SIEMENS

OPC 1

Communicating with controllers 2

SIMATIC HMI

WinCC (TIA Portal) WinCC Engineering V18 – Communication

System Manual

Online documentation

11/2022
Online documentation

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
Digital Industries
Postfach 48 48
90026 NÜRNBERG
GERMANY

Document order number: Online documentation
Ⓟ 11/2022 Subject to change

Copyright © Siemens AG 2022.
All rights reserved

Table of contents

1 OPC 13

1.1 OPC for Runtime Advanced 13
1.1.1 Basics 13
1.1.1.1 OPC 13
1.1.1.2 OPC Specifications 13
1.1.1.3 Compatibility 14
1.1.1.4 Using OPC in WinCC 14
1.1.2 Security concept of OPC UA 16
1.1.3 Configuring an OPC server 18
1.1.3.1 Configuring an HMI device as OPC DA server 18
1.1.3.2 Configuring an HMI device as an OPC UA server 20
1.1.3.3 Configuring DCOM user permissions in Windows 20
1.1.4 Configuring an OPC client 21
1.1.4.1 Creating a connection to a WinCC OPC DA server 21
1.1.4.2 Creating a connection to an OPC UA server 22
1.1.4.3 Accessing process values of an OPC server 23
1.1.4.4 OPC XML Gateway 24
1.1.4.5 Configuring an OPC XML Manager 25
1.1.5 Reference 26
1.1.5.1 Permitted data types (OPC) 26
1.1.5.2 Access to tags with OPC 27
1.1.5.3 Supported OPC UA services of the OPC UA client 28

1.2 OPC for Runtime Professional 29
1.2.1 Basics 29
1.2.1.1 OPC 29
1.2.1.2 OPC Specifications 29
1.2.1.3 Compatibility 30
1.2.1.4 Licensing 30
1.2.1.5 Using OPC in WinCC 30
1.2.2 Configuration 32
1.2.2.1 Basic settings for OPC 32
1.2.3 WinCC OPC server 32
1.2.3.1 Accessibility of WinCC OPC servers 32
1.2.3.2 WinCC OPC DA Server 33
1.2.3.3 WinCC OPC XML DA server 33
1.2.3.4 WinCC OPC HDA Server 36
1.2.3.5 WinCC OPC A&E server 44
1.2.3.6 WinCC OPC UA Server 61
1.2.4 WinCC OPC connection 79
1.2.4.1 Creating a connection to an OPC server 79
1.2.4.2 Accessing process values of an OPC server 80
1.2.5 WinCC OPC UA connection 80
1.2.5.1 Creating a connection to an OPC UA server 80
1.2.5.2 Setting up authentication via certificates. 82
1.2.5.3 Creating OPC UA tags 83

1.2.5.4 Creating array tags for OPC UA connection ... 84
1.2.5.5 Overview of the supported data types ... 86
1.2.5.6 Diagnostics options of the "OPC UA" channel ... 87
1.2.5.7 Supported OPC UA services of the OPC UA client ... 87
1.2.5.8 Description of Log File Entries ... 88
1.2.6 Commissioning OPC ... 89
1.2.6.1 Configuring DCOM user permissions in Windows ... 89
1.2.6.2 Adjusting Windows firewall settings ... 90

2 Communicating with controllers ... 91

2.1 Basics of communication ... 91
2.1.1 Communication between devices ... 91
2.1.2 Devices and networks in the automation system ... 93
2.1.3 Data exchange using tags ... 96
2.1.4 Data exchange using area pointers ... 97
2.1.5 Communication drivers ... 98
2.1.6 Secure communication ... 98

2.2 Editors for communication ... 101
2.2.1 "Devices & networks" editor ... 101
2.2.2 Network view ... 103
2.2.3 Network data ... 106
2.2.4 Diagnostics of online connections ... 109
2.2.5 Device view ... 110
2.2.6 Topology view ... 113
2.2.7 Inspector window ... 116
2.2.8 Hardware catalog ... 117
2.2.9 Information on hardware components ... 120

2.3 Networks and connections ... 121
2.3.1 SIMATIC communication networks ... 121
2.3.1.1 Communication networks ... 121
2.3.1.2 PROFINET and Ethernet ... 122
2.3.1.3 PROFIBUS ... 123
2.3.1.4 MPI ... 124
2.3.1.5 PPI ... 125
2.3.2 Configuring networks and connections ... 126
2.3.2.1 Networking devices ... 126
2.3.2.2 Configuring an integrated connection in the "Devices & Networks" editor ... 127
2.3.2.3 Special considerations of the "Devices & Networks" editor ... 129
2.3.2.4 Configuring a non-integrated connection in the "Connections" editor ... 131
2.3.2.5 Configuring a routed connection in the "Devices & Networks" editor ... 135
2.3.2.6 Setting up switch on/switch off of a connection in Runtime ... 138
2.3.2.7 Using local ID of HMI connections for communication ... 139

2.4 Data exchange ... 140
2.4.1 Data exchange using tags ... 140
2.4.1.1 Basics of tags ... 140
2.4.1.2 Overview of HMI tag tables ... 141
2.4.1.3 External tags ... 143
2.4.1.4 Addressing external tags ... 145
2.4.1.5 Internal Tags ... 148
2.4.2 Data exchange using area pointers ... 149
2.4.2.1 Basic information on area pointers ... 149

2.4.2.2 Area pointers for connections ... 150
2.4.2.3 Use of area pointers ... 151
2.4.2.4 Accessing data areas ... 153
2.4.2.5 Configuring area pointers ... 153

2.5 Device dependency ... 158
2.5.1 Basic Panel ... 158
2.5.1.1 Communication drivers for Basic Panels ... 158
2.5.1.2 Interfaces of the Basic Panels ... 163
2.5.1.3 Area pointers for Basic Panels ... 164
2.5.1.4 Restrictions ... 165
2.5.2 Comfort Panel ... 165
2.5.2.1 Communication drivers for Comfort Panels ... 165
2.5.2.2 Interfaces of Comfort Panels ... 173
2.5.2.3 Area pointers for Comfort Panel ... 174
2.5.3 Mobile Panel ... 175
2.5.3.1 Communication drivers for Mobile Panels ... 175
2.5.3.2 Interfaces of Mobile Panels ... 178
2.5.3.3 Area pointers for Mobile Panel ... 179
2.5.4 PC systems with WinCC Runtime ... 180
2.5.4.1 Communication drivers for WinCC Runtime ... 180
2.5.4.2 Interfaces for PC systems ... 183
2.5.5 Parallel communication ... 184

2.6 Communicating with SIMATIC S7 1500 ... 185
2.6.1 Communication with SIMATIC S7-1500 ... 185
2.6.2 Communication via PROFINET ... 186
2.6.2.1 Configuring an HMI connection ... 186
2.6.2.2 Configuring an HMI connection ... 189
2.6.2.3 PROFINET parameters ... 197
2.6.2.4 Setting port options ... 211
2.6.3 Communication via PROFIBUS ... 215
2.6.3.1 Configuring an HMI connection ... 215
2.6.3.2 Configuring an HMI connection ... 218
2.6.3.3 PROFIBUS parameters ... 225
2.6.4 Data exchange ... 238
2.6.4.1 Data exchange using area pointers ... 238
2.6.4.2 Trends ... 257
2.6.4.3 Alarms ... 259
2.6.4.4 LED mapping ... 262
2.6.5 Performance features of communication ... 263
2.6.5.1 S7-1500 device dependency ... 263
2.6.5.2 Valid data types for SIMATIC S7 1500 ... 270
2.6.5.3 Addressing ... 272
2.6.6 Configuring connections in the "Connections" editor ... 272
2.6.6.1 Creating a PROFINET connection ... 272
2.6.6.2 Creating a PROFIBUS connection ... 275
2.6.6.3 Parameters for the connection ... 277
2.6.7 Configuring connections in the "Connections" editor ... 281
2.6.7.1 Creating a connection ... 281
2.6.7.2 Parameters for the connection (SIMATIC S7 1500) ... 282
2.6.8 Configuring time synchronization ... 286
2.6.8.1 Time synchronization ... 286

2.6.8.2 Time synchronization restrictions ... 287
2.6.8.3 Configuring time synchronization for integrated connections ... 288
2.6.8.4 Configuring time synchronization for non-integrated connections ... 289
2.6.9 Querying the PLC status ... 291
2.6.10 Raw data tags of the "SIMATIC S7-1500" communication driver ... 291
2.6.10.1 Raw Data Tag ... 291
2.6.10.2 Raw data tags of the "SIMATIC S7-1500" communication driver ... 292
2.6.10.3 Configuring raw data tags for SIMATIC S7-1500 ... 293
2.6.10.4 Raw data tag as byte array ... 294
2.6.10.5 How to configure a raw data tag as byte array ... 296

2.7 Communicating with SIMATIC S7 1200 ... 298
2.7.1 Communication with SIMATIC S7-1200 ... 298
2.7.2 Communication via PROFINET ... 299
2.7.2.1 Configuring an HMI connection ... 299
2.7.2.2 Configuring an HMI connection ... 303
2.7.2.3 PROFINET parameters ... 310
2.7.2.4 Setting port options ... 322
2.7.3 Communication via PROFIBUS ... 327
2.7.3.1 Configuring an HMI connection ... 327
2.7.3.2 Configuring an HMI connection ... 330
2.7.3.3 PROFIBUS parameters ... 339
2.7.4 Data exchange ... 349
2.7.4.1 Data exchange using area pointers ... 349
2.7.4.2 Trends ... 368
2.7.4.3 Alarms ... 370
2.7.4.4 LED mapping ... 373
2.7.5 Performance features of communication ... 374
2.7.5.1 S7-1200 device dependency ... 374
2.7.5.2 Valid data types for SIMATIC S7-1200 ... 382
2.7.6 Configuring connections in the "Connections" editor ... 383
2.7.6.1 Creating a PROFINET connection ... 383
2.7.6.2 Creating a PROFIBUS connection ... 385
2.7.6.3 Parameters for the connection (SIMATIC S7 1200) ... 387
2.7.7 Configuring connections in the "Connections" editor ... 391
2.7.7.1 Creating a connection ... 391
2.7.7.2 Parameters for the connection (SIMATIC S7 1200) ... 393
2.7.8 Configuring time synchronization ... 396
2.7.8.1 Time synchronization ... 396
2.7.8.2 Time synchronization restrictions ... 397
2.7.8.3 Configuring time synchronization for integrated connections ... 397
2.7.8.4 Configuring time synchronization for non-integrated connections ... 398
2.7.9 Querying the PLC status ... 400

