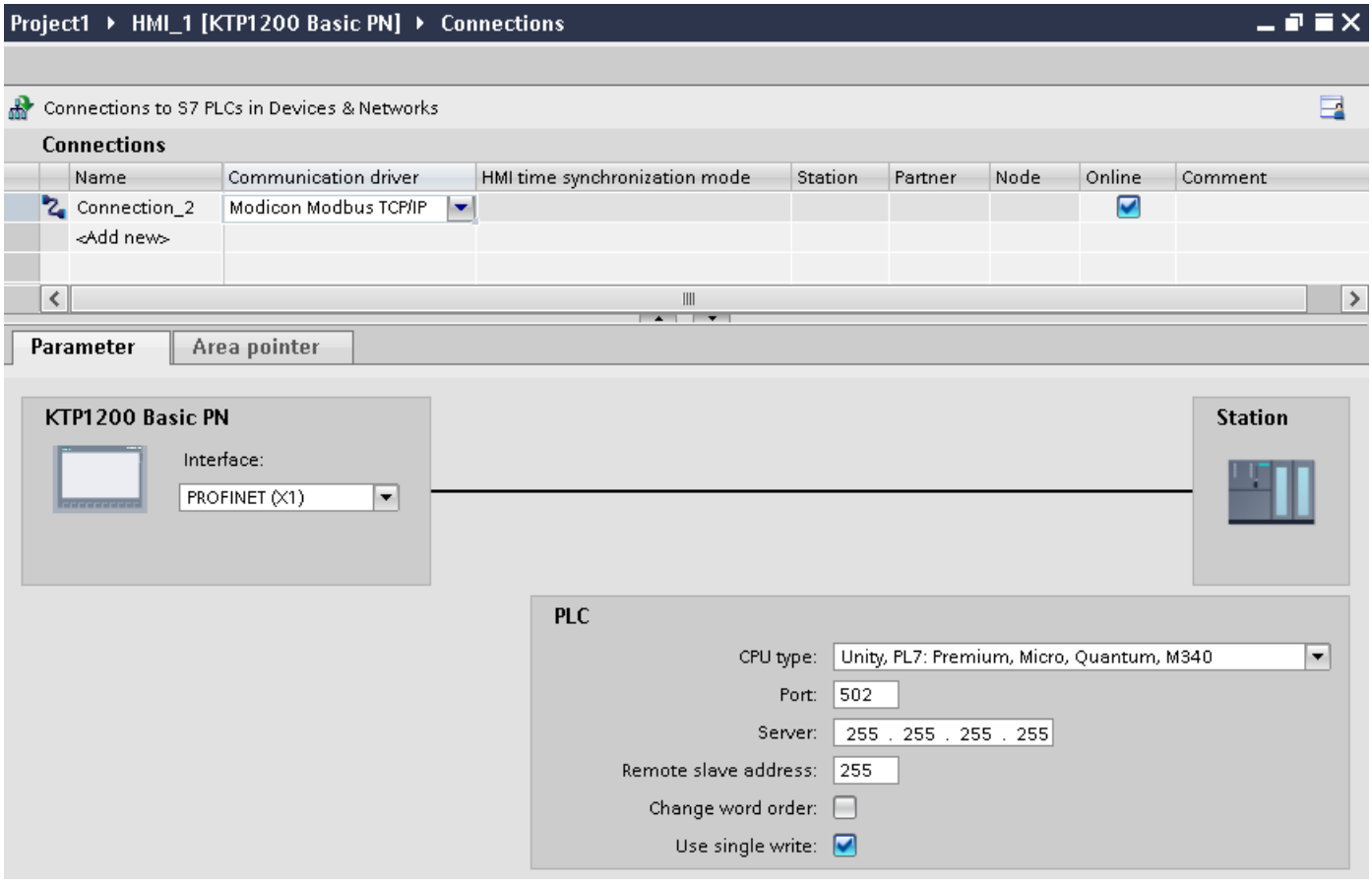
Parameters for the connection (Modicon Modbus TCP/IP) (Page 518)

Parameters for the connection (Modicon Modbus TCP/IP) (Basic Panels)

Parameters to be set

To set the connection parameters, such as addresses and profiles, click the connection that you have created in the "Connections" editor.

The communication partners are displayed schematically in the Inspector window under "Parameters". The "HMI device" and "PLC" areas are available for assigning parameters according to the interface used.



Parameters for the HMI device

You can select only one interface for the HMI device in the Inspector window under "Parameters". Depending on the HMI device, there are several interfaces available.

If you are directly connected to the HMI device during configuration, you can set up the IP address of the HMI device in WinCC. The IP address is transferred to the HMI device during project transfer.

Note

The IP address in the control panel will be overwritten upon subsequent loading if you have already set up the IP address in the HMI device control panel.

The IP address already set up in the control panel will be retained upon subsequent loading if you activate "Set IP address using a different method".

To set up the IP address of the HMI device:

1. Click the HMI device.
2. Open the "Device configuration" editor.

3. Click the Ethernet interface.
4. Assign the IP address in the inspector window under:
   "General > PROFINET interface > Ethernet addresses"

Parameters for the PLC

- "CPU type"
  For "CPU type", you set the Modicon PLC to which the HMI device is connected.
- "Port"
  For "Port", you set the port that is used for the TCP/IP connection. The port used by the Modicon PLCs is 502.
- "Server"
  You set the IP address or host name of the PLC under "Server". Only the IP address can be used on a Basic Panel.
- "Distributed slave address"
  For "Distributed slave address", set the slave address of the remote PLC, but only if a bridge is used.
  If a bridge is not used, the default value "255" (or "0") must be retained.
- "Change word order"
  The "Change word order " parameter only affects the word order of the 32-bit values display. The setting pertains to the data types Double, Double+/-, and Float. The byte order cannot be changed.
  - "Change word order" not activated
    The most significant byte is sent first.
    For double words, the least significant word is sent before the most significant word.
    This setting has been system-tested for all approved PLCs.
  - "Change word order" activated
    The most significant byte is sent first.
    For double words, the most significant word is sent before the least significant word.

    Note

    This setting must be used for the SIEMENS SENTRON PAC3200 and PAC4200 multi-function meters and can be used for PLCs of other manufacturers.
- "Use single write"
  If you deselect this function, only function codes 15H and 16H are used for writing into the PLC.
  If this function remains selected, the function codes 05H, 06H, 15H, and 16H are used.

See also

Modicon Modbus communication drivers (Page 515)

Configuring a connection via Modicon Modbus TCP/IP (Page 516)

Connecting HMI device to PLC (Basic Panels)

Connections via Modicon Modbus TCP/IP (Basic Panels)

Connection

The HMI device can be connected to the Modicon Modbus PLC using the following components:

- Existing Ethernet network that also contains the PLCs
- Cross-over Ethernet cable connected directly to the Ethernet interface of the CPU or the communication module

The connection of the HMI device to a Modicon Modbus PLC is limited primarily to the physical connection of the HMI device. Special blocks for the connection are not required in the PLC.

Note

Timeout response with TCP/IP (Ethernet)

Due to the use of the TCP/IP protocol, the breakdown of a connection is detected at the earliest after approximately one minute. Communication failure cannot be reliably detected if no tags are requested, for example, no output tags in the current screen.

Configure area pointer coordination for each PLC. This setting ensures that a communication failure is recognized after approximately two minutes, even in the aforementioned scenario.

See also

Configuring a connection via Modicon Modbus TCP/IP (Page 516)

Communication types (Basic Panels)

Approved communication types

The following communication types are system-tested and approved:

- Point-to-point coupling:
- Multiple point coupling of a HMI device (Modbus TCP/IP Client) with up to 4 PLCs, each with different couplings. CPU types can be mixed.
  The following couplings are possible:
  - Coupling the Ethernet CPU interface of the TSX Unity Quantum.
  - Coupling via the communication modules for Ethernet 140 NOE 771 01 for the TSX Quantum and TSX Unity Quantum series
  - Coupling via the Ethernet interface of the 171 CCC 980 30 CPU adapter of the Momentum series
  - Coupling the Ethernet CPU interface of the TSX Unity Premium.
  - Coupling via the Ethernet TCP/IP connect module TSX ETY 110 for the TSX Premium and TSX Unity Premium series
  - Coupling via the Ethernet TCP/IP connect module TSX ETY 410 for the Micro series
  - Coupling via the Ethernet TCP/IP Modbus Plus Bridge 174 CEV 200 40 to the Modbus Plus interface of the Compact, the TSX Quantum and the TSX Unity Quantum series

Via the TCP/IP Modbus Plus Bridge, 174 CEV 200 40, the PLCs can be accessed at their Remote Slave Address via the Ethernet interface of this bridge.

Note

Integration of the HMI device in a Modbus network via a bridge is not possible. The HMI device is the Modbus master.

Restrictions

The coupling of the HMI device to PLCs of other manufacturers who offer a Modbus TCP/IP interface is not system-tested and thus, not enabled.

However, if another PLC is to be used, observe the following instructions:

- Use the following CPU types, because these operate without address offset and in the usual bit count manner.
  - Unity, PL7: Premium, Micro, Quantum, M340
- The following function codes are used for the respective data areas:

Reading function codes		Address range	
01	ReadCoilStatus	0x / %M	DIGITAL_OUT
02	ReadInputStatus	1x / %I	DIGITAL_IN
03	ReadHoldingRegisters	4x / %MW	USERDATA

Reading function codes		Address range	
04	ReadInputRegisters	3x / %IW	ANALOG_IN
20 (14Hex)	ReadGeneralReference	6x / –	EXTENDEDMEMORY (not for all CPUs)

Writing function codes		Address range	
06 [1]	PresetSingleRegister	4x / %MW	USERDATA Single
16 (10Hex)	PresetMultipleRegisters	4x / %MW	USERDATA Multiple
05 [1]	ForceSingleCoil	0x / %M	DIGITAL_OUT with BIT
15 (0FHex)	ForceMultipleCoils	0x / %M	DIGITAL_OUT with 16 BIT GROUP
21 (15Hex)	WriteGeneralReference	6x / –	EXTENDEDMEMORY (not for all CPUs)

[1] Select use with "Use single write".

Connectable PLCs

Connections can be implemented for the following Modicon Modbus PLCs:

Modicon Modbus PLC	Supported protocol	
Modicon Modbus RTU [2]	Modicon Modbus TCP/IP	
TSX Compact	x	x [1]
TSX Quantum	x	x
Momentum	x	x
Premium	-	x
Micro	-	x
M340 20x0 (without 2010)	-	x

[1] Only via Ethernet TCP/IP Modbus Plus Bridge

[2] Communication via RS 232 is tested and enabled for the PLC. In the HMIs that only have a RS 422/485 interface, the RS 422/232 converter with the order number 6AV6 671-8XE00-0AX0 was tested and enabled.

See also

Configuring a connection via Modicon Modbus TCP/IP (Page 516)

Performance features of communication (Basic Panels)

Permissible data types for Modicon Modbus TCP/IP (Basic Panels)

Permitted data types

The table lists the data types that can be used when configuring tags and area pointers.

Note

If you change the Modicon Modbus RTU communication driver to Modicon Modbus TCP/IP, the string in the "String" data type may be different.

Permitted data types for CPU type "Unity, PLC: Premium, Micro, Quantum M340"

Data type	Operand type	Length
+/- Double	%MW	4 bytes
+/- Int	%MW, %IW	2 bytes
16-bit group	%MW, %I	2 bytes
ASCII	%MW	0 to 80 characters
Bit	%MW, %IW, %M, %I	1-bit
Double	%MW	4 bytes
Float	%MW	4 bytes
Int	%MW, %IW	2 bytes

Note

The ranges "%I" and "%IW" are not supported for the following CPU types:

- Premium
- Micro
- M340

Permitted data types for CPU type "Concept, ProWORX: Compact, Quantum, Momentum"

Data type	Operand type	Length
+/- Double	4x, 6x	4 bytes
+/- Int	3x, 4x, 6x	2 bytes
16-bit group	0x, 1x	2 bytes
ASCII	4x, 6x	0 to 80 characters
Bit	0x, 1x, 3x, 4x, 6x	1-bit
Double	4x, 6x	4 bytes

Data type	Operand type	Length
Float	4x, 6x	4 bytes
Int	3x, 4x, 6x	2 bytes

Bit counting method

The usual bit counting method "16 LSB - 1 MSB" in the following CPU types is only used in the "HMI tags" editor with the selected "Bit" data type:

- Concept, ProWORX: Compact, Quantum, Momentum

The following bit location assignment applies:

	Left byte								Right byte							
Counting with tags	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Format for "Signed"

The placeholder "+/-" stands for the data types "Signed Int" and "Signed Double".

See also

Modicon Modbus communication drivers (Page 515)

Supported CPU types for Modicon Modbus TCP/IP (Basic Panels)

CPU types

The following CPU types are supported for configuring the Modicon Modbus TCP/IP communication driver.

- Compact
- Momentum
- Quantum
  - Concept Quantum
  - Unity Quantum
- Micro
- Premium
- Modicon M340
  - 20x0 (except 2010)

See also

Modicon Modbus communication drivers (Page 515)

Address areas for Modicon Modbus TCP/IP (Basic Panels)

UnityPl7

Address areas	Data types							
	Bool	16-bit group	Int	+/- Int	DInt	+/- DInt	Float	ASCII
%1	%I0 - %I65535	%I65535 %I0 - %I65520	--	--	--	--	--	--
%M	%M0 - %M65535	%M65535 %M0 - %M65520	--	--	--	--	--	--
%IW	%IW0.0 - %IW65535.15	--	%IW0 - %IW65535	%IW0 - %IW65535	--	--	--	--
%MW	%MW0.0 - %MW65535.15	--	%MW0 - %MW65535	%MW0 - %MW65535	%MW0 - %MW65534	%MW0 - %MW65534	%MW0 - %MW65534	%MW0 - %MW65535

ConceptProWORX

Address areas	Data types							
	Bool	16-bit group	Int	+/- Int	DInt	+/- DInt	Float	ASCII
0x	0x1 - 0x65535	0x1 - 0x65520	--	--	--	--	--	--
1x	1x100001 - 1x165535	1x100001 - 1x165520	--	--	--	--	--	--
3x	3x300001.1 - 3x365535.16	--	3x300001 - 3x365535	3x300001 - 3x365535	--	--	--	--
4x	4x400001.1 - 4x465535.16	--	4x400001 - 4x465535	4x400001 - 4x465535	4x400001 - 4x465534	4x400001 - 4x465534	4x400001 - 4x465534	4x400001 - 4x465535
6x	6x60000.1:1 - 6x69999.16:10	--	6x60000:1 - 6x69999:10	6x60000:1 - 6x69999:10	6x60000:1 - 6x69998:10	6x60000:1 - 6x69998:10	6x60000:1 - 6x69998:10	6x60000:1 - 6x69999:10

Note

When you select the address format for Unity Pro when configuring the connection between the HMI device and a MODBUS Quantum controller, an offset of 1 is configured for each address.

This is due to different address formats of the controller and Unity Pro. The valid address area for MODBUS Quantum controllers begins with "0x400001", "0x300001", etc., while the address range for Unity Pro begins with "%MW0". For mapping, an offset of 1 is added for each read or write access by MODBUS Quantum controllers.

See also

Modicon Modbus communication drivers (Page 515)

Commissioning components (Basic Panels)

Transferring a project to the HMI device

1. Switch the HMI device to "transfer mode".
2. Set all necessary transfer parameters.
   - Interface
   - Transfer parameters
   - Target storage location
3. Start the project transfer.
   The project is compiled automatically.
   All compilation and transfer steps are logged to a message window.

Interconnecting the PLC with the HMI device

1. Interconnect the PLC with the HMI device using a suitable cable.
2. The message "Connection to PLC .... is established" is output to the HMI device.

Optimizing the configuration (Basic Panels)

Acquisition cycle and update time

The acquisition cycles for the "Area pointers" and of the tags defined in the configuration software are decisive in terms of the update times which can actually be achieved.

The update time is equivalent to the acquisition cycle + transmission time + processing time.

Items to observe when optimizing the update times in configuration data:

- Optimize the maximum and minimum size of the data areas.
- Acquisition cycles which are too short lead to unnecessary load on overall performance. Set the acquisition cycle according to the rate of change of the process values. The rate of temperature changes at a furnace, for example, is significantly slower compared to the speed rate of an electrical drive. A time of approx. 1 second is a benchmark for the acquisition cycle.
- Avoid any gaps when entering the alarm or screen tags in a data area.
- Changes in the PLC can only be detected reliably if these are available at least within the actual acquisition cycle.

Screens

The refresh rate of screens is determined by the type and volume of data to be visualized.

Only configure short acquisition cycles for objects which actually require shorter refresh cycles. This procedure reduces update times.

Trends

The HMI device always updates all bit-triggered trends whose group bit is set in the "Trend transfer area". It resets the bits in the next cycle.

The group bit in the PLC program can only be set again after the HMI device has reset all bits.

Job mailboxes

A high rate and volume of job mailboxes transferred may lead to overload in communication between the HMI device and the PLC.

The HMI device confirms acceptance of the job mailbox by entering the value zero in the first data word of the job mailbox. The HMI device now processes the job for which it requires a certain time slice. It may take the HMI device some time to process a new job mailbox which is transferred in immediate succession to the job mailbox. The next job mailbox is only accepted if sufficient computing resources are available.

Modicon Modbus RTU (Basic Panels)

Configuring a connection via Modicon Modbus RTU (Basic Panels)

Introduction

You configure a connection to a PLC with a Modicon Modbus RTU communication driver in the "Connections" editor of the HMI device.

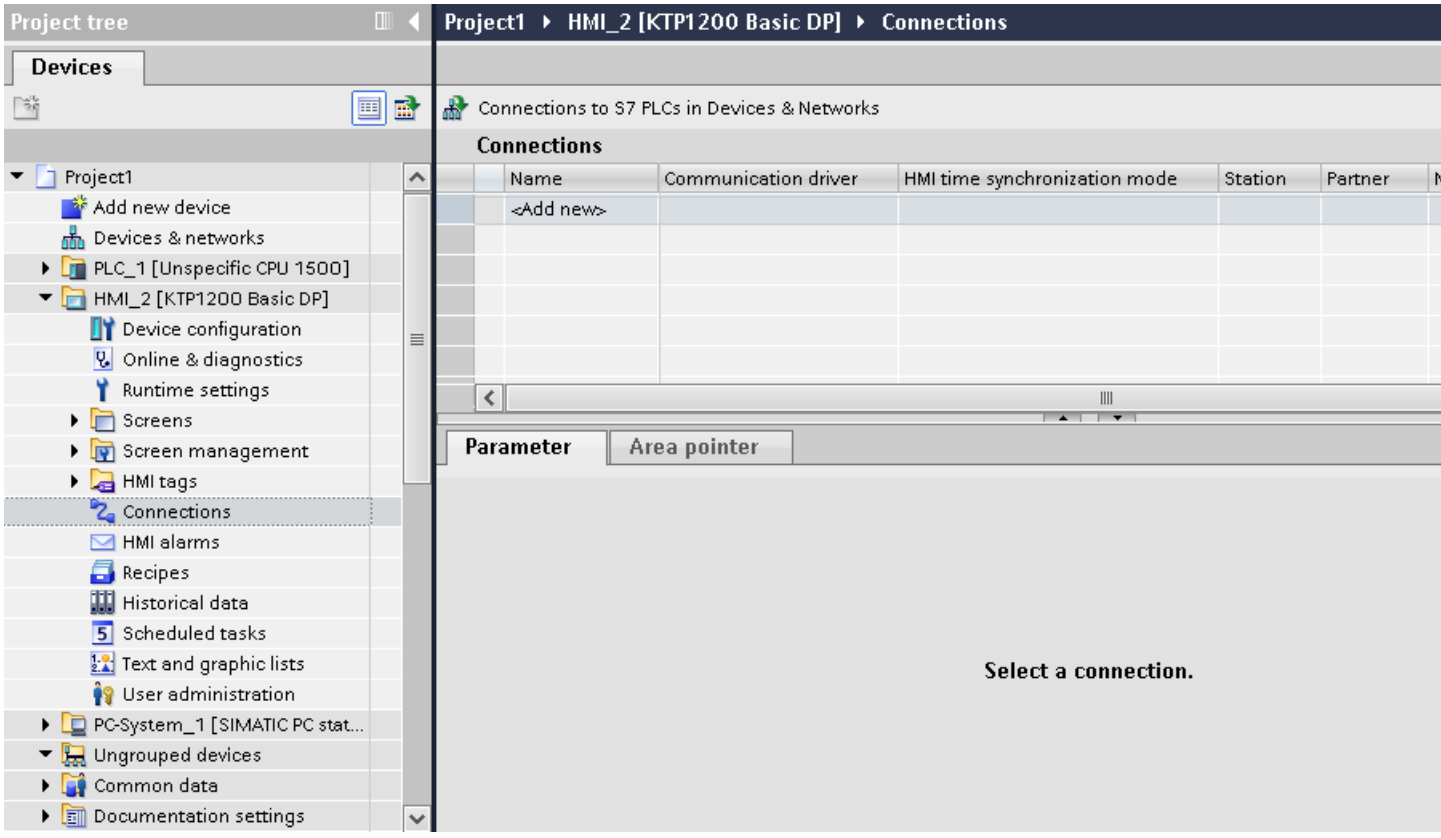
The interfaces are named differently depending on the HMI device.

Requirements

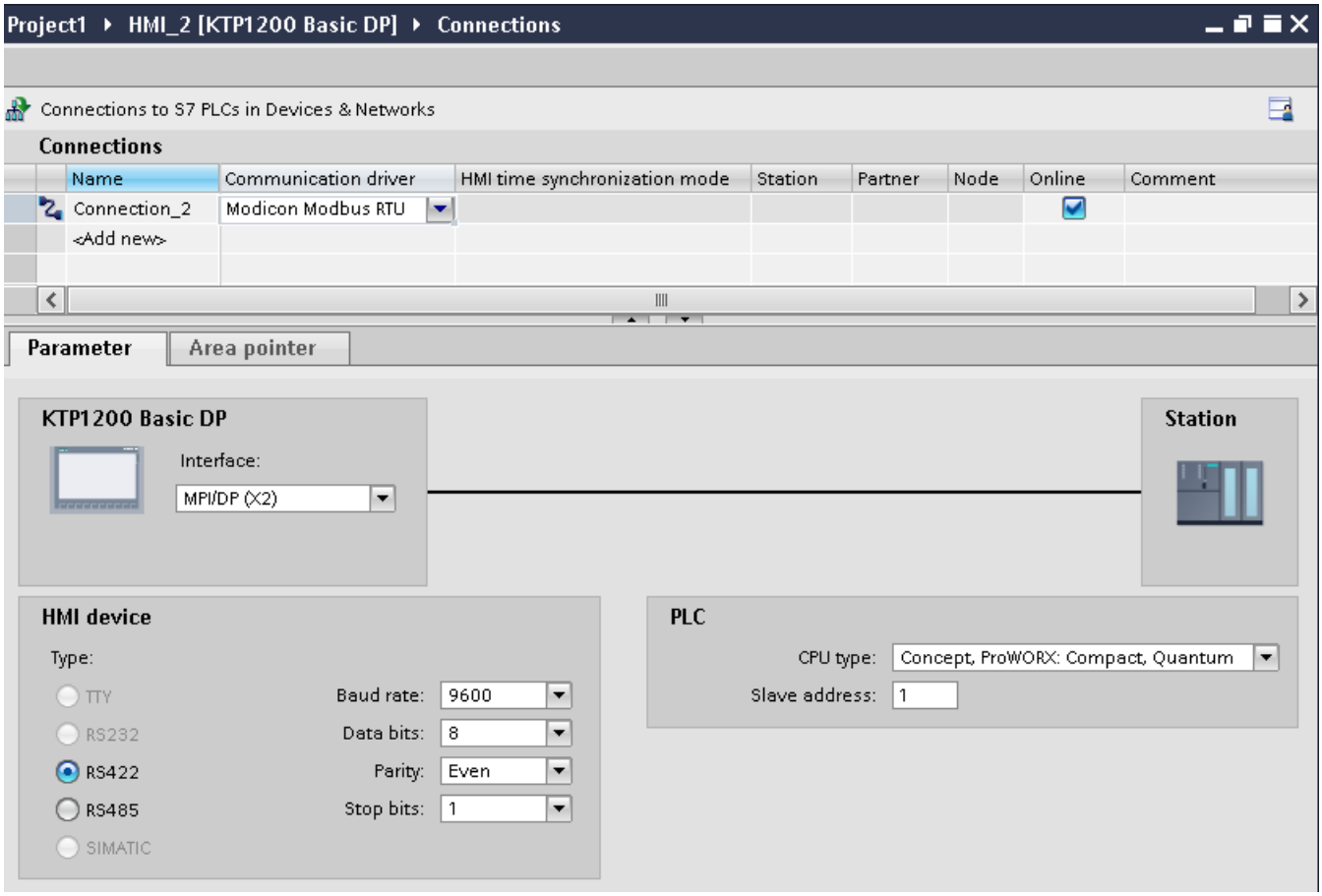
- A project is open.
- An HMI device has been created.

Procedure

1. Double-click the HMI device under "Devices" in the project tree.
2. Double-click the "Connections" item.
3. Double-click "<Add>" in the "Connections" editor.



4. In the "Communication drivers" column, select the "Modicon Modbus RTU" driver.



5. Select all necessary connection parameters for the interface in the Inspector window under "Parameters".

See also

Modicon Modbus communication drivers (Page 515)

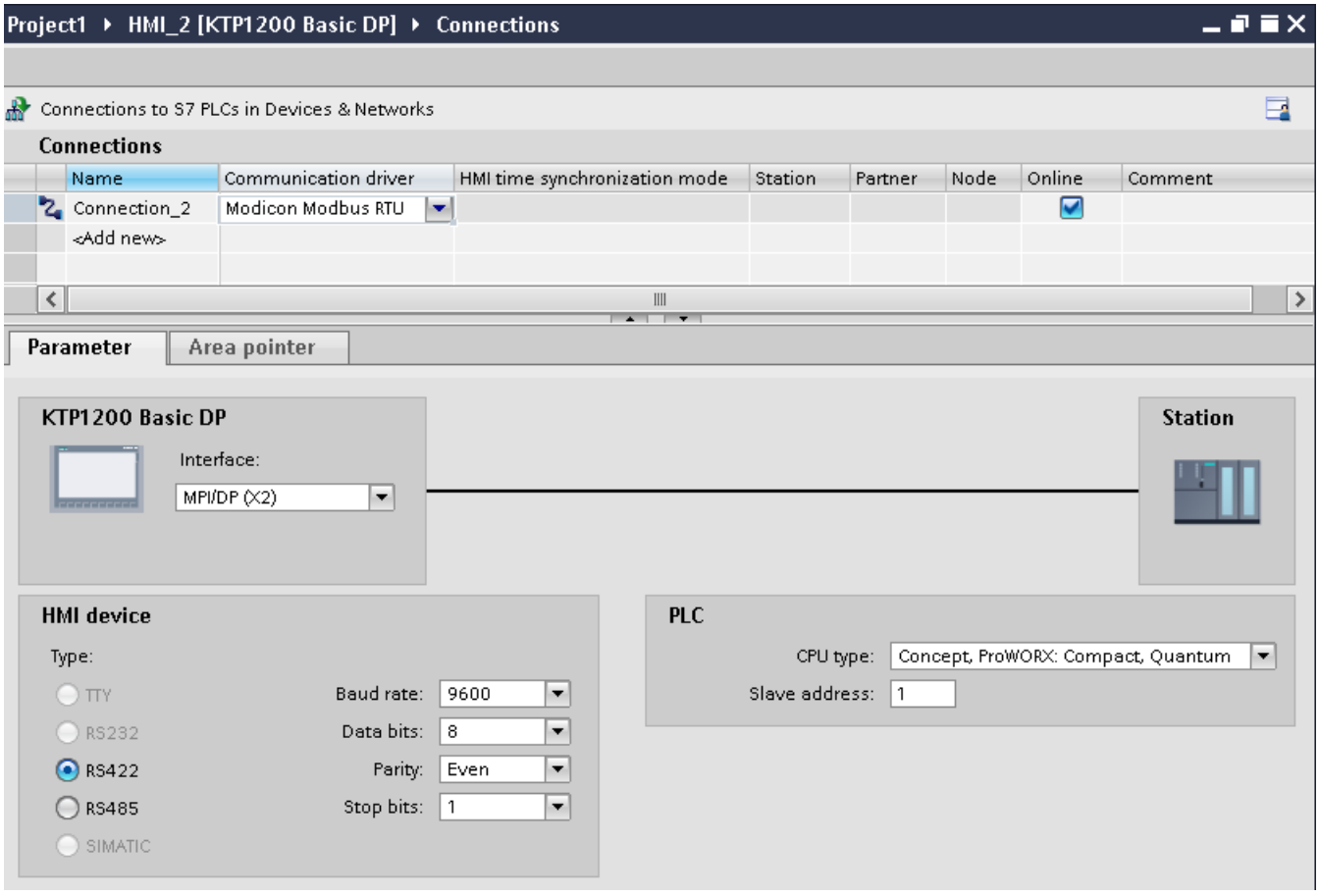
Parameters for the connection (Modicon Modbus RTU) (Page 530)

Parameters for the connection (Modicon Modbus RTU) (Basic Panels)

Parameters to be set

To set the connection parameters, such as addresses and profiles, click the connection that you have created in the "Connections" editor.

The communication partners are displayed schematically in the Inspector window under "Parameters". The "HMI device" and "PLC" areas are available for assigning parameters according to the interface used.



Parameters for the HMI device

You can select an interface for the HMI device in the Inspector window under "Parameters". Depending on the HMI device, there are several interfaces available.

- Type
  Only RS 232 is system-tested.
  No warranty is given for RS 485.

  Note

  RS 422 is only approved in combination with the RS 422-RS 232 converter.

  Order number: 6AV6 671-8XE00-0AX0

  Note

  If you use the IF1B interface, you must switch over the RS422 receive data additionally by 4 DIL-switches on the back of the HMI device.

- Baud rate
  For "Baud rate", select the transmission speed between the HMI device and Modicon PLC.
  A baud rate of 19200 or 9600 can be selected for the communication.
  A baud rate of 4800 can be selected for certain HMI devices.

- Data bits
  For "Data bits", only the value "8" can be selected.
- Parity
  For "Parity", you can choose from "None", "Even", and "Odd".
- Stop bits
  For "Stop bits", you can choose between 1 and 2 bits.

Parameters for the PLC

- CPU type
  For "CPU type", you set the Modicon PLC to which the HMI device is connected.
  You can select the following CPUs:
  – Concept, ProWORX: Compact, Quantum
- Slave address
  Under "Slave address" you set which slave address the CPU has.

See also

Modicon Modbus communication drivers (Page 515)

Configuring a connection via Modicon Modbus RTU (Page 528)

Connecting HMI device to PLC (Basic Panels)

Connections via Modicon Modbus RTU (Basic Panels)

Connection

Connect the HMI device to the Modicon Modbus RTU interface of the Modicon Modbus RTU slave.

The connection of the HMI device to Modicon is limited primarily to the physical connection of the HMI device. Special blocks for the connection are not required in the PLC.

Connection cable

The following connecting cables are available to connect the HMI device to Modicon Modbus.

Interface to the HMI device	Modicon PLC		
	directly via Modbus interface (RS232) 9-pin Sub D male connector	Via MB Bridge (RS 232)	directly via Modbus interface (RS232) 8-pin RJ45 connector
RS 232, 9-pin	PP1	PP1	PP2

The cable pin assignments can be found in Section "Connecting cables for Modicon Modbus RTU".

See also

Configuring a connection via Modicon Modbus RTU (Page 528)

Communication types (Page 533)

Communication types (Basic Panels)

Approved communication types

The following communication types are system-tested and approved:

- Ppoint-to-point connection only via the RS-232 interface.
- Multipoint connection from a HMI device (Modbus-Master) with up to 4 PLCs: The HMI device must be connected with a Modbus Plus Bridge or a Compact, Momentum CPU or TSX Quantum CPU which is configured as a Modbus Plus Bridge.
- You connect the other PLCs via the Modbus Plus connection on the first PLC. The PLCs can be reached under their address via the bridge functionality of the first PLC.

  Note

  It is not possible to integrate the HMI device into a Modbus network because the HMI device is Modbus-Master.

- Integration of the HMI device into a Modbus Plus network via the "bridge mode" of the Compact, Momentum or Quantum (logical point-to-point communication of the HMI device with a Compact, Momentum or Quantum).

Restrictions

The connection of the HMI device to PLCs of other manufacturers which offer a Modicon Modbus interface is not system-tested and therefore not approved.

If you use another PLC nevertheless, observe the following information:

- These drivers only work for tags with the bit counting method typical for Modicon PLCs from left (bit1 = most significant bit) to right (bit16 = least significant bit in data type INT).
- The address offset displayed in the configuring is subtracted at protocol level in the message frame. E.g. in Holding Register 4x the offset "40001". The configured address "40006" therefore becomes address "5" in the message frame. The address (e.g. "5") transferred in the message frame is transformed to the PLC-specific address range in the different Non-Modicon PLCs.
- A reply message frame without "ExceptionCode" is expected within 500 ms.
- The following function codes are used for the respective data areas:

Reading function codes		Address range	
01	ReadCoilStatus	0x	DIGITAL_OUT
02	ReadInputStatus	1x	DIGITAL_IN
03	ReadHoldingRegisters	4x	USERDATA

Reading function codes		Address range	
04	ReadInputRegisters	3x	ANALOG_IN
20 (14Hex)	ReadGeneralReference	6x	EXTENDEDMEMORY (not for all CPUs)

Writing function codes		Address range	
06	PresetSingleRegister	4x	USERDATA Single
16 (10Hex)	PresetMultipleRegisters	4x	USERDATA Multiple
05	ForceSingleCoil	0x	DIGITAL_OUT with data type Bit
15 (0FHex)	ForceMultipleCoils	0x	DIGITAL_OUT with data type 16 bit group
21 (15Hex)	WriteGeneralReference	6x	EXTENDEDMEMORY (not for all CPUs)

Connectable PLCs

Connections can be implemented for the following Modicon Modbus PLCs:

Modicon Modbus PLC	Supported protocol	
Modicon Modbus RTU [2]	Modicon Modbus TCP/IP	
TSX Compact	x	x [1]
TSX Quantum	x	x
Momentum	x	x
Premium	-	x
Micro	-	x
M340 20x0 (without 2010)	-	x

[1] Only via Ethernet TCP/IP-Modbus Plus Bridge

[2] Communication via RS 232 is tested and enabled for the PLC. In the HMI devices which only have an RS 422/485 interface, the RS 422/232 converter with the order number 6AV6 671-8XE00-0AX0 was tested and approved.

See also

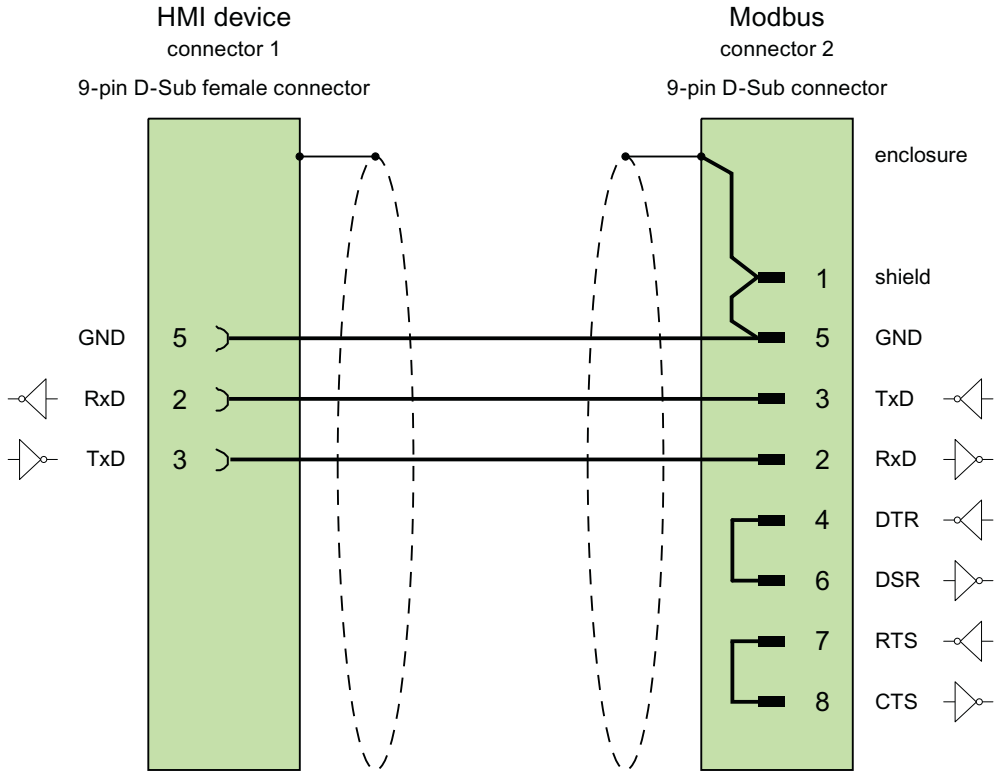
Configuring a connection via Modicon Modbus RTU (Page 528)

Connections via Modicon Modbus RTU (Page 532)

Connecting cables for Modicon Modbus RTU (Basic Panels)

Connecting cable PP1, RS-232, for Modicon (Basic Panels)

Point-to-point cable 1: PLC > PC ...