2.8 Communicating with SIMATIC S7 300/400 ... 401
2.8.1 Communication with SIMATIC S7 300/400 ... 401
2.8.2 Communication via PROFINET ... 402
2.8.2.1 Configuring an HMI connection ... 402
2.8.2.2 Configuring an HMI connection ... 405
2.8.2.3 Named connections ... 413
2.8.2.4 PROFINET parameters ... 415
2.8.2.5 Port options ... 423
2.8.3 Communication via PROFIBUS ... 427

2.8.3.1 Configuring an HMI connection ... 427
2.8.3.2 Configuring an HMI connection ... 430
2.8.3.3 PROFIBUS parameters ... 437
2.8.4 Communication via MPI ... 443
2.8.4.1 Configuring an HMI connection ... 443
2.8.4.2 Configuring an HMI connection ... 446
2.8.4.3 MPI parameters ... 453
2.8.5 Data exchange ... 459
2.8.5.1 Data exchange using area pointers ... 459
2.8.5.2 Trends ... 477
2.8.5.3 Alarms ... 479
2.8.5.4 LED mapping ... 482
2.8.6 Performance features of communication ... 483
2.8.6.1 S7 300/400 device dependency ... 483
2.8.6.2 Permitted data types for SIMATIC S7 300/400 ... 490
2.8.6.3 Permitted data types for SIMATIC S7 300/400 ... 491
2.8.7 Creating a connection in the "Connections" editor ... 492
2.8.7.1 Creating a PROFINET connection ... 492
2.8.7.2 Creating a PROFIBUS connection ... 495
2.8.7.3 Creating an MPI connection ... 497
2.8.7.4 Parameters for the connection ... 498
2.8.8 Creating a connection in the "Connections" editor ... 505
2.8.8.1 Creating a connection ... 505
2.8.8.2 Parameters for the connection (SIMATIC S7 300/400) ... 507
2.8.9 Troubleshooting of connection errors (commissioning) ... 515
2.8.9.1 Procedure for the localization of errors ... 515
2.8.9.2 Error codes ... 527
2.8.9.3 Internal error codes and constants ... 543
2.8.9.4 API error texts ... 561
2.8.10 Raw data tags of the "SIMATIC S7 300/400" communication driver ... 571
2.8.10.1 Raw Data Tag ... 571
2.8.10.2 Raw data tags of the "SIMATIC S7 300/400" communication driver ... 572
2.8.10.3 Raw data tag as byte array ... 573
2.8.10.4 How to configure a raw data tag as byte array ... 575
2.8.10.5 Raw data tag for BSEND/BRCV functions of S7 communication ... 578
2.8.10.6 How to configure a raw data tag for "BSEND/BRCV" functions ... 581

2.9 Communicating with the SIMATIC S7-1500 Software Controller ... 582
2.9.1 Communication with SIMATIC S7-1500 Software Controller ... 582
2.9.2 Communication via PROFINET ... 583
2.9.2.1 Basics of HMI connections to an SIMATIC S7-1500 Software Controller ... 583
2.9.2.2 Configuring an HMI connection ... 585
2.9.2.3 Configuring an HMI connection ... 588
2.9.2.4 PROFINET parameters ... 595
2.9.3 Communication via PROFIBUS ... 603
2.9.3.1 Configuring an HMI connection ... 603
2.9.3.2 Configuring an HMI connection ... 606
2.9.3.3 PROFIBUS parameters ... 613
2.9.4 Data exchange ... 624
2.9.4.1 Data exchange using area pointers ... 624
2.9.4.2 Trends ... 643
2.9.4.3 Alarms ... 646
2.9.4.4 LED mapping ... 648

2.9.5 Performance features of communication ... 649
2.9.5.1 Device dependencies of the SIMATIC S7-1500 software controller ... 649
2.9.5.2 Permitted data types ... 658
2.9.6 Configuring connections in the "Connections" editor ... 659
2.9.6.1 Creating a PROFINET connection ... 659
2.9.7 Configuring connections in the "Connections" editor ... 662
2.9.7.1 Creating a connection ... 662
2.9.7.2 Connection parameters ... 663
2.9.8 Configuring time synchronization ... 665
2.9.8.1 Time synchronization ... 665
2.9.8.2 Time synchronization restrictions ... 666
2.9.8.3 Configuring time synchronization for integrated connections ... 667
2.9.8.4 Configuring time synchronization for non-integrated connections ... 668

2.10 Communicating with the SIMATIC ET 200 CPU ... 670
2.10.1 Communication with SIMATIC ET 200 CPU ... 670
2.10.2 Communication via PROFINET ... 671
2.10.2.1 Configuring an HMI connection ... 671
2.10.2.2 Configuring an HMI connection ... 674
2.10.2.3 PROFINET parameters ... 681
2.10.2.4 Setting port options ... 687
2.10.3 Communication via PROFIBUS ... 692
2.10.3.1 Configuring an HMI connection ... 692
2.10.3.2 Configuring an HMI connection ... 694
2.10.3.3 PROFIBUS parameters ... 700
2.10.4 Data exchange ... 705
2.10.4.1 Data exchange using area pointers ... 705
2.10.4.2 Trends ... 723
2.10.4.3 Alarms ... 725
2.10.4.4 LED mapping ... 728
2.10.5 Performance features of communication ... 729
2.10.5.1 Valid data types for SIMATIC ET 200 CPU ... 729
2.10.5.2 Device dependencies of SIMATIC ET 200 CPU ... 730
2.10.6 Configuring connections in the "Connections" editor ... 737
2.10.6.1 Creating a PROFINET connection ... 737
2.10.6.2 Creating a PROFIBUS DP connection ... 740
2.10.6.3 Connection parameters ... 742
2.10.7 Configuring connections in the "Connections" editor ... 746
2.10.7.1 Creating a connection ... 746
2.10.7.2 Connection parameters ... 748
2.10.8 Configuring connections in the "Connections" editor ... 751
2.10.8.1 Creating a connection ... 751
2.10.8.2 Connection parameters ... 752
2.10.9 Configuring time synchronization ... 758
2.10.9.1 Time synchronization ... 758
2.10.9.2 Time synchronization restrictions ... 759
2.10.9.3 Configuring time synchronization for integrated connections ... 760
2.10.9.4 Configuring time synchronization for non-integrated connections ... 761

2.11 Communicating with SIMATIC S7 200 ... 763
2.11.1 Communication with SIMATIC S7 200 ... 763
2.11.2 Creating a connection to SIMATIC S7 200 ... 763
2.11.3 Parameters for the connection ... 766

2.11.3.1 Parameters for the connection (SIMATIC S7 200) ... 766
2.11.3.2 Ethernet parameters ... 767
2.11.3.3 PROFIBUS parameters ... 768
2.11.3.4 MPI parameters ... 770
2.11.3.5 PPI parameters ... 771
2.11.3.6 Cyclic operation ... 773
2.11.4 Data exchange ... 774
2.11.4.1 Data exchange using area pointers ... 774
2.11.4.2 Trends ... 791
2.11.4.3 Alarms ... 794
2.11.5 Performance features of communication ... 797
2.11.5.1 Device dependency S7 200 ... 797
2.11.5.2 Permitted data types for SIMATIC S7 200 ... 804

2.12 Communicating with SIMATIC LOGO! ... 805
2.12.1 Communication with SIMATIC LOGO! ... 805
2.12.2 Creating a connection to SIMATIC LOGO! ... 806
2.12.3 Connection parameters ... 808
2.12.3.1 Connection parameters ... 808
2.12.3.2 Ethernet parameters ... 809
2.12.3.3 Cyclic operation ... 810
2.12.4 Data exchange ... 811
2.12.4.1 Trends ... 811
2.12.4.2 Alarms ... 814
2.12.5 Performance features of communication ... 816
2.12.5.1 SIMATIC LOGO! device dependency ... 816
2.12.5.2 Valid data types for SIMATIC LOGO! ... 824

2.13 Configuring direct keys ... 825
2.13.1 Direct keys ... 825
2.13.2 Changing the operating mode of the HMI device ... 825
2.13.2.1 Changing the operating mode for a PROFINET connection ... 825
2.13.2.2 Changing the operating mode for a PROFIBUS connection ... 826
2.13.3 Configuring direct keys ... 827
2.13.3.1 Configuring direct keys for an HMI device with touch operation ... 827
2.13.4 PROFINET IO direct keys ... 828
2.13.4.1 PROFINET IO direct keys ... 828
2.13.4.2 HMI devices for the configuration of PROFINET IO direct keys ... 830
2.13.4.3 Restrictions for PROFINET IO direct keys ... 830
2.13.4.4 Inputs and outputs of HMI devices ... 832
2.13.5 PROFIBUS DP direct keys ... 843
2.13.5.1 PROFIBUS DP direct keys ... 843
2.13.5.2 HMI devices for the configuration of PROFIBUS DP direct keys ... 845
2.13.5.3 Restrictions for PROFIBUS DP direct keys ... 845
2.13.5.4 Inputs and outputs of HMI devices ... 847

2.14 Communication via SIMATIC HMI HTTP ... 853
2.14.1 Basic information on SIMATIC HMI HTTP ... 853
2.14.2 Configuring a connection via SIMATIC HMI HTTP ... 855
2.14.2.1 Configuring a connection via SIMATIC HMI HTTP ... 855
2.14.2.2 Configuring an HMI device as a server ... 856
2.14.2.3 Configuring an HMI device as a client ... 858
2.14.2.4 Assigning tag parameters in the client ... 861
2.14.2.5 Commissioning a connection ... 864

2.14.3 Performance features of communication ... 869
2.14.3.1 Permitted data types ... 869

2.15 Communication via OPC ... 870
2.15.1 Communication via OPC ... 870
2.15.2 Configuring a connection via OPC ... 871
2.15.3 Performance features of communication ... 873
2.15.3.1 Permitted data types for OPC DA ... 873
2.15.3.2 Permitted data types for OPC XML DA ... 874
2.15.3.3 Permitted data types for OPC UA ... 875
2.15.4 Performance features of communication ... 875
2.15.4.1 Permitted data types for OPC DA ... 875
2.15.4.2 Permitted data types for OPC XML DA ... 876

2.16 Communication via routing ... 877
2.16.1 Communication via routing ... 877
2.16.2 Example for communication via routing ... 879
2.16.3 Configuring communication via routing ... 881