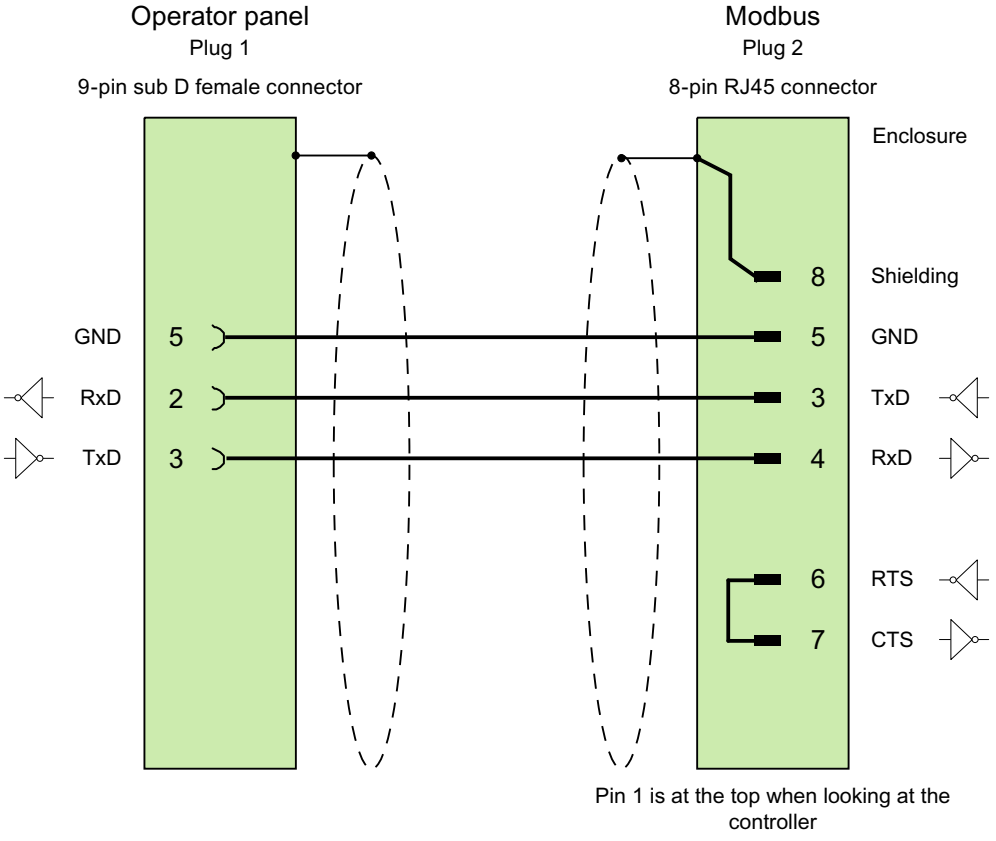
Cables: 3 x 0.14 mm^2, shielded,
max. length 15 m

See also

Connections via Modicon Modbus RTU (Page 532)

Connecting cable PP2, RS-232, for Modicon (Basic Panels)

Point-to-point cable 2: PLC (TSX Compact) > PC...



Cables: 3 x 0.14 mm^2, shielded,
max. length 15 m

See also

Connections via Modicon Modbus RTU (Page 532)

Performance features of communication (Basic Panels)

Permitted data types for Modicon Modbus RTU (Basic Panels)

Permitted data types

The table lists the data types that can be used when configuring tags and area pointers.

Data type	Operand type	Length
+/- Double	4x, 6x	4 bytes
+/- Int	3x, 4x, 6x	2 bytes
16-bit group	0x, 1x	2 bytes
ASCII	4x, 6x	0 to 80 characters
Bit [1]	0x, 1x, 3x, 4x, 6x	1-bit
Double	4x, 6x	4 bytes
Float	4x, 6x	4 bytes
Int	3x, 4x, 6x	2 bytes

[1] Note the following for write accesses:

For data type "Bit" with the operand types "4x" and "6x", the entire word is written back to the PLC following a change to the specified bit. There is no check to determine whether any other bits in the word have changed. As a result, the PLC only has read access to the specified word.

The usual bit counting method (16 LSB - 1 MSB) in the following CPU types is only used in the "HMI tags" editor with the selected "Bit" data type:

- Concept ProWORX: Compact, Quantum

The following bit location assignment applies:

	Left byte								Right byte							
Counting with tags	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Format for "Signed"

The placeholder "+/-" stands for the data types "Signed Int" and "Signed Double".

See also

Modicon Modbus communication drivers (Page 515)

Supported CPU types for Modicon Modbus RTU (Basic Panels)

CPU types

The following CPU types are supported in the configuration of the Modicon Modbus RTU communication driver.

- Compact
- Momentum
- Quantum

See also

Modicon Modbus communication drivers (Page 515)

Address areas for Modicon Modbus RTU (Basic Panels)

UnityPl7

Address areas	Data types							
	Bool	16-bit group	Int	+/- Int	DInt	+/- DInt	Float	ASCII
%1	%I0 - %I65535	%I65535 %I0 - %I65520	--	--	--	--	--	--
%M	%M0 - %M65535	%M65535 %M0 - %M65520	--	--	--	--	--	--
%IW	%IW0.0 - %IW65535.15	--	%IW0 - %IW65535	%IW0 - %IW65535	--	--	--	--
%MW	%MW0.0 - %MW65535.15	--	%MW0 - %MW65535	%MW0 - %MW65535	%MW0 - %MW65534	%MW0 - %MW65534	%MW0 - %MW65534	%MW0 - %MW65535

ConceptProWORX

Address areas	Data types							
	Bool	16-bit group	Int	+/- Int	DInt	+/- DInt	Float	ASCII
0x	0x1 - 0x65535	0x1 - 0x65520	--	--	--	--	--	--
1x	1x100001 - 1x165535	1x100001 - 1x165520	--	--	--	--	--	--

Address areas	Data types							
	Bool	16-bit group	Int	+/- Int	DInt	+/- DInt	Float	ASCII
3x	3x300001.1 - 3x365535.16	--	3x300001 - 3x365535	3x300001 - 3x365535	--	--	--	--
4x	4x400001.1 - 4x465535.16	--	4x400001 - 4x465535	4x400001 - 4x465535	4x400001 - 4x465534	4x400001 - 4x465534	4x400001 - 4x465534	4x400001 - 4x465535
6x	6x60000.1:1 - 6x69999.16:10	--	6x60000:1 - 6x69999:10	6x60000:1 - 6x69999:10	6x60000:1 - 6x69998:10	6x60000:1 - 6x69998:10	6x60000:1 - 6x69998:10	6x60000:1 - 6x69999:10

Note

When you select the address format for Unity Pro when configuring the connection between the HMI device and a MODBUS Quantum controller, an offset of 1 is configured for each address.

This is due to different address formats of the controller and Unity Pro. The valid address area for MODBUS Quantum controllers begins with "0x400001", "0x300001", etc., while the address range for Unity Pro begins with "%MW0". For mapping, an offset of 1 is added for each read or write access by MODBUS Quantum controllers.

See also

Modicon Modbus communication drivers (Page 515)

Commissioning components (Basic Panels)

Transferring a project to the HMI device

1. Switch the HMI device to "transfer mode".
2. Set all necessary transfer parameters.
   - Interface
   - Transfer parameters
   - Target storage location
3. Start the project transfer.
   The project is compiled automatically.
   All compilation and transfer steps are logged to a message window.

Interconnecting the PLC with the HMI device

1. Interconnect the PLC with the HMI device using a suitable cable.
2. The message "Connection to PLC .... is established" is output to the HMI device.

Optimizing the configuration (Basic Panels)

Acquisition cycle and update time

The acquisition cycles for the "Area pointers" and of the tags defined in the configuration software are decisive in terms of the update times which can actually be achieved.

The update time is equivalent to the acquisition cycle + transmission time + processing time.

Items to observe when optimizing the update times in configuration data:

- Optimize the maximum and minimum size of the data areas.
- Acquisition cycles which are too short lead to unnecessary load on overall performance. Set the acquisition cycle according to the rate of change of the process values. The rate of temperature changes at a furnace, for example, is significantly slower compared to the speed rate of an electrical drive. A time of approx. 1 second is a benchmark for the acquisition cycle.
- Avoid any gaps when entering the alarm or screen tags in a data area.
- Changes in the PLC can only be detected reliably if these are available at least within the actual acquisition cycle.

Screens

The refresh rate of screens is determined by the type and volume of data to be visualized.

Only configure short acquisition cycles for objects which actually require shorter refresh cycles. This procedure reduces update times.

Trends

The HMI device always updates all bit-triggered trends whose group bit is set in the "Trend transfer area". It resets the bits in the next cycle.

The group bit in the PLC program can only be set again after the HMI device has reset all bits.

Job mailboxes

A high rate and volume of job mailboxes transferred may lead to overload in communication between the HMI device and the PLC.

The HMI device confirms acceptance of the job mailbox by entering the value zero in the first data word of the job mailbox. The HMI device now processes the job for which it requires a certain time slice. It may take the HMI device some time to process a new job mailbox which is transferred in immediate succession to the job mailbox. The next job mailbox is only accepted if sufficient computing resources are available.

Data exchange (Basic Panels)

Area pointers for Modicon Modbus (Basic Panels)

Area pointers for connections via Modicon Modbus communication drivers

You use an area pointer to access a data area in the PLC.

For more detailed information on area pointers and their configuration, refer to Section "Data exchange using area pointers (Page 649)".

Special considerations for connections via Modicon communication drivers

You can configure the following area pointers

Area pointers	Modicon Modbus TCP/IP	Modicon Modbus RTU
Screen number	Yes	Yes
Date/time	Yes	Yes
Date/time PLC	Yes	Yes
Coordination	Yes	Yes
Project ID	Yes	Yes
Job mailbox	Yes	Yes
Data record	Yes	Yes

Restrictions Modicon Modbus TCP/IP

The following restrictions apply for configuring area pointers.

CPU type	Data types	File types
Concept, ProWORX: Compact, Quantum, Momentum	+/- Int, Int	4x, 6x
Unity, PL7: Premium, Micro, Quantum, M340	+/- Int, Int	%MW

Modicon Modbus RTU restrictions

The following restrictions apply for configuring area pointers.

CPU type	Data types	File types
Concept, ProWORX: Compact, Quantum, Momentum	+/- Int, Int	4x, 6x

See also

Data exchange using area pointers (Page 649)

Trends (Basic Panels)

Trends

A trend is the graphical representation of one or more values from the PLC. The value is read out time-triggered for Basic Panels.

For additional information see:

AUTOHOTSPOT

Time-triggered trends

The HMI device reads in the trend values cyclically at an interval specified in the configuration.

Time-triggered trends are suitable for continuous curves, such as the operating temperature of a motor.

Alarms (Basic Panels)

Configuring alarms (Basic Panels)

Configuring alarms for non-integrated connections

Several steps are necessary to configure alarms such as warnings, error messages and acknowledgement.

- Step 1: Create tags
- Step 2: Configure alarms
- Step 3: Configure acknowledgment

You can find additional information in the section:

AUTOHOTSPOT

Distinctive features when configuring alarms

If you are configuring connections of HMI devices to PLCs of other manufacturers, note the following distinctive features when configuring:

- Data types of the tags
- Addressing of tags
- How the bit positions are counted

Data types

For connections with a Modicon Modbus communication driver, the following data types are supported:

PLC	Permitted data types	
	Discrete alarms	Analog alarms
All Modicon series	Int, +/-Int	16 Bit Group, Int, +/-Int, Double, +/-Double, Float

Arrays and array tags cannot be used for discrete alarms.

How the bit positions are counted

For connections with a Modicon Modbus communication driver, the following counting method applies:

How the bit positions are counted	Left byte								Right byte							
In WinCC you configure:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

See also

Acknowledgment of alarms (Page 543)

Acknowledgment of alarms (Basic Panels)

Procedure

Create suitable tags on the PLC to acknowledge an error alarm. You assign these tags to an alarm in the "Bit messages" editor. You make the assignment in "Properties > Acknowledgment".

Distinction in terms of acknowledgment:

- Acknowledgment by the PLC
- Acknowledgment on the HMI device

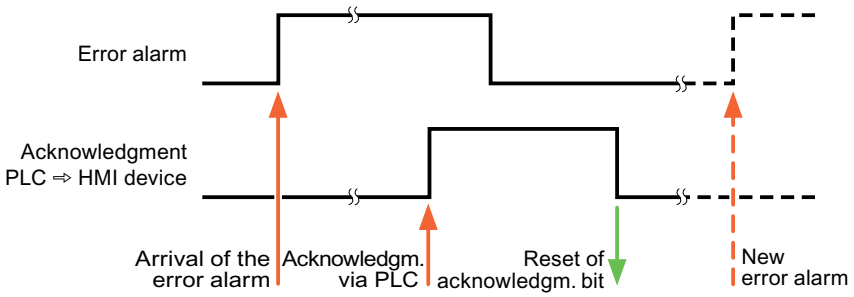
Acknowledgment by the PLC

In "PLC acknowledgment tag", configure the tag or array tag and the bit number that the HMI device uses to identify a PLC acknowledgment.

A bit set in the tag triggers acknowledgment of the assigned error alarm bit at the HMI device. This tag bit returns a function similar to acknowledgment on the HMI device which is triggered by pressing the "ACK" button, for example.

The acknowledgment bit must be located in the same tag as the bit for the error alarm.

Reset the acknowledgment bit before setting the bit in the alarm area again. The figure below shows the pulse diagram.



Acknowledgment on the HMI device

In the "HMI acknowledgment tag" area, configure the tag or array tag as well as the bit number that the HMI device writes to the PLC after acknowledgment. Make sure when you use an array tag that it is not longer than 6 words.

To always create a signal change when setting an assigned acknowledgment bit of a discrete alarm that must be acknowledged, the HMI device will reset the acknowledgment bit assigned to the alarm as soon as it detects an alarm subject to acknowledgment and write the acknowledgment tag in the PLC. There will be a certain delay between detecting the message and writing the acknowledgment tag in the PLC because the HMI device has to process the operations.

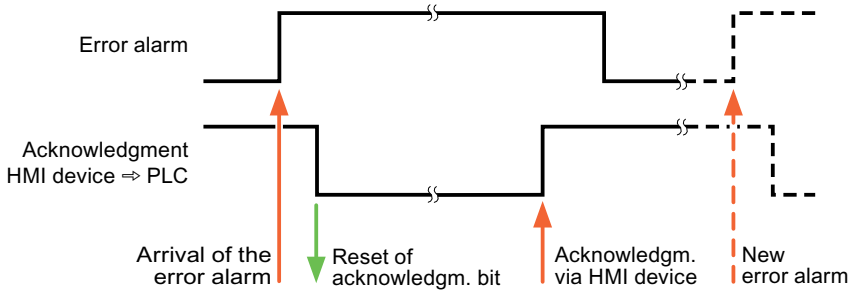
If a discrete alarm subject to acknowledgment is acknowledged by the HMI device, then the corresponding bit in the assigned acknowledgment tag will be set. The entire acknowledgment tag is then written to the PLC by the HMI device. This allows the PLC to recognize that a certain alarm message has been acknowledged at the HMI device.

Note

All alarm bits acknowledged since the last Runtime start will remain in the acknowledgment tag until a new incoming of the respective discrete alarms is detected.

This area should only be read by the PLC because the entire section of the HMI device will be overwritten once the next acknowledgment tag is written.

The figure below shows the pulse diagram.



See also

Configuring alarms (Page 542)

1.13.3.4 Omron (Basic Panels)

Omron communication drivers (Basic Panels)

Introduction

This section describes the communication between an HMI device and PLCs that use Omron communication drivers.

The following communication drivers are supported:

- Omron Host Link

Data exchange

Data is exchanged by means of tags or area pointers.

- Tags
  The PLC and the HMI device use process values for data exchange. You create tags in the configuration that point to addresses in the PLC. The HMI device reads the value from the defined address, and then displays it. The operator may also enter values on the HMI device, which are then written to the address in the PLC.
- Area pointers
  Area pointers are used to exchange specific data and are only set up when these data are used.

Omron Host Link (Basic Panels)

Configuring a connection via Omron Host Link (Basic Panels)

Introduction

You configure a connection to a PLC with an Omron Host Link communication driver in the "Connections" editor of the HMI device.

Note

Connection with Omron Host Link

A connection will not automatically be established when runtime is started if you have configured a connection via Omron.

A tag which is in the valid PLC memory area must be configured in the runtime start screen.

The connection will otherwise only be established once a corresponding screen has been selected.

This tag will be accessed when runtime is started and a connection will then be established.

The interfaces are named differently depending on the HMI device.

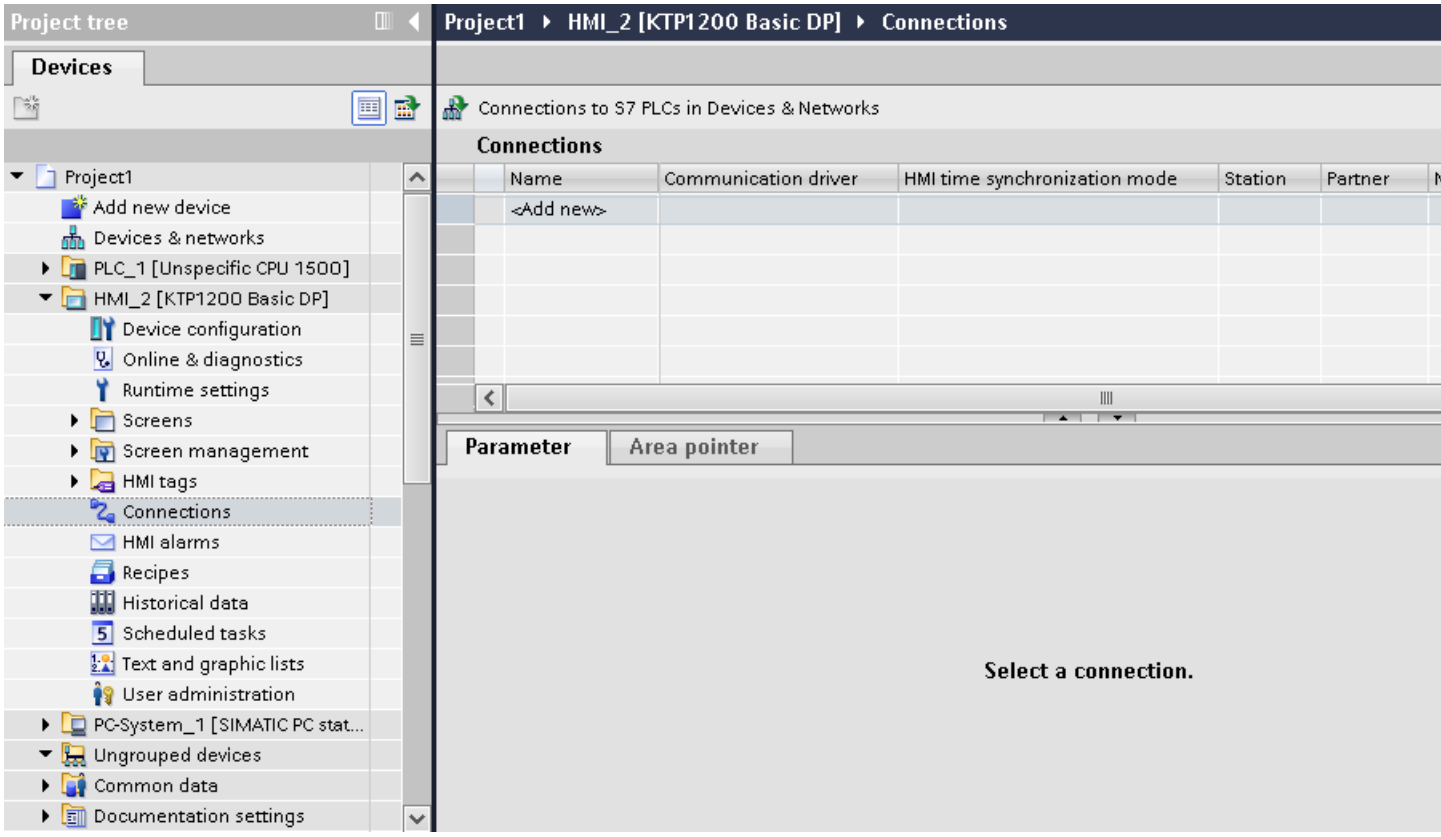
Requirements

- A project is open.
- An HMI device has been created.

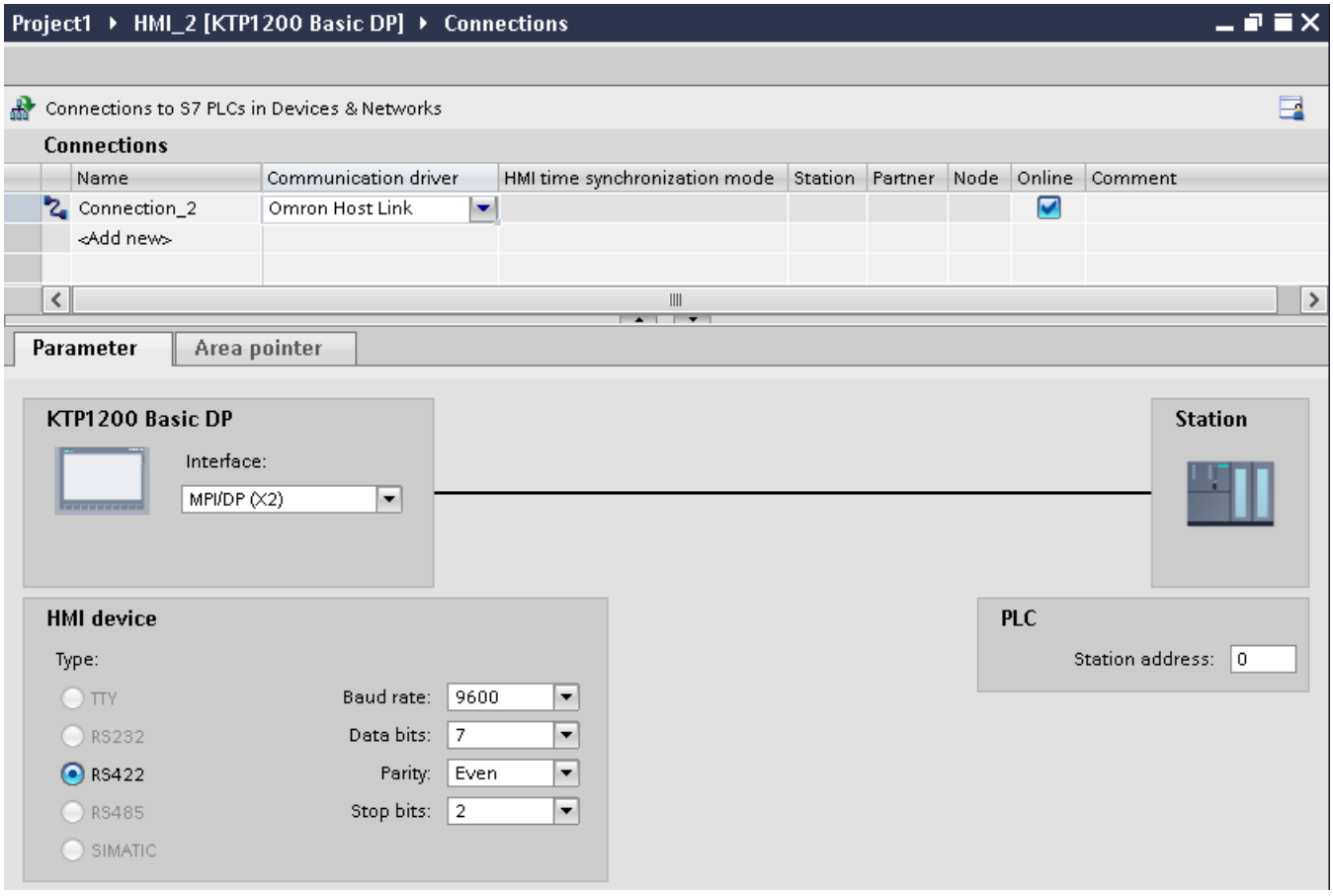
Procedure

1. Double-click the HMI device under "Devices" in the project tree.
2. Double-click the "Connections" item.

3. Double-click "<Add>" in the "Connections" editor.



4. Select the "Omron Host Link" driver in the "Communication driver" column.



5. Select all necessary connection parameters for the interface in the inspector window under "Parameters".

See also

Omron communication drivers (Page 545)

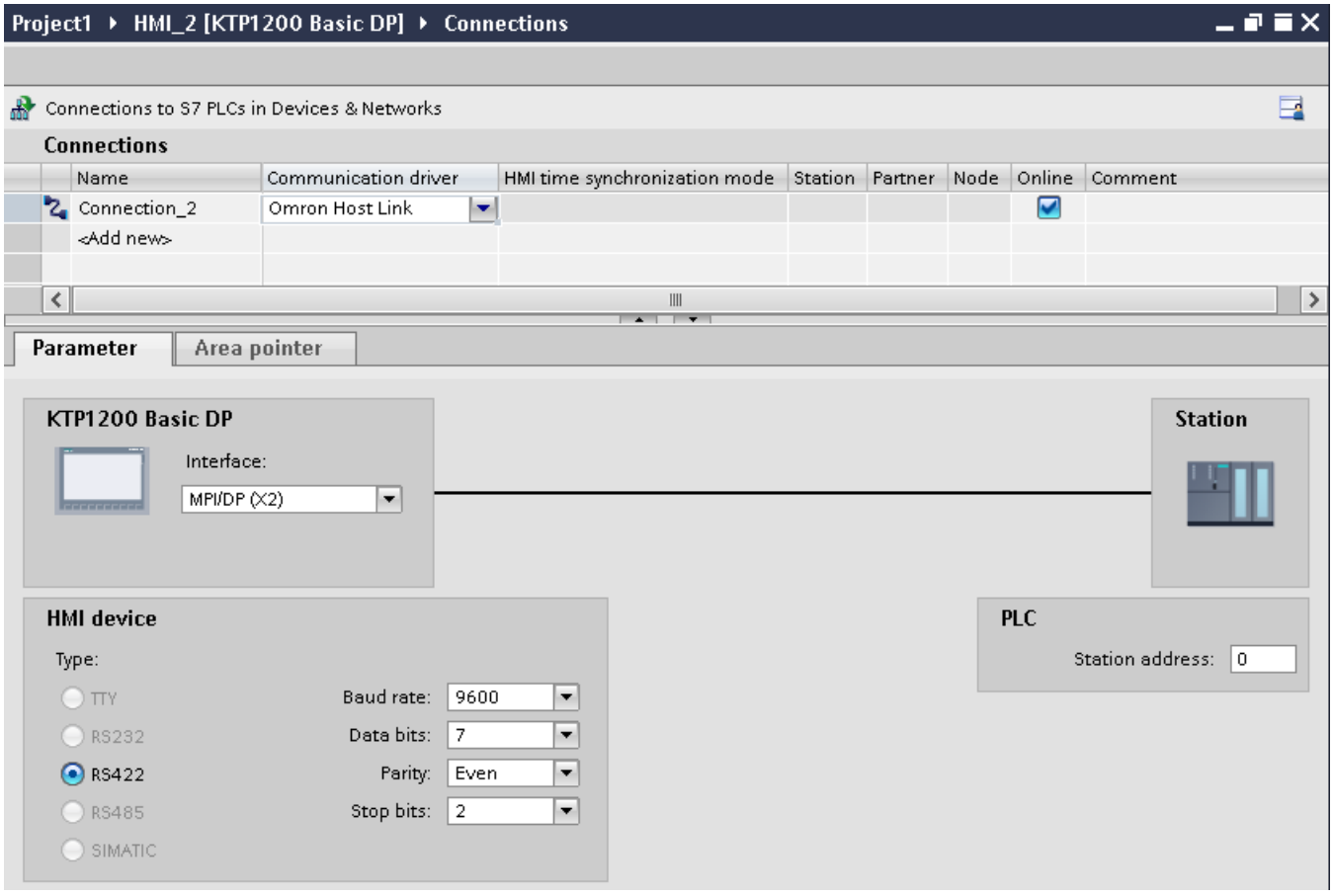
Parameters for the connection (Omron Host Link) (Page 548)

Parameters for the connection (Omron Host Link) (Basic Panels)

Parameters to be set

To set the connection parameters, such as addresses and profiles, click the connection that you have created in the "Connections" editor.

The communication partners are displayed schematically in the Inspector window under "Parameters". The "HMI device" and "PLC" areas are available for assigning parameters according to the interface used.



Parameters for the HMI device

You can select an interface for the HMI device in the Inspector window under "Parameters". Depending on the HMI device, there are several interfaces available.

- Type
  Specifies the physical connection used.
- Baud rate
  For "Baud rate", you set the transmission speed of the HMI device to OMRON. A baud rate of 19200 or 9600 can be selected for the communication.
- Data bits
  For "Data bits", you can choose between "7 bits" and "8 bits".
- Parity
  For "Parity", you can choose from "None", "Even", and "Odd".
- Stop bits
  For "Stop bits", you can choose between 1 and 2 bits.

Parameters for the PLC

- Station address
  For "Station address", set the station number of the connected PLC.

See also

Omron communication drivers (Page 545)

Configuring a connection via Omron Host Link (Page 546)

Connecting HMI device to PLC (Basic Panels)

Connections via Omron Host Link (Basic Panels)

Connection

The connection of the HMI device to an OMRON PLC is limited primarily to the physical connection of the HMI device. Special blocks for the connection are not required in the PLC.

Connection cable

The following connecting cables are available to connect the HMI device to an Omron PLC.

Interface to the HMI device	Omron PLC			
	RS232, 9-pin	RS232 I/O port	RS422, 9-pin	RS422, terminals/ pins
RS232, 9-pin	PP1	Programming cable (standard cable of Omron)	—	—
RS232 via converter	—	—	—	Multi-point cable 1
RS422, 9-pin	—	—	PP2	Multi-point cable 2

Refer to the relevant device manual to determine which HMI device interface is to be used.

See also

Configuring a connection via Omron Host Link (Page 546)

Communication types (Basic Panels)

Approved communication types

The connection from a HMI device to an OMRON-CPU with the Omron Host Link protocol via RS232 and via RS 422 is system-tested and approved by Siemens AG.

This concerns the following CPU types:

- CP1x (CP1L, CP1H, CP1E)
- CJ1x (CJ1M, CJ1H, CJ1G)
- CJ2H
- CS1x (CS1G, CS1H, CS1D)
- CPM2C

Note

Only the following CPU types have been tested and released for Basic Panels, TP 177A and OP 77A:

- CP1x (CP1L, CP1H, CP1E)
- CJ1x (CJ1M, CJ1H, CJ1G)

Multipoint connection

A multipoint connection to the up to 4 approved OMRON PLCs in a RS422-four-wire connection can be implemented with communication modules on the PLCs and is system-tested and approved by Siemens AG.

Note

The HMI device can only be operated as a master. Exactly one master is possible in the RS422-four-wire-Multidrop connection.

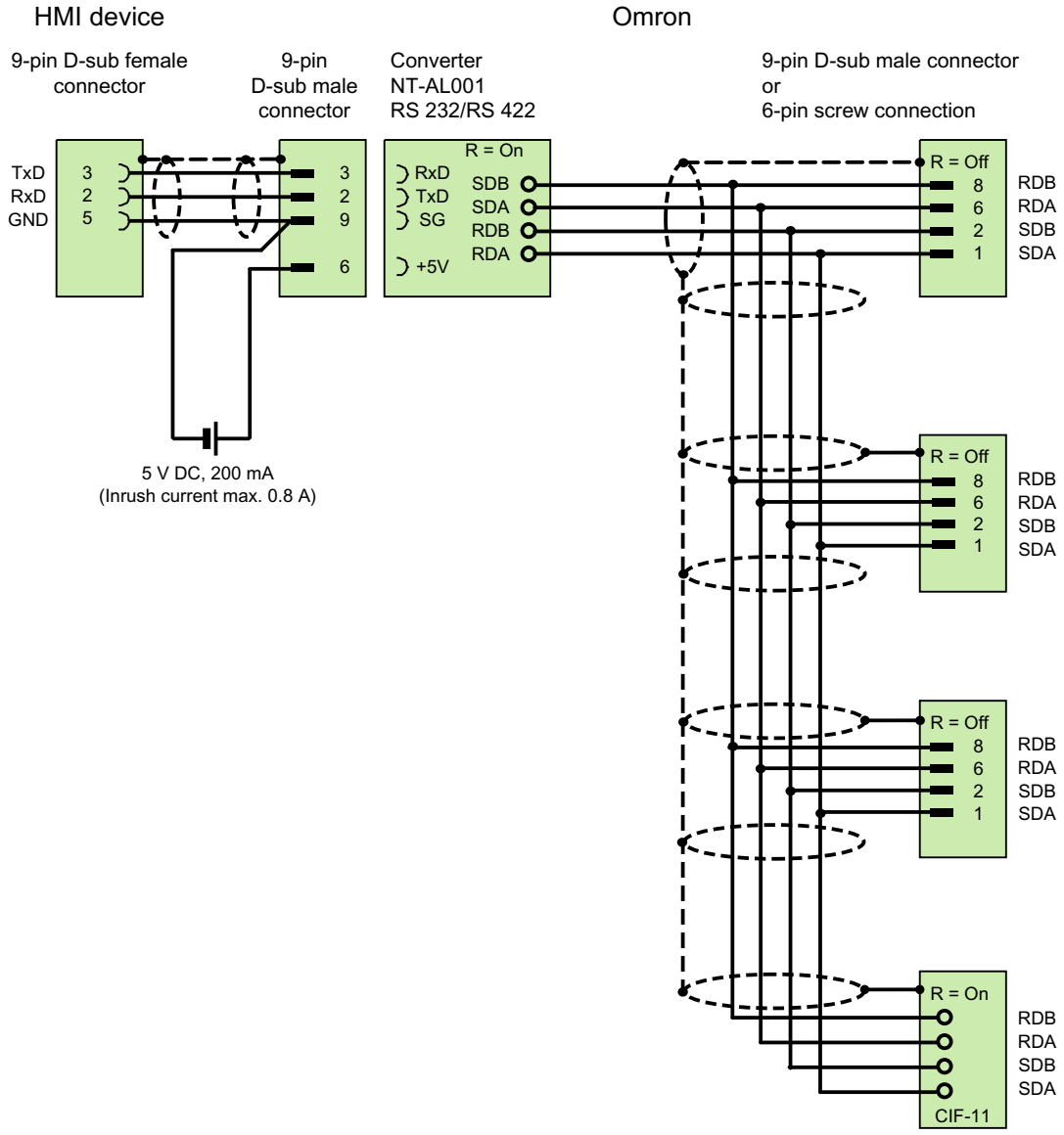
See also

Configuring a connection via Omron Host Link (Page 546)

Connecting cable (Basic Panels)

Connecting cable MP1, RS-232, over converter, for Omron (Basic Panels)

Multipoint cable 1: MP/TP/PC > PLC



[1] Inrush current max. 0.8 A

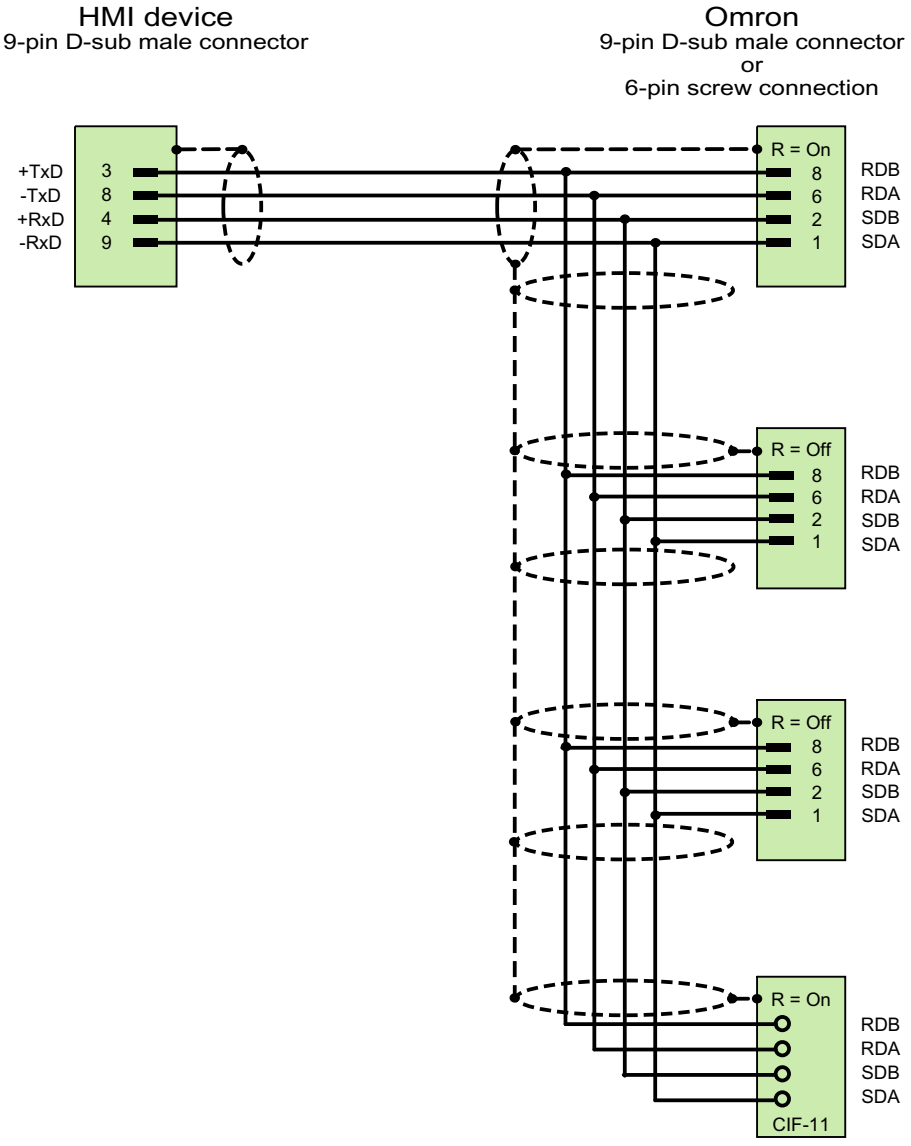
shielded, max. length 500 m

See also

Connections via Omron Host Link (Page 550)

Connecting cable MP2, RS-422, for Omron (Basic Panels)

Multipoint cable 2: RS422, MP/TP/PC > SPS_



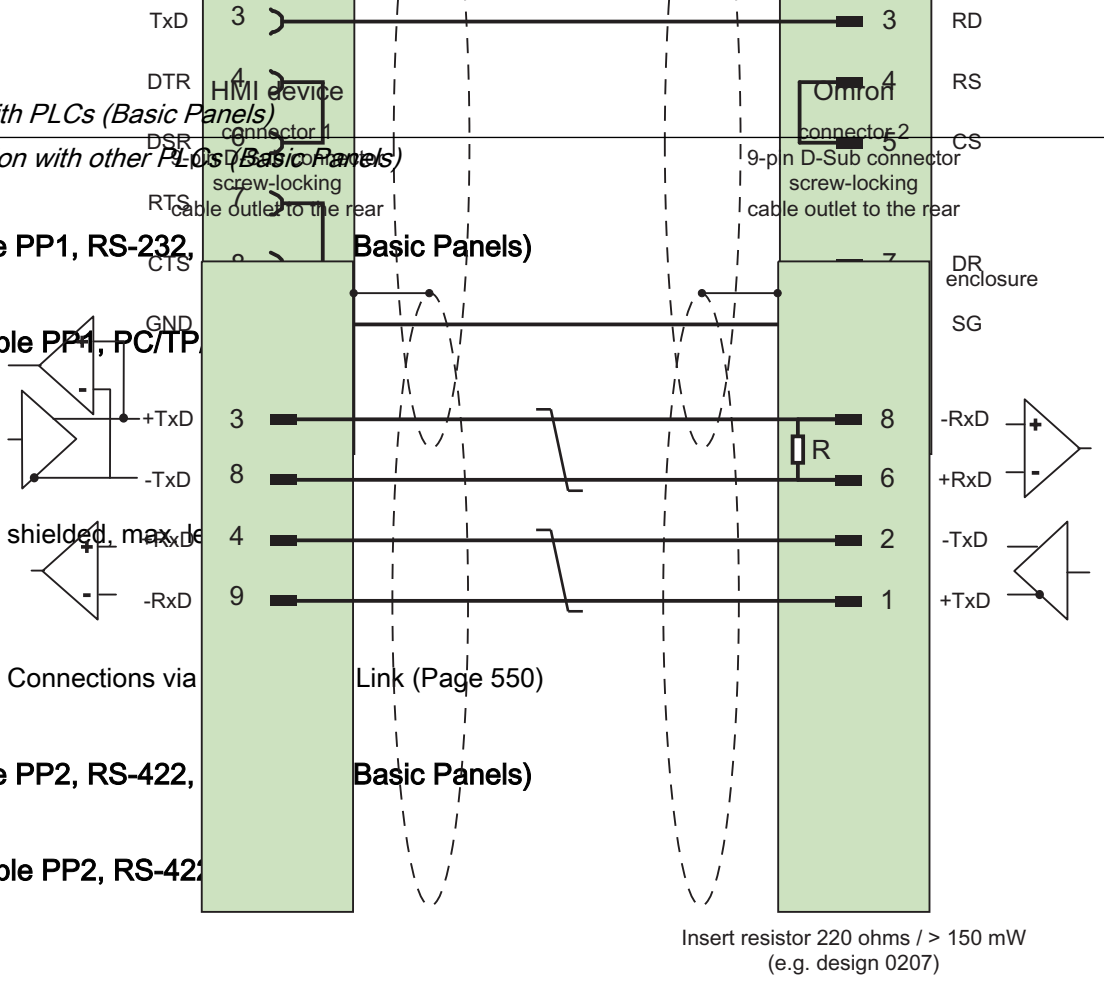
shielded, max. length 500 m

See also

Connections via Omron Host Link (Page 550)

Connecting cable PP1, RS-232, Basic Panels)

Point-to-point cable PP1, PC/TP



shielded, max. le

See also

Connections via Link (Page 550)

Connecting cable PP2, RS-422, Basic Panels)

Point-to-point cable PP2, RS-42

shielded, max. length 500 m

See also

Connections via Omron Host Link (Page 550)

Performance features of communication (Basic Panels)

Permissible data types for Omron Host Link (Basic Panels)

Permitted data types

The table lists the data types that can be used when configuring tags and area pointers.

Data type	Operand type	Length
Bool	I/O, HR, AR, LR, DM, T/C bit, CPU status	1-bit
Byte	CPU type	1 byte
DInt	HR, AR, LR, DM	4 bytes
Int	I/O, HR, AR, LR, DM, T/C Val	2 bytes
Real	HR, DM	4 bytes

Data type	Operand type	Length
String	HR, AR, LR, DM	0 to 80 characters
UDInt	HR, AR, LR, DM	4 bytes
UInt	I/O, HR, AR, LR, DM, T/C Val	2 bytes

Note

Read and write operations of all data areas in the OMRON PLC can only be reliably carried out in "STOP" or "MONITOR" mode.

"I/O" refers either to the IR/SR area or the CIO area depending on the PLC series. The operand types "LR", "HR" and "AR" are not available in all PLC series.

Note

Note the following for write accesses:

For the "Bool" data type with the operand types "I/O", "HR", "AR", "LR" and "DM", the entire word is written back into the PLC when the specified bit is changed. There is no check to determine whether any other bits in the word have changed. As a result, the PLC only has read access to the specified word.

Operand type old PLC	Operand type CS and CJ PLC
CPU Status	CPU Status
I/O	CIO
HR	H Range 0-511
AR	A
LR	n/a 1)
DM	D
T/C	T/C
CPU type	CPU type

1) You do not get an error message when you read or write the LR area in the following PLCs
- CS
- CJ
- CP

See also

Omron communication drivers (Page 545)

Configuring a connection via Omron Host Link (Page 546)

Supported CPU types for Omron Host Link (Basic Panels)

CPU types

The following CPU types are supported in the configuration of the Omron Host Link communication driver.

- CP1
  - CP1L
  - CP1H
  - CP1E
- CJ1
  - CJ1M
  - CJ1H
  - CJ1G
- CJ2
  - CJ2H
- CS1
  - CS1G
  - CS1H
  - CS1D
- CPM
  - CPM2C

See also

Omron communication drivers (Page 545)

Configuring a connection via Omron Host Link (Page 546)

Addressing in Omron Host Link (Basic Panels)

Addressing of PLCs in Omron Host Link

In PLCs of the series CS, CP and CJ, the timers 0-4095 are addressed with T/C 0-2047.

The counters 0-4095 must be addressed with an offset of 2048 (T/C 2048-4095 correspond to the counters 0-2047). Counters and timers with addresses > 2047 cannot be addressed via Host Link.

Counters and timers with addresses > 2047 cannot be addressed via Host Link.

Example:

If you want to address counter C20, you must address T/C 20+2048 = T/C 2068.

See also

Omron communication drivers (Page 545)

Configuring a connection via Omron Host Link (Page 546)

Address areas for Omron Host Link (Basic Panels)

Omron

Address areas	Data types							
	Bool	**Byte**	**UInt**	**Int**	**UDInt**	**DInt**	**Real**	**String**
I/O	I/O 0.0 - I/O 9999.15	--	I/O 0 - I/O 9999	I/O 0 - I/O 9999	--	--	--	--
HR	HR 0.0 - HR 9999.15		HR 0 - HR 9999	HR 0 - HR 9999	HR 0 - HR 9998	HR 0 - HR 9998	HR 0 - HR 9999	HR 0 - HR 9999
AR	AR 0.0 - AR 9999.15		AR 0 - AR 9999	AR 0 - AR 9999	AR 0 - AR 9998	AR 0 - AR 9998		AR 0 - AR 9999
LR	LR 0.0 - LR 9999.15		LR 0 - LR 9999	LR 0 - LR 9999	LR 0 - LR 9998	LR 0 - LR 9998		LR 0 - LR 9999
DM	DM 0.0 - DM 9999.15		DM 0 - DM 9999	DM 0 - DM 9999	DM 0 - DM 9998	DM 0 - DM 9998	DM 0 - DM 9999	DM 0 - DM 9999
T/C Bit	T/C Bit 0 - T/C Bit 4095							
T/C Val			T/C Val 0 - T/C Val 4095	T/C Val 0 - T/C Val 4095				
CPU Status	RUN, MONITOR							
CPU type		CPU type						

See also

Omron communication drivers (Page 545)

Configuring a connection via Omron Host Link (Page 546)

Commissioning components (Basic Panels)

Transferring a project to the HMI device

1. Switch the HMI device to "transfer mode".
2. Set all necessary transfer parameters.
   - Interface
   - Transfer parameters
   - Target storage location
3. Start the project transfer.
   The project is compiled automatically.
   All compilation and transfer steps are logged to a message window.

Interconnecting the PLC with the HMI device

1. Interconnect the PLC with the HMI device using a suitable cable.
2. The message "Connection to PLC .... is established" is output to the HMI device.

Optimizing the configuration (Basic Panels)

Acquisition cycle and update time

The acquisition cycles for the "Area pointers" and of the tags defined in the configuration software are decisive in terms of the update times which can actually be achieved.

The update time is equivalent to the acquisition cycle + transmission time + processing time.

Items to observe when optimizing the update times in configuration data:

- Optimize the maximum and minimum size of the data areas.
- Acquisition cycles which are too short lead to unnecessary load on overall performance. Set the acquisition cycle according to the rate of change of the process values. The rate of temperature changes at a furnace, for example, is significantly slower compared to the speed rate of an electrical drive. A time of approx. 1 second is a benchmark for the acquisition cycle.
- Avoid any gaps when entering the alarm or screen tags in a data area.
- Changes in the PLC can only be detected reliably if these are available at least within the actual acquisition cycle.

Screens

The refresh rate of screens is determined by the type and volume of data to be visualized.

Only configure short acquisition cycles for objects which actually require shorter refresh cycles. This procedure reduces update times.

Trends

The HMI device always updates all bit-triggered trends whose group bit is set in the "Trend transfer area". It resets the bits in the next cycle.

The group bit in the PLC program can only be set again after the HMI device has reset all bits.

Job mailboxes

A high rate and volume of job mailboxes transferred may lead to overload in communication between the HMI device and the PLC.

The HMI device confirms acceptance of the job mailbox by entering the value zero in the first data word of the job mailbox. The HMI device now processes the job for which it requires a certain time slice. It may take the HMI device some time to process a new job mailbox which is transferred in immediate succession to the job mailbox. The next job mailbox is only accepted if sufficient computing resources are available.

Data exchange (Basic Panels)

Area pointers for Omron (Basic Panels)

Area pointers in connections via Omron communication drivers

You use an area pointer to access a data area in the PLC.

For more detailed information on area pointers and their configuration, refer to Section:

"Data exchange using area pointers (Page 649)".

Special features of connections via Omron Host Link

Area pointers can only be created in the following "File types": "DM", "I/O", "HR", "AR", and "LR".

See also

Data exchange using area pointers (Page 649)

Trends (Basic Panels)

Trends

A trend is the graphical representation of one or more values from the PLC. The value is read out time-triggered for Basic Panels.

For additional information see:

AUTOHOTSPOT

Time-triggered trends

The HMI device reads in the trend values cyclically at an interval specified in the configuration.

Time-triggered trends are suitable for continuous curves, such as the operating temperature of a motor.

Alarms (Basic Panels)

Configuring alarms (Basic Panels)

Configuring alarms for non-integrated connections

Several steps are necessary to configure alarms such as warnings, error messages and acknowledgement.

- Step 1: Create tags
- Step 2: Configure alarms
- Step 3: Configure acknowledgment

You can find additional information in the section:

AUTOHOTSPOT

Distinctive features when configuring alarms

If you are configuring connections of HMI devices to PLCs of other manufacturers, note the following distinctive features when configuring:

- Data types of the tags
- Addressing of tags
- How the bit positions are counted

Data types

For connections with an Omron communication driver, the following data types are supported:

PLC	Permitted data types	
	Discrete alarms	Analog alarms
CP1, CJ1, CJ2, CS1, CPM	Uint, int	UInt, Int, UDInt, DInt

How the bit positions are counted

For connections with an Omron communication driver, the following counting method applies:

How the bit positions are counted	Left byte								Right byte							
In Omron PLCs	15							8	7							0
In WinCC you configure:	15							8	7							0

Only tags for the "DM", "I/O", "HR", "AR", and "LR" file types are allowed for use as a trigger tag for discrete alarms.

Configuring discrete alarms

Use arrays for discrete alarms and append each individual alarm to one bit of the array tags themselves and not to the individual subelements.

Only tags for the "DM", "I/O", "HR", "AR", LR" areas and the "Int" and "UInt" file types are permitted for discrete alarms and arrays.

See also

Acknowledgment of alarms (Page 561)

Acknowledgment of alarms (Basic Panels)

Procedure

Create suitable tags on the PLC to acknowledge an error alarm. You assign these tags to an alarm in the "Bit messages" editor. You make the assignment in "Properties > Acknowledgment".

Distinction in terms of acknowledgment:

- Acknowledgment by the PLC
- Acknowledgment on the HMI device

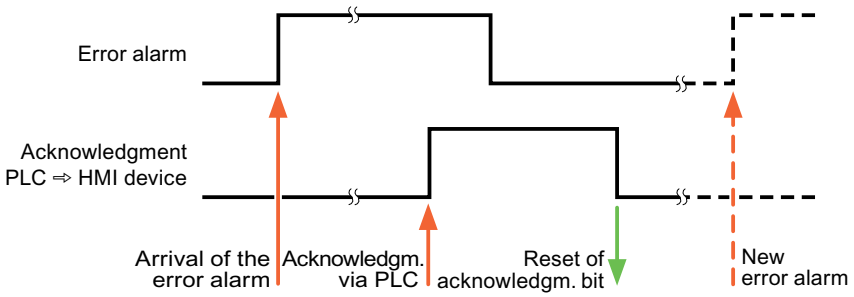
Acknowledgment by the PLC

In "PLC acknowledgment tag", configure the tag or array tag and the bit number that the HMI device uses to identify a PLC acknowledgment.

A bit set in the tag triggers acknowledgment of the assigned error alarm bit at the HMI device. This tag bit returns a function similar to acknowledgment on the HMI device which is triggered by pressing the "ACK" button, for example.

The acknowledgment bit must be located in the same tag as the bit for the error alarm.

Reset the acknowledgment bit before setting the bit in the alarm area again. The figure below shows the pulse diagram.



Acknowledgment on the HMI device

In the "HMI acknowledgment tag" area, configure the tag or array tag as well as the bit number that the HMI device writes to the PLC after acknowledgment. Make sure when you use an array tag that it is not longer than 6 words.

To always create a signal change when setting an assigned acknowledgment bit of a discrete alarm that must be acknowledged, the HMI device will reset the acknowledgment bit assigned to the alarm as soon as it detects an alarm subject to acknowledgment and write the acknowledgment tag in the PLC. There will be a certain delay between detecting the message and writing the acknowledgment tag in the PLC because the HMI device has to process the operations.

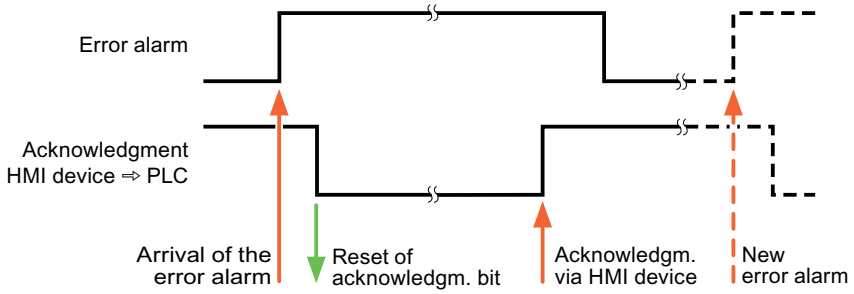
If a discrete alarm subject to acknowledgment is acknowledged by the HMI device, then the corresponding bit in the assigned acknowledgment tag will be set. The entire acknowledgment tag is then written to the PLC by the HMI device. This allows the PLC to recognize that a certain alarm message has been acknowledged at the HMI device.

Note

All alarm bits acknowledged since the last Runtime start will remain in the acknowledgment tag until a new incoming of the respective discrete alarms is detected.

This area should only be read by the PLC because the entire section of the HMI device will be overwritten once the next acknowledgment tag is written.

The figure below shows the pulse diagram.



See also

Configuring alarms (Page 560)

1.13.3.5 Channel support packages (Basic Panels)

CSP_1 (Basic Panels)

Parallel communication (Basic Panels)

Parallel communication of communication drivers

The following table shows an overview of which communication drivers you can use simultaneously on one HMI device.

Note

Parallel communication is not approved for Basic Panels.

Parallel communication over Ethernet interfaces

The approved combinations can be operated via the same Ethernet interface. Several Ethernet interfaces are not required.

Parallel communication only concerns the Ethernet-based communication drivers.

	Allen-Bradley Ether-Net/IP	Mitsubishi MC TCP/IP	Modicon Modbus TCPIP	OPC (DA/ XML DA)	OPC UA (DA)	SIMATIC LOGO !	SIMATIC S7 200	SIMATIC S7 300/400	SIMATIC S7 1200	SIMATIC S7 1500	SIMATIC HTTP protocol	Sinumerik NC
Allen-Bradley Ether-Net/IP	--	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Mitsubishi MC TCP/IP	No	--	No	Yes	Yes	No	No	No	No	No	Yes	No
Modicon Modbus TCPIP	No	No	--	Yes	Yes	No	No	Yes	No	No	Yes	No
OPC (DA/ XML DA)	Yes	Yes	Yes	--	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

	Allen-Bradley Ether-Net/IP	Mitsubishi MC TCP/IP	Modicon Modbus TCPIP	OPC (DA/ XML DA)	OPC UA (DA)	SIMATIC LOGO !	SIMATIC S7 200	SIMATIC S7 300/400	SIMATIC S7 1200	SIMATIC S7 1500	SIMATIC HTTP protocol	Sinumerik NC
OPC UA (DA)	Yes	Yes	Yes	Yes	--	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SIMATIC LOGO!	Yes	No	No	Yes	Yes	--	Yes	Yes	Yes	Yes	Yes	Yes
SIMATIC S7 200	Yes	No	No	Yes	Yes	Yes	--	Yes	Yes	Yes	Yes	Yes
SIMATIC S7 300/400	Yes	No	Yes	Yes	Yes	Yes	Yes	--	Yes	Yes	Yes	Yes
SIMATIC S7 1200	Yes	No	No	Yes	Yes	Yes	Yes	Yes	--	Yes	Yes	Yes
SIMATIC S7 1500	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes	--	Yes	Yes
SIMATIC HTTP protocol	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	--	Yes
Sinumerik NC	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	--

Parallel communication over serial interfaces

The following applies for parallel communication over serial interfaces:

- One communication driver per interface.
- One interface per communication driver.

SIMATIC 500 / 505 Serial and DP (Comfort Panels) (Basic Panels)

Communication with SIMATIC 505 (Basic Panels)

Communication partner (SIMATIC 505) (Basic Panels)

Introduction

This section describes communication between Comfort Panels and SIMATIC 505 PLCs.

With SIMATIC 505, connection takes place via protocols that are specific to the PLCs.

- NITP (Non-Intelligent Terminal Protocol)
  Serial connection with NITP takes place via RS 232 interface (with converter RS 422 to RS 232, MLFB 6AV6 671-8XE00-0AX0), RS-422 and RS-485 interfaces.
  To configure the baud rate for the PLC, use the DIP switch on the front of the PLC.
- PROFIBUS DP
  PROFIBUS DP is master-slave fieldbus with up to 122 slaves. Normally, a PROFIBUS DP network is operated by one master (SIMATIC 505 PLC). This master polls all slaves cyclically. The master could, for example, be a PLC with interface module that is compatible with the DP standard. Each HMI device is a slave and is assigned permanently to a master.
  To enable communication with the PLC, configure the PLC using a suitable control program.

Note

Only the SIMATIC 545 and SIMATIC 555 PLCs support communication via NITP and PROFIBUS DP protocols that are specific to the PLCs.

The following Comfort Panels communicate with SIMATIC 505 PLCs:

- KP/KTP400 Comfort
- KP/TP700 Comfort
- KP/TP900 Comfort
- KP/TP1200 Comfort
- KP/TP1500 Comfort
- TP1900 Comfort
- TP 2200 Comfort

Known restrictions

- The SIMATIC 520, SIMATIC 525, SIMATIC 530, SIMATIC 535, SIMATIC 560, SIMATIC 565 and SIMATIC 575 PLCs do not support communication over NITP and PROFIBUS DP protocols that are specific to the PLC.
- Communication with SIMATIC 505 DP is not supported in the PC simulation.

See also

Communication between HMI device and PLC (SIMATIC 505) (Page 566)

Requirements of communication (Page 571)

Requirements of communication (Page 573)

Connecting cable PP1, RS-232, for SIMATIC 505 (Page 579)

Communication between HMI device and PLC (SIMATIC 505) (Basic Panels)

Communications principle

The HMI device and the PLC communicate using tags and the user data areas.

Tags

The PLC and the HMI device exchange data using process values. In your configuration, create tags that point to an address on the PLC. The HMI device reads and displays the value from the defined address. The operator can also make an entry on the HMI device that is then written to the address on the PLC.

User data areas

User data areas are intended for the exchange of special data and are set up only when such data is used.

Data for which user data areas are required, for example:

- Job mailboxes
- Transfer of data records
- Date/time synchronization
- Sign-of-life monitoring

You create the user data areas while configuring in the TIA Portal. You assign the corresponding addresses in the PLC.

See also

Communication partner (SIMATIC 505) (Page 564)

Configuring the SIMATIC 505 communication driver (Basic Panels)

Permitted data types (SIMATIC 505) (Basic Panels)

Permitted data types

When configuring tags and area pointers, you can use the user data types listed in the following table. It is essential that these data areas are also set up for the CPU with TISOFT.

The following table lists the data types for the serial connection:

Name	Area	Data type
Discrete input	X	BIT
Discrete output	Y	BIT
Control relay	C	BIT

Name	Area	Data type
Variable memory	V	BIT [1)] +/– INT INT +/– DOUBLE DOUBLE REAL ASCII
Word input	WX	
Word output	WY	
Constant memory	K	
Status word memory	STW	
Timer/counter preset	TCP	+/– INT INT
Timer/counter current	TCC	
Analog alarm/alarm acknowledge flags	AACK	+/–INT, INT
Analog alarm deadband	AADB	+/–INT, INT, REAL
Most significant word of analog alarm C flags	ACFH	+/–INT, INT
Least significant word of analog alarm C flags	ACFL	+/–INT, INT
Analog alarm error	AERR	+/–INT, INT, REAL
Analog alarm high alarm limit	AHA	+/–INT, INT, REAL
Analog alarm high-high alarm limit	AHHA	+/–INT, INT, REAL
Analog alarm low alarm limit	ALA	+/–INT, INT, REAL
Analog alarm low-low alarm limit	ALLA	+/–INT, INT, REAL
Analog alarm orange deviation alarm limit	AODA	+/–INT, INT, REAL
Analog alarm process variable	APV	+/–INT, INT, REAL
Analog alarm process variable high limit	APVH	REAL
Analog alarm process variable low limit	APVL	REAL
Analog alarm rate of change alarm limit	ARCA	REAL
Analog alarm setpoint	ASP	+/–INT, INT, REAL
Analog alarm SP high limit	ASPH	+/–INT, INT, REAL
Analog alarm SP low limit	ASPL	+/–INT, INT, REAL
Analog alarm sample rate	ATS	REAL
Analog alarm flags	AVF	+/–INT, INT
Analog alarm yellow deviation alarm limit	AYDA	+/–INT, INT, REAL
Alarm peak elapsed time	APET	+/–INT, INT
Loop alarm/alarm acknowledge flags	LACK	+/–INT, INT
Loop alarm deadband	LADB	+/–INT, INT, REAL
Most significant word of loop C flags	LCFH	+/–INT, INT
Least significant word of loop C flags	LCFL	+/–INT, INT
Loop error	LERR	+/–INT, INT, REAL
Loop alarm high limit	LHA	+/–INT, INT, REAL
Loop alarm high-high limit	LHHA	+/–INT, INT, REAL
Loop gain	LKC	REAL
Loop derivative gain limiting coefficient	LKD	REAL
Loop low alarm limit	LLA	+/–INT, INT, REAL
Loop low-low alarm limit	LLLA	+/–INT, INT, REAL
Loop output	LMN	+/–INT, INT, REAL
Loop bias	LMX	+/–INT, INT, REAL
Loop orange deviation limit	LODA	+/–INT, INT, REAL

Name	Area	Data type
Loop process variable	LPV	+/–INT, INT, REAL
Loop PV high limit	LPVH	REAL
Loop PV low limit	LPVL	REAL
Loop rate of change alarm limit	LRCA	REAL
Loop ramp/soak flags	LRSF	+/–INT, INT
Loop ramp/soak step number	LRSN	+/–INT, INT
Loop setpoint	LSP	+/–INT, INT, REAL
Loop setpoint high point	LSPH	+/–INT, INT, REAL
Loop setpoint low limit	LSPL	+/–INT, INT, REAL
Loop rate	LTD	REAL
Loop reset	LTI	REAL
Loop sample rate	LTS	REAL
Loop V flags	LVF	+/–INT, INT
Loop yellow deviation alarm limit	LYDA	+/–INT, INT, REAL
Loop peak elapsed time	LPET	+/–INT, INT
SF program peak elapsed time	PPET	+/–INT, INT
SF subroutine peak elapsed time	SPET	+/–INT, INT

[1] In the case of write access with NITP protocol, note: With the "BIT" data type in the areas "WX", "WY", "V", "K" and "STW", the entire word is written back to the PLC after the specified bit is changed. There is no check to determine whether any other bits in the word have changed. As a result, the PLC only has read access to the specified word.

The following table lists the data types for connecting via PROFIBUS DP:

Name	Area	Data type
Discrete input	X	BIT
Discrete output	Y	BIT
Control relay	C	BIT
Variable memory	V	BIT [1] +/– INT INT +/– DOUBLE DOUBLE REAL ASCII
Word input	WX	
Word output	WY	
Constant memory	K	
Status word memory	STW	
Timer/counter preset	TCP	+/– INT INT
Timer/counter current	TCC	

Note

The supported data types depend on the function block used for DP communication.

Note

If you configure the address for array data types, use suitable operands. Not all operands that can be selected under "Operand type" are suitable for the selected array data type.

The following table lists operands that are suitable for the array data types:

Array data type	Operand	
	SIMATIC 505 Serial	SIMATIC 505 PROFI-BUS DP
Double+/- Double	WX, WY, V, K and STW	WX, WY, V, K and STW
Real	WX, WY, V, K, STW, LADB, LERR, LHA, LHHA, LKC, LKD, LLA, LLLA, LMN, LMX, LODA, LPVH, LPVL, LPV, LRCA, LSPH, LSPL, LSP, LTD, LTI, LTS, LYDA, AADB, AERR, AHA, AHHA, ALA, ALLA, AODA, APVH, APVL, ARCA, APV, ASPH, ASPL, ASP, ATS and AYDA	WX, WY, V, K and STW
Int & +/- Int	WX, WY, V, K, STW, TCP, TCC, LSPH, LSPL, ASPH, ASPL, PPET, SPET, LACK, LADB, LCFH, LCFL, LERR, LHA, LHHA, LLA, LLLA, LMN, LMX, LODA, LPV, LRSF, LRSN, LSP, ASP, LVF, LYDA, LPET, AACK, AADB, ACFH, ACFL, AERR, AHA, AHHA, ALA, ALLA, AODA, APV, AVF, AYDA and APET	WX, WY, V, K, STW, TCP and TCC

Optimizing the configuration (Basic Panels)

Acquisition cycle and update time

The acquisition cycles for the "Area pointers" and the acquisition cycles of the tags specified in the configuration software are decisive factors for the actual update times that can be achieved.

The update time is the sum of the acquisition cycle + transmission time + processing time.

To achieve optimum update times, remember the following points during configuration:

- Keep the individual data areas as small as possible and as large as necessary.
- Define data areas that belong together as belonging together. You can improve the update time by setting up one large data area instead of several small areas.
- If the acquisition cycles you select are too short, this is detrimental to the overall performance. Set the acquisition cycle to suit the rate of change of the process values. The rate of temperature changes at a furnace, for example, is significantly slower compared to the speed rate of an electrical drive. As a general guideline, the acquisition cycle should be approx. 1 second.
- Put the tags of an alarm or a screen in one data area without gaps.
- To allow changes in the PLC to be recognized reliably, these must be available at least during the actual acquisition cycle.
- Set the transmission rate to the highest possible value.

Discrete alarms

For discrete alarms, use tags and assign the individual alarms to one bit of the tags themselves and not to the individual sub-elements. Discrete alarms are not permitted for arrays. Only the tags of the "Int" and "+/- Int" data types can be used.

Screens

With screens, the update rate that can actually be achieved depends on the type and amount of data to be displayed.

Configure short acquisition cycles only for objects which actually require shorter refresh cycles.

Trends

When using bit-triggered trends, if the group bit is set in the "Trend transfer area", the HMI device always updates all the trends whose bit is set in this area. It then resets the bits.

The group bit in the PLC program can only be set again after all bits have been reset by the HMI device.

Job mailboxes

If large numbers of job mailboxes are sent in quick succession, this can lead to overload in the communication between the HMI device and PLC.

The HMI device confirms acceptance of the job mailbox by entering the value 0 in the first data word of the job mailbox. The HMI device now processes the job for which it requires time. If a new job mailbox is entered again immediately in the job mailbox, it may take some time before the HMI device can process the next job mailbox. The next job mailbox will only be accepted when there is computing capacity available.

Communication driver for the NITP protocol (Basic Panels)

Requirements of communication (Basic Panels)

Connector

To connect the HMI device to the SIMATIC 505, no additional communications module is required. Communication can be handled via one of the standard DP/MPI/PPI ports in the system. At the controller side, you connect the HMI device to the programming interface of the CPU (RS 232 with converter RS 422 to RS 232 (MLFB 6AV6 671-8XE00-0AX0) or RS 422).

The HMI device port to be used is defined in the corresponding manual.

Cable

The following cables are available to connect the HMI device to the PLC:

Interface	SIMATIC 505		
HMI devices	RS-232, 9-pin	RS 422, 9-pin [1]	RS 422, 9-pin [2]
RS 232 with converter RS 422 to RS 232 (MLFB 6AV6 671-8XE00-0AX0), 9-pin	PP1	Commercially available V.24/ RS 422 converter	Commercially available V.24/ RS 422 converter
RS-422, 9-pin	–	6XV1440–2M_ _ _	6XV1440–1M_ _ _

See also

Communication partner (SIMATIC 505) (Page 564)

Installing the communication driver (Page 571)

Configuring the controller type and protocol (Page 572)

Configuring protocol parameters (Page 572)

Installing the communication driver (Basic Panels)

Driver for the HMI device

The driver for the connection to SIMATIC 505 is supplied with the TIA Portal and is installed automatically.

No special blocks are required on the PLC for the connection.

See also

Requirements of communication (Page 571)

Configuring the controller type and protocol (Basic Panels)

Selecting the PLC

1. For a connection with a SIMATIC 505 controller over NITP protocol, open the "Connections" editor.
2. Go to the "Communication drivers" column and select the "SIMATIC 505 serial" protocol. The property view displays the parameters of the selected protocol.
3. To edit parameters, select the connection and edit its parameters in the properties dialog box.

Note

The settings on the HMI device and on the PLC must match.

See also

Requirements of communication (Page 571)

Configuring protocol parameters (Basic Panels)

Parameters to be set

To set the parameters, open the "Connections" editor. "SIMATIC 505 seriell" is selected in the "Communication drivers" column in the work area. Change the protocol parameters of the connection in the Properties window.

Device-dependent parameters

- Interface
  Under "Interface", you select the interface on the HMI device over which the connection to the SIMATIC 505 will be established.
  For further information, refer to the relevant device manuals.
- Type
  The type depends on the selected interface.
- Data bits
  Under "Data bit" set "7".
- Parity
  Under "Parity" set "Odd".

- Stop bits
  Under "Stop bits" set "1".
- Baud rate
  Under "Baud rate" define the transmission rate between the HMI device and the SIMATIC 505. Communication is possible with 2400 bps, 4800 bps, 9600 bps, 19200 bps (default value), 38400 bps, 57600 bps, 93750 bps and 115200 bps.

Note

The baud rates 300 bps, 600 bps and 1200 bps are not supported.

Note

Make sure that the transmission parameters on the HMI device and the SIMATIC 505 match.

See also

Requirements of communication (Page 571)

Communication driver for PROFIBUS DP (Basic Panels)

Requirements of communication (Basic Panels)

Hardware requirements

To include the HMI devices in an existing PROFIBUS DP network, you require the following hardware components:

- On the PLC: Installed DP interface
- For each device (HMI device or PLC): PROFIBUS DP bus connector or other approved component (except for FSK bus terminal, see configurator in the SIMATIC HMI catalog ST80.1).

Software requirements

For the PROFIBUS DP connection, you require the following software components:

- TIA Portal engineering tool
- Specific configuration software (e.g. COM PROFIBUS) for assigning parameter values complying with the DP standard to the interface module in the PLC

See also

Communication partner (SIMATIC 505) (Page 564)

Installing the communication driver (Page 574)

Configuring the controller type and protocol (Page 574)

Configuring protocol parameters (Page 575)

Assigning parameter values for the PROFIBUS DP network based on the example of COM PROFIBUS (Page 576)

Installing the communication driver (Basic Panels)

Driver for the HMI device

The driver for the connection to SIMATIC 505 is supplied with CSP TIA Portal and is installed automatically.

To connect the HMI device and the PLC, a physical connection and a control program are necessary.

See also

Requirements of communication (Page 573)

Configuring the controller type and protocol (Basic Panels)

Selecting the PLC

1. For a connection with a SIMATIC 505 PLC over PROFIBUS DP, open the "Connections" editor.
2. In the work area, go to the "Communication drivers" column and select the "SIMATIC 505 DP" protocol.
   The property view displays the parameters of the selected protocol.
3. To edit parameters, select the connection and edit its parameters in the properties dialog box.

Note

The settings on the HMI device and on the PLC must match.

See also

Requirements of communication (Page 573)

Configuring protocol parameters (Basic Panels)

Parameters to be set

To set the parameters, open the "Connections" editor. "SIMATIC 505 DP" is selected in the "Communication drivers" column in the work area. Change the protocol parameters of the connection in the Properties window.

Device-dependent parameters

- Interface
  Under "Interface", you select the interface on the HMI device over which the connection to the SIMATIC 505 will be established. The type is set automatically depending on the selected interface set under "Interface".
  For further information, refer to the relevant device manuals.
- Baud rate
  Under "Baud rate" you select the transmission rate at which the HMI device communicates with the SIMATIC 505 in the network configuration. Set the same baud rate for all devices in the network.
  Communication is possible with 93,750 bps, 187,500 bps (default value), 500,000 bps, 1,500,000 bps, 3,000,000 bps, 6,000,000 bps and 12,000,000 bps.

PLC-dependent parameters

- HMI address
  Under "HMI address" set the PROFIBUS DP address of the HMI device.
  Values from 3 through 125 are permitted.
- Block length
  Under "Block length" you specify the assignment of the I/O area used for communication between the HMI device and PLC. The size of the I/O area influences performance.
  The configuration is implemented according to class B (basic DP slave complying with EN 50170). You can select four different configurations:
  - Class B tiny
  - Class B small
  - Class B middle
  - Class B big

  The assignment of the I/O areas is fixed with the four various settings. The following table shows the assignment of the I/O areas:

Class	Inputs (bytes)	Outputs (bytes)
Class B tiny	32	22
Class B small	42	22
Class B middle	64	32
Class B big	128	64

The settings in WinCC must match the configuration data of the interface module, for example the DP interface on the PLC.

Note

To transfer large amounts of data, it is advisable to set a large I/O area. This means that the displays on the HMI device are updated more quickly since the data is fetched in one cycle.

See also

Requirements of communication (Page 573)

Assigning parameter values for the PROFIBUS DP network based on the example of COM PROFIBUS (Basic Panels)

DP interface on the PLC

To configure the installed DP interface on the PLC, you require the COM PROFIBUS configuration package. CSP TIA Portal is supplied with GSD files for HMI device slaves.

These GSD files are located in the directory "Siemens\Automation\Portal V13\ Data\Hmi\ Support Files\GSD". Different GSD files must be used depending on the particular HMI device:

GSD file	HMI device
Siem8191.gsd	KP/KTP400 Comfort
Siem8192.gsd	KP/TP700 Comfort, KP/TP900 Comfort, KP/TP1200 Comfort
Siem81a5.gsd	KP/TP1500 Comfort, TP1900 Comfort, TP 2200 Comfort

In the following cases, copy the files from the specified TIA Portal directory to COM PROFIBUS:

- The COM PROFIBUS does not yet support a new HMI device.

Then restart the COM PROFIBUS and select the "Read GSD files".

If you have already created a COM PROFIBUS configuration with an older file and want to use the newer GSD files, you will have to recreate the configuration.