2.17 PROFINET IO and IRT ... 883
2.17.1 PROFINET IO ... 883
2.17.1.1 What is PROFINET IO? ... 883
2.17.1.2 PROFINET IO systems ... 884
2.17.2 IRT communication ... 884
2.17.2.1 Introduction: Isochronous Realtime Ethernet ... 884
2.17.2.2 Overview of RT classes ... 886
2.17.2.3 Basic procedures for configuring IRT ... 886
2.17.2.4 Sync domain ... 887
2.17.3 Configuring the HMI device as IO device ... 887
2.17.4 Parameters for PROFINET IO and IRT ... 889
2.17.4.1 Real-time settings ... 889
2.17.5 Performance features of PROFINET IO and IRT ... 890
2.17.5.1 Configuration of PROFINET and IRT ... 890
2.17.5.2 Device dependency ... 891

2.18 Media redundancy ... 892
2.18.1 Media redundancy ... 892
2.18.2 Media Redundancy Protocol (MRP) ... 894
2.18.3 Configuring media redundancy for HMI devices ... 895
2.18.4 Parameters for media redundancy ... 897
2.18.5 Managing MRP domains ... 898
2.18.6 Restrictions with media redundancy ... 900

2.19 Communication with other controllers ... 901
2.19.1 Communication with other PLCs ... 901
2.19.2 Distinctive features when configuring ... 902
2.19.3 Parallel communication ... 903
2.19.4 Communication drivers ... 905
2.19.4.1 Allen-Bradley ... 905
2.19.4.2 Mitsubishi ... 958
2.19.4.3 Modicon Modbus ... 995
2.19.4.4 Omron ... 1032
2.19.5 Connection diagnostics ... 1053
2.19.5.1 Operating principle ... 1053
2.19.5.2 Diagnostics with "WinCC Channel Diagnosis" ... 1054

2.19.5.3 Diagnostics with the Channel diagnostics view 1056
2.19.5.4 Description of the log file 1057
2.19.5.5 Configuring the trace function..... 1058
2.19.5.6 Checking connections with performance tags..... 1060

Index 1063

OPC 1

1.1 OPC for Runtime Advanced

1.1.1 Basics

1.1.1.1 OPC

Introduction

OPC refers to standardized multi-vendor software interfaces for data exchange in automation engineering.

The OPC interfaces provide a standardized environment in which devices and applications from various manufacturers can be linked.

OPC is based on the Windows technology COM (Component Object Model) and DCOM (Distributed Component Object Model).

OPC UA (Unified Architecture) is the technology succeeding OPC. OPC UA is platform-independent and can use different reports as a communication medium.

Principle of operation

In WinCC, you can configure HMI devices as OPC servers or OPC clients. The particular HMI device determines which OPC servers and OPC clients are available.

1.1.1.2 OPC Specifications

OPC specifies interfaces to gain access to the following objects in WinCC:

- Process values (OPC Data Access 1.0, 2.05a)
- Process values (DataAccess ClientFacet (OPC UA)
- Process values (OPC XML-Data Access 1.01)

You can find additional information about the individual OPC specifications in the Internet at the website of OPC Foundation:.

- www.opcfoundation.org (www.opcfoundation.org)

See also

Supported OPC UA services of the OPC UA client (Page 28)

1.1.1.3 Compatibility

Support of the mentioned specifications is checked regularly by the "Compliance Test Tool" (CTT) of the OPC Foundation. Interoperability with OPC products of other manufacturers is ensured through the participation in "OPC Interoperability Workshops".

The test results submitted are published on the website of the OPC Foundation. The results can be called up from there using the search term "OPC Self-Certified Products".

1.1.1.4 Using OPC in WinCC

Configuration option

HMI devices configured with WinCC feature an OPC interface for data communication between automation devices or automation systems using the OPC communication driver.

You can use an HMI device as an OPC server or as an OPC client. As an OPC client, the HMI device can be connected to a maximum of eight OPC servers.

For the exchange of data via an XML connection, some OPC XML DA clients create several HTTP connections to the OPC XML DA server.

HMI device	Data exchange over	Operating system	OPC server	OPC client
PC, Panel PC (with WinCC Runtime Advanced)	DCOM or OPC UA binary (TCP/IP)	Windows 7 Windows 8 Windows 10	OPC DA server OPC UA server (DA only)	OPC DA client OPC UA client*
HMI devices with version V12.0: • Comfort Panels	SOAP	Windows CE	OPC XML DA server	-
HMI devices with version V13.0 (or higher): • Comfort Panels • Mobile Panels KTP 400F / 700(F) / 900(F)	SOAP or OPC UA binary (TCP/IP)	Windows CE	OPC UA server (DA only)	OPC UA client
*: The OPC UA client exchanges data via TCP/IP.				

A communication via DCOM is only possible between HMI devices of the "PC" or the "Panel PC" type. In order for a control room PC to display the process values of Windows CE HMI devices in the plant, for example, install "OPC-XML-Gateway" on the control room PC using the WinCC Runtime Advanced Setup. The "OPC-XML-Gateway" supports the communication within the SIMATIC HMI devices family between an OPC DA client and an OPC XML DA server.

The area pointer is supported by any OPC connection.

OPC server accessibility

The following table lists the available OPC servers and how you can access them:

OPC server	OPC server name
OPC DA server	OPC.SimaticHMI.CoRtHmiRTm (ProgID)
OPC XML DA server	http://<xxx>/soap/OpcXml (URL) [1] <xxx>: IP address or DNS name of the HMI device
OPC DA server	OPC.Siemens.XML (ProgID) for accessing the OPC XML DA server via the OPC XML gateway from an OPC DA client
OPC UA server	Server URL: opc.tcp://[HostName]:4870
[1]: For the access of an external OPC DA XML client.	

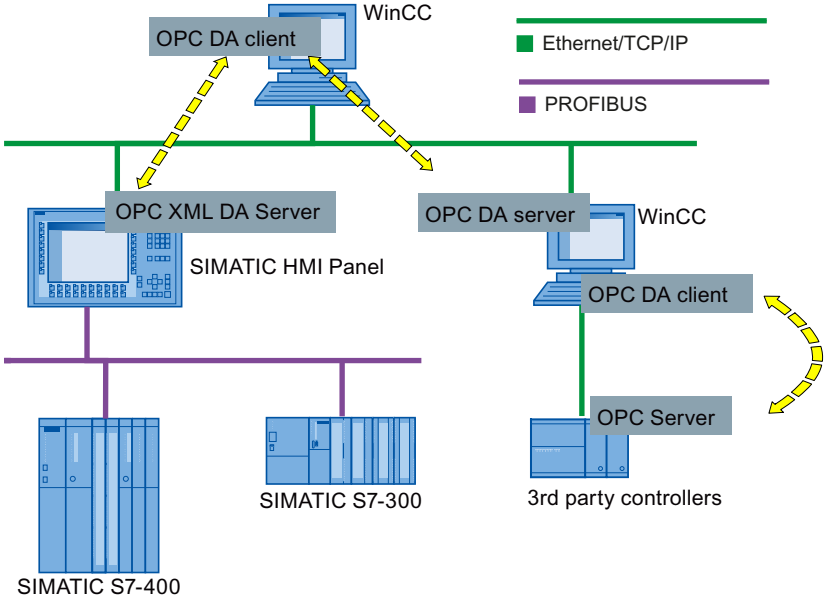
HMI device as OPC client

To use an HMI device as an OPC client, create an "OPC" connection in the WinCC project. The OPC client accesses the tags of the OPC server over this connection. You need a separate connection for each OPC server.

The following exception applies here:

If you want to connect your OPC DA client to multiple panels, you only need one OPC connection to the OPC XML gateway in your project. The multiplexing of this connection to multiple OPC XML connections to individual panels takes place in the OPC XML gateway. You configure the connections to the panels using he OPC XML Manager.

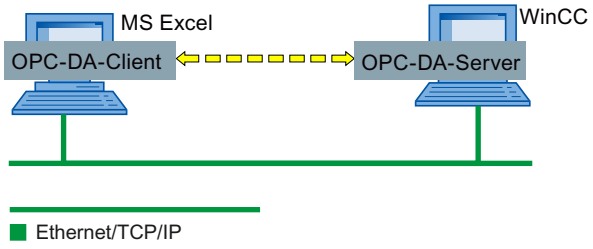
The following figure shows the use of an HMI device as a central operator control and monitoring device:



HMI device as OPC server

An HMI device as OPC server makes the data available to other applications. The applications can run on the same HMI device or on HMI devices in the connected network environment.

The following schematic diagram shows the use of MS Excel as an OPC client that displays process values of the OPC server:



See also

Configuring an HMI device as OPC DA server (Page 18)

Creating a connection to a WinCC OPC DA server (Page 21)

OPC XML Gateway (Page 24)

1.1.2 Security concept of OPC UA

Introduction

The WinCC OPC UA uses the TCP/IP protocol for data exchange. Instance certificates are exchanged for authorization between WinCC OPC UA server and OPC UA client. In addition, you can encode the data transfer.

Security concept

The WinCC OPC UA server and each OPC UA client authorize themselves mutually by exchanging certificates.

By default, the WinCC OPC UA server creates during installation a self signed instance certificate. The server sends this instance certificate to the client. You can alternatively replace this instance certificate with a project-specific instance certificate.

Note

Private key and own certificates

If you have an own certification center, you can create your own certificates and make them available for all communication partners. In this case, delete the instance certificate created by WinCC OPC UA server.

The instance certificate of the client that is allowed to communicate with the server must be stored on both the server and the client in the appropriate directory.

Storage locations of the instance certificates

The instance certificate is stored on the OPC UA server in the following path:

- On PCs: "\ProgramData\Siemens\CoRtHmiRTm\MiniWeb1X.X.X\SystemRoot\SSL"
- On Panels: "\flash\simatic\SystemRoot\SSL"

The instance certificate is stored on the OPC UA client in the following path:

- On PCs: "ProgramData\Siemens\CoRtHmiRTm\OPC\PKI\CA\"
- On Panels: "flash\Simatic\SystemRoot\OPC\PKI\CA\"

If you configure two OPC connections in your project, create the following directories under [...]\OPC\PKI\CA\:

- "Default"
- "!Conn_1"
- "!Conn_2"

Each directory contains the subdirectories "certs", "rejected" and "private". Self-signed instance certificates and private keys are stored under "Default\certs" or "Default\private". The OPC-UA connections only use these certificates.

The client always stores the sent server certificates first in the "rejected" directory and does not allow a connection to the server. To allow connection to the server at the client end, you need to move the certificate in question from the "rejected" directory into the "certs" directory.

The server also initially stores the client certificate in the "rejected" directory under [...] \SystemRoot\SSL. To allow connection to the server at the server end, you need to move the certificate in question from the "rejected" directory into the "certs" directory under [...] \SystemRoot\SSL.

Configuring the OPC UA client for the security mode

If the OPC UA server is running and the OPC UA client was started with an encrypted connection, set up the communication between the OPC UA server and the OPC UA client in security mode.