Parameter

Before the PLC and the HMI device can communicate with each other, the following parameters must be set in COM PROFIBUS:

- Station type: HMI
- Station number: 3-125
  The value entered here must match the HMI device address specified when the HMI device is configured.

- Desired configuration:
  The desired configuration is set by selecting the class and the symbolic name of the configuration. The following desired configurations can be set:
  - Class B tiny
  - Class B small
  - Class B middle
  - Class B big
- Address ID:
  The address ID is assigned automatically according to the selected configuration and must not be changed.
- I and Q address:
  The address must match the address stored in the PLC program.

Integrating the COM PROFIBUS configuration into TISOFT

The TISOFT documentation describes in detail how to integrate the COM PROFIBUS configuration into the TISOFT program. The main steps are simply outlined below:

1. Create a binary file in COM PROFIBUS using the "Export" function.
2. Merge the binary file into your TISOFT program with "CONFIO" > "PRO-DP" > "MERGE".
3. Set the connection to the CPU to "Online PLC" mode.
4. Specify the I/O addresses in TISOFT with "CONFIO" > "PRO-DP" > "CONFIG".
5. Transfer the program to the CPU with "UPDATE".

See also

Requirements of communication (Page 573)

User data areas (Basic Panels)

Trend request and trend transfer (Basic Panels)

Function

A trend is the graphic visualization of one or more values from the PLC. The value is read either time- or bit-triggered, depending on the configuration.

Time-triggered trends

The HMI device reads in the trend values cyclically at an interval specified in the configuration. Time-triggered trends are suitable for continuous processes, for example, the operating temperature of a motor.

Bit-triggered trends

By setting a trigger bit in the trend transfer tag, the HMI device either reads in a trend value or an entire trend buffer. This setting is defined in configuration data. Bit-triggered trends are

normally used to visualize rapidly changing values. One example might be the injection pressure in the production of plastic parts.

To trigger bit-triggered trends, you must create suitable external tags in the TIA Portal "Tags" editor during the configuration. The external tags must be linked with the trend areas. The HMI device and PLC then communicate with each other over these trend areas.

The following areas are available for trends:

- Trend request area
- Trend transfer area 1
- Trend transfer area 2 (required only with switch buffers)

Tags of the "V" or "K" types are permitted. They must be of the data type "Int", "+/- Int", or an array tag of the data type "Int", "+/-Int". During configuration you assign a bit to a trend. This sets a unique bit assignment for all areas.

Trend request area

If a screen is opened on the HMI device with one or more trends, the HMI device sets the corresponding bits in the trend request area. After deselecting the screen, the HMI device resets the relevant bits in the trend request area.

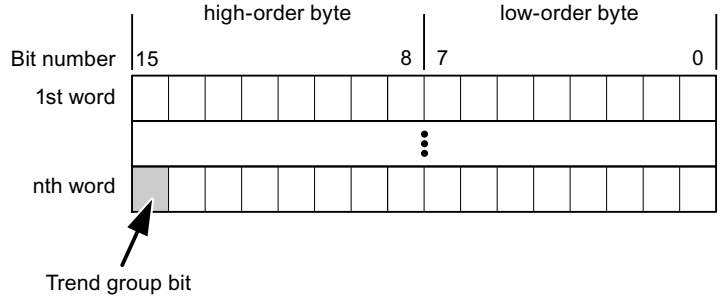
Using the trend request area, the PLC can recognize which trend is currently displayed on the HMI device. Trends can also be triggered without evaluation of the trend request area.

Trend transfer area 1

This area is used to trigger trends. In your PLC program, you set the bit assigned to the trend in the trend transfer area and set the trend group bit. The trend group bit is the last bit in the trend transfer area.

The HMI device detects the trigger. The HMI device reads either a value or the entire buffer from the PLC. It then resets the trend bit and the trend group bit.

The following picture shows the structure of a trend transfer area.



The trend transfer area must not be modified by the PLC program until the trend group bit has been reset.

Trend transfer area 2

Trend transfer area 2 is required for trends configured with a switch buffer. The trend transfer areas 1 and 2 have a similar structure.

Switch buffer

The switch buffer is a second buffer for the same trend that can be set up during configuration.

While the HMI device reads the values from buffer 1, the controller writes to buffer 2. When the HMI device reads buffer 2, the controller writes to buffer 1. This prevents the trend values being overwritten by the PLC while the trend is being read out by the HMI device.

LED mapping (Basic Panels)

Function

The keyboard units of Comfort Panels (CP), Operator Panel (OP), Multi Panel (MP) and Panel PC have LEDs in the function keys. These LEDs can be activated from the PLC. Thus, it is possible, for example, to signal to the user which key should be pressed in a given situation by lighting up an LED.

Requirements

In order to activate an LED, an LED tag or an array tag must be set up in the PLC and specified as an LED tag during configuration.

LED assignment

The assignment of the individual LEDs to the bits in the LED tags is specified when the function keys are configured. In this process, the "LED tag" and the assigned "Bit" are specified for each function key in the "General" group of the properties window.

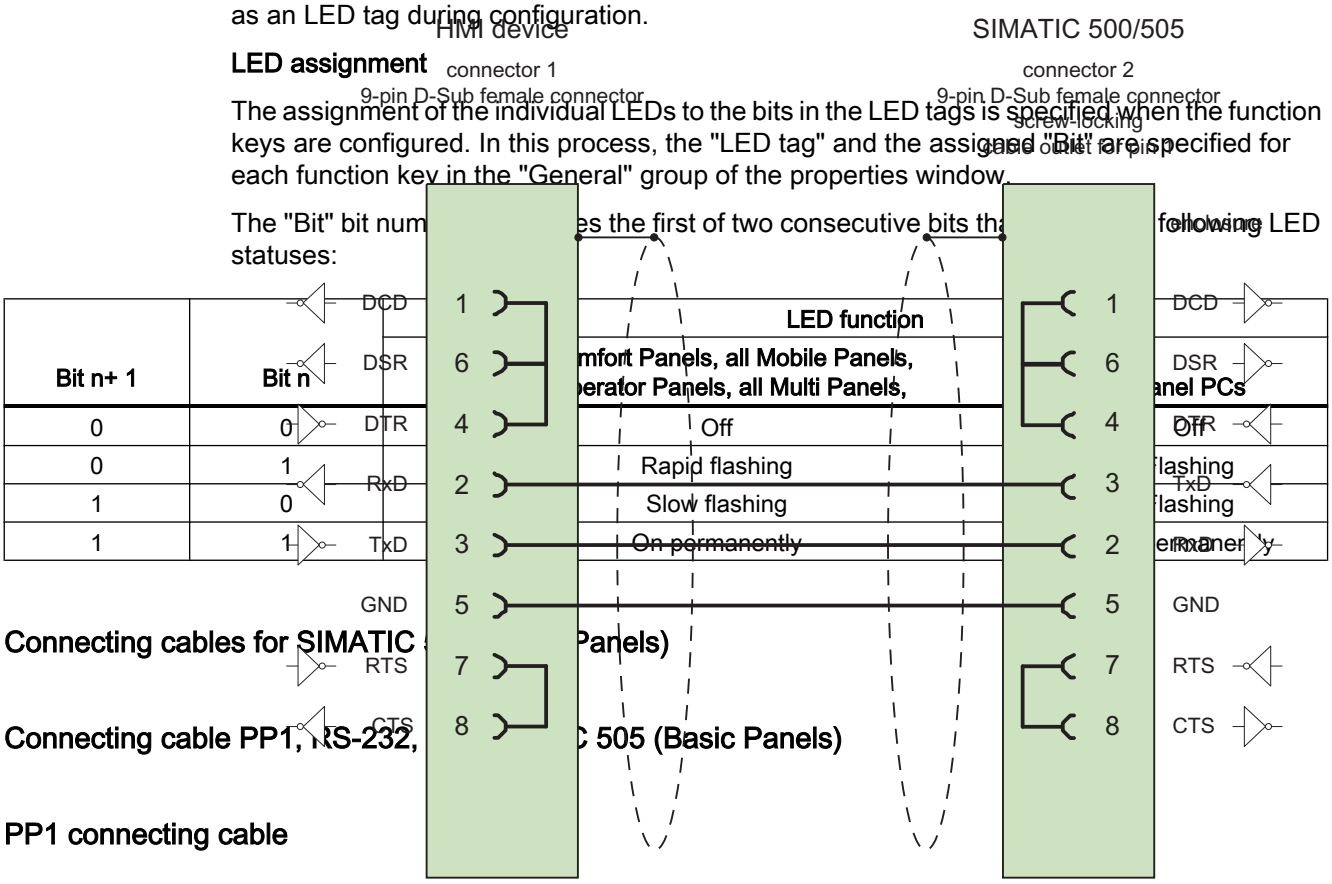
The "Bit" bit num es the first of two consecutive bits tha following LED statuses:

Bit n+ 1	Bit n	LED function
		mfort Panels, all Mobile Panels, erator Panels, all Multi Panels, anel PCs
0	0	Off
0	1	Rapid flashing
1	0	Slow flashing
1	1	On permanently

Connecting cables for SIMATIC Panels)

Connecting cable PP1, RS-232, C 505 (Basic Panels)

PP1 connecting cable



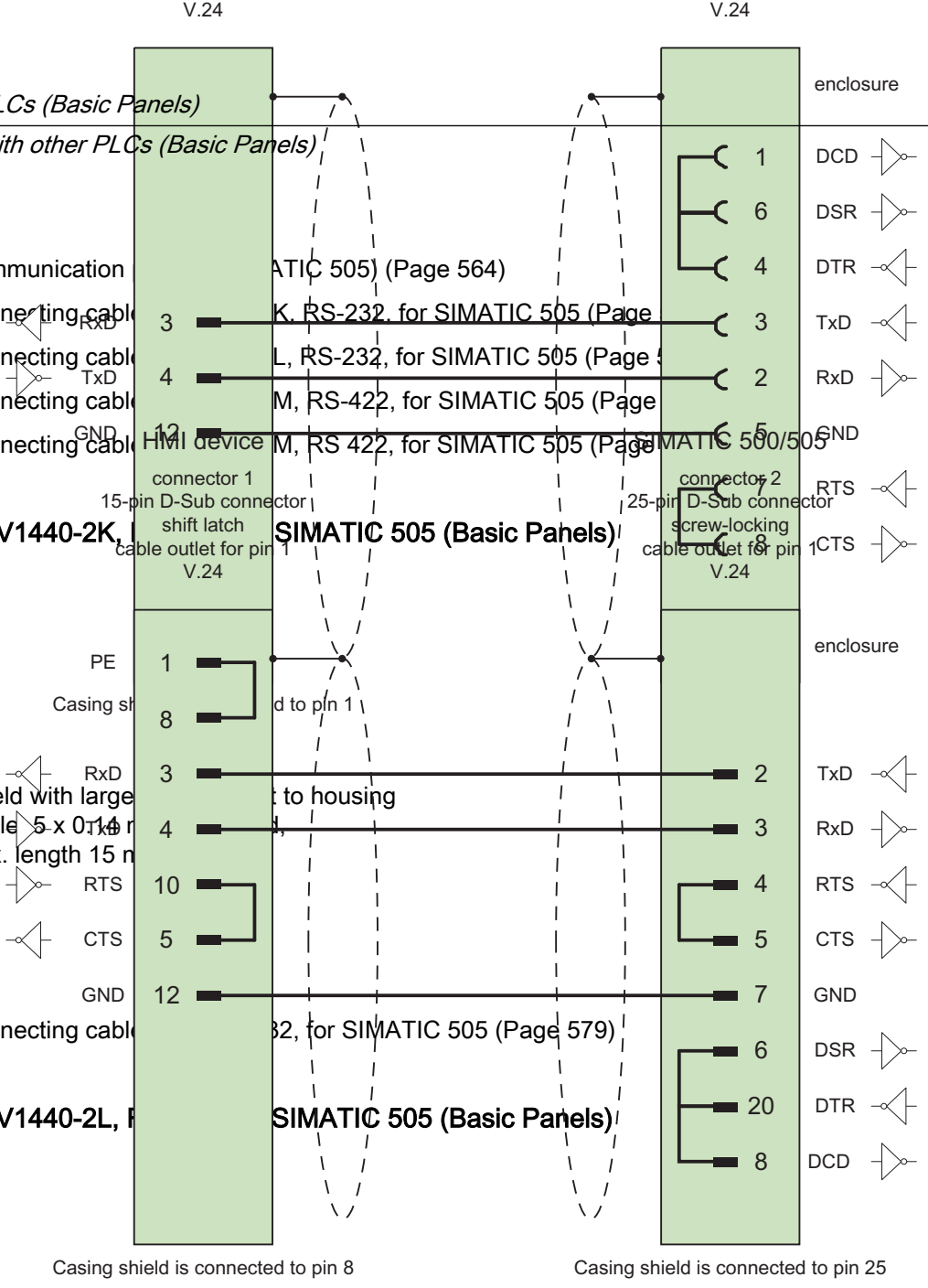
Shield with large-area contact to housing
Cable: 3 x 0.14 mm², shielded,
max. length 15 m

See also

Communication ATIC 505) (Page 564)
Connecting cable K, RS-232, for SIMATIC 505 (Page
Connecting cable L, RS-232, for SIMATIC 505 (Page 5
Connecting cable M, RS-422, for SIMATIC 505 (Page
Connecting cable HMI device M, RS 422, for SIMATIC 505 (Page SIMATIC 500/505

Connecting cable 6XV1440-2K, SIMATIC 505 (Basic Panels)

6XV1440-2K_ _ _



Shield with large-area contact to housing
Cable: 5 x 0.14 mm², shielded,
max. length 15 m

See also

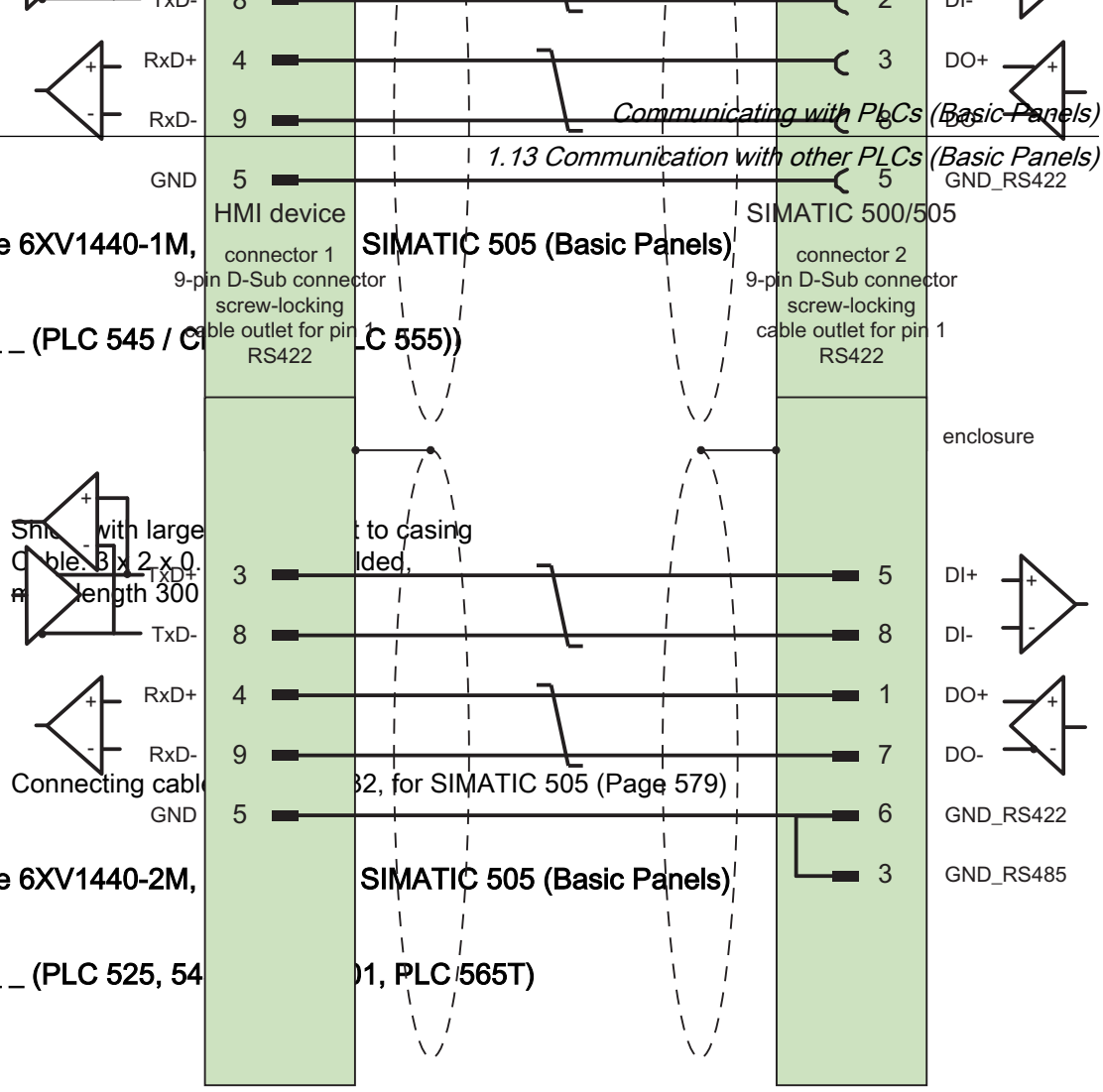
Connecting cable PP1, RS-232, for SIMATIC 505 (Page 579)

Connecting cable 6XV1440-2L, SIMATIC 505 (Basic Panels)

6XV1440-2L _ _ _

See also

Connecting cable PP1, RS-232, for SIMATIC 505 (Page 579)



Connecting cable 6XV1440-1M, SIMATIC 505 (Basic Panels)

6XV1440-1M _ _ _ (PLC 545 / C LC 555))

Shield with large-area contact to casing
Cable: 3 x 2 x 0.14 mm², shielded,
max. length 300 m

See also

Connecting cable PP1, RS-232, for SIMATIC 505 (Page 579)

Connecting cable 6XV1440-2M, SIMATIC 505 (Basic Panels)

6XV1440-2M _ _ _ (PLC 525, 54 01, PLC 565T)

Shield with large-area contact to casing
Cable: 3 x 2 x 0.14 mm², shielded,
max. length 300 m

See also

Connecting cable PP1, RS-232, for SIMATIC 505 (Page 579)

Allen-Bradley DH485 (Comfort Panels) (Basic Panels)

Communication using DH485 protocol (Comfort Panels) (Basic Panels)

Communication partners (Basic Panels)

Introduction

This section describes communication between Comfort Panels and Allen-Bradley PLCs. The DH485 protocol is used for communication purposes.

The DH485 protocol is installed together with the CSP Allen-Bradley DH485 (Comfort Panels) in the TIA Portal (V13 SP1 Update 4).

Note

No migration of WinCC flexible projects is planned for CSP Allen-Bradley DH485 (Comfort Panels).

Note

Runtime and the simulation for Comfort Panels are not supported in connection with the CSP Allen-Bradley DH485.

Available communication partners

The following Allen-Bradley PLCs support communication via the DH485 protocol:

- SLC500
- SLC5/01 (you must not connect more than one HMI device to this controller)
- SLC5/02
- SLC5/03
- SLC5/04
- SLC5/05
- MicroLogix

The following Comfort Panels communicate with Allen-Bradley PLCs via DH485 protocol:

- KTP400
- KP400
- TP700
- KP700
- TP900
- KP900
- TP1200
- KP1200
- TP1500
- KP1500
- TP1900
- TP2200

Note

Communication via DH485 protocol is supported for Comfort Panels with device version 13.00.01.04.

See also

Requirements of communication (Page 583)

Connecting cables (Page 592)

Requirements of communication (Basic Panels)

Connector

The connection between the HMI device and the Allen-Bradley PLC is limited mainly to the setting of interface parameters and the bus address. No special blocks are required on the PLC for the connection.

Note

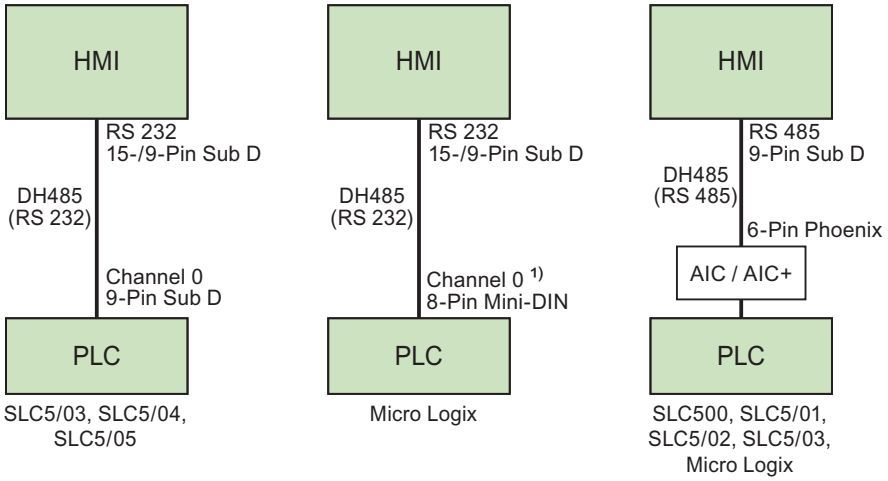
Allen-Bradley offers a variety of communication adapters for integrating devices for the DH485, DH, and DH+ networks. Since no Allen-Bradley communications adapters have been system tested together with the Siemens communications driver for DH485 protocol, none have therefore been released.

Point-to-point connection with DH485 protocol

Point-to-point and multiple-point connections can be established with the DH485 protocol.

A HMI device connected via the RS-232 interface can be connected directly to the PLC.

For a DH485 connection via the RS-485 interface, you have to use an Isolated Link Coupler (AIC) or an Advanced Interface Converter (AIC+) for electrical isolation. Please refer to the Allen-Bradley documentation for details on this.



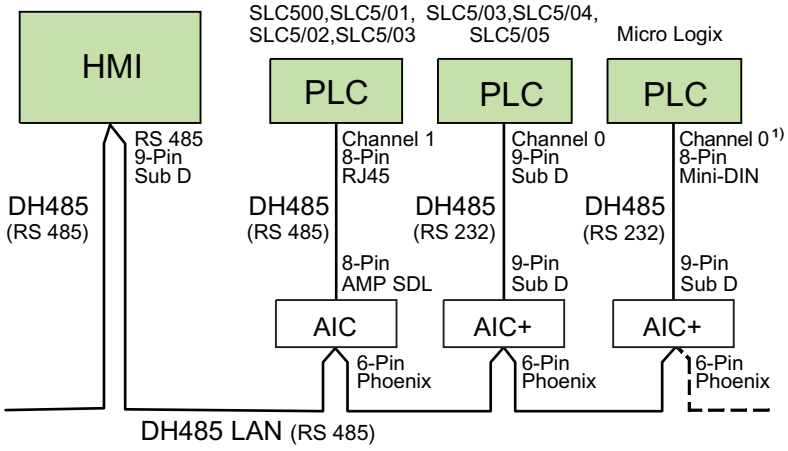
1) For Micro Logix ML1500 LRP, Channel 1 (9-pin D-sub) is also possible.

Connecting cable

HMI panel interface used	Connection to SLC5/03, SLC5/04, SLC5/05	Micro Logix	SLC500, SLC5/01, SLC5/02, SLC5/03, Micro Logix
For RS-232 15-pin	6XV1440-2K _ _ _	PP1 (RS-232 – Micro Logix)	—
For RS-232 9-pin	Allen-Bradley cable 1747-CP3	Allen-Bradley cable 1761-CBL-PM02	—
For RS-485 9-pin	—	—	PP4 (RS-485 - AIC)

The cable pin assignments can be found in the section "Connecting cables".

Multi-point connection with DH485 protocol



1) For Micro Logix ML1500 LRP, Channel 1 (9-pin D-sub) is also possible.

Note

The multi-point connection allows a connection with up to four Comfort Panels and up to four PLCs per HMI device.

Connecting cable

HMI panel interface used	For connection to AIC+ interface module	For connection to LAN (RS485)
For RS-232 9-pin	to 9-pin D-sub: • 1761-CP3 • RS-232 - AIC+ to 8-pin D-sub: • 1761-CBL-PM02 • RS-232 - AIC+	—
For RS-485 9-pin	—	RS-485 - DH485 LAN

Refer to the relevant product manual to determine which HMI device interface is to be used.

See the Allen-Bradley documentation for details about the cable connection.

The cable pin assignments can be found in the section "Connecting cables".

See also

Communication partners (Page 581)

Configuring the controller type and protocol (Page 586)

Configuring protocol parameters (Page 587)

Permitted data types (Allen-Bradley DH485) (Page 588)

Optimizing the configuration (Page 589)

Installation (Basic Panels)

Installing CSP Allen-Bradley DH485 (Comfort Panels)

The following installations are required to install CSP Allen-Bradley DH485 (Comfort Panels) :

- WinCC V13 SP1 Update 4
- The "CSP_V1301_001_001_WinCCTIAPortal_ALLENBRADLEY_DH485.isp13" file must be available on the configuration PC.

1. Open the TIA Portal with administrator rights.
2. Select "Tools > Support Packages" from the menu.
   The "Detailed information > Installation of support packages" dialog box opens.
3. Click the "Add from file system" button.
4. Select the directory of the "CSP_V1301_001_001_WinCCTIAPortal_ALLENBRADLEY_DH485.isp13" file in the file system.
5. Click "Open."
   The file is added in the TIA Portal.
6. Activate "CSP_V1301_001_001_WinCCTIAPortal_ALLENBRADLEY_DH485.isp13" in the "Detail information" dialog.
7. Click "Install".
   A dialog opens.
8. Close the TIA Portal.
9. Click the "Next" button.
   Installation is started.

Note

CSP Allen-Bradley DH485 can be uninstalled.

The uninstalling is handled automatically by a setup program. Once started, the setup program guides you step-by-step through the entire removal procedure.

Installing driver for HMI devices

The communication driver for Allen-Bradley PLCs, for which WinCC supports a connection with DH485 protocol, is supplied with the CSP Allen-Bradley DH485 TIA Portal.

The Comfort Panels are installed automatically.

Configuring the controller type and protocol (Basic Panels)

Selecting the PLC

1. For a connection with an Allen-Bradley PLC via DH485 protocol, double-click "Connections" in the Project window of the HMI device.
2. In the work area, select the Allen-Bradley DH485 protocol in the "Communication driver" column.
   The Properties window displays the parameters of the selected protocol.

For subsequent changes of the parameters, double-click "Connections" in the Project window of the HMI device. Select the connection and edit its parameters in the Properties window.

Note

The settings on the HMI device and on the PLC must match.

See also

Requirements of communication (Page 583)

Configuring protocol parameters (Basic Panels)

Parameters to be set

To set the parameters, double-click "Connections" in the Project window of the HMI device. "Allen-Bradley DH485" is selected in the "Communication drivers" column. You can now enter or modify the protocol parameters in the Properties window:

Device-dependent parameters

- Interface
  Select the HMI port to which the Allen-Bradley PLC is connected at the "Interface" entry. For more detailed information, refer to the HMI device manual.
- Type
  You can select IF1B and RS 485, depending on the HMI device and the selected interface.
- Baud rate
  Define the transmission rate between the HMI device and the PLC under "Baud rate".
- Data bits
  "8-bit" is set under "Data bits".
- Parity
  "Even" is set under "Parity".
- Stop bits
  "1-bit" is set under "Stop bits".

Network parameters

- HMI address
  Set the HMI address at "HMI Address". You can select any address from 1 to 31.
- Max. bus address
  Set the highest bus address used at the "Max. Bus Address" parameter. The bus address is evaluated for token passing.
  You can set addresses 2 through 31.

PLC-dependent parameters

- Destination address
  Define the PLC address at the "Destination address" parameter.
- CPU type
  Set the PLC type to which the HMI device is connected at the "CPU type" parameter.
  For the PLCs SLC 5/03, SLC 5/04 or SLC 5/05, select "SLC50x".

See also

Requirements of communication (Page 583)

Permitted data types (Allen-Bradley DH485) (Basic Panels)

Permitted data types

The table lists the user data types that can be used when configuring tags and area pointers.

Designation	File type	Data type
ASCII [1]	A	ASCII
Binary	B	BOOL, UNSIGNED INT
Counter	C	BOOL, SIGNED INT, UNSIGNED INT
Float [1]	F	REAL
Digital Input	I	BOOL, UNSIGNED INT
Data Register (Integer)	N	BOOL, SIGNED INT, UNSIGNED INT, SIGNED LONG, UNSIGNED LONG, REAL
Digital Output	O	BOOL, UNSIGNED INT
Control	R	BOOL, UNSIGNED INT
Status	S	BOOL, UNSIGNED INT
Timer	T	BOOL, SIGNED INT, UNSIGNED INT

[1] Applies to SLC 5/03, SLC 5/04 and SLC 5/05.

Display in WinCC

The data formats in WinCC are abbreviated as follows.

- UNSIGNED INT = UInt
- UNSIGNED LONG = UDInt

- SIGNED INT = Int
- SIGNED LONG = DInt

Distinctive features for connections with Allen-Bradley DH485

Area pointers can only be created in "File Type" "N", "O", "I", "S", and "B".

Only tags whose "File type" is "N", "O", "I", "S" and "B" are allowed for use as a "trigger tag" for discrete alarms. These tags are only valid for the data types "Int" and "UInt".

Array tags may only be used for discrete alarms and trends. Therefore, you can only create array tags using tags whose "File Type" is "N", "O", "I", "S", or "B" and whose data type is "Int" or "UInt".

Note

I/O modules with 8 or 16 ports occupy one data word on the PLC. I/O modules with 24 or 32 ports occupy two data words. The HMI device does not output an error message if using non-existent bits.

Therefore, you should always make sure that I/O modules with 8 or 24 ports only occupy the bits that are actually assigned to a port.

Note

If you configure the address for array data types, use suitable data types. Not all data types that can be selected under "Data type" are suitable for the selected array data type.

The following table lists data types that are suitable for the array data types:

Array data type	Data type
Int	N, S
UInt	N, B, I, O
Dint	N
UDint	N
Real	N

See also

Requirements of communication (Page 583)

Optimizing the configuration (Basic Panels)

Acquisition cycle and update time

The acquisition cycles for the "Area pointers" and the acquisition cycles of the tags specified in the configuration software are decisive factors for the actual update times that can be achieved.

The update time is the sum of the acquisition cycle + transmission time + processing time.

To achieve optimum update times, remember the following points during configuration:

- Keep the individual data areas as small as possible and as large as necessary.
- Define data areas that belong together as belonging together. You can improve the update time by setting up one large data area instead of several small areas.
- If the acquisition cycles you select are too short, this is detrimental to the overall performance. Set the acquisition cycle to suit the rate of change of the process values. The rate of temperature changes of a furnace, for example, is significantly slower compared to the speed rate of an electrical drive. As a general guideline, the acquisition cycle should be approx. 1 second.
- Put the tags of an alarm or a screen in one data area without gaps.
- To allow changes in the PLC to be recognized reliably, these must be available at least during the actual acquisition cycle.
- Set the transmission rate to the highest possible value.
- Set the "Token Hold Factor" in the PLC (channel configuration of the PLC) as high as possible.

Discrete alarms

Use arrays for discrete alarms and append each individual alarm to a bit of the array tags themselves and not to the individual sub-elements. Only tags of the "N", "O", "I", "S" and "B" file types and the "Int" and "UInt" data types are permitted for discrete alarms and arrays.

Screens

The refresh rate of screens is determined by the type and volume of data to be visualized.

Only configure short acquisition cycles for objects which actually need to be updated quickly. This procedure reduces update times.

Trends

The HMI device always updates all bit-triggered trends whose group bit is set in the "Trend transfer area". It then resets the bits.

The group bit in the PLC program can only be set again after all bits have been reset by the HMI device.

Job mailboxes

If large numbers of PLC jobs are sent in quick succession, this can lead to overload in the communication between the HMI device and PLC.

The HMI device confirms acceptance of the PLC job by entering the value 0 in the first data word of the job mailbox. The HMI device now processes the job, for which it requires time. If a new PLC job is entered again immediately in the job mailbox, it may take some time before the HMI device can process the next PLC job. The next PLC job will only be accepted when there is computing capacity available.

Network parameter assignment

The HMI device and controller network devices all have basically the same priority at the DH485. The owner of a so-called "token" controls the bus until it has passed the token to the

device with the next highest node number. To ensure optimum bus parameter assignment, observe the following:

- The bus participants must occupy the addresses from 1 with no gaps, if possible, so that there are no interruptions in passing on the token.
- The "Max. bus address" parameter must correspond to the highest bus node address used.

See also

Requirements of communication (Page 583)

Commissioning components (Allen-Bradley DH485) (Basic Panels)

Transferring the PLC program to the PLC

1. Connect the PC to the CPU using the appropriate cable.
2. Download the program files to the CPU.
3. Then set the CPU to RUN.

Transferring the project to the HMI device

Two scenarios exist for transferring the project to the HMI device:

- Initial commissioning
  The HMI device does not yet contain any configuration data in the initial commissioning phase. The project data and runtime software required for operation must be transferred from the configuration computer to the device:
- Recommissioning
  Recommissioning means that you overwrite existing project data on the HMI device.

1. To download the project to the HMI device, start "Start Center" on the HMI device.
2. In the "Start Center" under "Settings", set the protocol to be used for loading the project.
3. On the HMI device, set the transfer mode to "Automatic" or "Manual".
4. Check whether the alarm settings in your WinCC project meet your requirements.
5. Select the "Download to device > Software" command from the shortcut menu of an HMI device.
6. If the "Extended download to device" dialog is open, configure the "Settings for download". Make sure that the "Settings for download" correspond to the "Transfer settings in the HMI device".
   – Select the protocol used, for example, Ethernet or HTTP.
   – Configure the relevant interface parameters on the configuration PC.
   – Make any interface-specific or protocol-specific settings required in the HMI device.
   – Click "Load".

   The "Load preview" dialog opens. The project is compiled at the same time. The result is displayed in the "Load preview" dialog.

7. Check the displayed defaults and change them as necessary.
8. Click "Load".
   After successful download, the project can be executed on the HMI device.

   Note

   You can open the "Extended download to device" dialog at any time with the menu command "Online > Extended download to device...".

   Please refer to the documentation for the HMI device used for more detailed information on transfer settings.

Interconnecting the PLC with the HMI device

1. Interconnect the PLC and the HMI device using a suitable cable.
2. The message "Connection to PLC .... is established" is output to the HMI device. Note that users can edit the system event texts in WinCC.

   Note

   Always observe the safety-related information in the HMI device manual when commissioning the device.

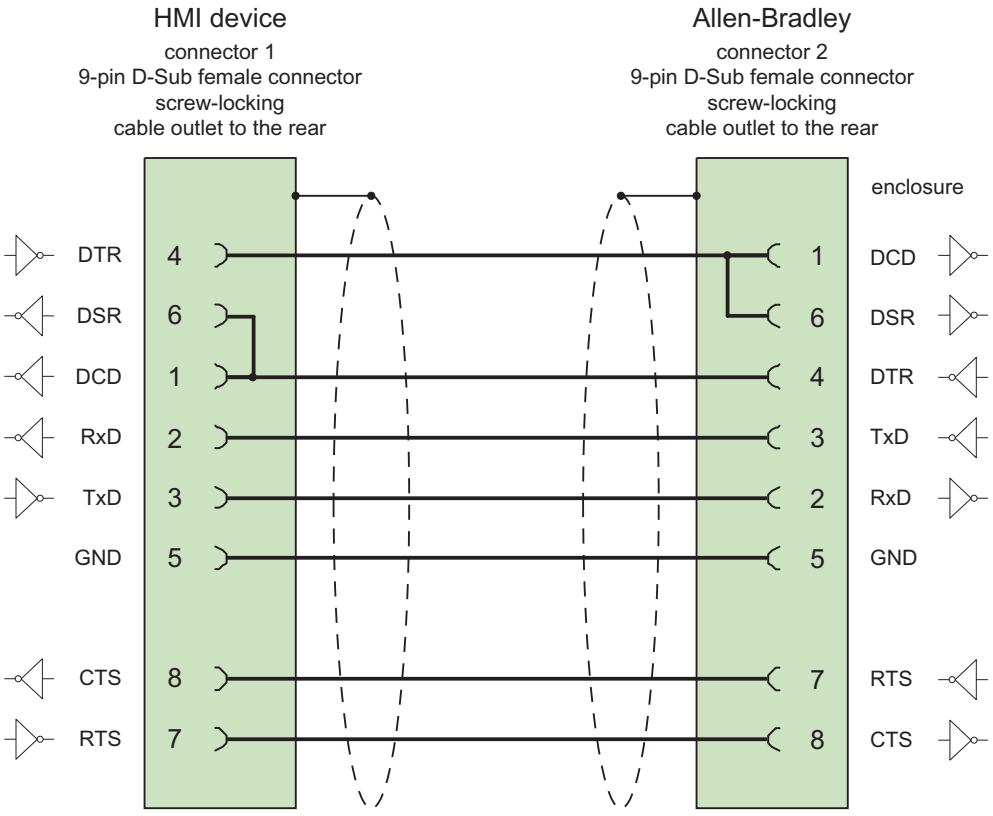
   Note that RF radiation emitted from devices such as cell phones may cause unwanted operating states.

Connecting cables (Basic Panels)

Connecting cable 1747-CP3, RS 232 (Basic Panels)

Allen-Bradley cable 1747-CP3

For interconnecting HMI device (RS 232, 9-pin D-sub) - SLC5/03, SLC5/04, SLC5/05 (Channel 0), AIC+



Shield with large-area contact to housing at both ends
Cable: 5 x 0.14 mm², shielded,
max. length 3 m

See also

Communication partners (Page 581)

Connecting cable 1761-CBL-PM02, RS 232 (Page 593)

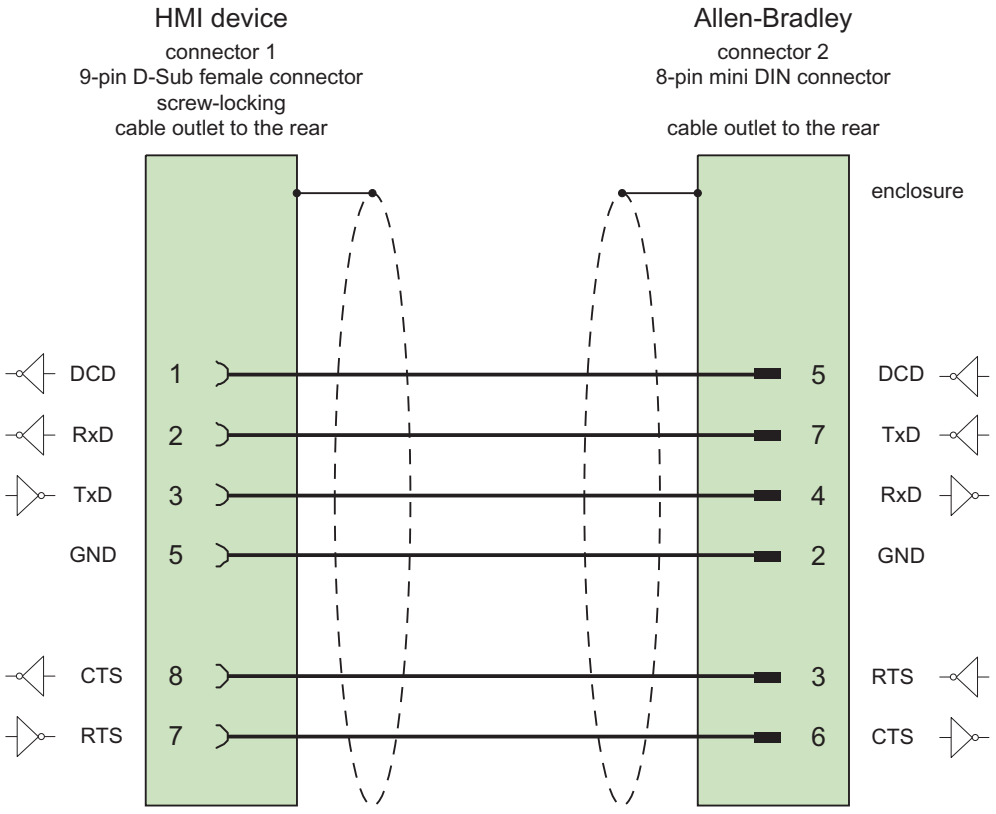
Connecting cable PP4, RS 485 (Page 594)

Connecting cable MP1, RS 485 (Page 596)

Connecting cable 1761-CBL-PM02, RS 232 (Basic Panels)

Allen-Bradley cable 1761-CBL-PM02

For interconnecting the HMI device (RS 232, 9-pin sub D) - Micro Logix, AIC+



Shield with large-area contact to housing at both ends
Cable: 5 x 0.14 mm^2, shielded,
max. length 15 m

See also

Connecting cable 1747-CP3, RS 232 (Page 592)

Connecting cable PP4, RS 485 (Basic Panels)

PP4 connecting cable

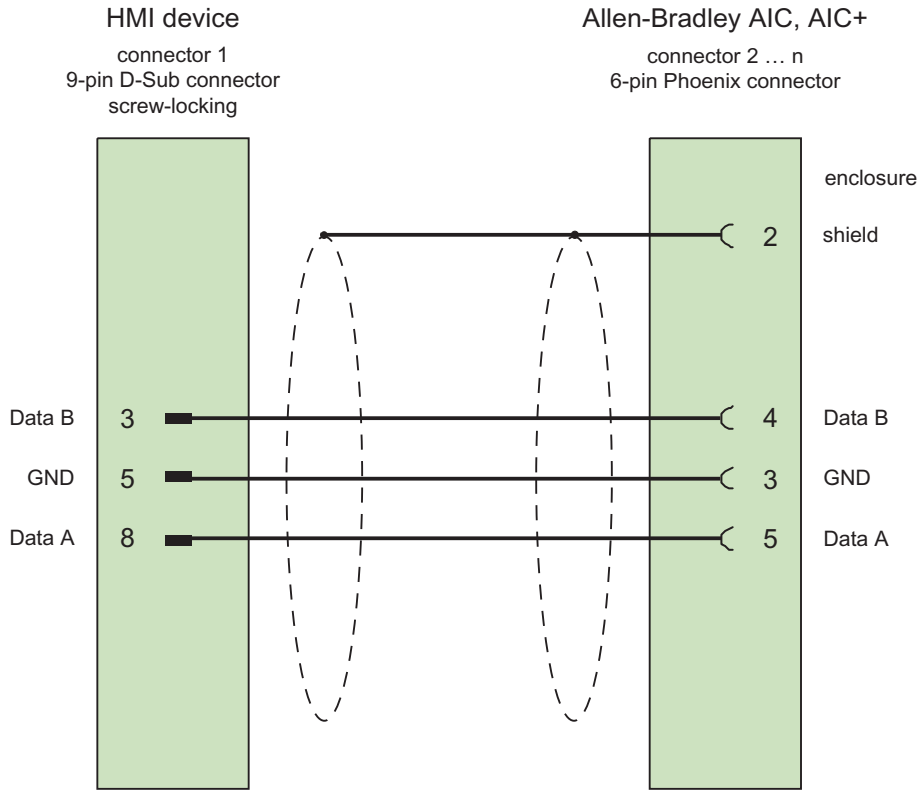
For interconnecting the HMI device (RS 485, 9-pin sub D) - AIC, AIC+

Items to observe when defining the connection concept:

- Minimum cable length = 1 m
- Maximum cable length = 1220 m
- Terminating resistance of 120 ohms between the data lines Data A and Data B only for longer cables.

Note

The shield of the cable may not be connected to the HMI device housing.



Cables: 5 x 0.14 mm^2, shielded,
min. length 1 m
max. bus length 1500 m

See also

Connecting cable 1747-CP3, RS 232 (Page 592)

Connecting cable MP1, RS 485 (Basic Panels)

MP1 connecting cable

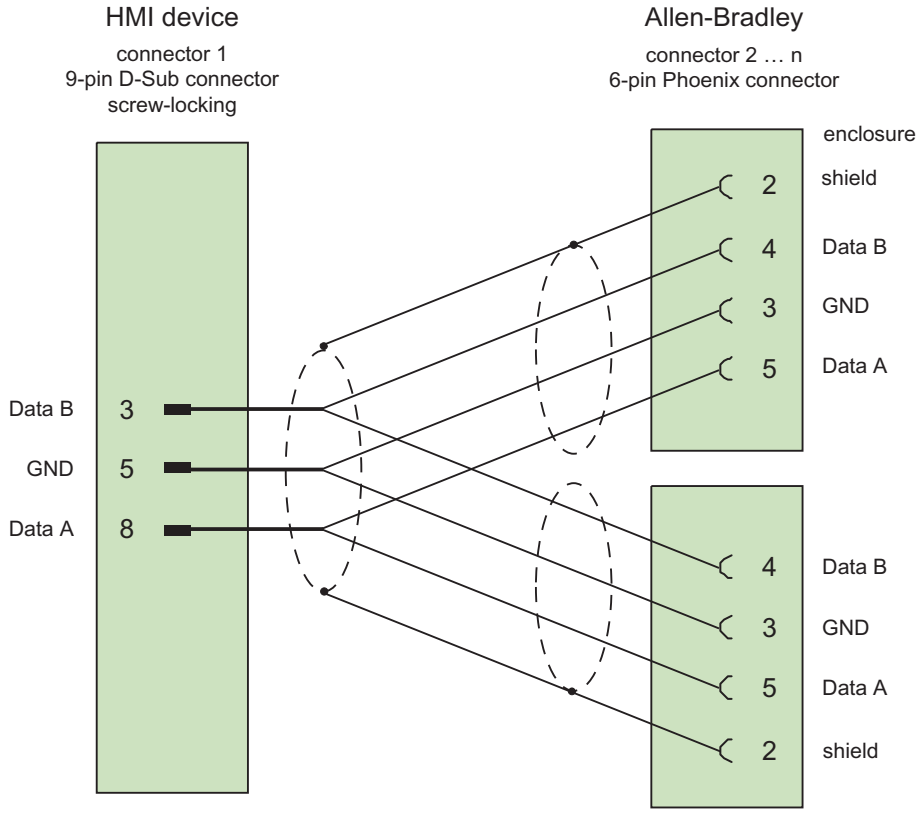
For connection HMI device (RS 485, 9-pin sub D) - DH485-LAN (AIC, AIC+)

When planning the network attachment, remember the following points:

- The HMI device must not be attached at the start or end of the LAN
- Both ends of the bus must be terminated. Refer to the Allen-Bradley documentation on installing the RS-485 network (for example, Allen-Bradley 1761-6.4).
- Cable length of the entire DH485 network: max. 1,220 m

Note

The shield of the cable must not be connected to the casing of the HMI device.



Cables: 5 x 0.14 mm², shielded,
max. length 1220 m

See also

Connecting cable 1747-CP3, RS 232 (Page 592)

Standard Modbus (Basic Panels)

Communication with standard Modbus (Basic Panels)

Communication partner (standard Modbus) (Basic Panels)

Introduction

This document describes specific extensions for the "Standard Modbus" Communication Service Package. For general information on communication between PLCs and HMI devices, refer to the "Communicating with PLCs" chapter in the TIA Portal help.

The "Standard Modbus" Communication Service Package is available for Basic Panels 2nd Generation with TIA V13 SP1 Update 8 or higher.

Note

The "Standard Modbus" Communication Service Package is not compatible with projects that were migrated from earlier versions of the TIA Portal, from WinCC Classic or from WinCC flexible.

The coupling with the PLCs and with the weighing modules SIWAREX WP231 and SIWAREX WP241 is implemented using the PLC's own protocols as follows:

- Standard Modbus RTU
- Standard Modbus TCP/IP

Connectable controllers

Coupling can be implemented for the following standard Modbus PLCs:

Standard Modbus controller	Supported protocol	
	Standard Modbus RTU	Standard Modbus TCP/IP
Modbus 984	x	x[1)]
TSX Compact	x	x[1)]
TSX Quantum	x	x
Momentum	-	x
Premium	-	x
Micro	-	x
M340	x	x

1) Only via Ethernet TCP/IP Modbus Plus Bridge

SIWAREX weighing modules	Supported protocol	
	Standard Modbus RTU	Standard Modbus TCP/IP
SIWAREX WP231	x	x
SIWAREX WP241	x	x

Note

Generic addressing

The project engineer is alone responsible for the correct configuration and the availability of addresses in the PLC.

Note

The addressing must be assigned to the corresponding standard addresses (for example, to the address area 4x for MW%).

Approved communication types with standard Modbus RTU

The following types of communication have been system-tested and approved:

- Point-to-point connection only via the RS 232 interface using an RS 232-RS 422 convertor.
- Multipoint connection from an HMI device (Modbus master) with up to 4 controllers: The HMI device must be connected with a Modbus Plus Bridge or a Modbus 984-CPU or TSX Quantum-CPU which is configured as a Modbus Plus Bridge.
- Connect the other PLCs via the Modbus plus connection to the first PLC. The PLCs can be reached under their address via the bridge functionality of the first PLC.

  Note

  It is not possible to integrate the HMI device into a Modbus network because the HMI device is Modbus master.

- The integration of the HMI device in a Modbus Plus network via Modbus Plus Bridge type BM85-000 (logical point-to-point communication of the HMI device with a Modbus 984 or TSX Quantum).
- Integration of the HMI device into a Modbus Plus network via the bridge function of the Modbus 984 or TSX Quantum, logical point-to-point communication of the HMI device with a controller.
- SIWAREX weighing modules communicate with Standard Modbus RTU only via the RS 485 interface.

Note

The "Change word order" parameter must be enabled for communication with SIWAREX WP231 via the Standard Modbus RTU protocol.

Restrictions

Note

Controllers which have not been tested and which support the Standard Modbus RTU protocol can also be connected to the HMI device. However, full functionality is not guaranteed.

If you nevertheless use a different PLC, you will find several useful tips here:

- These drivers only work for tags with the bit counting method typical for Modbus PLCs from left (bit1 = most significant bit) to right (bit16 = least significant bit in data type INT).
- The address offset displayed in the configuring is subtracted at protocol level in the message frame. For example, the offset is "400001" at the holding register 4x. As a result, the configured address "400006" becomes address "5" in the frame. The address (for example, "5") transferred in the message frame is transformed to the PLC-specific address area in the different Non-Modbus PLCs.
- A response frame without "ExceptionCode" is expected within 500 ms.
- The following function codes are used for the respective data areas:

Reading function codes		Address area	
01	ReadCoilStatus	0x	DIGITAL_OUT
02	ReadInputStatus	1x	DIGITAL_IN
03	ReadHoldingRegisters	4x	USERDATA
04	ReadInputRegisters	3x	ANALOG_IN
20 (14Hex)	ReadGeneralReference	6x	EXTENDEDMEMORY (not on all CPUs)

Writing function codes		Address area	
06	PresetSingleRegister	4x	USERDATA Single
16 (10Hex)	PresetMultipleRegisters	4x	USERDATA Multiple
05	ForceSingleCoil	0x	DIGITAL_OUT with BIT
15 (0FHex)	ForceMultipleCoils	0x	DIGITAL_OUT with 16 BIT GROUP
21 (15Hex)	WriteGeneralReference	6x	EXTENDEDMEMORY (not on all CPUs)

Approved communication types with standard Modbus TCP/IP

The following types of communication have been system-tested and approved:

- Point-to-point connection:
- Multipoint connection from an HMI device (Modbus TCP/IP client) with up to 4 controllers, with differing connections in each case.
  The following types of connection are possible:
  - Connection to the Ethernet CPU interface of the TSX Unity Quantum
  - Connection via the communication module for Ethernet 140 NOE 771 01 for the series TSX Quantum and TSX Unity Quantum
  - Connection via the Ethernet interface of the CPU adapter 171 CCC 980 20 of the Momentum series
  - Connection to the Ethernet CPU interface of the TSX Unity Quantum
  - Connection via the Ethernet TCP/IP switch-on module TSX ETY 110 for the series TSX Premium and TSX Unity Premium
  - Connection via the Ethernet TCP/IP switch-on module TSX ETY 410 for the Micro series
  - Connection via the Ethernet TCP/IP Modbus Plus Bridge 174 CEV 200 40 to the Modbus Plus interface of the Compact, the TSX Quantum, the TSX Unity Quantum and the 984 series (apart from 984A, 984B and 984X)

  Via bridge the controllers can be reached via their remote slave address via the Ethernet interface of the bridge.

Note

It is not possible to integrate the HMI device in a Modbus network via a bridge, as the HMI device is a Modbus master.

Restrictions

Note

Controllers which have not been tested and which support the Standard Modbus TCP/IP protocol can also be connected to the HMI device. However, full functionality is not guaranteed.

If you nevertheless use a different PLC, you will find several useful tips here:

- Use the CPU type "Premium/Micro" as this works without an address offset and with the standard bit method of counting.
- The following function codes are used for the respective data areas:

Reading function codes		Address area	
01	ReadCoilStatus	0x / %M	DIGITAL_OUT
02	ReadInputStatus	1x / %I	DIGITAL_IN
03	ReadHoldingRegisters	4x / %MW	USERDATA
04	ReadInputRegisters	3x / %IW	ANALOG_IN
20 (14Hex)	ReadGeneralReference	6x / –	EXTENDEDMEMORY (not on all CPUs)

Writing function codes		Address area	
06	PresetSingleRegister	4x / %MW	USERDATA Single
16 (10Hex)	PresetMultipleRegisters	4x / %MW	USERDATA Multiple
05	ForceSingleCoil	0x / %M	DIGITAL_OUT with BIT
15 (0FHex)	ForceMultipleCoils	0x / %M	DIGITAL_OUT with 16 BIT GROUP
21 (15Hex)	WriteGeneralReference	6x / –	EXTENDEDMEMORY (not on all CPUs)

See also

Communication between HMI deice and controller (standard Modbus) (Page 601)

Requirements of communication (Page 603)

General information on area pointers (standard Modbus) (Page 626)

General information on operational messages, alarm messages and acknowledgments (Page 642)

Commissioning components (Page 609)

Connecting cable 6XV1440-1K, RS 232, for standard Modbus (Page 619)

Installing CSP standard Modbus (Page 602)

Communication between HMI deice and controller (standard Modbus) (Basic Panels)

Communications principle

The HMI device and the PLC communicate using tags and area pointers.

Note

The area pointers are not available for SIWAREX modules.

Tags

The PLC and the HMI device use process values for data exchange. In your configuration, create tags that point to an address on the PLC. The HMI device reads and displays the value from the defined address. The operator can also make an entry on the HMI device that is then written to the address on the PLC.

Area pointers

Area pointers are used to exchange specific data and are only set up when this data is used.

For example, area pointers are required for the following data:

- PLC jobs
- Data record transfer
- Date/time synchronization
- Sign-of-life monitoring

You create the area pointers during configuration. You assign the corresponding addresses in the PLC.

See also

Communication partner (standard Modbus) (Page 597)

Installing CSP standard Modbus (Page 602)

Installing CSP standard Modbus (Basic Panels)

Installing CSP standard Modbus

The CSP standard Modbus requires the following installations:

- WinCC V13 SP1 Update 8
- The "CSP_V1301_004_001_WinCCTIAPortal_STD_MODBUS.isp13" file must be on the configuration PC.

1. Open the TIA Portal with administrator rights.
2. Select "Options > Support packages" from the menu.
   The "Detailed information > Installation of support packages" dialog box opens.
3. Click the "Add from file system" button.
4. Select the directory of the file "CSP_V1301_004_001_WinCCTIAPortal_STD_MODBUS.isp13" in the file system.
5. Click "Open."
   The file is added in the TIA Portal.
6. Enable "CSP_V1301_004_001_WinCCTIAPortal_STD_MODBUS.isp13" in the "Detailed information" dialog.
7. Click the "Install" button.
   A dialog opens.
8. Close the TIA Portal.
9. Click "Next".
   Installation is started.

Note

The CSP standard Modbus can be uninstalled.

The uninstalling is handled automatically by a setup program. Once started, the setup program guides you step-by-step through the entire removal procedure.

See also

Communication partner (standard Modbus) (Page 597)

Communication between HMI deice and controller (standard Modbus) (Page 601)

Configuration of the communications driver standard Modbus (Basic Panels)

Communication via Modbus RTU protocol (Basic Panels)

Requirements of communication (Basic Panels)

Connector

Connect the HMI device to the programming interface of the CPU (RS 232).

The connection of the HMI device to standard Modbus is limited primarily to the physical connection of the HMI device. No special blocks are required on the PLC for the connection.

Cable

The following connecting cables are available to connect the HMI device to standard Modbus.

Interface on the HMI device	Standard Modbus controller		
	Directly via Modbus interface (RS 232)	Via MB bridge (RS 232)	TSX Compact Point-to-point coupling
RS-232, 9-pin	PP1	PP1	PP2
RS 232, 15-pin	6XV1 440-1K...	6XV1 440-1K...	PP3

... = Length code (see catalog)

SIWAREX

The SIWAREX WP231 and SIWAREX WP241 weighing modules can be connected to a HMI device via the RS485 serial interface.

The SIWAREX modules can be used in the following topologies:

- Coupling between two stations: Point-to-point connection
- Coupling between multiple stations: Multi-point connection

See also

Installing the communication driver (Page 604)

Configuring the PLC type and protocol (Page 604)

Configuring protocol parameters (Page 604)

Permitted data types (standard Modbus RTU) (Page 607)

Optimizing the configuration (Page 608)

Commissioning components (Page 609)

Installing the communication driver (Basic Panels)

Driver for the HMI device

The driver for connecting to a standard Modbus controller via Modbus RTU is supplied with TIA Portal V13 SP1 Update 8 and installed automatically.

No special blocks are required on the PLC for the connection.

See also

Requirements of communication (Page 603)

Configuring the PLC type and protocol (Basic Panels)

Select the PLC

1. For a connection with a standard Modbus controller via the Modbus RTU protocol, double-click "Connections" in the project window of the HMI device.
2. In the work area, select the standard Modbus RTU protocol in the "Communication driver" column.
   The Properties window displays the parameters of the selected protocol.

For subsequent changes of the parameters, double-click "Connections" in the project window of the HMI device. Select the connection and edit its parameters in the properties dialog box.

Note

The settings on the HMI device and on the PLC must match.

See also

Requirements of communication (Page 603)

Configuring protocol parameters (Basic Panels)

Parameters to be set

To set the parameters, double-click "Connections" in the project window of the HMI device. "Standard Modbus RTU" is selected in the "Communication drivers" column in the work area. You can now enter or modify the protocol parameters in the Properties window:

Device-dependent parameters

- Interface
  Under "Interface", select the HMI interface to which the standard Modbus controller is connected.
  For more detailed information, refer to the HMI device manual.
- Type
  RS 232, RS 485 and RS 422 are system-tested.
- Baud rate
  Under "Baud rate", you set the transmission speed between the HMI device and standard Modbus controller. Communication is possible at 9600 bps and 19200 bps.
- Data bits
  Under "Data bits" you can only select "8".
- Parity
  Select "None", "Even" or "Odd" under "Parity".
- Stop bits
  Under "Stop bits" you can select 1 or 2.

Network parameters

- Framing
  Under "Framing" you can set that RTU (standard) is used as framing.

PLC-dependent parameters

- Slave address
  Under "Slave address" you set the slave address of the PLC.
- Change word order
  The "Change word order" parameter only modifies the word order for the representation of 32-bit values. The setting has an effect on the data types Double, Double+/- and Float. The byte sequence cannot be altered.
  - "Change word order" disabled
    The Most Significant Byte (MSB) is sent first.
    For double words, the "Least Significant Word" is sent before the "Most Significant Word".
    This setting has been system-tested for all approved PLCs.
  - "Change word order" enabled
    The Most Significant Byte (MSB) is sent first.
    For double words, the "Most Significant Word" is sent before the "Least Significant Word".
- Change bit order
  The "Change bit order" parameter has an effect on the bit data types for the areas 4x and 6x.
  - "Change bit order" disabled
    The standard bit counting method (16 LSB - 1 MSB) used for controllers of the 984, Compact, Quantum and Momentum series will only be used for these CPUs in the "Tags" editor with selected data type "Bit". The following bit position assignment applies:

	Left byte								Right byte							
How the tags are counted	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

  - "Change bit order" enabled
    This bit counting method (0 LSB - 15 MSB) is only used for selected controllers (for example, the Premium and Micro series) in the "Tags" editor when the "Bit" data type is selected. The following bit position assignment applies:

How the bit positions are counted	Left byte								Right byte							
In the engineering system, you configure:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

See also

Requirements of communication (Page 603)

Permitted data types (standard Modbus RTU) (Basic Panels)

Permitted data types

The table lists the user data types that can be used when configuring tags and area pointers.

Name	Area	Data type
Coil (discrete output)	0x	Bit, 16-bit group
Discrete input	1x	Bit, 16-bit group
Input register	3x	Array, Bit, +/- Int, Int
Holding register (output)	4x	Array, Bit [1], +/- Int, Int, +/- Double, Double, Float, ASCII
Extended memory (Only available with the "TSX Quantum" CPU)	6x	Array, Bit [1], +/- Int, Int, +/- Double, Double, Float, ASCII

[1] In the case of write accesses note:
With the "bit" data type in the "4x" and "6x" areas, after changing the specified bit the entire word is written back to the PLC. There is no check to determine whether any other bits in the word have changed. As a result, the PLC only has read access to the specified word.

The standard bit counting method (16 LSB - 1 MSB) used with controllers of the 984, Compact and Quantum series will only be used in the "Tags" editor with selected data type "bit". The following bit location assignment applies:

	Left byte								Right byte							
How the tags are counted	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

When entering bit numbers elsewhere in WinCC, for example discrete alarms, LED maps, "SetBitInTag" system function and graphics lists, the following bit allocation by the engineering system (0 LSB - 15 MSB) applies:

How the bit positions are counted	Left byte								Right byte							
In the engineering system, you configure:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Format for "Signed"

The placeholder "+/-" stands for the data types "Signed Int" and "Signed Double".

Special considerations for connections with standard Modbus

Area pointers can only be used in the areas "4x" and "6x".

Only tags in the areas "4x" and "6x" and only the data types "Int" and "+/-Int" are permitted as trigger tags for discrete alarms.

Arrays and array tags are not permitted for discrete alarms and trends.

See also

Requirements of communication (Page 603)

Optimizing the configuration (Basic Panels)

Acquisition cycle and update time

The acquisition cycles for the "Area pointers" and the acquisition cycles of the tags specified in the configuration software are decisive factors for the actual update times that can be achieved.

Note

The area pointer is not available for SIWAREX modules.

The update time is the sum of the acquisition cycle + transmission time + processing time.

To achieve optimum update times, remember the following points during configuration:

- Keep the individual data areas as small as possible and as large as necessary.
- Define data areas that belong together as belonging together. You can improve the update time by setting up one large data area instead of several small areas.
- If the acquisition cycles you select are too short, this is detrimental to the overall performance. Set the acquisition cycle to suit the rate of change of the process values. The rate of temperature changes of a furnace, for example, is significantly slower than the speed rate of an electrical drive. As a general guideline, the acquisition cycle should be approx. 1 second.
- Put the tags of an alarm or a screen in one data area without gaps.
- To allow changes in the PLC to be recognized reliably, these must be available at least during the actual acquisition cycle.
- Set the baud rate to the highest possible value.

Discrete alarms

Use arrays for discrete alarms and append each individual alarm to a bit of the array tags themselves and not to the individual sub-elements. Only tags of "4x" and "6x" reference and "Int" and "+/-Int" data types are permitted for discrete alarms and arrays.

Screens

With screens, the update rate that can actually be achieved depends on the type and amount of data to be displayed.

In the interest of short update times during configuration, make sure that you configure short acquisition cycles only for objects which actually need to be updated quickly.

Trends

When using bit-triggered trends, if the group bit is set in the "Trend transfer area", the HMI device always updates all the trends whose bit is set in this area. It then resets the bits.

The group bit in the PLC program can only be set again after all bits have been reset by the HMI device.

PLC jobs

If large numbers of PLC jobs are sent in quick succession, this can lead to overload in the communication between the HMI device and PLC.

The HMI device confirms acceptance of the PLC job by entering the value 0 in the first data word of the job mailbox. The HMI device now processes the job, for which it requires time. If a new PLC job is entered again immediately in the job mailbox, it may take some time before the HMI device can process the next PLC job. The next PLC job will only be accepted when there is computing capacity available.

See also

Requirements of communication (Page 603)

Commissioning components (Basic Panels)

Transferring the PLC program to the PLC

1. Connect the PC to the CPU using the appropriate cable.
2. Load the program files to the CPU.
3. Then set the CPU to RUN.

Transferring the project to the HMI device

Two scenarios exist for transferring the project to the HMI device:

- Initial commissioning
  The HMI device does not yet contain any configuration during initial commissioning. The project data and Runtime software required for operation must be transferred from the configuration computer to the device: The HMI device automatically changes to transfer mode.
  The transfer dialog with a connection message appears on the HMI device.
- Recommissioning
  Recommissioning means that you overwrite an existing configuration on the HMI device.

1. To load the project to the HMI device, start "Start Center" on the HMI device.
2. In the "Start Center" under "Settings", set the protocol to be used for loading the project.
3. On the HMI device, set the transfer mode to "Automatic" or "Manual".
4. Check whether the alarm settings in your WinCC project meet your requirements.
5. Select the "Download to device > Software" command from the shortcut menu of an HMI device.

6. If the "Extended download to device" dialog is open, configure the "Settings for loading". Make sure that the "Settings for loading" correspond to the "Transfer settings in the HMI device".
   - Select the protocol used, for example, Ethernet or HTTP.
   - Configure the relevant interface parameters on the configuration PC.
   - Make any interface-specific or protocol-specific settings required in the HMI device.
   - Click "Load".

   The "Load preview" dialog opens. The project is compiled at the same time. The result is displayed in the "Load preview" dialog.
7. Check the displayed presettings and change them as necessary.
8. Click "Load".
   After successful download, the project can be executed on the HMI device.

   Note

   You can open the "Extended download to device" dialog at any time with the menu command "Online > Extended download to device...".

   Please refer to the documentation for the HMI device used for more detailed information on transfer settings.

Interconnecting the PLC with the HMI device

1. Interconnect the PLC and the HMI device using a suitable connecting cable.
2. The message "Connection to PLC .... is established" is output to the HMI device. Note that users can edit the system event texts in WinCC.

   Note

   Always observe the safety-related information in the HMI device manual when commissioning the device.

   Note that RF radiation emitted from devices such as cell phones may cause unwanted operating states.

See also

Requirements of communication (Page 603)

Communication via Modbus TCP/IP protocol (Basic Panels)

Requirements of communication (Basic Panels)

Connector

The connection of the HMI device to the Modbus controller can take place via the following components:

- Existing Ethernet network that also contains the PLCs
- Cross-over Ethernet cable connected directly to the Ethernet interface of the CPU or the communication module

The connection of the HMI device to a Modbus controller is limited primarily to the physical connection of the HMI device. No special blocks are required on the PLC for the connection.

SIWAREX

The SIWAREX WP231 and SIWAREX WP241 weighing modules can be connected to a HMI device via the RS485 serial interface.

The SIWAREX modules can be used in the following topologies:

- Coupling between two stations: Point-to-point connection
- Coupling between multiple stations: Multi-point connection

See also

Installing the communication driver (Page 611)

Configuring the PLC type and protocol (Page 612)

Configuring protocol parameters (Page 612)

Permitted data types (standard Modbus TCP/IP) (Page 614)

Optimizing the configuration (Page 615)

Installing the communication driver (Basic Panels)

Driver for the HMI device

The driver for connecting to a Modbus controller via Modbus TCP/IP is supplied with TIA Portal V13 SP1 Update 8 and installed automatically.

No special blocks are required on the PLC for the connection.

See also

Requirements of communication (Page 611)

Configuring the PLC type and protocol (Basic Panels)

Select the PLC

1. For a connection with a standard Modbus controller via the Modbus TCP/IP protocol, double-click "Connections" in the project window of the HMI device.
2. In the work area, select the standard Modbus TCP/IP protocol in the "Communication driver" column.
   The Properties window displays the parameters of the selected protocol.

For subsequent changes of the parameters, double-click "Connections" in the project window of the HMI device. Select the connection and edit its parameters in the properties dialog box.

See also

Requirements of communication (Page 611)

Configuring protocol parameters (Basic Panels)

Parameters to be set

To set the parameters, double-click "Connections" in the project window of the HMI device. "Standard MODBUS TCP/IP" is selected in the "Communication drivers" column in the work area. You can now enter or modify the protocol parameters in the Properties window:

Device-dependent parameters

- Interface
  Under "Interface", select the HMI interface with which the HMI device is connected to the network. In this case, set "Ethernet".
  For more detailed information, refer to the HMI device manual.
- Type
  The protocol type "IP" is set as default.

  Note

  You hence need to configure the IP address and the subnet mask manually on the HMI device.

PLC-dependent parameters

- Server
  Under "Server" you set the IP address (or host name) for the PLC.
- Port
  Under "Port" set the port that is to be used for the TCP/IP connection. The port used by the Modbus controllers is 502.

- Remote slave address
  Under "Remote slave address", only when using a bridge, set the slave address for the removed PLC.
  If no bridge is being used, the default value 255 (or 0) must remain.
- Change word order
  The "Change word order" parameter only modifies the word order for the representation of 32-bit values. The setting has an effect on the data types Double, Double+/- and Float. The byte sequence cannot be altered.
  - "Change word order" disabled
    The Most Significant Byte (MSB) is sent first.
    For double words, the "Least Significant Word" is sent before the "Most Significant Word".
    This setting has been system-tested for all approved PLCs.
  - "Change word order" enabled
    The Most Significant Byte (MSB) is sent first.
    For double words, the "Most Significant Word" is sent before the "Least Significant Word".
- Change bit order
  The "Change bit order" parameter has an effect on the bit data types for the areas 4x and 6x.
  - "Change bit order" disabled
    The standard bit counting method (16 LSB - 1 MSB) used for controllers of the 984, Compact, Quantum and Momentum series will only be used for these CPUs in the "Tags" editor with selected data type "Bit". The following bit position assignment applies:

	Left byte								**Right byte**							
How the tags are counted	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

  - "Change bit order" enabled
    This bit counting method (0 LSB - 15 MSB) is only used for selected controllers (for example, the Premium and Micro series) in the "Tags" editor when the "Bit" data type is selected. The following bit position assignment applies:

How the bit positions are counted	**Left byte**								**Right byte**							
In the engineering system, you configure:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

See also

Requirements of communication (Page 611)

Permitted data types (standard Modbus TCP/IP) (Basic Panels)

Permitted data types

The table lists the user data types that can be used when configuring tags and area pointers.

Name	Area with CPU 984, Compact, Quantum, Momentum	Data type
Coil (discrete output)	0x	Bit, 16-bit group
Discrete input	1x	Bit, 16-bit group
Input register	3x	Array, Bit, +/- Int, Int
Holding register (output)	4x	Array, Bit [2], +/- Int, Int, +/- Double, Double, Float, ASCII
Extended memory (only available with "Quantum/Momentum" CPU)	6x	Array, Bit [2], +/- Int, Int, +/- Double, Double, Float, ASCII

[1] Due to a system property of the third-party controller, it is not possible to access the last x bits at the end of the address area.

[2] Note the following for write access:
For data type "Bit" in the areas "4x" and "6x", the entire word is written back to the controller following a change to the specified bit. There is no check to determine whether any other bits in the word have changed. As a result, the PLC only has read access to the specified word.

The standard bit counting method (16 LSB - 1 MSB) used for controllers of the 984, Compact, Quantum and Momentum series will only be used for these CPUs in the "Tags" editor with selected data type "Bit". The following bit location assignment applies:

	Left byte								Right byte							
How the tags are counted	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

When entering bit numbers elsewhere in WinCC, for example discrete alarms, LED maps, "SetBitInTag" system function and graphics lists, the bit allocation by the engineering system (0 LSB - 15 MSB) applies:

How the bit positions are counted	Left byte								Right byte							
In the engineering system, you configure:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

This bit counting method also applies to Premium and Micro controllers

Format for "Signed"

The placeholder "+/-" stands for the data types "Signed Int" and "Signed Double".

Special considerations for connections with Modbus TCP/IP protocol

Area pointers can only be used in the "4x" or "6x" "area".

Only tags in the areas "4x" or %MW and "6x" and only the data types "Int" and "+/-Int" are permitted as trigger tags for discrete alarms.

Array tags may only be used for discrete alarms and trends. Therefore, array tags can only be created from tags of the "4x" or "%MW" and "6x" areas and have "Int" and "+/-Int" data types.

If an existing Modbus RTU project is switched over to Modbus TCP/IP protocol, the characters in the string may differ.

See also

Requirements of communication (Page 611)

Optimizing the configuration (Basic Panels)

Acquisition cycle and update time

The acquisition cycles for the "Area pointers" and the acquisition cycles of the tags specified in the configuration software are decisive factors for the actual update times that can be achieved.

Note

The area pointer is not available for SIWAREX modules.

The update time is the sum of the acquisition cycle + transmission time + processing time.

To achieve optimum update times, remember the following points during configuration:

- Keep the individual data areas as small as possible and as large as necessary.
- Define data areas that belong together as belonging together. You can improve the update time by setting up one large data area instead of several small areas.
- If the acquisition cycles you select are too short, this is detrimental to the overall performance. Set the acquisition cycle to suit the rate of change of the process values. The rate of temperature changes of a furnace, for example, is significantly slower than the speed rate of an electrical drive. As a general guideline, the acquisition cycle should be approx. 1 second.
- Put the tags of an alarm or a screen in one data area without gaps.
- To allow changes in the PLC to be recognized reliably, these must be available at least during the actual acquisition cycle.
- Set the baud rate to the highest possible value.

Discrete alarms

Use arrays for discrete alarms and append each individual alarm to a bit of the array tags themselves and not to the individual sub-elements. Only tags of "4x" or "6x" area and "Int" and "+/-Int" data types are permitted for discrete alarms and arrays.

Screens

With screens, the update rate that can actually be achieved depends on the type and amount of data to be displayed.

In the interest of short update times during configuration, make sure that you configure short acquisition cycles only for objects which actually need to be updated quickly.

Trends

When using bit-triggered trends, if the group bit is set in the "Trend transfer area", the HMI device always updates all the trends whose bit is set in this area. It then resets the bits.

The group bit in the PLC program can only be set again after all bits have been reset by the HMI device.

PLC jobs

If large numbers of PLC jobs are sent in quick succession, this can lead to overload in the communication between the HMI device and PLC.

The HMI device confirms acceptance of the PLC job by entering the value 0 in the first data word of the job mailbox. The HMI device now processes the job, for which it requires time. If a new PLC job is entered again immediately in the job mailbox, it may take some time before the HMI device can process the next PLC job. The next PLC job will only be accepted when there is computing capacity available.

Timeout response with TCP/IP (Ethernet)

Due to the use of the TCP/IP protocol, the breakdown of a connection is detected at the earliest after approximately one minute. Connection breakdowns cannot be reliably detected if no tags are requested, for example, when there are no output tags in the current screen.

Configure area pointer coordination for each PLC. This setting ensures that a communication failure is recognized after approximately two minutes, even in the aforementioned scenario.