1. Move the instance certificate sent from the server from the "rejected" directory to the "certs" directory on the OPC UA client.
2. Move the instance certificate sent from the client from the "rejected" directory to the "certs" directory on the OPC UA server.

Once the client and the server have authorized each other, they communicate in security mode.

Note

If you want to use certificates of third-party vendors for the OPC UA client, stop Runtime and copy the third-party certificates and private keys into the directories "!Conn1\certs" and "!Conn1\private". Then copy the certificate from the "certs" directory into the "SSL" directory on the server. Then start Runtime again.

Security settings

The following table lists the security settings supported by the WinCC OPC UA server:

SecurityPolicy	Message Security Mode		
None[1]	None		
Basic128Rsa15 [2]	None [3]	Sign [4]	SignAndEncrypt [5]
1: The certificate exchange is switched off. Every OPC UA client can log on to the WinCC OPC UA server.			
2: Certificate exchange with depth of encryption of 128 bit.			
3: The data packages are exchanged after certificate check unsecured between client and server.			
4: The data packages are signed with the certificates, but not encoded			
5: The data packages are signed with the certificates and encoded			

1.1.3 Configuring an OPC server

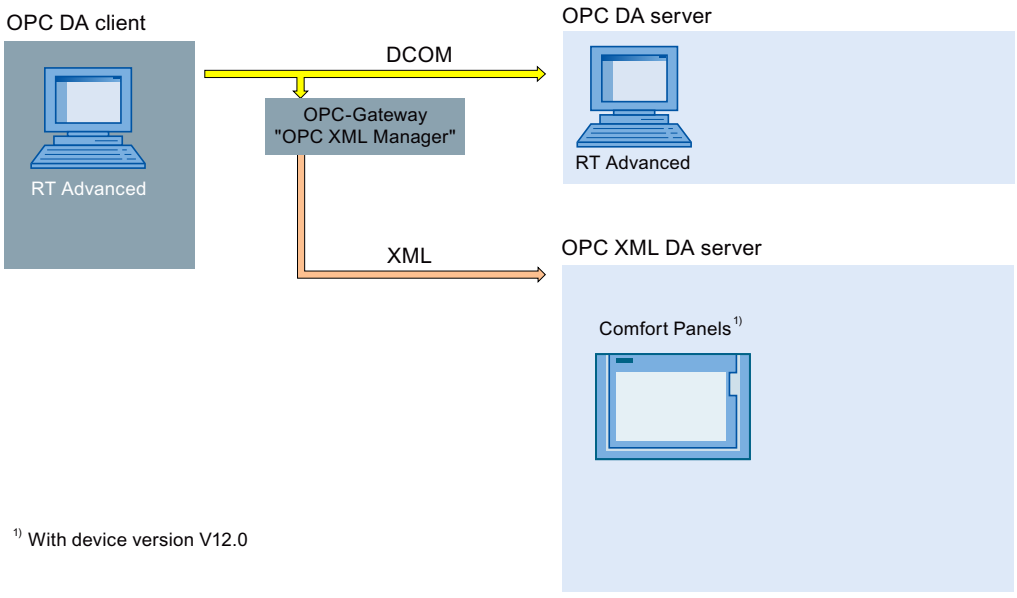
1.1.3.1 Configuring an HMI device as OPC DA server

Introduction

Which OPC server is used depends on the HMI device:

- HMI devices with RT Advanced as OPC DA server
- HMI devices with version V12.0 as OPC-XML-DA server
- HMI devices with version V13.0 as OPC XML DA server

The following figure shows the two methods of accessing an OPC server:



Procedure

To configure an HMI device as an OPC server, proceed as follows:

1. Open the "Runtime settings" of the HMI device in the project tree.
2. Select the "Operate as OPC server" option in the "Runtime settings" under "Services > Write/ read tags".
3. Save the project.
4. Download the project to the HMI device.
5. Start runtime on the HMI device.

Result

The OPC server is accessible. If an OPC client connects to the OPC server, the OPC server on the HMI device is started.

See also

Using OPC in WinCC (Page 14)

Creating a connection to a WinCC OPC DA server (Page 21)

1.1.3.2 Configuring an HMI device as an OPC UA server

Introduction

The following HMI devices can be used as OPC UA servers:

- Comfort Panels
- RT Advanced

Procedure

1. Open the "Runtime settings" of the HMI device in the project tree.
2. Select the "Operate as OPC server" option under "Service" in the "Runtime settings".
3. Configure the server settings at "OPC settings > OPC Unified Architecture Server configuration":
   - Change the "Port number", if required.
   - Activate at least one "SecurityPolicy" and the respective "Message security mode".

   Note

   Unsecured communication between client and server possible

   Use the "none" setting only for test and diagnostics purposes.
4. Save the project.
5. Download the project to the HMI device.
6. Start runtime on the HMI device.

Result

The OPC server is accessible.

If an OPC client connects to the OPC server, the OPC server on the HMI device is started.

1.1.3.3 Configuring DCOM user permissions in Windows

Introduction

The OPC DA client and OPC DA server are DCOM applications, whose security settings must be set in compliance with the DCOM security mechanisms:

- The OPC client needs launch/activation rights and access rights for the OPC DA server.
- The OPC DA server only needs access rights for the OPC DA client

The following must be known on the PCs of the OPC DA server and the OPC DA client respectively:

- The user account for which the OPC DA client is executed

Requirement

You have administrator rights.

Procedure

The procedure for configuring DCOM user rights is described on the Internet on the website of the OPC Foundation (https://opcfoundation.org/).

For more information on assigning user rights, see the Windows documentation.

1.1.4 Configuring an OPC client

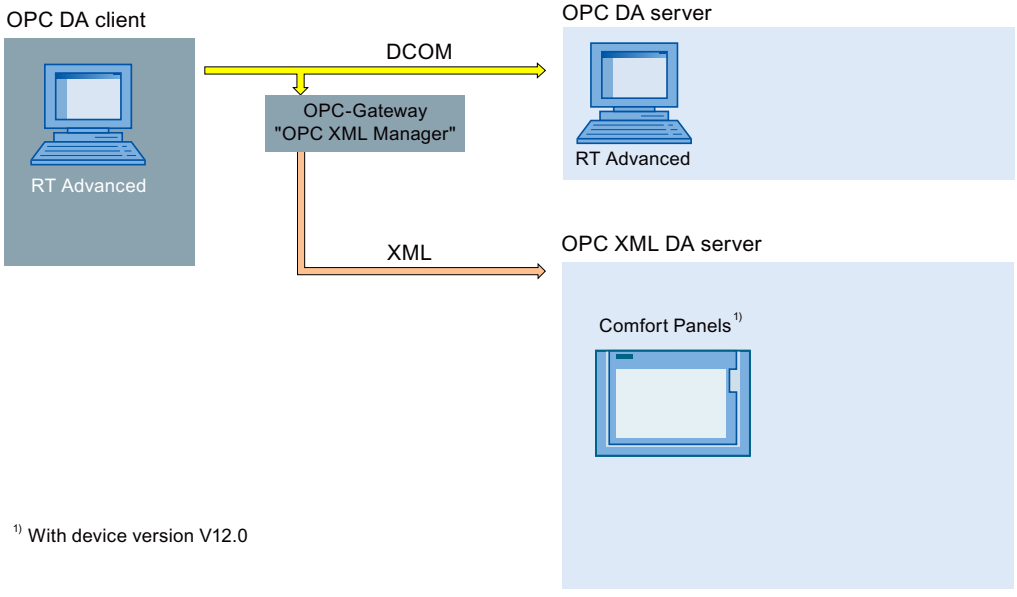
1.1.4.1 Creating a connection to a WinCC OPC DA server

Introduction

For an HMI device to access the OPC server data of another HMI device, you need to create a connection to this OPC server in the WinCC project.

Depending on the target HMI device used, either an OPC DA server or an OPC XML DA server is employed.

The following figure shows the two methods for accessing the OPC servers of the HMI devices:



1) With device version V12.0

If a connection to one or more OPC XML DA servers is created using an OPC XML gateway, also enter the OPC XML DA servers in the "OPC XML Manager".

Requirement

- An HMI device is configured as an OPC server.
- The project on the HMI device is in runtime.
- For connections to a OPC XML DA server: "OPC-XML-Gateway" is installed on the configuration PC and the HMI device with the OPC client.

Procedure

To create a connection to an OPC server of an HMI device, follow these steps:

1. Open the "Connections" editor on the configuration PC in the WinCC project of the OPC client.
2. Create a new connection and enter a meaningful name.
3. Select the entry "OPC" as the "Communication driver".
4. Enter the communication partner under "Parameters > OPC server" in the work area:
   - If the connection communicates directly with the OPC server or with the OPC server on a PC-based HMI device, select the "OPC.SimaticHMI.CoRtHmiRTm" item from the list.
   - If the connection communicates directly with the OPC server of a Windows CE HMI device, select "OPC.Siemens.XML" from the list.
   - If the OPC server is installed on a remote computer, enter the computer's IP address or name under "Remote computer name".

Result

The OPC connection is configured. To access the data on the WinCC OPC DA server, create tags.

See also

Using OPC in WinCC (Page 14)

Configuring an HMI device as OPC DA server (Page 18)

OPC XML Gateway (Page 24)

Configuring an OPC XML Manager (Page 25)

1.1.4.2 Creating a connection to an OPC UA server

Introduction

The OPC UA client can access process data in the hierarchical name space of an OPC UA server.

For the OPC UA client to access the process values of an OPC UA server, the OPC UA server and the OPC UA client authorize each other by exchanging certificates. In addition, you can encode the data transfer.

The OPC UA client usually classifies each certificate of an OPC UA server as a "trustworthy". How an OPC UA server responds to a connection request of the OPC UA client depends on the configuration of the OPC UA server.

In order to establish communication to an OPC UA server, inform yourself from the OPC UA server operator about the following:

- URL of the OPC UA server
- Security settings
- Required certificates

Requirement

URL and security settings of the OPC UA server are known.

Procedure

To create a connection to an OPC UA server, proceed as follows:

1. Open the "Connections" editor on the HMI device.
2. Create a new connection and enter a meaningful name.
3. Select the entry "OPC UA" as the "Communication driver".
4. In the work area, under "Parameters", configure the "OPC server":
   - Specify the "Discovery URL" of the OPC UA server or select the OPC UA server from the list.
   - Select the "Security policy"
   - Select the "Message security mode"

Result

The OPC UA connection is configured. You create tags to access data from the OPC UA server.