See also

Requirements of communication (Page 611)

Commissioning components (Basic Panels)

Transferring the PLC program to the PLC

1. Connect the PC to the CPU using the appropriate cable.
2. Load the program files to the CPU.
3. Then set the CPU to RUN.

Transferring the project to the HMI device

Two scenarios exist for transferring the project to the HMI device:

- Initial commissioning
  The HMI device does not yet contain any configuration during initial commissioning. The project data and Runtime software required for operation must be transferred from the configuration computer to the device: The HMI device automatically changes to transfer mode.
  The transfer dialog with a connection message appears on the HMI device.
- Recommissioning
  Recommissioning means that you overwrite an existing configuration on the HMI device.

1. To load the project to the HMI device, start "Start Center" on the HMI device.
2. In the "Start Center" under "Settings", set the protocol to be used for loading the project.
3. On the HMI device, set the transfer mode to "Automatic" or "Manual".
4. Check whether the alarm settings in your WinCC project meet your requirements.
5. Select the "Download to device > Software" command from the shortcut menu of an HMI device.
6. If the "Extended download to device" dialog is open, configure the "Settings for loading". Make sure that the "Settings for loading" correspond to the "Transfer settings in the HMI device".
   – Select the protocol used, for example, Ethernet or HTTP.
   – Configure the relevant interface parameters on the configuration PC.
   – Make any interface-specific or protocol-specific settings required in the HMI device.
   – Click "Load".

   The "Load preview" dialog opens. The project is compiled at the same time. The result is displayed in the "Load preview" dialog.
7. Check the displayed presettings and change them as necessary.
8. Click "Load".
   After successful download, the project can be executed on the HMI device.

Note

You can open the "Extended download to device" dialog at any time with the menu command "Online > Extended download to device...".

Please refer to the documentation for the HMI device used for more detailed information on transfer settings.

Interconnecting the PLC with the HMI device

1. Connect the controller (CPU or communication module) and the HMI device using a suitable connecting cable.
2. The message "Connection to PLC .... is established" is output to the HMI device. Note that users can edit the system event texts in WinCC.

Note

Always observe the safety-related information in the HMI device manual when commissioning the device.

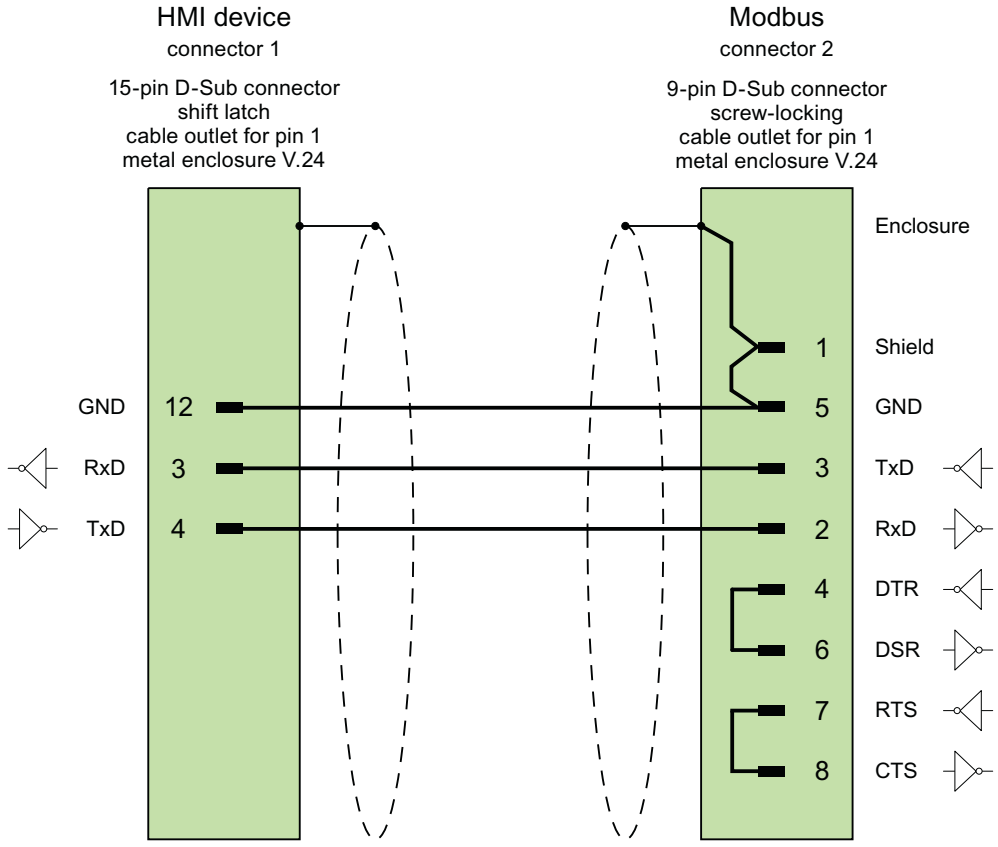
Note that RF radiation emitted from devices such as cell phones may cause unwanted operating states.

Connecting cable for standard Modbus (Basic Panels)

Connecting cable for standard Modbus RTU protocol (Basic Panels)

Connecting cable 6XV1440-1K, RS 232, for standard Modbus (Basic Panels)

Order number: 6XV1440 -1K...



Shield with large-area contact to casing
Cable: 2 x 0.14 mm^2, shielded,
max. length 3.7 m

See also

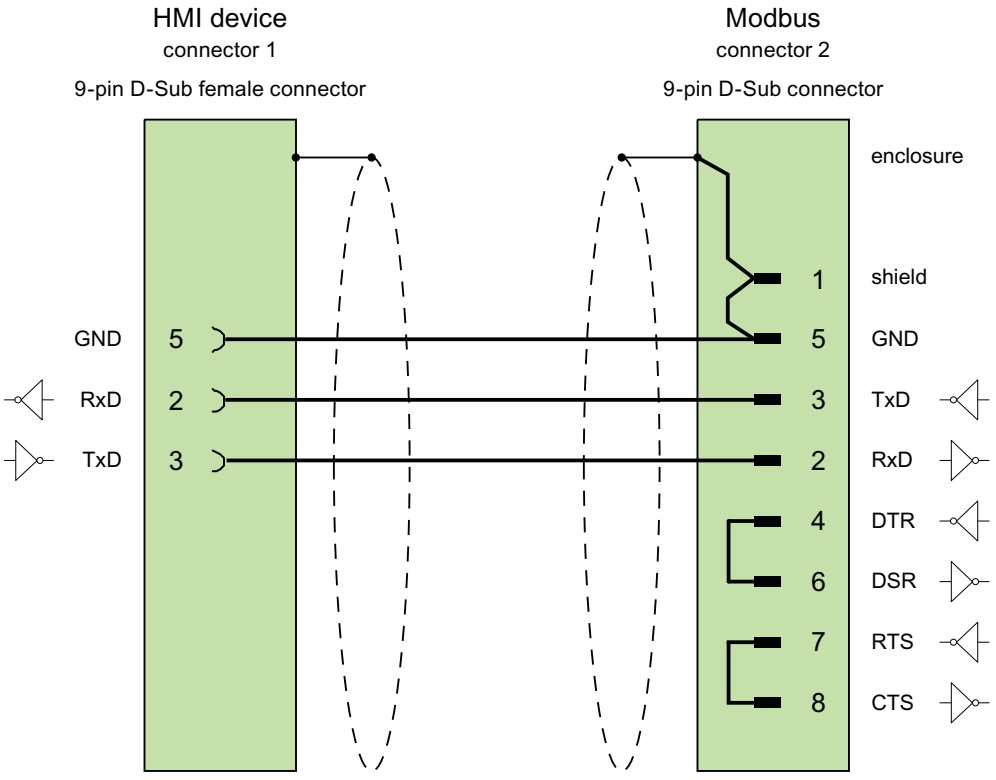
Connecting cable PP1, RS 232, for standard Modbus (Page 620)

Connecting cable PP2, RS 232, for standard Modbus (Page 621)

Connecting cable PP3, RS 232, for standard Modbus (Page 622)

Connecting cable PP1, RS 232, for standard Modbus (Basic Panels)

Point-to-point cable 1: PLC > PC ...



Cables: 3 x 0.14 mm², shielded,
max. length 15 m

See also

Connecting cable 6XV1440-1K, RS 232, for standard Modbus (Page 619)

Connecting cable PP2, RS 232, for standard Modbus (Basic Panels)

Point-to-point cable 2: PLC (TSX Compact) > PC...

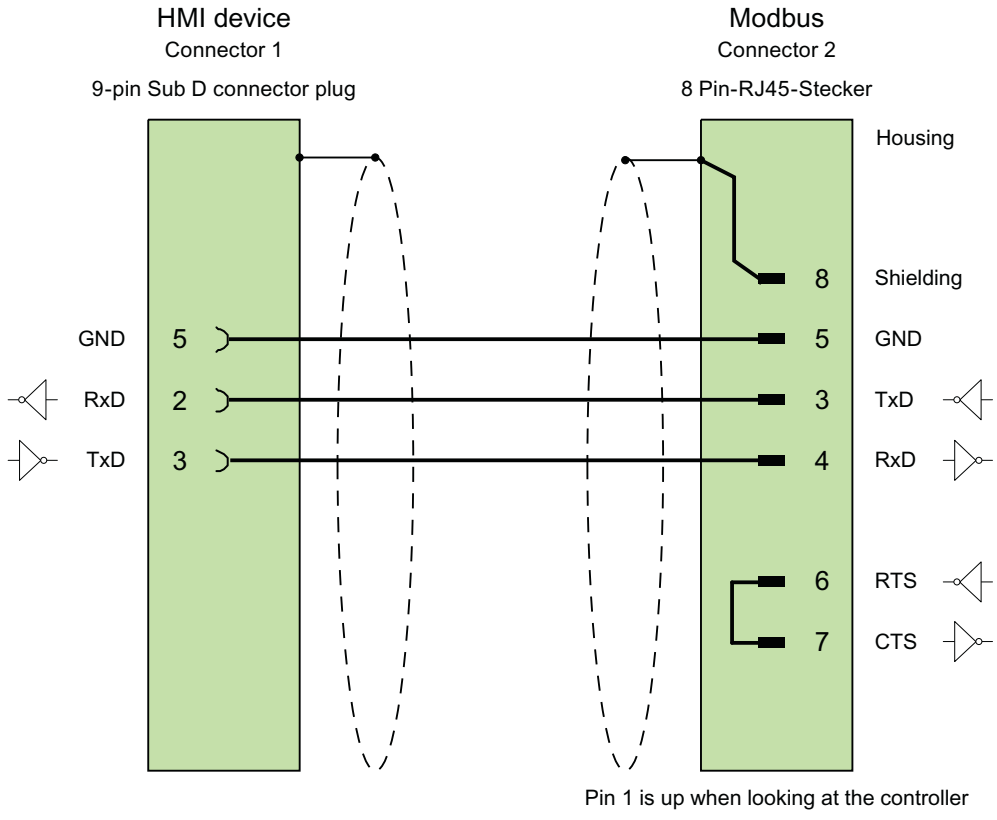


Figure 1-1 Modbus_Connecting_Cable_PP2_RS232_9-pin>8-pin RJ45

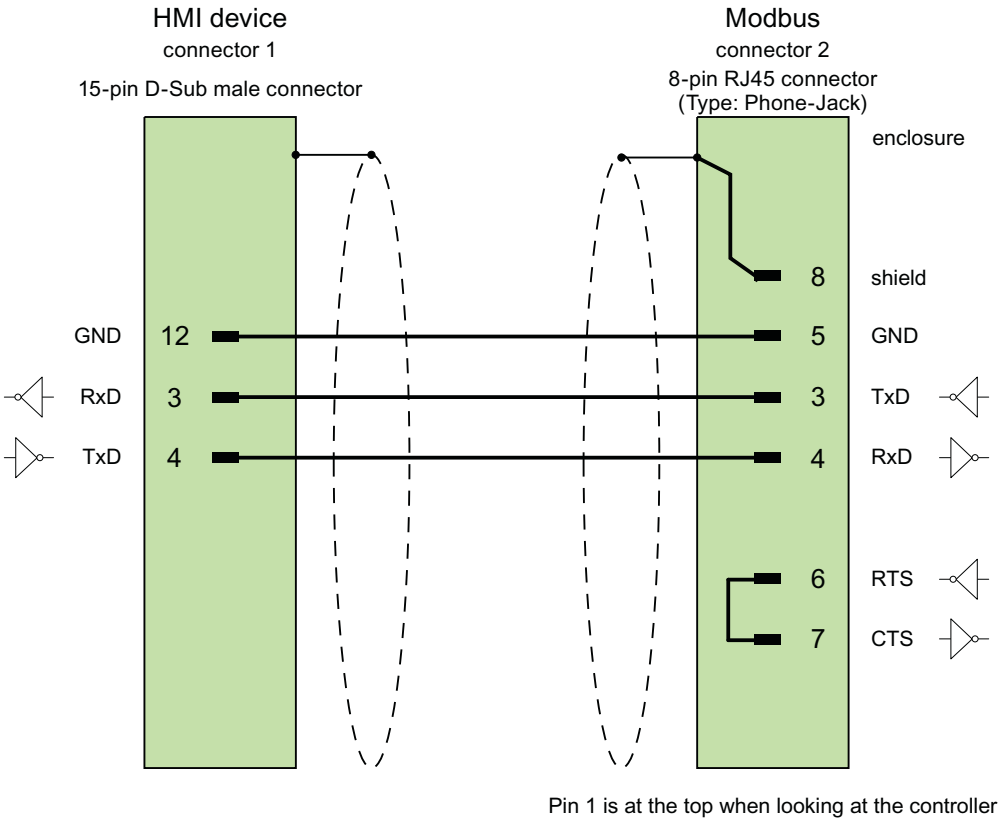
Cable: 3 x 0.14 mm^2, shielded,
max. Length 15 m

See also

Connecting cable 6XV1440-1K, RS 232, for standard Modbus (Page 619)

Connecting cable PP3, RS 232, for standard Modbus (Basic Panels)

Point-to-point cable 3: PLC (TSX Compact) > Multi Panel...



Cables: 3 x 0.14 mm^2, shielded,
max. length 15 m

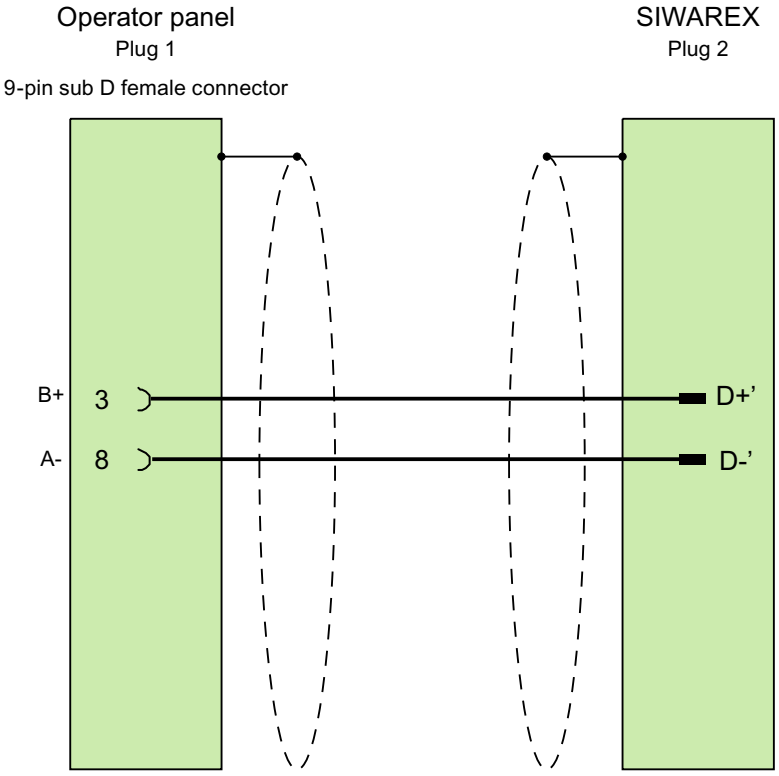
See also

Connecting cable 6XV1440-1K, RS 232, for standard Modbus (Page 619)

SIWAREX: Connection via RS485 (Basic Panels)

SIWAREX connecting cable

The SIWAREX WP231 and SIWAREX WP241 weighing modules can be connected to a HMI device via the RS485 serial interface.



Cable: Fieldbus cable 2-wire, shielded, ≤ 115 Kbps, e.g. 6XV1830-0EH10
max. Length 1,000 m

Note

You can find more information in the manual of the respective SIWAREX module.

See also

Connecting cable for standard Modbus RTU protocol (Page 619)

CSP_5 (Basic Panels)

User data areas (Basic Panels)

Trend request and trend transfer (Basic Panels)

Function

A trend is the graphic visualization of one or more values from the PLC. The value is read either time- or bit-triggered, depending on the configuration.

Time-triggered trends

The HMI device reads in the trend values cyclically at an interval specified in the configuration. Time-triggered trends are suitable for continuous processes, for example, the operating temperature of a motor.

Bit-triggered trends

By setting a trigger bit in the trend request tag, the HMI device either reads in a trend value or an entire trend buffer. This setting is defined in configuration data. Bit-triggered trends are normally used to visualize rapidly changing values. One example might be the injection pressure in the production of plastic parts.

To trigger bit-triggered trends, appropriate external tags must be created in the "Tags" editor during the configuration. The tags must be linked to trend areas. The HMI device and PLC then communicate with each other over these trend areas.

The following areas are available for trends:

- Trend request area
- Trend transfer area 1
- Trend transfer area 2 (required only with switch buffers)

Tags of "Reference", "4x", or "6x" are permitted. They must have data type "Int" or "+/-Int" or be an array tag with data type "Int" or "+/-Int". During configuration, you assign a bit to a trend. This sets a unique bit assignment for all areas.

Trend request area

The HMI device sets corresponding bits in the trend request area when you open a screen which contains one or more trends on the HMI device. After deselecting the screen, the HMI device resets the relevant bits in the trend request area.

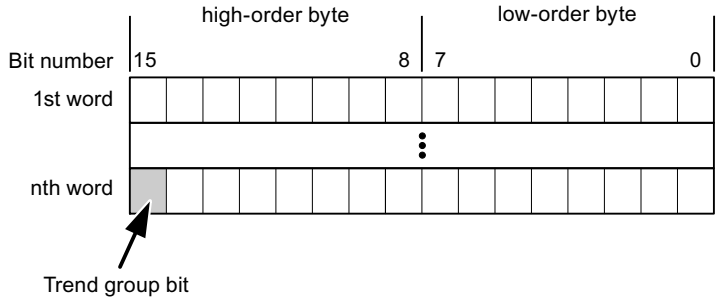
Using the trend request area, the PLC can recognize which trend is currently displayed on the HMI device. Trends can also be triggered without evaluation of the trend request area.

Trend transfer area 1

This area is used to trigger trends. In your PLC program, you set the bit assigned to the trend in the trend transfer area and set the trend group bit. The trend group bit is the last bit in the trend transfer area.

The HMI device detects the trigger. The HMI device reads either a value or the entire buffer from the PLC. It then resets the trend bit and the trend group bit.

The following picture shows the structure of a trend transfer area.



The trend transfer area must not be modified by the PLC program until the trend group bit has been reset.

Trend transfer area 2

Trend transfer area 2 is required for trends configured with a switch buffer. Trend transfer area 2 has exactly the same structure as trend transfer area 1.

Switch buffer

The switch buffer is a second buffer for the same trend that can be set up during configuration.

While the HMI device reads the values from buffer 1, the controller writes to buffer 2. When the HMI device reads buffer 2, the controller writes to buffer 1. This prevents the trend values being overwritten by the PLC while the trend is being read out by the HMI device.

LED mapping (Basic Panels)

Function

The keyboard units of Operator Panel (OP), Multi Panel (MP) and Panel PC have LEDs in the function keys. These LEDs can be activated from the PLC. Thus, it is possible, for example, to signal to the user which key should be pressed in a given situation by lighting up an LED.

Requirements

In order to activate an LED, an LED tag or an array tag must be set up in the PLC and specified as an LED tag during configuration.

LED assignment

The assignment of the individual LEDs to the bits in the LED tags is specified when the function keys are configured. In this process, the "LED tag" and the assigned "Bit" are specified for each function key in the "General" group of the properties window.

The "Bit" bit number designates the first of two consecutive bits that control the following LED statuses:

Bit n+ 1	Bit n	LED function	
		All Mobile Panels, all Operator Panels, all Multi Panels	Panel PCs
0	0	Off	Off
0	1	Rapid flashing	Flashing

Bit n+ 1	Bit n	LED function	
		All Mobile Panels, all Operator Panels, all Multi Panels	Panel PCs
1	0	Slow flashing	Flashing
1	1	On permanently	On permanently

Area pointer (Basic Panels)

General information on area pointers (standard Modbus) (Basic Panels)

Introduction

Area pointers are parameter fields. WinCC Runtime receives information about the location and size of data areas in the PLC from these parameter fields. During communication, the PLC and the HMI device alternately access those data areas for read and write operations. The PLC and the HMI device trigger defined interactions based on the evaluation of stored data.

The area pointers reside in controller memory. Their addresses are configured in the "Range Pointers" dialog of the "Connections" editor.

WinCC uses the following area pointers:

- Control request
- Project ID
- Screen number
- Data record
- Date/time
- Date/time PLC
- Coordination

Device dependency

Availability of the area pointer depends on the HMI device used.

Use

Configure and enable the area pointer in "Communication > Connections" before you put it into use.

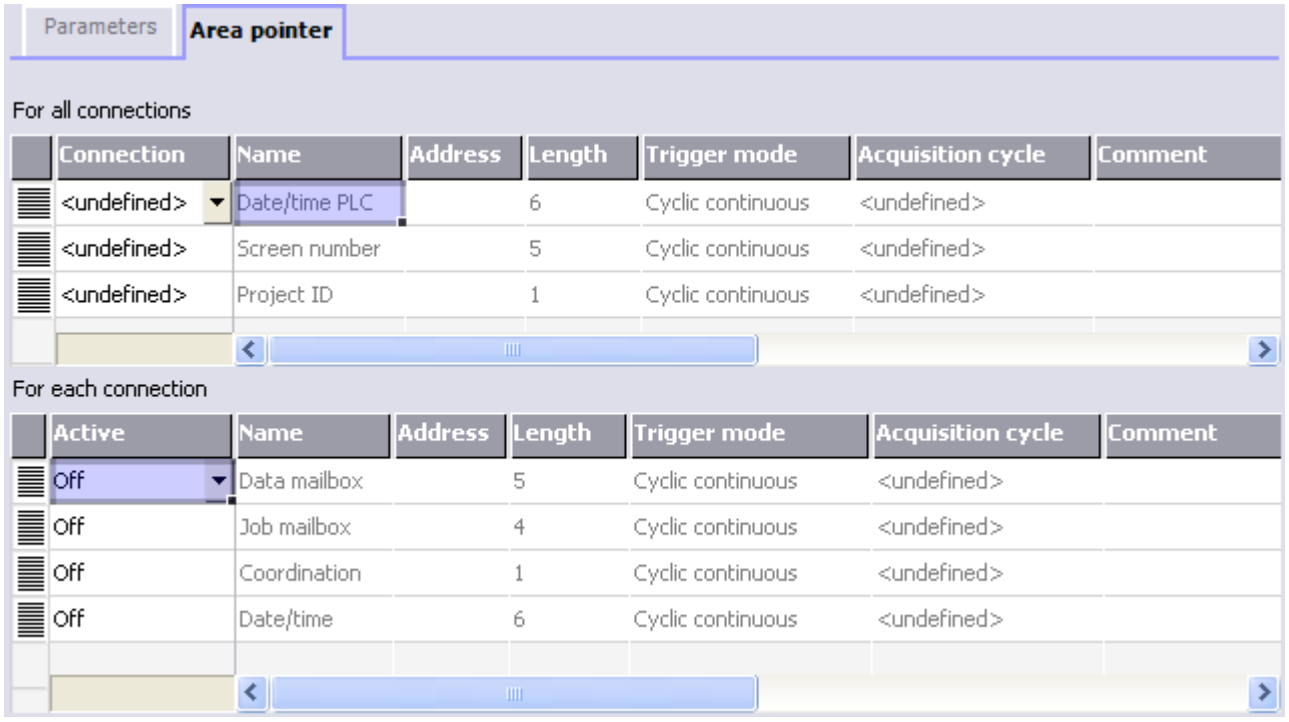


Figure 1-2 Enabling an area pointer based on the example of a SIMATIC S7 PLC

- Active
  Enables the area pointer.
- Name
  Name of the area pointer specified by WinCC.
- Address
  Tag address of the area pointer in the PLC.
- Length
  The length of the area pointer is specified by WinCC.
- Acquisition cycle
  Define an acquisition cycle in this field to allow cyclic reading of the area pointer in Runtime. An extremely short acquisition time may have a negative impact on HMI device performance.
- Comment
  Save a comment, for example, to describe the purpose of the area pointer.

Accessing data areas

The table shows how the controller and HMI device handle read (R) and write (W) access to the data areas.

Data area	Required for	HMI device	Controller
Screen number	Evaluation by the PLC in order to determine the active screen.	W	R
Data record	Transfer of data records with synchronization	R/W	R/W
Date/time	Transfer of the date and time from the HMI device to the controller	W	R
Date/time PLC	Transfer of the date and time from the controller to the HMI device	R	W
Coordination	Requesting the HMI device status in the control program	W	R
Project ID	Runtime checks for consistency between the WinCC project ID and the project in the controller	R	W
Control request	Triggering of HMI device functions by the control program	R/W	R/W

The next sections describe the area pointers and their associated controller jobs.

See also

"Data mailbox" area pointer (Page 635)

"Screen number" area pointer (Page 628)

"Date/time" area pointer (Page 629)

"Date/time controller" area pointer (Page 630)

"Coordination" area pointer (Page 631)

"Project ID" area pointer (standard Modbus) (Page 632)

"PLC job" area pointer (Page 633)

"Screen number" area pointer (Basic Panels)

Function

The HMI device saves information about the screen called on the HMI device to the "Screen number" area pointer.

This allows the transfer of the current screen contents from the HMI device to the PLC. The PLC can trigger specific reactions such as the call of a different screen.

Use

Configure and enable the area pointer in "Communication > Connections" before you put it into use. You can create only **one** instance of the "Screen number" area pointer and only on **one** PLC.

The screen number is transferred spontaneously to the PLC. That is, it is always transferred when a new screen is activated on the HMI device. It is therefore unnecessary to configure an acquisition cycle.

Structure

The area pointer is a data area in the memory of the PLC with a fixed length of 5 words.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st word	Current screen type															
2nd word	Current screen number															
3rd word	Reserved															
4th word	Current field number															
5th word	Reserved															

- Current screen type
  "1" for the root screen or
  "4" for the permanent window
- Current screen number
  1 to 32767
- Current field number
  1 to 32767

See also

General information on area pointers (standard Modbus) (Page 626)

"Date/time" area pointer (Basic Panels)

Function

This area pointer is used to transfer the date and time from the HMI device to the PLC.

The PLC writes control job "41" to the job mailbox.

When it evaluating the control job, the HMI device saves its current date and time to the data area configured in the "Date/time" area pointer. All definitions are coded in BCD format.

The "Date/Time" area pointer when used in a project which contains multiple connections must be enabled for each configured connection.

The date/time data area has the following structure:

Data word	Left byte							8	Right byte							0	
	15								7								
n+0	Reserved								Hour (0 to 23)								Time
n+1	Minute (0 to 59)								Second (0 to 59)								
n+2	Reserved								Reserved								
n+3	Reserved								Weekday (1 to 7, 1 = Sunday)								Date
n+4	Day (1 to 31)								Month (1 to 12)								
n+5	Year (80 to 99/0 to 29)								Reserved								

Note

The entry of values from 80 to 99 in the "Year" data area returns the years 1980 through 1999; values from 0 to 29 return the years 2000 through 2029.

See also

General information on area pointers (standard Modbus) (Page 626)

"Date/time controller" area pointer (Basic Panels)

Function

This area pointer is used to transfer the date and time from the PLC to the HMI device. Use this area pointer if the PLC is the time master.

The PLC loads the data area of the area pointer. All definitions are coded in BCD format.

The HMI device reads the data cyclically within the configured acquisition cycle and synchronizes itself.

Note

Set an acquisition cycle of sufficient length for the Date/time area pointer in order to avoid any negative impact on the HMI device performance.
Recommended: Acquisition cycle of 1 minute if your process allows this.

The date/time data area has the following structure:

DATE_AND_TIME format (in BCD code)

Data word	Left byte			Right byte		
	15	. . . . .	8	7	. . . . .	0
n+0	Year (80 to 99/0 to 29)			Month (1 to 12)		
n+1	Day (1 to 31)			Hour (0 to 23)		
n+2	Minute (0 to 59)			Second (0 to 59)		
n+3	Reserved			Reserved		Weekday (1 to 7, 1=Sunday)
n+4 [1)]	Reserved			Reserved		
n+5 [1)]	Reserved			Reserved		

1) The two data words must exist in the data area to ensure that the data format matches WinCC and to avoid reading false information.

Note

When making entries in the "Year" data area, you should note that values 80 to 99 result in years 1980 through 1999, while the values 0 to 29 result in the years 2000 through 2029.

See also

General information on area pointers (standard Modbus) (Page 626)

"Coordination" area pointer (Basic Panels)

Function

The "Coordination" area pointer is used to implement the following functions:

- Detecting the startup of the HMI device in the control program
- Detecting the current operating mode of the HMI device in the control program
- Detecting whether the HMI device is ready to communicate in the control program

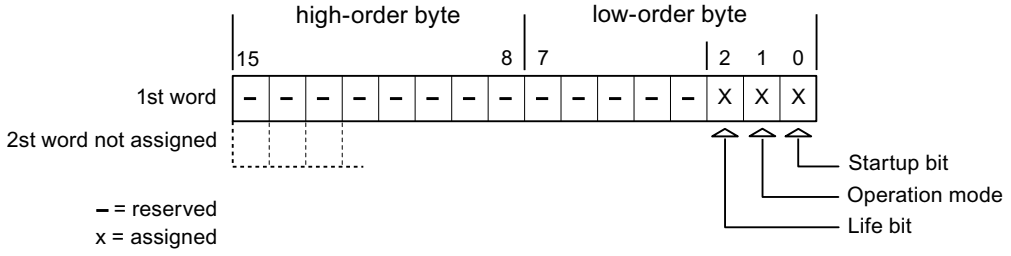
The "Coordination" area pointer has a length of two words.

Use

Note

The HMI device always writes the entire coordination area when updating the area pointer. The control program can for this reason not make changes to the coordination area.

Assignment of the bits in the "Coordination" area pointer



Startup bit

The startup bit is set briefly to "0" by the HMI device during startup. It sets the bit permanently to "1" when startup is completed.

Operating mode

The operating mode bit is set to 1 as soon as the user switches the HMI device offline. The status of the operating mode bit is "0" during normal operation of the HMI device. You can

determine the current operating mode of the HMI device by reading this bit in the control program.

Life bit

The HMI device inverts the life bit at intervals of approximately one second. You can check whether or not there is still a connection to the HMI device by querying this bit in the control program.

See also

General information on area pointers (standard Modbus) (Page 626)

"Project ID" area pointer (standard Modbus) (Basic Panels)

Function

You can check whether the HMI device is connected to the correct PLC at the start of runtime. This check is important when operating with several HMI devices.

The HMI device compares a value stored on the PLC with the value specified in configuration. This ensures compatibility of configuration data with the control program. If discrepancy is detected, a system alarm is displayed on the HMI device and runtime is stopped.

Use

To use this area pointer, set up the following during the configuration:

- Define the version of configuration. Possible values between 1 and 255.
  Enter the version in the "Device settings > Device settings" editor in "Project ID".
- Data address of the value for the version that is stored in the PLC:
  Enter the data address in the "Communication > Connections" editor in "Address".

Connection failure

A connection failure to a device on which the "project ID" area pointer is configured results in all the other connections in the project being switched to "offline".

This behavior has the following prerequisites:

- You have configured several connections in a project.
- You are using the "project ID" area pointer in at least one connection.

Causes which may set connections "offline":

- The PLC is not available.
- The connection has been switched offline in the engineering system.

See also

General information on area pointers (standard Modbus) (Page 626)

"PLC job" area pointer (Basic Panels)

Function

The PLC can use the job mailbox to transfer jobs to the HMI device to trigger corresponding actions on the HMI device. These functions include, for example:

- Display screen
- Set date and time

Data structure

The first word of the job mailbox contains the job number. Depending on the job mailbox, up to three parameters can be transferred.

Word	Left byte	Right byte
n+0	0	Job number
n+1	Parameter 1	
n+2	Parameter 2	
n+3	Parameter 3	

The HMI device evaluates the job mailbox if the first word of this job is unequal to zero. This means that the parameters must be entered in the job mailbox first, followed by the job number.

When the HMI device accepts the job mailbox, the first word is set to 0 again. The execution of the job mailbox is generally not completed at this point in time.

Job mailboxes

All job mailboxes and their parameters are listed below. The "No." column contains the job number of the job mailbox. Job mailboxes can only be triggered by the PLC when the HMI device is online.

Note

Please note that not all HMI devices support job mailboxes. TP 170A and Micro Panel do not support PLC jobs, for example.

No.	Function	
14	**Setting the time (BCD coded)**	
	Parameter 1	Left byte: - Right byte: hours (0-23)
	Parameter 2	Left byte: minutes (0-59) Right byte: seconds (0-59)
	Parameter 3	-
15	**Setting the date (BCD coded)**	
	Parameter 1	Left byte: - Right byte: weekday (1-7: Sunday-Saturday)

No.	Function	
14	**Setting the time (BCD coded)**	
	Parameter 2	Left byte: day (1-31) Right byte: month (1-12)
	Parameter 3	Left byte: year
23	**User logon**	
	Logs the user on with the name "PLC user" at the HMI device with the group number transferred in Parameter 1. The logon is possible only when the transferred group number exists in the project.	
	Parameter 1	Group number 1 to 255
	Parameter 2, 3	-
24	**User logoff**	
	Logs off the current user. The function corresponds to the "logoff" system function)	
	Parameter 1, 2, 3	-
40	**Transfer date/time to PLC**	
	(in the S7 format DATE_AND_TIME) An interval of at least 5 seconds must be maintained between two successive jobs to prevent overload of the HMI device.	
	Parameter 1, 2, 3	-
41	**Transfer date/time to PLC**	
	(In OP/MP format) An interval of at least 5 seconds must be maintained between successive jobs in order to prevent overload of the HMI device.	
	Parameter 1, 2, 3	-
46	**Update tags**	
	Causes the HMI device to read the current value of the PLC tags whose update ID matches the value transferred in Parameter 1. (Function corresponds to the "UpdateTag" system function.)	
	Parameter 1	1 - 100
49	**Clear process alarm buffer**	
	Parameter 1, 2, 3	-
50	**Clear alarm buffer**	
	Parameter 1, 2, 3	-
51	**Screen selection** [1)]	
	Parameter 1	Screen number
	Parameter 2	-
	Parameter 3	Field number
69	**Read data record from PLC**	
	Parameter 1	Recipe number (1-999)
	Parameter 2	Data record number (1-65535)
	Parameter 3	0: Do not overwrite existing data record 1: Overwrite existing data record
70	**Write data record to PLC**	
	Parameter 1	Recipe number (1-999)

No.	Function	
14	**Setting the time (BCD coded)**	
	Parameter 2	Data record number (1-65535)
	Parameter 3	-

1) OP 73, OP 77A and TP 177A HMI devices also execute the "Screen selection" job mailbox if the on-screen keyboard is active.

See also

General information on area pointers (standard Modbus) (Page 626)

"Data mailbox" area pointer (Basic Panels)

"Data mailbox" area pointer (Basic Panels)

Function

When data records are transferred between the HMI device and PLC, both partners access common communications areas on the PLC.

Transmission modes

There are two ways of transferring data records between the HMI device and PLC:

- Transfer without synchronization
- Transfer with synchronization over the data mailbox

Data records are always transferred directly, which means that the tag values are read straight from an address or written straight to an address configured for this tag without being redirected via an interim memory.

Initiating the transfer of data records

There are three ways of triggering the transfer:

- Operator input in the recipe view
- Job mailboxes
  The transfer of data records can also be triggered by the PLC.
- Triggering by configured functions

If the transfer of data records is triggered by a configured function or by a job mailbox, the recipe view on the HMI device remains operable. The data records are transferred in the background.

Simultaneous processing of several transfer requests is, however, not possible. In this case, the HMI device rejects the other transfer requests with a system event.

See also

General information on area pointers (standard Modbus) (Page 626)

Transfer without synchronization (Page 636)

Possible causes of error when transferring data records (Page 638)

Sequence of the transfer when triggered by a configured function (Page 638)

Sequence of the transfer triggered by a PLC job (Page 639)

Sequence of a transfer started by the operator in the recipe display (Page 641)

Transfer with synchronization (standard Modbus) (Page 637)

Transfer without synchronization (Basic Panels)

If you select asynchronous transfer of data records between the HMI device and PLC, there is no coordination over the common data areas. It is therefore unnecessary to set up a data area during configuration.

Asynchronous data record transfer can be a useful alternative, for example when:

- The system is capable of excluding the risk of uncontrolled overwriting of data by the communication peer.
- The PLC does not require information about the recipe number and data record number.
- The transfer of data records is triggered by the operator of the HMI device.

Reading values

When a read job is triggered, the values are read from the PLC addresses and transferred to the HMI device.