See also

Supported OPC UA services of the OPC UA client (Page 28)

1.1.4.3 Accessing process values of an OPC server

Requirement

- The OPC server to be addressed is ready-to-operate and in the "running" status
- A connection to the OPC-Server is created

Procedure

To access process values of an OPC-Server via the OPC connection, follow these steps:

1. On the configuration PC in the project navigation, open the "HMI tags" editor under the HMI device that you use as an OPC client.
2. Create a tag with the same data type as the tag on the OPC server.

3. Select the OPC connection for "Connection".
4. Enter the "Address", or select the desired tag on the OPC server via the object list.

Result

If you launch Runtime on the HMI device, the process value from the OPC server will be written to the tag on the HMI device via the OPC connection.

See also

Permitted data types (OPC) (Page 26)

Access to tags with OPC (Page 27)

1.1.4.4 OPC XML Gateway

Usage

The OPC DA client uses the "OPC XML Gateway" so that the OPC DA client can communicate with the OPC XML DA server. The "OPC XML Gateway" compiles the data in the respective "language" of the corresponding standard. The "OPC XML Gateway" communicates exclusively with the OPC XML DA server which runs on an SIMATIC HMI device.

Installation

To install "OPC XML Gateway", activate the "OPC XML Gateway" entry during installation of WinCC runtime advanced in the component selection. To install "OPC XML Gateway" afterwards, re-execute the setup of WinCC Runtime Advanced.

Proxy setting for the OPC XML Gateway

The configuration settings for the OPC XML Gateway can be found in the "SOPCSRVR.ini" file in the section "[Configuration]". The OPC XML Gateway is configured by default on the PC as the HMI device, so that a proxy server configured in the Internet settings of Internet Explorer is ignored:

NOPROXY=1

If this entry is set to "0", the OPC XML Gateway uses a configured proxy server for HTTP connections.

You can find the "SOPCSRVR.ini" file in the folder "C:\Program Files\ \Siemens\Automation\WinCC RT Advanced".

Note

When data is requested from an HMI device that cannot be accessed over the configured proxy server, the OPC XML Gateway uses a direct connection after a Timeout. The direct connection is set up again on every request for data and slows down OPC communication significantly.

See also

Configuring an OPC XML Manager (Page 25)

Using OPC in WinCC (Page 14)

1.1.4.5 Configuring an OPC XML Manager

Introduction

The OPC XML DA server, to which the OPC DA client has access, is administered in the "OPC XML Manager". The OPC XML Manager can be found in the Windows Start menu under "SIMATIC > OPC-XML-Gateway > OPC XML Manager".

To enter an OPC server, you need the following information:

- Server prefix
  Any string that is used in the name of the OPC tag. Use an abbreviation of the server, for example. You can find the characters permitted for tag names in the reference.
- Name or IP address of the OPC XML DA server

Requirement

"OPC XML Manager" is open.

Procedure

To configure the OPC XML Manager, proceed as follows:

1. To enter a new OPC XML DA server, click on the button "Add".
   The "Add/Edit Webservice" dialog is opened.
2. Enter the server prefix and the name or IP address of the OPC XML DA server.
3. Close the two dialogs with "OK".

Result

The OPC XML DA server is entered. The OPC DA client can now access the data of the OPC XML DA server via the "OPC XML Gateway".

Editing or removing an OPC server

To edit or delete a configured OPC XML DA server, select the required OPC XML DA server. Then click either "Edit" or "Remove" .

See also

OPC XML Gateway (Page 24)

1.1.5 Reference

1.1.5.1 Permitted data types (OPC)

Permitted data types

The following table lists the data types supported by the WinCC OPC servers:

OPC data type	WinCC data type
VT_BOOL	BOOL
VT_I1	CHAR
VT_UI1	BYTE
VT_I2	SHORT
VT_UI2	WORD
VT_UI4	DWORD
VT_I4	LONG
VT_R4	FLOAT
VT_R8	DOUBLE
VT_DATE	DATE
VT_BSTR	STRING

The following table lists the ranges of values of the OPC data types:

OPC data type	Range of values
VT_BOOL	0 or -1
VT_I1	-128 to 127
VT_UI1	0 to 255
VT_I2	-32768 to 32767
VT_UI2	0 to 65535
VT_I4	-2147483648 to 2147483647
VT_UI4	0 to 4294967295
VT_R4	3.402823466 e-38 through 3.402823466 e+38
VT_R8	1.7976931486231e-308 to 1.7976931486231e+308
VT_Date	1 January 100 to 31 December 9999

Special features in communication with the OPC-DA server

The array tag in the OPC-DA server belonging to the area pointer must be of the data type SHORT (VT_I2).

Special features in communication with the OPC-XML server

Array tags are not supported by OPC-XML servers.

1.1.5.2 Access to tags with OPC

Introduction

When accessing tags with OPC, note the following when configuring tags:

- Permitted characters in tag names
- Permitted cycle times (OPC XML)
- Special features of the data type "STRING" (OPC XML)
- Special features of the data type "Date/Time" (OPC XML)

The OPC XML DA server is tested and enabled for eight connections, each with 2 groups and consisting of 35 tags.

Permitted characters in tag names

Only use the following characters in tag names:

- Letters from "a" to "z" (no umlauts)
- Numbers from "0" to "9"
- Special characters: "-" and "_"

Permitted cycle times (OPC XML)

OPC XML connections are designed for the exchange of small volumes of data:

- For this reason use cycle times that are greater than one second
- In general, only request a small number of tags; a maximum of 30 per screen

Special features of the data type "STRING" (OPC XML)

Only valid ASCII values from 0x20hex to 0x7Fhex are supported in the data type "STRING".

Special features of the data type "Date/Time" (OPC XML)

Values of the data type "Date/Time" are always expected as UTC (Universal Time Coordinated) by the OPC-Gateway. If a tag of the type "Date/Time" is read by the OPC client, the returned value is a time in UTC. If a value is written to the tag, the value is treated as UTC. The time including the time zone and daylight saving time are shown as "local time" on an HMI device.

Example:

The time zone GMT+1 and daylight saving time is set on the HMI device.

OPC DA client (UTC time): 01.01.2005 16:00

Display on the HMI device (OPC server): 01.01.2005 18:00

1.1.5.3 Supported OPC UA services of the OPC UA client

OPC UA Services support

The OPC UA client supports the following OPC UA services:

- SecurityPolicy - Basic128Rsa15
- SecurityPolicy - Basic256
- SecurityPolicy - None
- DataAccess ClientFacet

You can additional information about OPC UA services in the "OPC UA Part 3 - Address Space Model 1.01 Specification" document under "§5.6".

Explanation of the security settings

The following table lists the security settings supported by the OPC UA client:

SecurityPolicy	Message Security Mode	
None[1]	None	
Basic128Rsa15[2]		SignAndEncrypt[4]
Basic256[3]	Sign	
1: The certificate exchange is disabled. Every OPC UA client can log on to the WinCC OPC UA server. This setting can be disabled on each OPC UA server.		
2: Certificate exchange with depth of encryption of 128 bit.		
3: Certificate exchange with depth of encryption of 256 bit.		
4: The data packages are signed with the certificates and encoded.		

Note

Unsecured communication between client and server possible

Use the "none" setting only for test and diagnostics purposes.

For a secure communication between client and server, use in operating mode at least the following settings:

- SecurityPolicy: Basic128Rsa15
- Message Security Mode: Sign

See also

Creating a connection to an OPC UA server (Page 22)

1.2 OPC for Runtime Professional

1.2.1 Basics

1.2.1.1 OPC

OPC refers to standardized multi-vendor software interfaces for data exchange in automation engineering.

The OPC interfaces provide a standardized environment in which devices and applications from various manufacturers can be linked.

OPC is based on the Windows technology COM (Component Object Model) and DCOM (Distributed Component Object Model).

In addition, OPC XML DA offers a further software interface that is based on the Internet standards XML, SOAP, and HTTP.

OPC UA (Unified Architecture) is the technology succeeding OPC. OPC UA is platform-independent and can use different reports as a communication medium.

See also

Using OPC in WinCC (Page 30)

OPC Specifications (Page 29)

Compatibility (Page 30)

1.2.1.2 OPC Specifications

OPC specifies interfaces to gain access to the following objects in WinCC:

- Process values (OPC Data Access 2.05a, 3.0; OPC XML Data Access 1.01; OPC UA 1.02)
- Logged process values (OPC Historical Data Access 1.20; OPC UA 1.02)
- Historical alarms (OPC Historical Alarms and Events v1.10)*
- Alarms (OPC Alarms and Events 1.10); OPC UA Alarms and Conditions 1.02

For detailed information about the individual OPC specifications, refer to the website of OPC Foundation.

See also

Using OPC in WinCC (Page 30)

OPC Foundation (www.opcfoundation.org)

1.2.1.3 Compatibility

Support of the mentioned specifications is checked regularly by the "Compliance Test Tool" (CTT) of the OPC Foundation. Interoperability with OPC products of other manufacturers is ensured through the participation in "OPC Interoperability Workshops".

The test results submitted are published on the website of the OPC Foundation. The results can be called up from there using the search term "OPC Self-Certified Products".

See also

OPC (Page 29)

1.2.1.4 Licensing

Licensing

For the operation of one of the supported WinCC OPC servers, the following license must be installed:

- WinCC Runtime Professional

1.2.1.5 Using OPC in WinCC

Introduction

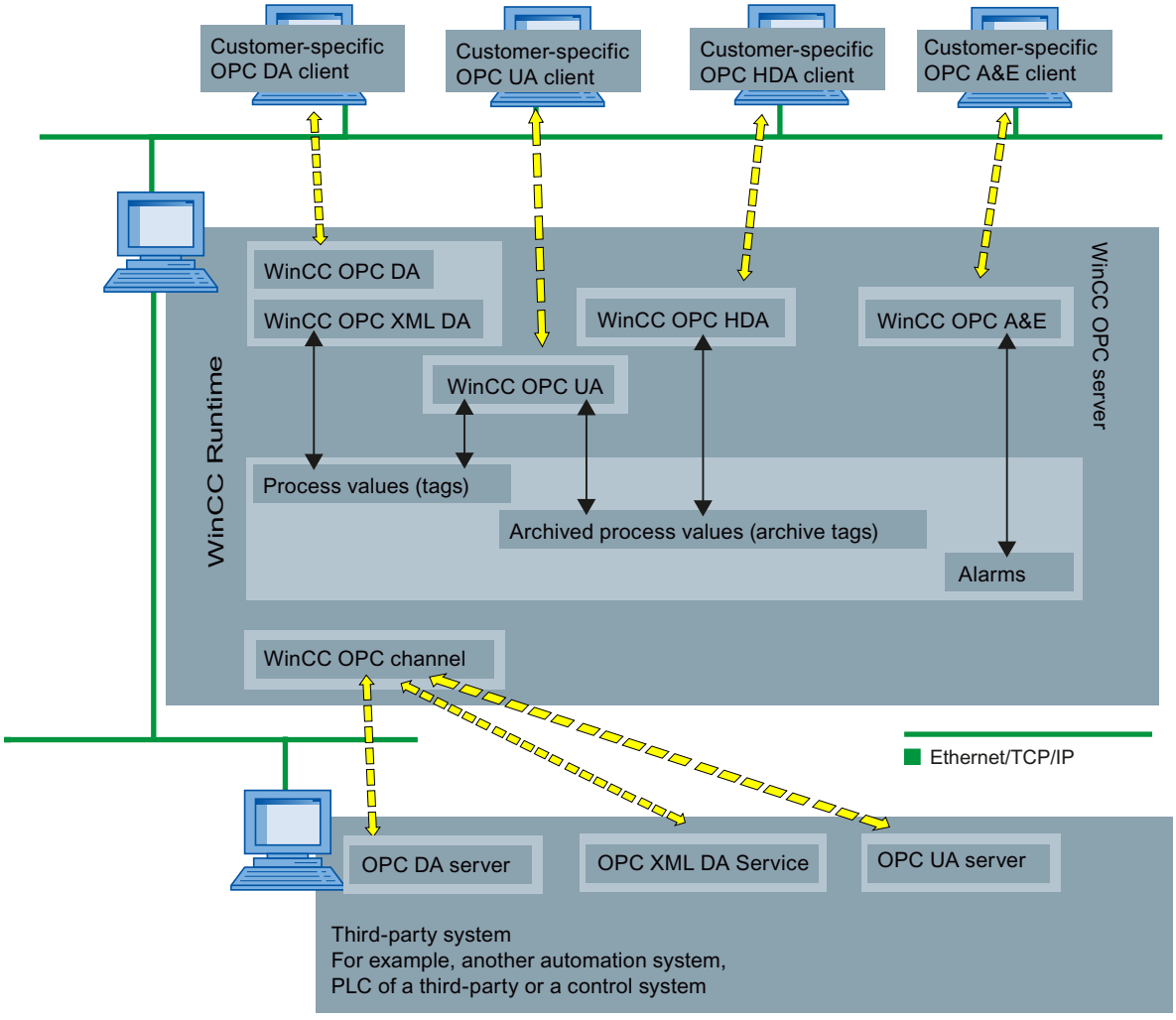
On a HMI device with "WinCC Runtime Professional" there are server for the following OPC interfaces available:

- OPC Data Access / OPC XML Data Access: Access to the data management of WinCC
- OPC Historical Data Access: Access to the WinCC log system
- OPC Alarms&Events: Access to the WinCC alarm system
- OPC Unified Architecture: Access to the WinCC data management, alarm system and log system

In addition, WinCC contains as standard an OPC channel that can access corresponding servers with data access support as client via OPC DA, OPC XML DA or OPC UA.

OPC communication concept of WinCC

The following screen shows the OPC communication concept of WinCC:



See also

Licensing (Page 30)

Functionality of the WinCC OPC DA Server (Page 33)

Functionality of WinCC OPC XML DA server (Page 33)

Functionality of the WinCC OPC HDA server (Page 36)

Functionality of the WinCC OPC A&E server (Page 44)

Principle of operation the WinCC OPC UA Server (Page 61)

Creating a connection to an OPC server (Page 79)

1.2.2 Configuration

1.2.2.1 Basic settings for OPC

Introduction

For the following WinCC OPC servers, you configure the basic settings on the configuration PC for OPC in the HMI device runtime settings:

- OPC HDA
- OPC A&E
- OPC UA

Basic settings for OPC

For HMI devices with Runtime Professional, you define the following in the runtime settings:

- OPC settings
  You configure the following in this category:
  - Exception rules for the validation of writing access to log data
  - Security concept of the WinCC OPC UA server
- Alarms
  In this category you configure the mapping of the WinCC alarm system for the alarm behavior of the following WinCC OPC servers:
  - OPC A&E

See also

Configure WinCC OPC HDA Server (Page 43)

Configure WinCC OPC A&E server (Page 54)

Configure WinCC OPC UA Server (Page 77)

1.2.3 WinCC OPC server

1.2.3.1 Accessibility of WinCC OPC servers

Client-specific OPC clients data from the WinCC data management are available for the WinCC OPC servers, for example, process values or alarms.

In order to access a WinCC OPC server, you must activate the related WinCC project on the HMI device.

1.2.3.2 WinCC OPC DA Server

Functionality of the WinCC OPC DA Server

Principle of operation

The WinCC OPC DA server enables any OPC DA client to access process values from the data management of WinCC.

Supported specifications

The WinCC OPC A&E Server supports OPC Data Access 2.05a and 3.0.

ProgID of the WinCC OPC DA servers

You access the WinCC OPC DA server via the following ProgID: "OPCServer.WinCC_SCADA".

Quality codes

The "Quality Code" delivers information on the status and quality of the process values. For more information, refer to the "OPC Data Access 3.0" specification.

See also

Using OPC in WinCC (Page 30)

Accessibility of WinCC OPC servers (Page 32)

Licensing (Page 30)

1.2.3.3 WinCC OPC XML DA server

Functionality of WinCC OPC XML DA server

Principle of operation

WinCC's OPC XML DA Server is implemented as a web service of Microsoft Internet Information Server (IIS). The OPC XML DA client accesses the OPC XML DA server via an internet protocol to which process values from the data management of WinCC are available.

Supported specifications

The WinCC OPC XML DA Server supports the specification OPC XML Data Access 1.01.

"OPC XML Data Access" is a standard that supports communication over the Internet by means of platform-independent protocol (SOAP/XML). The data access via OPC XML Data Access has an OPC Data Access leaning on the scope of operation.

URL of the WinCC OPC XML DA servers

You access the WinCC OPC XML DA server via the following URL: "http://<xxx>/WinCC-OPC-XML/DAWebservice.asmx"

<xxx>: IP address or DNS name of the OPC XML DA server

Note

If automatic detection of settings is enabled on a PC for the internet options under Connections > LAN Settings", access to OPC XML DA through the web service will be slowed considerably.

Quality codes

The "Quality Code" delivers information on the status and quality of the process values. For more information, refer to the "OPC XML Data Access1.01" specification.

Special features of tags of "String" type

If you use tags of "string" type that logically stand for floating point values, there can be problems when OPC clients want to write and read.

An OPC client writes to a string tag and does not enter the new value in the form of a string but as float, double or decimal. The comma that indicated the decimal value (in German) can be lost. This will result in a wrong value. This behavior also occurs when string tags are read if the value being read is requested in float, double, or decimal format.

Only use the respective floating point tags for the floating point values. Get string tags only in string format.

See also

Using OPC in WinCC (Page 30)

Accessibility of WinCC OPC servers (Page 32)

Installing WinCC OPC XML DA Server (Page 34)

Installing WinCC OPC XML DA Server

Introduction

To install the OPC XML DA server automatically in the framework of the WinCC setup, the following settings on the HMI device must be fulfilled:

- Internet Information Service (IIS) is installed
- Microsoft .NET-Framework 4.5 or higher is installed

If the requirements are not met, you must install the OPC XML DA server separately after you have installed the IIS and the .Net-Framework . To avoid problems during installation, follow in any case the steps described below.

Also note the installation requirements for WinCC Runtime Professional.

Installation of the Internet Information Server (IIS)

Under Windows Server 2008 R2 / 2012 R2 / 2016 , configure the settings of the Server Manager using the "Web server (IIS)" role in the corresponding role services.

Enable the following settings:

- Web management tools:
  - IIS management service
  - IIS management console
  - IIS management scripts and tools
- WWW Services > Common HTTP Features:
  - Standard document
  - Static content
- WWW services > Application development features:
  - .NET extension capability
  - ASP.NET
  - ISAPI extensions
  - ISAPI filters
- WWW services > Security:
  - Requirement filtering
  - Standard authentication
  - Windows authentication

Note

Always install Microsoft Internet Information Service (IIS) with ASP.NET

Always install ASP.NET together with Microsoft Internet Information Service (IIS).

Note

The web service of the WinCC OPC XML DA server communicates via port: 80 (HTTP).

Make sure that the firewall rule "WWW services (HTTP)" is activated and enabled for the required network areas.

Installation of the WinCC OPC XML server

To configure the OPC XML DA server on the HMI device, proceed as follows:

1. Install the "Microsoft Internet Information Service (IIS)".
2. Install the "Microsoft .NET-Framework" or higher.

3. Install the WinCC OPC XML DA Server via the WinCC setup.
4. Test the installation on the OPC server:
   – Launch Internet Explorer on the PC running as the WinCC OPC XML Server.
   – On the address bar enter the URL "http://localhost/WinCC-OPC-XML/DAWebservice.asmx".

   If a page with the supported operations is displayed in Internet Explorer, the installation was successful.

See also

Functionality of WinCC OPC XML DA server (Page 33)

1.2.3.4 WinCC OPC HDA Server

Functionality of the WinCC OPC HDA server

Principle of operation

The WinCC OPC HDA server enables any OPC HDA client to access process values from the data management of the WinCC log system.

Supported specifications

The WinCC OPC HDA Server supports the OPC Historical Data Access 1.20 specification.

ProgID of the WinCC OPC HDA server

You access the WinCC OPC HDA server via the following ProgID:
"OPCServerHDA.WinCC_SCADA".

Quality codes

The "Quality Code" delivers information on the status and quality of the raw data. For more information, refer to the "OPC Historical Data Access 1.20" specification.

Documentation contents

The WinCC OPC HDA Server documentation contains the following information:

- An overview of the data structure
- An overview of supported attributes, aggregates, and functions

For more information about the OPC HDA Server, refer to the "OPC Historical Data Access 1.20" specification.

See also

Using OPC in WinCC (Page 30)

Accessibility of WinCC OPC servers (Page 32)

Overview of the supported attributes (Page 40)

Overview of the supported assemblies (Page 41)

Overview of the supported interfaces and functions (Page 42)

Configure WinCC OPC HDA Server (Page 43)

Supported Write-Accesses

Introduction

By default, any OPC HDA client can change process values in a data log. The tag log validates the data and inserts them to their corresponding time stamp.

If you frequently change a lot of process values, you can use the configuration dialog on the conifguration PC to deactivate the validation of the process values during writing to the data log. You speed up the large writing of process values in this way. The OPC HDA client must sort the data themselves chronologically, otherwise the data is not accepted.

Note

Deactivate writing access

You can completely deactivate the writing via the WinCC OPC HDA server from log data via the (user-defined) WinCC setup.