- Triggering by the operator in the recipe view:
  The values are downloaded to the HMI device. You can then, for example, process, edit, or save these values in the HMI device.
- Triggering by a function or job mailbox:
  The values are saved immediately to the data storage medium.

Writing values

When a write job is triggered, the values are written to the PLC addresses.

- Triggering by the operator in the recipe view:
  The current values are written to the PLC.
- Triggering by a function or job mailbox:
  The current values are written to the PLC from the data storage medium.

See also

"Data mailbox" area pointer (Page 635)

Transfer with synchronization (standard Modbus) (Basic Panels)

If you select synchronous transfer, both communication partners set status bits in the common data area. You can use this mechanism to prevent uncontrolled overwriting of data in either direction in your control program.

Application

Synchronous data record transfer can be a useful solution, for example, when:

- The PLC is the "active partner" in the transfer of data records.
- The PLC evaluates the information about the recipe number and data record number.
- The transfer of data records is triggered by means of a Job mailbox.

Requirements

In order to synchronize transfer of data records between the HMI device and the PLC, the following requirements must be met during configuration:

- An area pointer has been set up: "Communication > Connections" editor in "Area pointer".
- The PLC with which the HMI device synchronizes transfer of data records is specified in the recipe: "Recipes" editor, properties view of the recipe, "Properties" group in "Transfer".

Structure of the data area

The data area has a fixed length of 5 words. Structure of the data area:

	15		0
1st word	Current recipe number (1 - 999)		
2nd word	Current data record number (0 - 65535)		
3rd word	Reserved		
4th word	Status (0, 2, 4, 12)		
5th word	Reserved		

- Status
  The status word (word 4) can adopt the following values:

Value		Meaning
Decimal	Binary	
0	0000 0000	Transfer permitted, data record free
2	0000 0010	Transfer is busy.
4	0000 0100	Transfer completed without error
12	0000 1100	Transfer completed with error

See also

"Data mailbox" area pointer (Page 635)

Possible causes of error when transferring data records (Basic Panels)

Possible causes of error

The section below shows possible causes of errors which lead to a data record transfer being terminated with errors:

- Tag address not set up on the PLC
- Overwriting data records not possible
- Recipe number does not exist
- Data record number does not exist

Note

The status word may only be set by the HMI device. The PLC may only reset the status word to zero.

Note

The PLC may only evaluate the recipe and data record numbers when data inconsistency is detected if one of the conditions outlined below has been met:

- The data mailbox status is set to "Transfer completed".
- The data mailbox status is set to "Transfer completed with error".

Reaction to an aborted transfer due to errors

If the transfer of data records is aborted due to errors, the HMI device reacts as follows:

- Triggering by the operator in the recipe view
  Information in the status bar of the recipe view and output of system alarms
- Triggered by function
  Output of system alarms
- Triggering by job mailbox
  No return message on the HMI device

You can nonetheless evaluate the status of the transfer by querying the status word in the data mailbox.

See also

"Data mailbox" area pointer (Page 635)

Sequence of the transfer when triggered by a configured function (Basic Panels)

Reading from the PLC using a configured function

Step	Action	
1	Check: Status word = 0?	
	Yes	No

Step	Action	
2	The HMI device enters the recipe and data record number specified in the function and the status "Transferring" in the data record.	Abort with system event.
3	The HMI device reads the values from the PLC and stores them in the data record specified in the function.	
4	• If "Yes" was selected for the "Overwrite" function, an existing data record is overwritten without any prompt for confirmation. The HMI device sets the status "Transfer completed." • If "No" was selected for the "Overwrite" function and the data record already exists, the HMI device aborts the job and enters 0000 1100 in the status word of the data mailbox.	
5	The control program must reset the status word to zero in order to enable further transfers.	

Writing to the PLC by means of configured function

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the function and the status "Transferring" in the data record.	Abort with system event.
3	The HMI device fetches the values of the data record specified in the function from the data storage medium and transfers the values to the PLC.	
4	The HMI device sets the status "Transfer completed."	
5	The control program can now evaluate the transferred data. The control program must reset the status word to zero in order to enable further transfers.	

See also

"Data mailbox" area pointer (Page 635)

Sequence of the transfer triggered by a PLC job (Basic Panels)

The transfer of data records between the HMI device and the PLC can be initiated by either one of these stations.

The two job mailboxes No. 69 and No. 70 are available for this type of transfer.

No. 69: Read data record from PLC ("PLC → DAT")

Job mailbox no. 69 transfers data records from the PLC to the HMI device. The job mailbox is structured as follows:

	Left byte (LB)	Right byte (RB)
Word 1	0	69
Word 2	Recipe number (1-999)	
Word 3	Data record number (1 to 65535)	
Word 4	Do not overwrite existing data record: 0: Overwrite existing data record: 1	

No. 70: Write data record to PLC ("DAT → PLC")

Job mailbox No. 70 transfers data records from the HMI device to the PLC. The job mailbox is structured as follows:

	Left byte (LB)	Right byte (RB)
Word 1	0	70
Word 2	Recipe number (1-999)	
Word 3	Data record number (1 to 65535)	
Word 4	—	

Sequence when reading from the PLC with job mailbox "PLC → DAT" (no. 69)

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the job and the status "Transferring" in the data record.	Abort without return message.
3	The HMI device reads the values from the PLC and saves these to the data record defined in the job mailbox.	
4	• If "Overwrite" was selected in the job, an existing data record is overwritten without any prompt for confirmation. The HMI device sets the status "Transfer completed." • If "Do not overwrite" was selected in the job, and the data record already exists, the HMI device aborts the job and enters 0000 1100 in the status word of the data mailbox.	
5	The control program must reset the status word to zero in order to enable further transfers.	

Sequence of writing to the PLC using job mailbox "DAT → PLC" (no. 70)

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the job and the status "Transferring" in the data record.	Abort without return message.

Step	Action	
3	The HMI device fetches the values of the data record specified in the function from the data storage medium and writes the values to the PLC.	
4	The HMI device sets the status "Transfer completed."	
5	The control program can now evaluate the transferred data. The control program must reset the status word to zero in order to enable further transfers.	

See also

"Data mailbox" area pointer (Page 635)

Sequence of a transfer started by the operator in the recipe display (Basic Panels)

Reading from the PLC started by the operator in the recipe view

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe number to be read and the status "Transferring" in the data mailbox and sets the data record number to 0.	Abort with system event.
3	The HMI device reads the values from the PLC and displays them in the recipe view. If the recipes have synchronized tags, the values from the PLC are also written to the tags.	
4	The HMI device sets the status "Transfer completed."	
5	The control program must reset the status word to zero in order to enable further transfers.	

Writing to the PLC started by the operator in the recipe view

Step	Action	
	Check: Status word = 0?	
1	Yes	No
	The HMI device enters the recipe and data record number to be written and the status "Transferring" in the data mailbox.	Abort with system event.
2	The HMI device writes the current values to the PLC. If the recipes have synchronized tags, the changed values are synchronized between the recipe view and tags and then written to the PLC.	
3	The HMI device sets the status "Transfer completed."	
4	If required, the control program can now evaluate the transferred data.	
5	The control program must reset the status word to zero in order to enable further transfers.	

Note

The status word may only be set by the HMI device. The PLC may only reset the status word to zero.

Note

The PLC may only evaluate the recipe and data record numbers when data inconsistency is detected if one of the conditions outlined below has been met:

- The data mailbox status is set to "Transfer completed".
- The data mailbox status is set to "Transfer completed with error".

See also

"Data mailbox" area pointer (Page 635)

Events, alarms, and acknowledgments (Basic Panels)

General information on operational messages, alarm messages and acknowledgments (Basic Panels)

Function

Alarms provide the user on the HMI device information about operating states or faults of the controller or the HMI device. The alarm texts consist of configurable texts and/or tags with actual values.

Alarms are basically divided into system alarms and fault alarms. The project engineer defines what constitutes a system alarm and a fault alarm.

System alarm

A system alarm indicates a status. Example:

- Motor switched on
- Controller in manual mode

Fault alarm

A fault alarm indicates an operational fault. Example:

- Valve does not open.
- Motor temperature too high

The fault alarms indicate exceptional operational states and must be acknowledged.

Acknowledgment

The fault alarms are acknowledged as follows:

- Operator input on the HMI device
- Sets an acknowledgment bit from the controller.

Alarm trigger

An alarm is triggered in the controller as follows:

- By setting a bit in a tag
- With a limit violation of a measured value

The location of the tag or tag array is defined in WinCC. The tag or array must be configured in the controller.

See also

Step 1: Creating tags or an array (Page 643)

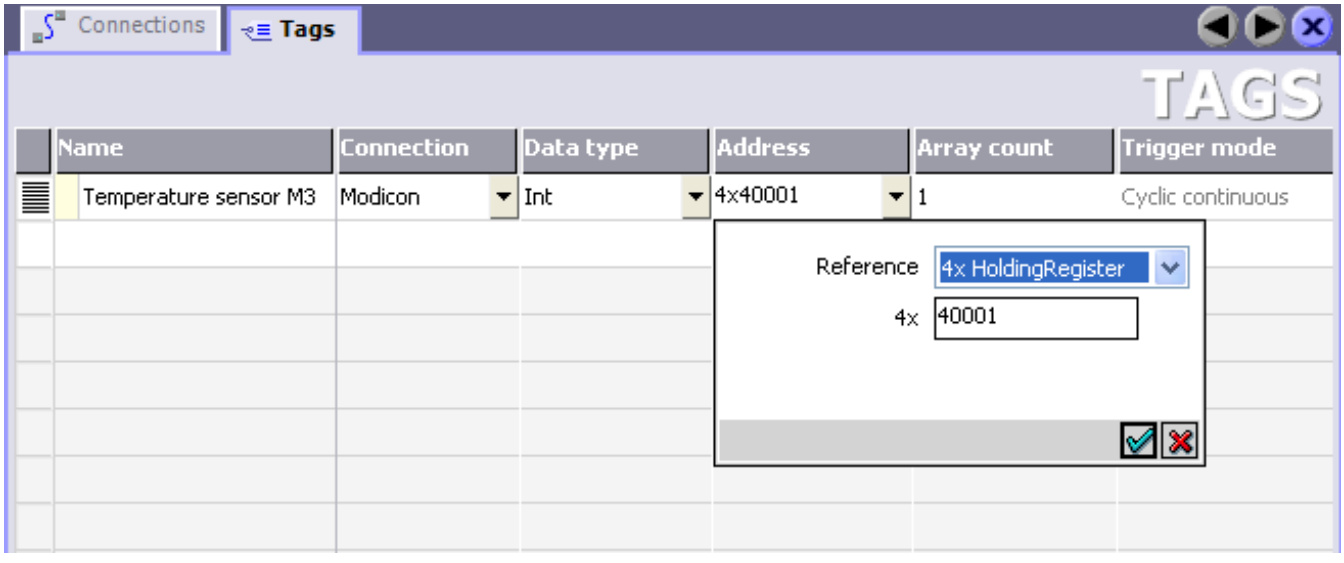
Step 2: Configuring an alarm (Page 644)

Step 3: Configuring the acknowledgment (Page 646)

Step 1: Creating tags or an array (Basic Panels)

Procedure

You create tags or arrays in the "Tags" editor. The dialog is shown in the following figure.



- Specify the name of the tag or array.
- Check the connection to the PLC.
  The connection must be configured in the "Connections" editor.
- Select the data type.
  The data types available for selection depend on the controller. If an invalid data type is selected, this has the result that the tag will not be available in the "Discrete alarms" and "Analog alarms" editors.
  The following data types are supported for standard Modbus controllers:

Controller	Permitted data types	
	Discrete alarms	Analog alarms
All standard Modbus series	Int, +/-Int	Bit, 16-bit group, Int, +/-Int, Double, +/-Double, Float

- Enter an address.
  The addressed tag contains the bit that triggers the alarm.
  As soon as the bit of the tag is set in the controller and transmitted to the HMI device in the configured acquisition cycle, it recognizes the alarm as "incoming".
  Conversely, the HMI device recognized the alarm as "outgoing" after resetting the same bit in the PLC.
- Select the array elements.
  If the number of array elements is increased, you can select correspondingly more bit numbers in the "Discrete alarms" editor. For example, 48 alarm bits are available for an array with 3 words.

See also

General information on operational messages, alarm messages and acknowledgments (Page 642)

Step 2: Configuring an alarm (Page 644)

Step 2: Configuring an alarm (Basic Panels)

Procedure

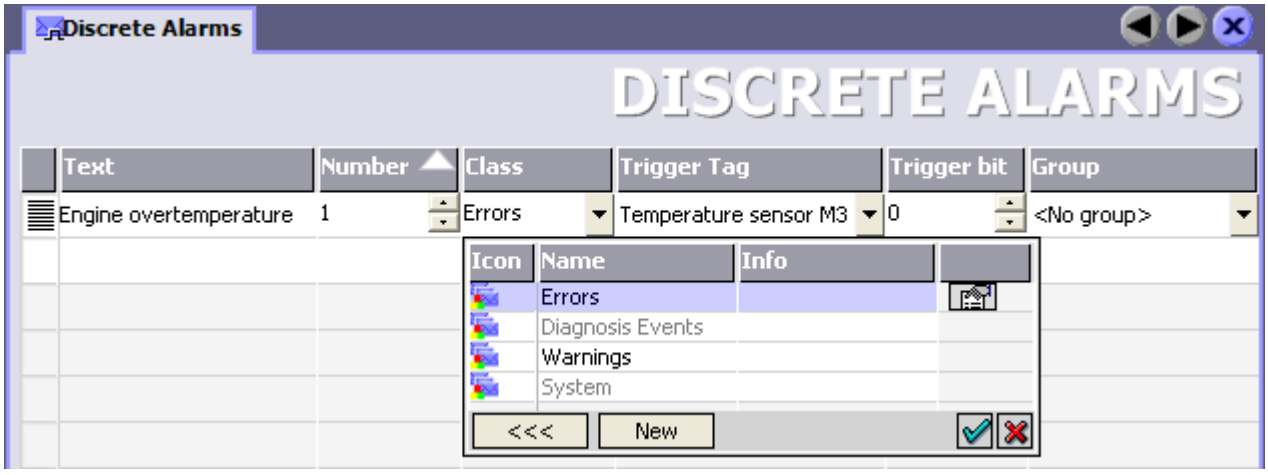
We differentiate between the following alarms:

- Discrete alarms
- Analog alarms

Alarms are created in the "Discrete alarms" and "Analog alarms" editors.

Discrete alarms

The editor is shown in the following figure.



- Edit text
  Enter the text that is displayed in Runtime. The text can be formatted character by character and contain fields for outputting tags.
  For example, the text appears in the alarm display if an alarm view has been configured in the "Screens" editor.
- Set number
  Each alarm has a number that must be unique in the project. It is used to uniquely identify the alarm and is generated in Runtime.
  Permissible values are between 1 and 100,000.
  The alarm number is assigned sequentially by WinCC. You can change them if required, for example, if you want to divide the numbers into groups.
- Specify alarm classes
  Available alarm classes are:
  - Fault alarms
    This class must be acknowledged.
  - System alarms
    This class signals events with incoming and outgoing alarms.
- Assign trigger tag
  In the trigger tag column, you link the configured alarm with the tag created in step 1. The selection list contains all tags with valid data types.
- Set bit number
  In the "Bit number" column, specify the relevant bit position in the created tag.
  It must be noted here that the way the bit position is counted does **not** depend on the controller. For standard Modbus controllers, the following assignment applies:

How the bit positions are counted	Left byte								Right byte							
In controllers of the series 984, Compact, Quantum and Momentum	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
In WinCC and CPUs of the Premium and Micro series, configure:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Analog alarms

Analog alarms only differ from discrete alarms in that a limit is configured instead of a bit number. If this limit is violated, the alarm is triggered. The outgoing alarm is triggered by a violation of the low limit, and perhaps with allowances for any configured hysteresis.

See also

General information on operational messages, alarm messages and acknowledgments (Page 642)

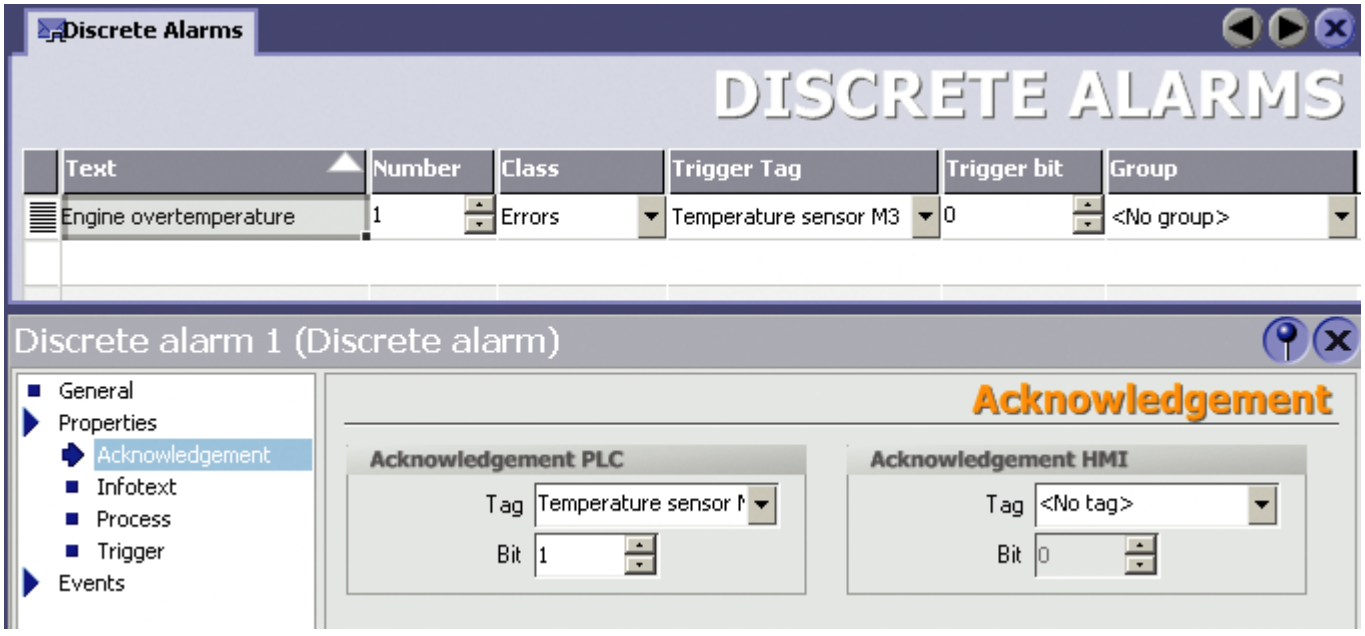
Step 3: Configuring the acknowledgment (Page 646)

Step 3: Configuring the acknowledgment (Basic Panels)

Procedure

Create suitable tags on the PLC to acknowledge an error alarm. You assign these tags to an alarm in the "Bit messages" editor. You make the assignment in "Properties > Acknowledgment".

The following figure shows the dialog for configuring an acknowledgment.



Distinction in terms of acknowledgment:

- Acknowledgment on the HMI device
- Acknowledgment by the PLC

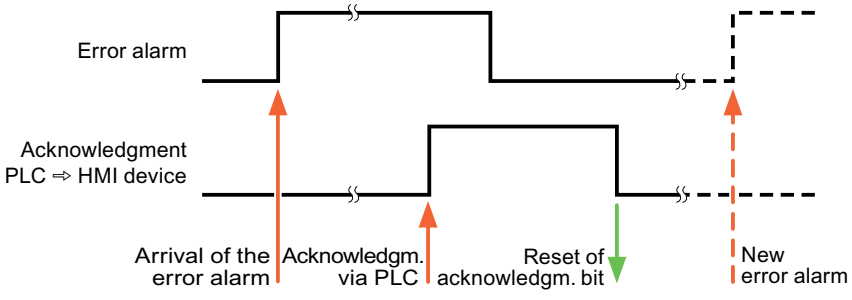
Acknowledgment by the PLC

In "Acknowledgment PLC tag", you configure the tag or the array tag and the bit number based on which the HMI device can recognize an acknowledgment by the PLC.

A bit set in the tag triggers acknowledgment of the assigned error alarm bit at the HMI device. This tag bit returns a function similar to acknowledgment on the HMI device which is triggered by pressing the "ACK" button, for example.

The acknowledgment bit must be located in the same tag as the bit for the error alarm.

Reset the acknowledgment bit before setting the bit in the alarm area again. The figure below shows the pulse diagram.



Acknowledgment on the HMI device

In "Ack read tag", you configure the tag or the array tag and the bit number that is written to the PLC after acknowledgment from the HMI device. Make sure when you use an array tag that it is not longer than 6 words.

To always create a signal change when setting an assigned acknowledgment bit of a discrete alarm that must be acknowledged, the HMI device will reset the acknowledgment bit assigned to the alarm as soon as it detects an alarm subject to acknowledgment and write the acknowledgment tag in the PLC. There will be a certain delay between detecting the message and writing the acknowledgment tag in the PLC because the HMI device has to process the operations.

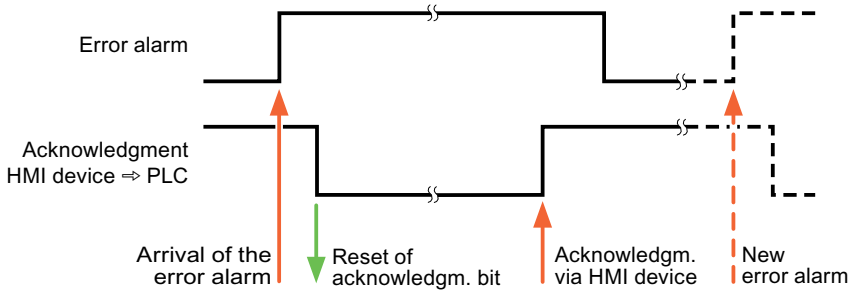
If a discrete alarm subject to acknowledgment is acknowledged by the HMI device, then the corresponding bit in the assigned acknowledgment tag will be set. The entire acknowledgment tag is then written to the PLC by the HMI device. This allows the PLC to recognize that a certain alarm message has been acknowledged at the HMI device.

Note

All alarm bits acknowledged since the last Runtime start will remain in the acknowledgment tag until a new incoming of the respective discrete alarms is detected.

This area should only be read by the PLC because the entire section of the HMI device will be overwritten once the next acknowledgment tag is written.

The figure below shows the pulse diagram.



See also

General information on operational messages, alarm messages and acknowledgments (Page 642)

CSP_6 (Basic Panels)

CSP_7 (Basic Panels)

CSP_8 (Basic Panels)

CSP_9 (Basic Panels)

CSP_10 (Basic Panels)

1.13.4 Data exchange using area pointers (Basic Panels)

1.13.4.1 General information on area pointers (Basic Panels)

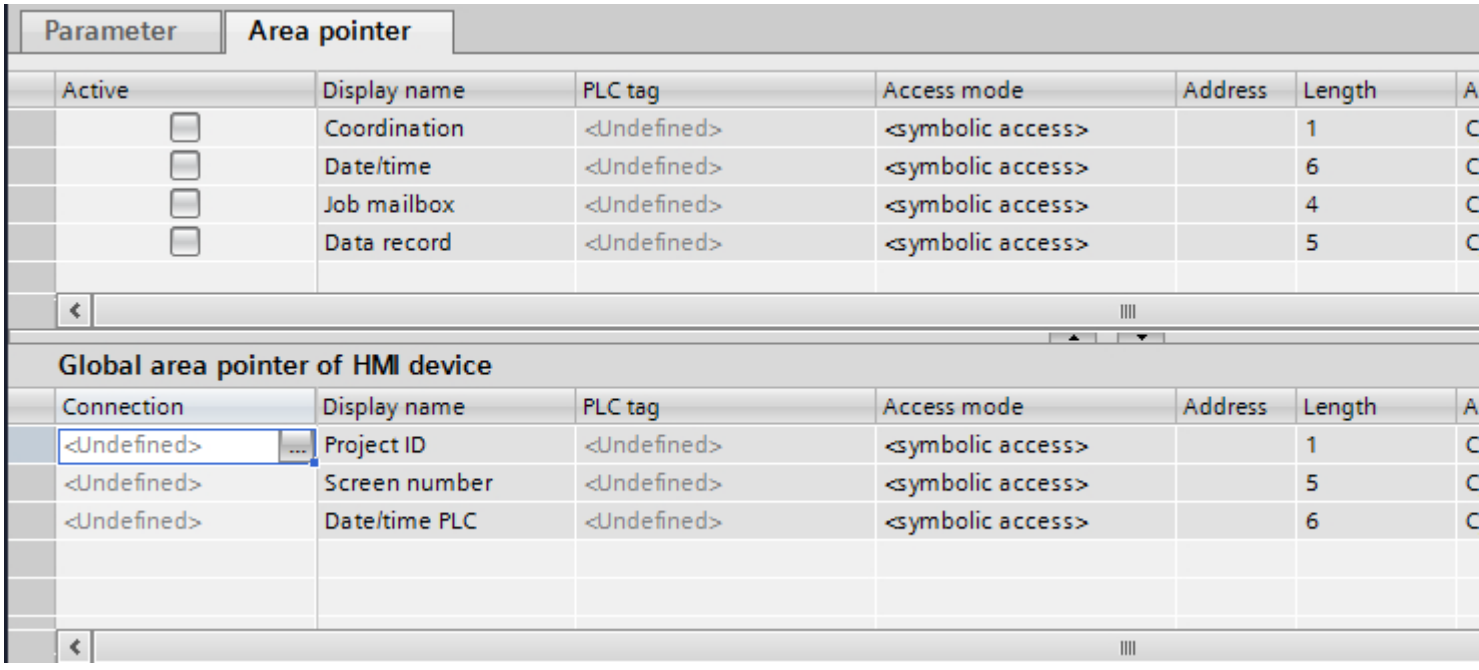
Introduction

You use an area pointer to access a data area in the PLC. During communication, the PLC and the HMI device alternately access these data areas for read and write operations.

The PLC and the HMI device trigger defined interactions based on the evaluation of stored data.

Configuration of area pointers

Before you use the area pointer, you enable it in "Connections ▸ Area pointers". You then assign the area pointer parameters.



- Active
  Enables the area pointer.
- Pointer name
  Name of the area pointer specified by WinCC.
- PLC tag
  Here you select the PLC tag or the tag array that you have configured as the data area for the area pointer.
- Address
  No address is entered into this field because of the symbolic access.
- Length
  WinCC specifies the length of the area pointer.
- Acquisition cycle
  You specify the acquisition cycle in this field for area pointers that are read by the HMI device. Note that a very short acquisition time may have a negative impact on HMI device performance.
- Comment
  Enter a comment, for example, to describe the purpose of the area pointer.

See also

Accessing data areas (Page 651)
"Screen number" area pointer (Page 651)
"Date/time" area pointer (Page 652)
"Date/time PLC" area pointer (Page 653)
"Coordination" area pointer (Page 654)
"Project ID" area pointer (Page 655)
"PLC job" area pointer (Page 656)
"Data mailbox" area pointer (Page 659)

1.13.4.2 Accessing data areas (Basic Panels)

Accessing data areas

The following table shows how HMI devices and PLCs access individual data areas for read (R) or write (W) operations.

Data area	Required for	HMI device	PLC
Screen number	Evaluation by the PLC in order to determine the active screen.	W	R
Data record	Transfer of data records with synchronization	R/W	R/W
Date/time	Transfer of the date and time from the HMI device to the PLC	W	R
Date/time PLC	Transfer of the date and time from the PLC to the HMI device	R	W
Coordination	Requesting the HMI device status in the PLC program	W	R
Project ID	Runtime checks for consistency between the WinCC project ID and the project in the PLC	R	W
Job mailbox	Triggering of HMI device functions by the PLC program	R/W	R/W

See also

General information on area pointers (Page 649)

1.13.4.3 "Screen number" area pointer (Basic Panels)

Function

The HMI devices store information about the screen called up on the HMI device in the "Screen number" area pointer.

This allows the transfer of the current screen contents from the HMI device to the PLC. The PLC can trigger specific reactions such as the call of a different screen.

Use

Configure and enable the area pointer in "Communication > Connections" before you put it into use. You can create only **one** instance of the "Screen number" area pointer and only on **one** PLC.

The screen number is always transferred to the PLC when a new screen is activated or when the focus within a screen changes from one screen object to another.

Structure

The area pointer is a data area in the memory of the PLC with a fixed length of 5 words.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st word	Current screen type															
2nd word	Current screen number															
3rd word	Reserved															
4th word	Current field number															
5th word	Reserved															

- Current screen type
  "1" for root screen or
  "4" for permanent area
- Current screen number
  1 to 32767
- Current field number
  1 to 32767

See also

General information on area pointers (Page 649)

1.13.4.4 "Date/time" area pointer (Basic Panels)

Function

This area pointer is used to transfer the date and time from the HMI device to the PLC.

The PLC writes control job "41" to the job mailbox.

When it evaluates the control job, the HMI device writes its current date and the time in the data area configured in the "Date/time" area pointer. All definitions are coded in BCD format.

The "Date/Time" area pointer when used in a project which contains multiple connections must be enabled for each configured connection.

Note

You cannot use the "Date/Time PLC" area pointer if you have configured the "Date/Time" area pointer.

The date/time data area has the following structure:

Data word	Left byte								Right byte								
	15							8	7							0	
n+0	Reserved								Hour (0 to 23)								Time
n+1	Minute (0 to 59)								Second (0 to 59)								
n+2	Reserved								Reserved								
n+3	Reserved								Weekday (1 to 7, 1=Sunday)								Date
n+4	Day (1 to 31)								Month (1 to 12)								
n+5	Year (80 to 99/0 to 29)								Reserved								

Note

When making entries in the "Year" data area, you should note that values 80 to 99 result in years 1980 through 1999, while the values 0 to 29 result in the years 2000 through 2029.

See also

General information on area pointers (Page 649)

1.13.4.5 "Date/time PLC" area pointer (Basic Panels)

Function

This area pointer is used to transfer the date and time from the PLC to the HMI device. Use this area pointer if the PLC is the time master.

The PLC loads the data area of the area pointer. All definitions are coded in BCD format.

The HMI device reads the data cyclically within the configured acquisition cycle and synchronizes itself.

Note

Set an acquisition cycle of sufficient length for the Date/time area pointer in order to avoid any negative impact on HMI device performance.
Recommended: Acquisition cycle of 1 minute if your process can handle it.

"Date/Time PLC" is a global area pointer and may be configured only once per project.

Note

You cannot use the "Date/Time" area pointer if you have configured the "Date/Time PLC" area pointer.

The "Date/time PLC" data area has the following structure:

Data word	Left byte			Right byte		
	15	. . . . .	8	7	. . . . .	0
n+0	Year (80 to 99/0 to 29)			Month (1 to 12)		
n+1	Day (1 to 31)			Hour (0 to 23)		
n+2	Minute (0 to 59)			Second (0 to 59)		
n+3	Reserved			Reserved		Weekday (1 to 7, 1=Sunday)
n+4 [1)]	Reserved			Reserved		
n+5 [1)]	Reserved			Reserved		

1) The two data words must exist in the data area to ensure that the data format matches WinCC and to avoid reading false information.

Note

When making entries in the "Year" data area, you should note that values 80 to 99 result in years 1980 through 1999, while the values 0 to 29 result in the years 2000 through 2029.

See also

General information on area pointers (Page 649)

1.13.4.6 "Coordination" area pointer (Basic Panels)

Function

The "Coordination" area pointer is used to implement the following functionality:

- Detecting the startup of the HMI device in the control program
- detection in the control program of the current HMI device operating mode
- detection in the control program of the HMI devices ready to communicate state

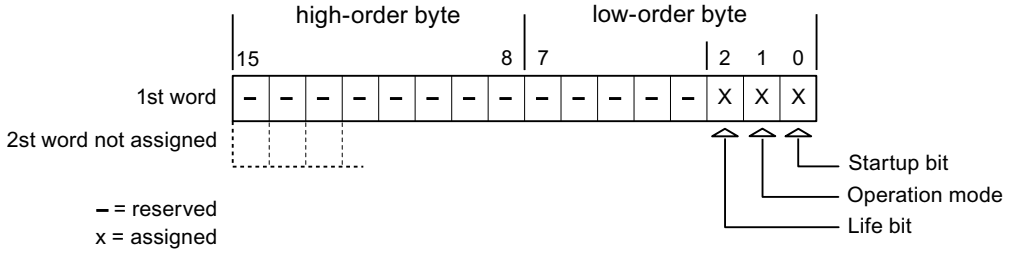
The "Coordination" area pointer has a length of one word.

Use

Note

The HMI device always writes the entire coordination area when updating the area pointer. The control program may not make changes to the coordination area for this reason.

Assignment of bits in the "Coordination" area pointer



Startup bit

The startup bit is set briefly to "0" by the HMI device during startup. It sets the bit permanently to "1" when startup is completed.

Operating mode

The operating mode bit is set to 1 as soon as the user switches the HMI device offline. The state of the operating mode bit is "0" during normal operation of the HMI device. You can determine the current operating mode of the HMI device by reading this bit.

Life bit

The HMI device inverts the life bit at intervals of approximately one second. You can check whether or not the connection to the HMI device is still up by querying this bit in the control program.

See also

General information on area pointers (Page 649)

1.13.4.7 "Project ID" area pointer (Basic Panels)

Function

You can check whether the HMI device is connected to the correct PLC at the start of Runtime. This check is important when operating with several HMI devices.

The HMI device compares a value stored on the PLC with the value specified in configuration. This ensures compatibility of configuration data with the control program. If discrepancy is detected, a system event is displayed on the HMI device and Runtime is stopped.

Application

To use this area pointer, set up the following during the configuration:

- Define the version of configuration. Possible values between 1 and 255.
  You enter the version in the editor "Runtime settings > General" in the "Identification" area.
- Data address of the value for the version that is stored in the PLC:
  You enter the data address in the editor "Communication > Connections".

Connection failure

A connection failure to a device on which the "Project ID" area pointer is configured results in all the other connections of the device being switched to "offline".

This behavior has the following prerequisites:

- You have configured several connections in a project.
- You are using the "project ID" area pointer in at least one connection.

Causes which may set connections "offline":

- The PLC is not available.
- The connection has been switched offline in the engineering system.

See also

General information on area pointers (Page 649)

1.13.4.8 "PLC job" area pointer (Basic Panels)

Function

The PLC can use the job mailbox to transfer jobs to the HMI device to trigger corresponding actions on the HMI device. These functions include, for example:

- Display screen
- Set date and time

Data structure

The first word of the job mailbox contains the job number. Depending on the job mailbox, up to three parameters can be transferred.

Word	Left byte	Right byte
n+0	0	Job number
n+1	Parameter 1	
n+2	Parameter 2	
n+3	Parameter 3	

The HMI device evaluates the job mailbox if the first word of this job is not equal to zero. This means that the parameters must be entered in the job mailbox first, followed by the job number.

When the HMI device accepts the job mailbox, the first word is set to 0 again. The execution of the job mailbox is generally not completed at this point in time.

Job mailboxes

All job mailboxes and their parameters are listed below. The "No." column contains the job number of the job mailbox. Job mailboxes can only be triggered by the PLC when the HMI device is online.

Note

Please note that not all HMI devices support job mailboxes.

No.	Function	
14	**Setting the time (BCD coded)**	
	Parameter 1	Left byte: - Right byte: hours (0-23)
	Parameter 2	Left byte: minutes (0-59) Right byte: seconds (0-59)
	Parameter 3	-
15	**Setting the date (BCD coded)**	
	Parameter 1	Left byte: - Right byte: weekday (1-7: Sunday-Saturday)
	Parameter 2	Left byte: day (1-31) Right byte: month (1-12)
	Parameter 3	Left byte: year
23	**User logon**	
	Logs the user on with the name "PLC user" at the HMI device with the group number transferred in Parameter 1. The logon is possible only when the transferred group number exists in the project.	
	Parameter 1	Group number 1 to 255
	Parameter 2, 3	-

No.	Function	
14	**Setting the time (BCD coded)**	
24	**User logoff**	
	Logs off the current user. (The function corresponds to the "logoff" system function)	
	Parameter 1, 2, 3	-
40	**Transfer date/time to PLC**	
	(in the S7 format DATE_AND_TIME) An interval of at least 5 seconds must be maintained between two successive jobs to prevent overload of the HMI device.	
	Parameter 1, 2, 3	-
41	**Transfer date/time to PLC**	
	(In OP/MP format) An interval of at least 5 seconds must be maintained between successive jobs in order to prevent overload of the HMI device.	
	Parameter 1, 2, 3	-
46	**Update tags**	
	Causes the HMI device to read the current value of the tags•from the PLC whose update ID matches the value transferred in parameter 1. (Function corresponds to the "UpdateTag" system function.)	
	Parameter 1	1 - 100
49	**Delete alarm buffer**	
	Deletes all analog alarms and discrete alarms of the "Warnings" class from the alarm buffer.	
	Parameter 1, 2, 3	-
50	**Delete alarm buffer**	
	Deletes all analog alarms and discrete alarms of the "Errors" class from the alarm buffer.	
	Parameter 1, 2, 3	-
51	**Screen selection** [1]	
	Parameter 1	Screen number
	Parameter 2	-
	Parameter 3	Field number
69	**Read data record from PLC**	
	Parameter 1	Recipe number (1-999)
	Parameter 2	Data record number (1-65535)
	Parameter 3	0: Do not overwrite existing data record 1: Overwrite existing data record
70	**Write data record to PLC**	
	Parameter 1	Recipe number (1-999)
	Parameter 2	Data record number (1-65535)
	Parameter 3	-

[1] OP 73, OP 77A and TP 177A HMI devices also execute the "Screen selection" job mailbox if the on-screen keyboard is active.