Write Accesses

An OPC HDA client does not have generally writing access to the following data logs.

- Compressed logs
- Data logs in main memory
- Swapped logs

The following table shows the supported writing accesses for a process value log on the hard disk of a HMI device:

Action	Writing in process value log allowed?
Adding process values later	Yes [1)]
Adding process values in Runtime	Yes [2)]
Inserting future process values	No
Replace process values (Replace)	Yes
Deleting process values	Yes [1)]
1: If the time period is contained in the circular log.	
2: The process value is added in the data buffer currently valid for the process value log.	

See also

Special features of the OPC HDA server in WinCC for acyclic logging (Page 38)

Configure WinCC OPC HDA Server (Page 43)

Special features of the OPC HDA server in WinCC for acyclic logging

Introduction

In WinCC. tags are logged cyclically or acyclically. The WinCC OPC HDA Server works differently depending on the logging method for tags:

- For all cyclically logged values, the OPC HDA Server operates in conformity to the HDA specification of the OPC foundation. The OPC aggregates are linearly interpolated.
- Acyclically logged tags are not included in the HDA specification of the OPC Foundation. The OPC aggregates are interpolated incrementally. If a tag value has not changed for a long period of time, no data are available during this time due to the interpolation.
  To obtain valid data in spite of this, take note of the following particularities:

Note

The OPC HDA Server is not OPC-compliant for acyclically logged tags. The OPC Foundation's HDA specification does not cover acyclically logged tags and therefore does not support log servers for acyclically logged tags. The supported aggregates are calculated in conformity to the OPC HDA specification. No non-explicitly called functions are supported.

Configuration of acyclically logged tags

When configuring acyclically logged tags, enable the "Log after segment change" option. This enters the most recent valid value in the the new log when a segment changes.

Supported aggregates of the WinCC OPC HDA Server for acyclically logged tags

The OPC HDA Server supports the following aggregates:

- OPCHDA_MINIMUM
- OPCHDA_MAXIMUM
- OPCHDA_AVERAGE
- OPCHDA_END
- OPCHDA_INTERPOLATIVE
- OPCHDA_TIMEAVERAGE
- OPCHDA_TOTAL

- OPCHDA_DURATIONGOOD
- OPCHDA_PERCENTGOOD

Supported functions of the WinCC OPC HDA Server for acyclically logged tags

- ReadRaw only with "Boundings". To find the last value actually saved in an area where value changes are not logged, always use "Boundings" when executing "ReadRaw" on a tag.
- ReadProcessed
- DeleteRaw
- DeleteAtTime
- Insert
- InsertReplace
- Replace

Calculating the aggregates for acyclically logged tags

The calculation of the aggregates is based on an extended data record of raw data. The extended data record contains not only actually stored values, but also virtual data points for calculation. The WinCC OPC HDA Server prepares the raw data received according to the requirements of "ReadProcessed". The virtual data points needed for the calculation are formed from the bordering real data points. The following significant points are generated for virtual data points:

- Value for "StartTime"
- Value for "EndTime"
- Value for interval limits

Example

The values for "00:59:00", "01:02:00" and "01:03:00" are stored for an acyclical logging tag. An OPC HDA Client uses "ReadProcessed" to request an aggregate with the following parameters:

- StartTime = 01:00:00
- EndTime = 01:04:00
- Interval = 00:02:00

Note

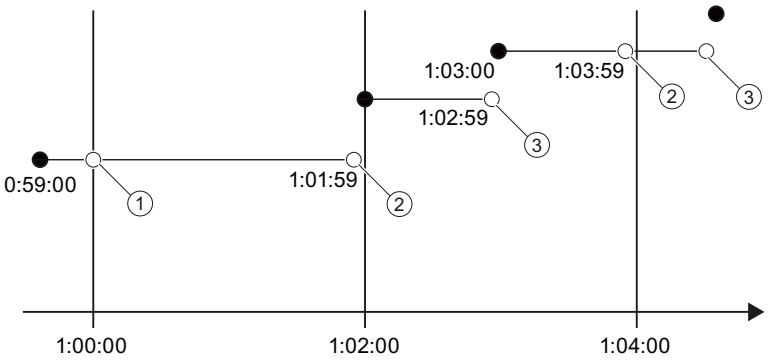
When generating virtual values at limits ("EndTime"/"Interval"), the time frame used for the calculation is always 1 µs less than the time stamp at the limit.

For the sake of clarity, a delta of 1 second is used in the following table. The OPC server uses the following raw data to calculate the aggregate:

Number	Time stamp	Real stored values	Generated virtual values
1	00:59:00	1.00	
2	01:00:00		1.00

Number	Time stamp	Real stored values	Generated virtual values
3	01:01:59		1.00
4	01:02:00	2.00	
5	01:02:59		2.00
6	01:03:00	3.00	
7	01:03:59		3.00

The following figure elucidates the example:



● Real values
○ Virtual values
① Interval start
② Interval end
③ Value change

See also

Supported Write-Accesses (Page 37)

Overview of the supported attributes (Page 40)

Overview of the supported assemblies (Page 41)

Overview of the supported interfaces and functions (Page 42)

Overview of the supported attributes

The following table contains the attributes supported by the WinCC OPC HDA Server. For more information, refer to the "OPC Historical Data Access 1.20" specification.

Attributes

Attribute	Attribute ID	Description
ItemID	OPCHDA_ITEMID	Indicates the WinCC logging tag to be accessed.
Item Datatype	OPCHDA_DATA_TYPE	Indicates the data type of the WinCC logging tag.

Attribute	Attribute ID	Description
Description	OPCHDA_DESCRIPTION	Returns a description of the WinCC logging tag. You define the description in the "HMI Tags" editor.
Engineering Units	OPCHDA_ENG_UNITS	Sets the display of measurement units. You define the display in the "HMI Tags" editor.

See also

Functionality of the WinCC OPC HDA server (Page 36)

Special features of the OPC HDA server in WinCC for acyclic logging (Page 38)

Overview of the supported assemblies

The following table lists the aggregates supported by the WinCC OPC HDA Server. For more information, refer to the "OPC Historical Data Access 1.20" specification.

Assemblies

Assembly	Description
OPCHDA_COUNT	Detects the raw data count for the specified time interval.
OPCHDA_START	Detects the initial value of the raw data at the beginning of the start time interval.
OPCHDA_END	Detects the end value of the raw data at the end time interval.
OPCHDA_AVERAGE	Detects the raw data mean value for the specified time interval.
OPCHDA_TIMEAVERAGE	Detects the time-weighted raw data mean value for the specified time interval.
OPCHDA_TOTAL	Returns the sum total value for the specified time interval.
OPCHDA_STDEV	Detects the raw data standard deviation for the specified time interval.
OPCHDA_MINIMUMACTUALTIME	Returns the minimum value of the raw data time stamp for the specified time interval.
OPCHDA_MINIMUM	Detects the raw data minimum value for the specified time interval.
OPCHDA_MAXIMUMACTUALTIME	Detects the maximum value and the raw data time stamp for the specified time interval.
OPCHDA_MAXIMUM	Detects the raw data maximum value for the specified time interval.
OPCHDA_DELTA	Detects the difference between the first and the last value of the raw data for the specified time interval.
OPCHDA_REGSLOPE	Detects the slope of the regression line of the raw data for the specified time interval.
OPCHDA_REGCONST	Detects the regression line value of the raw data at the starting point.
OPCHDA_REGDEV	Detects the slope of the regression line of the raw data for the specified time interval.
OPCHDA_VARIANCE	Detects the raw data variance for the specified time interval.

Assembly	Description
OPCHDA_RANGE	Detects the difference between OPCHDA_MAXIMUM and OPCHDA_MINIMUM of the raw data for the specified time interval.
OPCHDA_DURATIONGOOD	Detects the period of time in which the quality of the raw data was good. The period is indicated in seconds.
OPCHDA_DURATIONBAD	Detects the period of time in which the quality of the raw data was bad. The period is indicated in seconds.
OPCHDA_PERCENTGOOD	Detects the percentage of the raw data of good quality.
OPCHDA_PERCENTBAD	Detects the percentage of the raw data of bad quality.
OPCHDA_WORSTQUALITY	Detects the raw data worst quality for the specified time interval.

See also

Functionality of the WinCC OPC HDA server (Page 36)

Special features of the OPC HDA server in WinCC for acyclic logging (Page 38)

Overview of the supported interfaces and functions

The WinCC OPC HDA Server supports the following interfaces:

- IOPCHDA_Server
- IOPCHDA_SyncRead (no "ReadModified" method)
- IOPCHDA_SyncUpdate
- IOPCHDA_AsyncRead (no "ReadModified" method)
- IOPCHDA_AsyncUpdate
- IOPCCommon

For more information about the interfaces, refer to the "OPC Historical Data Access 1.20" specification.

See also

Functionality of the WinCC OPC HDA server (Page 36)

Special features of the OPC HDA server in WinCC for acyclic logging (Page 38)

Configure WinCC OPC HDA Server

Introduction

On the configuration PC, define in the configuration settings for the WinCC OPC HDA server whether specific users or OPC HDA clients may change process values in the data logs without validation.

Note

The configuration settings apply for the WinCC OPC HDA server as well as for the OPC UA server.

Note

Deactivated writing access

If the writing access of OPC HDA clients was deactivated on the data logs in (user-defined) WinCC setup, these settings are ignored.

Procedure

To configure WinCC OPC HDA server, proceed as follows:

1. Open the "Runtime settings" of the HMI device in the project tree.
2. Define under "OPC settings" which users or OPC clients may change process values in the data logs without validation:
   - To fully disable validation, select "Disable validation of the data during write access for selected OPC clients".
   - To deactivate the validation for single users or OPC clients, double-click "Insert" in the corresponding table. Enter the name of the user or the OPC client.
   - To re-select the validation for a single user or OPC client, clear the option in front of the respective user or OPC client.

See also

Functionality of the WinCC OPC HDA server (Page 36)

Basic settings for OPC (Page 32)

1.2.3.5 WinCC OPC A&E server

Functionality of the WinCC OPC A&E server

Principle of operation

The WinCC OPC A&E server enables the access to alarms of the WinCC alarm system. The OPC A&E Client is kept informed of status changes for WinCC alarms via Subscriptions. To avoid displaying all alarms and attributes on the OPC A&E client, configure in the Subscription filter.

Supported specifications

OPC Alarm & Events is a specification for the transmission of alarms and events. The WinCC OPC A&E Server supports the OPC Alarms&Events 1.10 specification.

In addition, the WinCC OPC A&E Server supports access to historical alarms in accordance with the Siemens draft "OPC Historical Alarms and Events v1.10".