See also

General information on area pointers (Page 649)

1.13.4.9 "Data record" area pointer (Basic Panels)

"Data mailbox" area pointer (Basic Panels)

Function

When data records are transferred between the HMI device and PLC, both partners access common communications areas on the PLC.

Data transfer types

There are two ways of transferring data records between the HMI device and PLC:

- Transfer without synchronization
- Transfer with synchronization over the data record

Data records are always transferred directly. That is, the tag values are read from an address or written to an address configured for this tag directly, without redirecting the values by means of interim memory.

Initiating the transfer of data records

There are three ways of triggering the transfer:

- Operator input in the recipe view
- Job mailboxes
  The transfer of data records can also be triggered by the PLC.
- Triggering by configured functions

If the transfer of data records is triggered by a configured function or by a job mailbox, the recipe view on the HMI device remains operable. The data records are transferred in the background.

Simultaneous processing of several transfer requests is, however, not possible. In this case, the HMI device rejects the other transfer requests with a system event.

See also

General information on area pointers (Page 649)

Transfer without synchronization (Page 660)

Transfer with synchronization (Page 660)

Sequence of a transfer started by the operator in the recipe display (Page 662)

Sequence of the transfer triggered by a PLC job (Page 663)

Sequence of the transfer when triggered by a configured function (Page 664)

Possible causes of error when transferring data records (Page 665)

Transfer without synchronization (Basic Panels)

If you select asynchronous transfer of data records between the HMI device and PLC, there is no coordination over the common data areas. It is therefore unnecessary to set up a data area during configuration.

Asynchronous data record transfer can be a useful alternative, for example, when:

- The system is capable of excluding the risk of uncontrolled overwriting of data by the communication peer.
- The PLC does not require information about the recipe number and data record number.
- The transfer of data records is triggered by the operator of the HMI device.

Reading values

When a read job is triggered, the values are read from the PLC addresses and transferred to the HMI device.

- Triggering by the operator in the recipe view:
  The values are downloaded to the HMI device. You can then process, edit, or save these values, for example.
- Triggering by a function or job mailbox:
  The values are saved immediately to the data volume.

Writing values

When a write job is triggered, the values are written to the PLC addresses.

- Triggering by the operator in the recipe view:
  The current values are written to the PLC.
- Triggering by a function or job mailbox:
  The current values are written to the PLC from the data medium.

See also

"Data mailbox" area pointer (Page 659)

Transfer with synchronization (Basic Panels)

If you select synchronous transfer, both communication partners set status bits in the common data area. You can use this mechanism to prevent uncontrolled overwriting of data in either direction in your control program.

Application

Synchronous data record transfer can be a useful solution, for example, when:

- The PLC is the "active partner" in the transfer of data records.
- The PLC evaluates the information about the recipe number and data record number.
- The transfer of data records is triggered by means of a Job mailbox.

Requirements

In order to synchronize transfer of data records between the HMI device and the PLC, the following requirements must be met during configuration:

- An area pointer has been set up: "Communication > Connections" editor in "Area pointer".
- The PLC with which the HMI device synchronizes transfer of data records is specified in the recipe:
  "Recipes" editor in the inspector window the option "Coordinated transfer of data records" under "General > Synchronization > Settings"

Structure of the data area

The data area has a fixed length of 5 words. Structure of the data area:

	15		0
1. Word	Current recipe number (1 - 999)		
2. Word	Current data record number (0 - 65535)		
3. Word	Reserved		
4. Word	Status (0, 2, 4, 12)		
5. Word	Reserved		

- Status
  The status word (word 4) can adopt the following values:

Value		Meaning
Decimal	Binary	
0	0000 0000	Transfer permitted, data record free
2	0000 0010	Transferring.
4	0000 0100	Transfer completed without error
12	0000 1100	Transfer completed with error

See also

"Data mailbox" area pointer (Page 659)

Sequence of a transfer started by the operator in the recipe display (Basic Panels)

Reading from the PLC started by the operator in the recipe view

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe number to be read and the status "Transferring" in the data record and sets the data record number to 0.	Abort with system event.
3	The HMI device reads the values from the PLC and displays them in the recipe view. If the recipes have synchronized tags, the values from the PLC are also written to the tags.	
4	The HMI device sets the status "Transfer completed."	
5	The control program must reset the status word to zero in order to enable further transfers.	

Writing to the PLC started by the operator in the recipe view

Step	Action	
	Check: Status word = 0?	
1	Yes	No
	The HMI device enters the recipe and data record number to be written and the status "Transferring" in the data record.	Abort with system event.
2	The HMI device writes the current values to the PLC. If the recipes have synchronized tags, the changed values are synchronized between the recipe view and tags and then written to the PLC.	
3	The HMI device sets the status "Transfer completed."	
4	If required, the control program can now evaluate the transferred data.	
5	The control program must reset the status word to zero in order to enable further transfers.	

Note

The status word may only be set by the HMI device. The PLC may only reset the status word to zero.

Note

The PLC may only evaluate the recipe and data record numbers when data inconsistency is detected if one of the conditions outlined below has been met:

- The data mailbox status is set to "Transfer completed".
- The data mailbox status is set to "Transfer completed with error".

See also

"Data mailbox" area pointer (Page 659)

Sequence of the transfer triggered by a PLC job (Basic Panels)

The transfer of data records between the HMI device and the PLC can be initiated by either one of these stations.

The two job mailboxes No. 69 and No. 70 are available for this type of transfer.

No. 69: Read data record from PLC ("PLC → DAT")

Job mailbox no. 69 transfers data records from the PLC to the HMI device. The job mailbox is structured as follows:

	Left byte (LB)	Right byte (RB)
Word 1	0	69
Word 2	Recipe number (1-999)	
Word 3	Data record number (1 to 65535)	
Word 4	Do not overwrite existing data record: 0 Overwrite existing data record: 1	

No. 70: Write data record to PLC ("DAT → PLC")

Job mailbox No. 70 transfers data records from the HMI device to the PLC. The job mailbox is structured as follows:

	Left byte (LB)	Right byte (RB)
Word 1	0	70
Word 2	Recipe number (1-999)	
Word 3	Data record number (1 to 65535)	
Word 4	—	

Sequence when reading from the PLC with job mailbox "PLC → DAT" (no. 69)

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the job and the status "Transferring" in the data record.	Abort without return message.
3	The HMI device reads the values from the PLC and saves these to the data record defined in the job mailbox.	

Step	Action	
4	• If "Overwrite" was selected in the job, an existing data record is overwritten without any prompt for confirmation. The HMI device sets the status "Transfer completed." • If "Do not overwrite" was selected in the job, and the data record already exists, the HMI device aborts the job and enters 0000 1100 in the status word of the data record.	
5	The control program must reset the status word to zero in order to enable further transfers.	

Sequence of writing to the PLC with job mailbox "DAT → PLC" (no. 70)

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the job and the status "Transferring" in the data record.	Abort without return message.
3	The HMI device fetches the values of the data record specified in the function from the data medium and writes the values to the PLC.	
4	The HMI device sets the status "Transfer completed."	
5	The control program can now evaluate the transferred data. The control program must reset the status word to zero in order to enable further transfers.	

See also

"Data mailbox" area pointer (Page 659)

Sequence of the transfer when triggered by a configured function (Basic Panels)

Reading from the PLC using a configured function

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the function and the status "Transferring" in the data record.	Abort with system event.
3	The HMI device reads the values from the PLC and stores them in the data record specified in the function.	

Step	Action	
4	• If "Yes" was selected for the "Overwrite" function, an existing data record is overwritten without any prompt for confirmation. The HMI device sets the status "Transfer completed." • If "No" was selected for the "Overwrite" function and the data record already exists, the HMI device aborts the job and enters 0000 1100 in the status word of the data record.	
5	The control program must reset the status word to zero in order to enable further transfers.	

Writing to the PLC by means of configured function

Step	Action	
1	Check: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe and data record number specified in the function and the status "Transferring" in the data record.	Abort with system event.
3	The HMI device fetches the values of the data record specified in the function from the data medium and transfers the values to the PLC.	
4	The HMI device sets the status "Transfer completed."	
5	The control program can now evaluate the transferred data. The control program must reset the status word to zero in order to enable further transfers.	

See also

"Data mailbox" area pointer (Page 659)

Possible causes of error when transferring data records (Basic Panels)

Possible causes of error

The section below shows possible error causes which lead to the cancellation of data record transfer:

- Tag address not set up on the PLC
- Overwriting data records not possible

- Recipe number does not exist
- Data record number does not exist

Note

The status word may only be set by the HMI device. The PLC may only reset the status word to zero.

Note

The PLC may only evaluate the recipe and data record numbers when data inconsistency is detected if one of the conditions outlined below has been met:

- The data mailbox status is set to "Transfer completed".
- The data mailbox status is set to "Transfer completed with error".

Reaction to an aborted transfer due to errors

If the transfer of data records is aborted due to errors, the HMI device reacts as follows:

- Triggering by the operator in the recipe view
  Information in the status bar of the recipe view and output of system alarms
- Triggered by function
  Output of system alarms
- Triggering by job mailbox
  No return message on the HMI device

You can nonetheless evaluate the status of the transfer by querying the status word in the data record.

See also

"Data mailbox" area pointer (Page 659)

Index

A

Absolute Addressing
  of a tag, 58
Access level, 88, 105, 298
Accessible devices, 96, 160, 223, 288, 345
Acknowledgment, 642
Acquisition cycle
  Area pointer, 64
Active
  Area pointer, 64
Address
  Area pointer, 64
  Mitsubishi, 499
  Omron Host Link, 556
Addressing
  Allen-Bradley, 455, 475
  Allen-Bradley Ethernet IP, 451
  Ethernet/IP, 455, 475
  MPI, 240
  Valid, 457
Alarm message
  Acknowledgment by the PLC, 127, 191, 258, 322, 373, 424, 435, 485, 514, 543, 561, 647
  Acknowledgment on the HMI device, 128, 192, 259, 323, 374, 425, 436, 486, 514, 544, 562, 648
  Configure acknowledgment, 127, 191, 258, 322, 373, 424, 435, 485, 514, 543, 561
  Configuring the acknowledgment, 646
Alarms
  Configuring, 126, 190, 257, 321, 372, 423, 434
  Data types, 126, 190, 257, 321, 372, 423, 434
  Modicon Modbus, 542
  Non-integrated connection, 542
  Restriction, 513
Allen-Bradley, 439, 465, 467
  Allen-Bradley DF1 communication driver, 465, 467
  Analog alarm, 484
  Area pointer, 482
  Basic Panel, 74
  Communication drivers, 441
  Communication partners, 582
  Data type, 484
  DF1, 441
  Ethernet/IP, 457
  Mitsubishi, 513
Allen-Bradley DF1
  Configuring a connection, 461
  Connection, 461, 465
  Connection parameters, 463
  CPU type, 475
  KF2 module, 465, 467
  KF3 module, 465, 467
  Valid data type, 473
Allen-Bradley DH485
  Connector, 583
  Device-dependent parameters, 587
  Installing communication drivers, 586
  Multi-point connection, 584
  Network parameters, 587
  Optimizing the configuration, 589
  Permitted data type, 588
  PLC-dependent parameters, 588
  Point-to-point connection, 583
  Protocol parameters, 587
  Selecting the PLC, 586
Allen-Bradley Ethernet IP
  Address multiplexing, 455
  Addressing, 451
  Addressing type, 453
Allen-Bradley EtherNet/IP
  Configuring a connection, 442
  Connection, 442, 447
  Connection parameters, 444
  Data type, 448
Analog alarm
  Allen-Bradley, 484
  Mitsubishi, 513
  Omron, 560
Area pointer, 61, 110, 173, 241, 303, 353, 407, 626, 649
  Acquisition cycle, 64
  Active, 64
  Address, 64
  Allen-Bradley, 482
  Basic Panel, 76
  Comment, 64
  Connection, 62, 69
  Connections editor, 63
  Coordination, 631, 654
  Data record, 119, 183, 249, 313, 363, 415, 635, 659
  Date/time, 173, 176, 353, 355, 629
  Date/time PLC, 630

Job mailbox, 117, 181, 247, 311, 360, 413, 633, 656
Length, 64
Modicon Modbus, 541
PLC tag, 64
Pointer name, 64
Project ID, 116, 180, 246, 309, 359, 412, 632, 655
Screen number, 111, 175, 178, 242, 304, 358, 407, 628, 651
Tab, 64
Area pointers
Configuring, 66
Coordination, 115, 177, 245, 308, 356, 411
Creating, 66
Date/time, 112, 243, 305, 408, 652
Date/time PLC, 113, 244, 306, 409, 653
Length, 67
Mitsubishi, 511
Omron, 559
Asynchronous
Transferring data, 120, 184, 250, 314, 364, 420, 636, 660
Automation system, 10, 36
Setting up, 11
Autonegotiation
Disabled, 94, 158, 222, 287, 343

B

Basic Panel
Area pointer, 76
Communication drivers, 74
ETHERNET, 75
IF1B, 75
Interface, 75
Mitsubishi, 74
Modicon Modbus, 74
Omron, 74
OPC, 74
Boundaries, 96, 160, 223, 288, 345

C

Cabling rules (PROFINET), 94, 158, 222, 287, 343
Change bit order
Standard MODBUS TCP/IP, 606, 613
Change word order
Modicon MODBUS TCP/IP, 520
Standard MODBUS TCP/IP, 606, 613
CJ1, 556
CJ2, 556
Comment
Area pointer, 64
Communication, 9
Basics, 10
Definition, 9
ET 200 CPU, 331
S7 1200, 140
S7 1500, 77
S7 200, 396
S7 300, 209
S7 400, 209
SIMATIC LOGO!, 428
SIMATIC S7-1500 Software Controller, 275
Third-party drivers, 439
using area pointers, 15
Communication drivers, 16
Allen-Bradley, 441
Basic Panel, 74
Mitsubishi, 486
Modicon Modbus, 515
Omron, 545
Communication network, 36
PROFINET, 37
Communication partners
Allen-Bradley, 582
HMI device, 9
Networking, 41
PLC, 9
SIMATIC 505, 564
SIMATIC S7, 135, 203, 272, 389, 403, 406
Communication types
Allen-Bradley, 447
Connectable PLCs, 523, 534
Enabled, 493, 506, 522, 533, 550
EtherNet/IP, 447
Restriction, 522, 533
Communications principle, 566, 601
Compact, 525, 538
CompactLogix, 450
Components
Commissioning, 459, 481, 501, 510, 527, 539, 558
Configure acknowledgment
Alarm message, 127, 191, 258, 322, 373, 424, 435, 485, 514, 543, 561
Configure alarms
Allen-Bradley, 484
Data types, 513
Non-integrated connection, 512
Special considerations, 513

Configuring
Alarms, 126, 190, 321, 372, 423, 434
Connections, 262, 265, 267, 396, 429
Configuring a connection
Allen-Bradley DF1, 461
Allen-Bradley EtherNet/IP, 442
Mitsubishi FX, 503
Mitsubishi MC TCP/IP, 487
Modicon Modbus RTU, 528
Modicon Modbus TCP, 516
Omron Host Link, 546
Configuring alarms
Modbus, 644
Configuring the acknowledgment
Alarm message, 646
Configuring the network with Ethernet, 150, 218, 340
Creating private subnets, 151, 219, 341
Linking networks, 152, 220, 341
Relationship between IP address and subnet mask, 151, 219, 341
Setting the IP address, 150, 218, 340
Setting the subnet mask, 150, 218, 340
Connecting
HMI device with PLC, 592, 610
PLC, 447, 492, 506, 521, 550
Connecting cable
6XV1440 - 2P for Mitsubishi PG protocol, 507
Allen-Bradley cable 1761-CBL-PM02, 593
Allen-Bradley cable 1784-CP10, 471
Mitsubishi FX, 507
PP4 connecting cable, 594
Connecting cable 9-pole
Sub D, RS 422, 470
Connecting cables
Allen-Bradley cable 1747-CP3, 592
Modbus, 603
MP1 connecting cable, 596
Point-to-point cable 1,
Point-to-point cable 2, 536, 621
Point-to-point cable 3,
SIWAREX, 623
Connection, 14
Allen-Bradley DF1, 461, 465
Allen-Bradley EtherNet/IP, 442
Area pointer, 62, 69
Configuring, 42, 46, 50, 442, 461, 487, 503, 516, 528, 546
Creating, 42, 46, 50
Highlighting, 44
Integrated, 42
integrated connection, 58
Mitsubishi MC TCP/IP, 487
Modicon, 532
Modicon Modbus RTU, 528
Modicon Modbus TCP, 516
offline, 116, 180, 246, 310, 359, 412
Omron communication drivers, 559
Omron Host Link, 546
Parameters, 444, 463, 489, 504, 518, 530, 548
Password, 92, 109, 156, 172, 285, 302
Routed, 50
S7 200, 396, 429
Table, 41
Connection cable
6XV1440 -1K, 619
6XV1440–1M _ _ _ (PLC 545 / CPU 1102, PLC 555), 581
6XV1440-2K _ _ _ for SIMATIC 505, 580
6XV1440-2L _ _ _ for SIMATIC 505, 580
6XV1440–2M _ _ _ (PLC 525, 545 / CPU 1101, PLC 565T), 581
Allen-Bradley cable 1747-CP3, 472
Allen-Bradley cable 1761-CBL-PM02, 472
Modicon, 532
Multipoint cable 1:MP/TP/PC, 552
Multipoint cable 2:RS422, MP/TP/PC, 553
Omron Host Link, 550
Point-to-point cable PP2 for Omron, 554
PP1 for SIMATIC 505, 579
SIMATIC 505 NITP, 571
Connection information
Diagnostics, 24
Connection parameters
Allen-Bradley DF1, 463
Allen-Bradley EtherNet/IP, 444
Mitsubishi FX, 504
Mitsubishi MC TCP/IP, 489
Modicon Modbus RTU, 530
Modicon Modbus TCP, 518
Omron Host Link, 548
Connections
Allen-Bradley DF1, 465, 467
Configuring, 262, 265, 267, 396, 429
Editor, 262, 265, 267
Connector
Allen-Bradley DH485, 583
Modbus, 603, 611
SIMATIC 505 NITP, 571
ControlLogix, 450
Coordination, 15
CP1, 556
CPM, 556
CPU type
Allen-Bradley DF1, 475

Allen-Bradley EtherNet/IP, 450
FX3 series, 499
Mitsubishi FX, 509
Mitsubishi MC TCP/IP, 499
Modicon Modbus RTU, 538
Modicon Modbus TCP, 525
Omron Host Link, 556
Create
Connection, 142, 332
Create array
Modbus, 643
Creating
Connection, 42, 46, 50, 79, 210, 277
Creating a tag
Modbus, 643
CS1, 556
CSP
installation, 602
CSP Allen-Bradley DH485
Installing, 585
Uninstalling, 586, 602
Cyclic operation, 274, 399, 433

D

Data areas
Area pointer, 66, 651
Reading, 66, 651
Writing, 66, 651
Data exchange, 9
Tags, 14
using area pointers, 15
Data record, 15
Data type
Allen-Bradley, 484
Allen-Bradley EtherNet/IP, 448
Discrete alarm, 484, 513, 560
Internal tag, 60
Mitsubishi, 513
Mitsubishi FX, 508
Mitsubishi MC TCP/IP, 497
Modicon Modbus RTU, 537
Modicon Modbus TCP, 524
Omron, 560
Omron Host Link, 554
Valid, 448, 497, 508, 524, 537, 554
Data types
ET 200 CPU, 382
S7 1500, 130
S7 200, 427
S7-1200 V1, 195
S7-300/400, 262
Valid, 130, 195, 262, 382, 427
Date/time, 15
Area pointer, 173, 353
Date/time PLC, 15
Device
Inserting, 12
Device dependency
S7 200, 425
S7-1200 V1, 193
S7-1200 V2, 193
S7-1200 V3, 193
S7-1200 V4, 193
S7-300/400, 260
SIMATIC LOGO!, 437
SIMATIC S7-1500 V1.0, 128, 324
Device information
Diagnostics, 24
Device view
Hardware and network editor, 26
Device-dependent parameters
Allen-Bradley DH485, 587
Modbus, 605, 612
SIMATIC 505 NITP, 572
SIMATIC 505 PROFIBUS DP, 575
Devices
Connecting, 14
Networking, 14, 41
Devices & Networks, 14, 41
Devices and networks, 14, 41
Connections, 42, 50
HMI connections, 42, 50
Diagnostics
Connection, 24
Connection information, 24
Device, 24
Device information, 24
Discrete alarm
Allen-Bradley, 484
Mitsubishi, 513
Omron, 560
DP, 168, 231
DP interface, 576

E

Editor
Connection, 46
Connections, 262, 265, 267
Devices and networks, 44
Enable autonegation, 93, 157, 221, 286, 342

End of discovery of accessible devices, 96, 160, 223, 288, 345
End of sync domain, 96, 160, 223, 288, 345
End of topology discovery, 96, 160, 223, 288, 345
ET 200 CPU
    Communication, 331
    Data types, 382
    HMI connection, 332
Ethernet
    Basic Panel,
    Parameters, 133, 201, 270, 390, 401, 432
Ethernet/IP
    Allen-Bradley, 457
External tags
    Data exchange, 14

F

FAQs, 35
Fault alarm, 642
Functions
    Updating tag value, 57
FX1 series, 509
FX2 series, 509
FX3
    PLC, 494
FX3 series, 499

G

Global data block
    Configuring, 67
    Creating, 67

H

Hardware
    PROFIBUS DP, 573
Hardware and network editor
    Device view, 26
    Network view, 19
    Topology view, 29
Hardware catalog
    Task card, 33
Hardware editor
    Components, 17
    Function, 16
    Hardware catalog, 33
    Inspector window, 31
Highlighting
    Connection, 44
HMI connection, 14
    Creating, 79, 142, 210, 277, 332
    ET 200 CPU, 332
    MPI, 233
    MPI parameters, 235
    Password, 92, 109, 156, 172, 285, 302
    PROFIBUS, 98, 224, 291
    PROFIBUS parameters, 100, 164, 227, 293, 348
    PROFINET, 78, 79, 141, 142, 209, 210, 276, 277, 331, 332
    PROFINET parameters, 81, 145, 212, 278, 335
    S7 1200, 142
    S7 1500, 79
    S7 300/400, 210
    SIMATIC S7-1500 Software Controller, 277
HMI connections
    Devices & Networks, 42, 50
HMI device
    Commissioning, 591, 609, 616
    MPI parameters, 237
    PROFIBUS parameters, 101, 165, 228, 294, 350
    PROFINET parameters, 83, 146, 214, 280, 336
    Transferring the project, 591, 609, 616
    with the PLC, 592, 610, 618
HMI device on a PLC
    Commissioning, 459, 481, 501, 510, 527, 539, 558

I

IF1B
    Basic Panel, 75
Industrial Ethernet, 36
Inspector window
    Hardware and network editor, 31
Installing
    CSP Allen-Bradley DH485, 585
Installing communication drivers
    Allen-Bradley DH485, 586
    Modbus, 604, 611
    SIMATIC 505 NITP, 571
    SIMATIC 505 PROFIBUS DP, 574
Integrated
    Connection, 42, 46
Integrated connection, 58
Interface
    Basic Panel, 75
Internal
    Ethernet port, 497
Internal Ethernet port
    Open settings, 497

Internal tags
Data exchange, 14
IP protocol, 148, 338
iQ series, 499

J

Job mailbox, 15
Transferring data, 122, 186, 253, 317, 367, 417, 639, 663

L

LED
Assignment, 324, 374
Function, 324, 374
Image, 324, 374
LED image, 579, 625
Length
Area pointer, 64
Area pointers, 67
LLDP (Link Layer Discovery Protocol), 96, 160, 223, 288, 345

M

Manuals, 35
Micro, 525
MicroLogix, 450, 475
Mitsubishi, 439
Address, 499
Analog alarm, 513
Basic Panel, 74
Communication drivers, 486
FX, 486
TCP/IP, 486
Mitsubishi FX
Configuring a connection, 503
Connection, 503, 506
Connection parameters, 504
CPU type, 509
Data type, 508
Mitsubishi MC TCP/IP, 493, 506
Mitsubishi MC TCP/IP
Configuring a connection, 487
Connection, 487, 492
Connection parameters, 489
CPU type, 499
Data type, 497
Modbus
Configuring alarms, 644
Connecting cables, 603
Connector, 603, 611
Create array, 643
Creating a tag, 643
Device-dependent parameters, 605, 612
Network parameters, 605
Optimizing the configuration, 608
Permitted data type, 607, 614
PLC-dependent parameters, 606, 612
Protocol parameters, 604, 612
Modicon
Approved communication with Modbus RTU, 533
Connection, 532
Connection cable, 532
Restrictions with Modbus RTU, 533
Modicon M340, 525
Modicon Modbus, 439
Basic Panel, 74
Communication drivers, 515
RTU, 515
TCP, 515
Modicon Modbus connection
Data types, 543
Modicon Modbus RTU, 538
Configuring a connection, 528
Connection, 528
Connection parameters, 530
Data type, 537
Modicon Modbus TCP, 525
Configuring a connection, 516
Connection, 516, 521
Connection parameters, 518
Data type, 524
Modicon MODBUS TCP/IP
Change word order, 520
Momentum, 525, 538
Monitor, 93, 157, 221, 286, 342
Moving the view
Overview navigation, 20, 27, 30
MPI, 37
Addressing, 240
Connection, 267
HMI connection, 233
Network, 39
Network architecture, 39
Parameters, 235, 237, 239
S7 200, 273, 404
S7 300/400, 232, 233
MPI address
S7 300, 240
S7 400, 241

MPI communication
S7 300/400, 232
MPI parameters
S7 300/400, 235, 237, 239
Multiplexing
Address multiplexing Allen-Bradley Ethernet IP, 455
Multi-point connection
Allen-Bradley DF1, 468, 469

N

Network, 9
Ethernet, 37
MPI, 39
PPI, 40
PROFIBUS, 38
PROFINET, 37
Network architecture
PPI, 40
Network editor
Components, 17
Function, 16
Hardware catalog, 33
Inspector window, 31
Network parameters
Allen-Bradley DH485, 587
Modbus, 605
Network view, 41
Hardware and network editor, 19
Networking
Communication partners, 41
Devices, 41
Non-integrated
Connection, 46

O

Omron, 439
Analog alarm, 560
Basic Panel, 74
Communication drivers, 545
Host Link, 545
Omron Host Link, 556
Address, 556
Configuring a connection, 546
Connection, 546, 550
Connection cable, 550
Connection parameters, 548
Data type, 554
OP 73
Baud rate on PROFIBUS, 134, 202, 388
OP 77A
Baud rate on PROFIBUS, 134, 202, 388
OP 73
Baud rate on PROFIBUS, 271, 273, 402, 404, 405
OP 77A
Baud rate on PROFIBUS, 271, 273, 402, 404, 405
OPC
Basic Panel, 74
Operator input in the recipe view
Transferring data, 185, 252, 316, 366, 416, 641, 662
Optimizing the configuration, 615
Allen-Bradley DF1, 460, 481, 502, 510, 527, 540, 558
Allen-Bradley DH485, 589
Modbus, 608
SIMATIC 505, 569
Order number
6XV1830-0EH10, 623
Other PLCs
Addressing, 440
Data types, 440
Special features, 440
Overview navigation, 20, 27, 30

P

Parallel communication
Communication drivers, 563
Parameter
SIMATIC LOGO!, 431
Parameters
Connection, 444, 463, 489, 504, 518, 530, 548
MPI, 235, 237, 239
PROFIBUS, 100, 101, 103, 164, 165, 167, 227, 228, 230, 293, 294, 296, 348, 350, 351
S7 1200, 200
S7 1500, 131
S7 200, 268, 273, 399, 404, 405
Password
HMI connection, 92, 109, 156, 172, 285, 302
Password protection, 88, 105, 153, 155, 170, 298
Permitted data type
Allen-Bradley DH485, 588
Modbus, 607, 614
SIMATIC 505, 566
Pin assignment
6XV1440 - 2P for Mitsubishi PG protocol, 507

6XV1440 -1K, 619
6XV1440–1M _ _ _ (PLC 545 / CPU 1102, PLC 555), 581
6XV1440-2K _ _ _ for SIMATIC 505, 580
6XV1440-2L _ _ _ for SIMATIC 505, 580
6XV1440–2M _ _ _ (PLC 525, 545 / CPU 1101, PLC 565T), 581
Allen-Bradley, 470
Allen-Bradley cable 1747-CP3, 472, 592
Allen-Bradley cable 1761-CBL-PM02, 472, 593
Allen-Bradley cable 1784-CP10, 471
MP1 connecting cable, 596
Multipoint cable 1:MP/TP/PC, 552
Multipoint cable 2:RS422, MP/TP/PC, 553
Point-to-point cable 1,
Point-to-point cable 2, 536, 621
Point-to-point cable 3,
Point-to-point cable PP1 for Omron, 554
Point-to-point cable PP2 for Omron, 554
PP1 for SIMATIC 505, 579
PP4 connecting cable, 594
PLC
Connecting, 447, 492, 506, 521, 550
MPI parameters, 239
PROFIBUS parameters, 103, 167, 230, 296, 351
PROFINET parameters, 85, 148, 216, 282, 338
PLC5, 475
PLC-dependent parameters
Allen-Bradley DH485, 588
Modbus, 606, 612
SIMATIC 505 PROFIBUS DP, 575
Pointer name
Area pointer, 64
Point-to-point connection
Allen-Bradley DF1, 467
Port options, 94, 158, 222, 287, 343
Enable autonegation, 93, 157, 221, 286, 342
Monitor, 93, 157, 221, 286, 342
Transmission rate / duplex, 93, 157, 220, 286, 342
Possible cause of error
Transferring data, 124, 188, 256, 320, 370, 419, 638, 665
PPI, 37
Network, 40
Network architecture, 40
S7 200, 405
Premium, 525
Product support, 35
PROFIBUS, 36, 168, 231
HMI connection, 98, 162, 224, 291, 347
Network, 38
OP 73, 134, 202, 388
OP 77A, 134, 202, 388
OP 73, 271, 273, 402, 404, 405
OP 77A, 271, 273, 402, 404, 405
Parameters, 100, 101, 103, 134, 164, 165, 167, 202, 227, 228, 230, 271, 293, 294, 296, 348, 350, 351, 388, 402
S7 1200, 161
S7 1500, 97, 290
S7 300/400, 98, 224, 291
S7-1200, 162
SIMATIC ET 200 CPU, 346, 347
Standard, 168, 231
Universal, 168, 231
PROFIBUS communication
S7 1200, 161
S7 1500, 97, 290
S7 300/400, 224
SIMATIC ET 200 CPU, 346
PROFIBUS DP, 38
Connection, 265
PROFIBUS parameters
S7 1200, 164, 165, 167, 348, 350, 351
S7 1500, 100, 101, 103, 293, 294, 296
S7 300/400, 227, 228, 230
PROFIBUS profile, 168, 231
Different profiles on the same subnet, 168, 231
Effect on the transmission rate, 168, 231
Meaning of profiles, 169, 232
PROFINET, 36
Connection, 262
HMI connection, 78, 79, 141, 142, 209, 210, 276, 277, 331, 332
S7 1200, 141
S7 1500, 78
S7 300/400, 209
SIMATIC ET 200 CPU, 331
WinAC 1500, 276
PROFINET IO
Port options, 93, 157, 220, 286, 342
PROFINET parameters
HMI connection, 81, 145, 212, 278, 335
HMI device, 83, 146, 214, 280, 336
PLC, 85, 148, 216, 282, 338
S7 1200, 145, 148, 335, 338
S7 1500, 81, 83, 85, 278, 280, 282
S7 300/400, 212, 214, 216
Project
Multiple connections, 116, 180, 246, 310, 359, 412
Project ID, 15
Protection concept, 89, 106, 153, 170, 299

Protection level, 88, 105, 153, 155, 170, 298
Protocol parameters
Allen-Bradley DH485, 587
Modbus, 604, 612
SIMATIC 505 NITP, 572
SIMATIC 505 PROFIBUS DP, 575

Q

Q
PLC, 495
Q series, 499
Q0xUDEH CPU
PLC, 497
Quantum, 525

R

Read-only, 155
Restriction
Modicon Modbus RTU, 541
Modicon Modbus TCP/IP, 541
Routed
Connection, 50

S

S7 1200
Communication, 140
Connection parameters, 200
HMI connection, 142
Parameters, 200
PROFIBUS, 161
PROFIBUS parameters, 164, 165, 167, 348, 350, 351
PROFINET, 141
S7 1500
Communication, 77
Connection parameters, 131
Data types, 130
HMI connection, 79
Parameters, 131
PROFIBUS, 97, 98, 290, 291
PROFIBUS parameters, 100, 101, 103, 293, 294, 296
PROFINET, 78, 276
PROFINET parameters, 81, 83, 85, 278, 280, 282
S7 200
Communication, 396
Connection, 396, 429
Connection parameters, 399
Data types, 427
MPI, 273, 404
Parameters, 273, 399, 404, 405
PPI, 405
S7 300
Communication, 209
MPI address, 240
S7 300/400
Connection parameters, 268
HMI connection, 210
MPI, 232, 233
MPI parameters, 235, 237, 239
Parameters, 268
PROFIBUS, 224
PROFIBUS parameters, 227, 228, 230
PROFINET, 209
PROFINET parameters, 212, 214, 216
S7 400
Communication, 209
MPI address, 241
S7-1200
PROFIBUS, 162
S7-1200 V2
Data types, 195
S7-300/400
Data types, 262
Screen number, 15
Script
Updating tag value, 57
Select the PLC
Standard Modbus, 604, 612
Selecting the PLC
Allen-Bradley DH485, 586
SIMATIC 505 NITP, 572
Service & Support, 35
SIMATIC 505, 564
Communication partners, 564
Optimizing the configuration, 569
Permitted data type, 566
SIMATIC 505 NITP
Connector, 571
Device-dependent parameters, 572
Installing communication drivers, 571
Protocol parameters, 572
Selecting the PLC, 572
SIMATIC 505 PROFIBUS DP, 573
Device-dependent parameters, 575
Installing communication drivers, 574
PLC-dependent parameters, 575
Protocol parameters, 575
Selecting the PLC, 574

SIMATIC LOGO!
Communication, 428
Connection parameters, 431
Parameter, 431
SIMATIC S7
Communication partners, 135, 203, 272, 389, 403, 406
SIMATIC S7-1500 Software Controller
Communication, 275
HMI connection, 277
SIMATIC ET 200 CPU
PROFIBUS, 346, 347
PROFINET, 331
SLC, 475
Special features
Omron Host Link, 559
Standard
PROFIBUS, 168, 231
Standard Modbus
Approved communication with standard Modbus RTU, 598
cleared communication with Modbus TCP/IP, 600
installation, 602
Installing communication drivers, 604, 611
Restrictions with Modbus TCP/IP, 600
Select the PLC, 604, 612
Standard MODBUS TCP/IP
Change bit order, 606, 613
Change word order, 606, 613
Switchport, 94, 158, 222, 287, 343
Symbolic addressing
of a tag, 59
Sync domain, 96, 160, 223, 288, 345
Synchronize, 57
Synchronous
Transferring data, 121, 251, 315, 365, 421, 637, 660
Syntax
Addressing, 452
System alarm, 642

T

Table
Connection, 41
Tag
absolute addressing, 58
External tag, 56
Internal tag, 60
symbolic addressing, 59
Tag table
Default, 54
for HMI devices, 54
user-defined, 54
Tags
Basics, 53
Data exchange, 14
Task card
Hardware catalog, 33
Third-party drivers
Communication, 439
Special features, 440
Time synchronization, 140, 204, 326, 391
Configuring, 137, 138, 206, 207, 328, 329, 393, 394
integrated connection, 137, 206, 328, 393
Non-integrated connection, 138, 207, 329, 394
Restriction, 136, 205, 327, 392
System limits, 136, 205, 327, 392
Topology discovery, 96, 160, 223, 288, 345
Topology view
Hardware and network editor, 29
Transferring data
"Date/time" area pointer, 629
Area pointer, 61, 110, 173, 241, 303, 353, 407, 626, 649
Area pointer screen number, 111, 175, 178, 242, 304, 358, 407, 628, 651
Coordination area pointer, 115, 177, 245, 308, 356, 411, 631, 654
Data record are pointer, 659
Data record area pointer, 119, 183, 249, 313, 363, 415, 635
Date/time area pointer, 112, 173, 176, 243, 305, 353, 355, 408, 652
Date/time PLC area pointer, 113, 244, 306, 409, 630, 653
Job mailbox, 122, 186, 253, 317, 367, 417, 639, 663
Job mailbox area pointer, 117, 181, 247, 311, 360, 413, 633, 656
Operator input in the recipe view, 185, 252, 316, 366, 416, 641, 662
Possible cause of error, 124, 188, 256, 320, 370, 419, 638, 665
Project ID area pointer, 116, 180, 246, 309, 359, 412, 632, 655
Triggering by means of a configured function, 123, 187, 255, 319, 369, 418, 638, 664
With synchronization, 121, 251, 315, 365, 421, 637, 660
Without synchronization, 120, 184, 250, 314, 364, 420, 636, 660

Transferring the project
HMI device, 591, 609
Transmission rate / duplex, 93, 157, 220, 286, 342
Trend request, 624
Trend transfer, 624
Triggering by means of a configured function
Transferring data, 123, 187, 255, 319, 369, 418, 638, 664

U

Uninstalling
CSP Allen-Bradley DH485, 586, 602
Universal
PROFIBUS, 168, 231

V

Valid
Data type, 448, 497, 508, 524, 537, 554
Data types, 130, 195, 262, 382, 427
Valid data type
Allen-Bradley DF1, 473

Z

Zoom
Adjusting the zoom setting, 20, 27, 30