ProgID of the WinCC OPC A&E server

You access the WinCC OPC A&E server via the following ProgID: "OPCServerAE.WinCC_SCADA".

Quality codes

The "Quality Code" delivers information on the status and quality of an alarm. For more information, refer to the "OPC Alarms&Events 1.10" specification.

OPC A&E Server mapping rules

To map the WinCC alarm system on OPC Alarms&Events, the WinCC OPC A&E server supports three mapping rules "Mode 1" to "Mode 3".

"Mode 1" and "Mode 2" are supported because of compatibility reasons to WinCC OPC A&E server version 3.52 and earlier. "Mode 1" and "Mode 2" do not have a restricted scope of operation.

"Mode 3" widens the restricted scope of operation of "Mode 1" and "Mode 2".

Documentation contents

The WinCC OPC A&E Server documentation contains the following information:

- A mapping of the alarm system on OPC A&E
- An overview of supported attributes

For more information, refer to the "OPC Alarm&Events 1.10" specification.

See also

Using OPC in WinCC (Page 30)
Accessibility of WinCC OPC servers (Page 32)
OPC A&E Server mapping rules (Page 45)
Mapping rule "Mode 1": (Page 50)
Mapping rules "Mode 2": (Page 51)
Mapping rules "Mode 3": (Page 51)
Restrictions for Mode 1 and Mode 2 (Page 52)
Access to logged events (Page 56)

OPC A&E Server mapping rules

General rules

OPC A&E Server mapping rules

During the configuration of the WinCC message system, settings are made to determine which process events generate a message. This message is shown as an alarm in OPC A&E.

The following table shows the most important parameters of the alarm and how the WinCC alarm system prepares the information:

OPC	WinCC message system	Type of event
Source	Source where the alarm was issued. The source has the format "<server prefix>::@LOCALMACHINE::".	S, C
Time	Time of the event. Issues a time stamp in UTC (Universal Time Coordinated).	S, C
EventType	Type of event. The WinCC OPC A&E Server supports "Simple Events" and "Condition Related Events".	S, C
Severity	Priority of the alarm from WinCC	S, C
EventCategory	Category of the alarms. For more information on this topic, refer to "Displaying Message Classes and Types".	S, C
Message	Alarm text for the corresponding alarm number.	S, C
ConditionName	Specified text that is delivered to the alarm additionally. Which text is returned, depends on the configured display rules: • "Mode 1" and "Mode 2": Alarm number. • "Mode 3": Alarm method, for example "Analog alarm".	C
SubConditionName	Corresponds "ConditionName", because WinCC supports no Multi-State Conditions.	C
ChangeMask	Changed status of the alarm. You can find additional information under "State machine".	C
NewState	Status of the alarm. You can find additional information under "State machine".	C

OPC	WinCC message system	Type of event
ConditionQuality	Returns the quality of the message. For more information, refer to "Quality Codes".	C
AckRequired	Indicates whether the message requires acknowledgment.	C
ActiveTime	Returns the time stamp for received messages.	C
EventAttribute	Lists the attributes required for the respective message. For more information, refer to "Attributes of the WinCC Message System".	C
Quality	Returns the quality code of the message.	C
Cookie	Returns the cookie from the OPC A&E Server.	C
ActorID	Returns the logged user.	C
S = "Simple Event" C = "Condition Related Event"		

See also

State machine (Page 47)

Quality Codes for OPC A&E (Page 48)

Attributes (Page 49)

Priorities (Page 53)

Alarm classes and alarm methods (Page 53)

OPC A&E types of event (Page 46)

OPC A&E types of event

Introduction

The WinCC OPC A&E Server supports "Condition Related Events" and "Simple Events".

Condition Related Events

"Condition Related Events" are alarms triggered by a condition. such as the limit violation of a tag, which is then output as an alarm on the HMI device. A "Condition Related Event"contains additionally the "Alarm" attribute.

By default, each alarm as "Condition Related Event" is sent to the OPC A&A client.

Simple Events

As "Simple Events" all other alarms are treated, which cannot be assigned to the "Condition Related Event" type of event.

In order for an alarm to be treated as a "Simple Event", the following prerequisites must be met when the alarm class is configured:

- "Acknowledgment Came In" is not activated.
- "Alarm without outgoing state" is activated.

Note

Note the following while using redundant systems:

"Simple Events" that are connected to internal tags are sent twice during tag comparison.

The first alarm is triggered by master, the second by the standby server.

See also

State machine (Page 47)

OPC A&E Server mapping rules (Page 45)

Mapping rule "Mode 1": (Page 50)

Mapping rules "Mode 2": (Page 51)

Mapping rules "Mode 3": (Page 51)

State machine

Introduction

For WinCC, the state machine is how an alarm is displayed and processed from "incoming" to "outgoing". On the WinCC OPC A&E Server, this alarm status is managed in the "ChangeMask" and "NewState" parameters.

ChangeMask

The "ChangeMask" parameter keeps track of where the alarm status changed. Possible parameter values are:

- OPC_CHANGE_ACTIVE_STATE
- OPC_CHANGE_ENABLE_STATE
- OPC_CHANGE_ACK_STATE

NewState

The "NewState" parameter indicates the alarm status after a change. Possible parameter values are:

- OPC_CONDITION_ACTIVE
- OPC_CONDITION_ENABLED
- OPC_CONDITION_ACKED

Overview

The following table shows how WinCC alarm statuses are mapped to the "NewState" and "ChangeState" OPC parameters:

WinCC	NewState	ChangeState
Incoming alarm	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
Outgoing alarm acknowledged	OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
Outgoing alarm without acknowledgment	OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
Acknowledged alarms (alarm pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Acknowledged alarms (alarm no longer pending)	OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Locked alarm	-	OPC_CHANGE_ENABLED_STATE
Unlocked alarm	OPC_CONDITION_ENABLED	OPC_CHANGE_ENABLED_STATE
Incoming, acknowledged alarm	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
Incoming, outgoing alarm with acknowledgment	OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Incoming, outgoing alarm without acknowledgment	OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Alarm acknowledged by the system (alarm pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Alarm acknowledged by the system (alarm no longer pending)	OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Emergency-acknowledged alarm (alarm pending)	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
Emergency-acknowledged alarm (alarm no longer pending)	OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE

See also

OPC A&E types of event (Page 46)

Quality Codes for OPC A&E

The "Quality Code" is required to evaluate the status and quality of an alarm.

Quality codes

OPC A&E's quality codes are presented in the following table:

Code	Quality	Status
0xC0	OPC_GOOD	OK
0x40	OPC_UNCERTAIN	Discrepancies, resulting from delayed acknowledgment, for example
0x00	OPC_BAD	Connection to source is interrupted

Attributes

The following table lists the OPC attributes of the WinCC message system. You can configure the attributes in the "Alarms" editor. Some attributes are intended for internal use in WinCC only and are therefore not relevant to an OPC A&E Client. These attributes are not listed.

OPC attribute	WinCC message system	Data type
CLASSNAME	Returns the alarm class name.	VT_BSTR
TYPENAME	Returns the alarm method type name.	VT_BSTR
FORECOLOR	Returns the text color for received, sent and acknowledged alarms.	VT_I4
BACKCOLOR	Returns the background color for received, sent and acknowledged alarms.	VT_I4
FLASHCOLOR	Returns the flashing color.	VT_I4
FLAGS	Indicates whether the message requires acknowledgment.	VT_I4
TEXT01	Returns the content of UserTextBlock01.	VT_BSTR
TEXT02	Returns the content of UserTextBlock02.	VT_BSTR
TEXT03	Returns the content of UserTextBlock03.	VT_BSTR
TEXT04	Returns the content of UserTextBlock04.	VT_BSTR
TEXT05	Returns the content of UserTextBlock05.	VT_BSTR
TEXT06	Returns the content of UserTextBlock06.	VT_BSTR
TEXT07	Returns the content of UserTextBlock07.	VT_BSTR
TEXT08	Returns the content of UserTextBlock08	VT_BSTR
TEXT09	Returns the content of UserTextBlock09	VT_BSTR
TEXT10	Returns the content of UserTextBlock10	VT_BSTR
PROCESSVALUE01	Returns the content of ProcessValueBlock01	VT_VARIANT
PROCESSVALUE02	Returns the content of ProcessValueBlock02	VT_VARIANT
PROCESSVALUE03	Returns the content of ProcessValueBlock03	VT_VARIANT
PROCESSVALUE04	Returns the content of ProcessValueBlock04	VT_VARIANT
PROCESSVALUE05	Returns the content of ProcessValueBlock05	VT_VARIANT
PROCESSVALUE06	Returns the content of ProcessValueBlock06	VT_VARIANT
PROCESSVALUE07	Returns the content of ProcessValueBlock07	VT_VARIANT
PROCESSVALUE08	Returns the content of ProcessValueBlock08	VT_VARIANT
PROCESSVALUE09	Returns the content of ProcessValueBlock09	VT_VARIANT
PROCESSVALUE10	Returns the content of ProcessValueBlock10	VT_VARIANT
STATETEXT	Returns the status alarm	VT_BSTR
INFOTEXT	Returns the information text for the alarm	VT_BSTR

OPC attribute	WinCC message system	Data type
LOOPINALARM	Indicates if loop-in alarm has been configured.	VT_I4
CLASSID	Returns the alarm class ID	VT_I4
TYPEID	Returns the message type ID	VT_I4
MODIFYSTATE	Returns the value of the status tag of the alarm	VT_I4
AGNR	Returns the number of the automation system that generated the alarm	VT_I2
CPUNR	Returns the number of the CPU that generated the alarm	VT_I2
DURATION	Indicates the period of time between alarm received, sent and acknowledged	VT_I4
COUNTER	Returns the number of alarms after the start of Runtime	VT_I4
QUITSTATETEXT	Indicates whether the alarm has been acknowledged	VT_BSTR
QUITCOUNT	Returns the number of active, unacknowledged alarms	VT_I4
PARAMETER	Outputs the message parameter. (Image of the message configuration)	VT_BSTR
BLOCKINFO	Returns the current content of the alarm block	VT_BSTR
ALARMCOUNT	Outputs the number of alarms pending	VT_I4
LOCKCOUNT	Returns the number of locked alarms	VT_I4
PRIORITY	Indicates the configured priority of the alarm	VT_I4
APPLICATION	Returns the application which triggered the alarm	VT_BSTR
COMPUTER	Returns the name of the computer which processed the alarm	VT_BSTR
USER	Returns the name of the user who processed the alarm	VT_BSTR
COMMENT	Returns the alarm comment	VT_BSTR
HIDDEN-COUNT*	Number of locked alarms	VT_I4
OS-HIDDEN*	Indicates whether the alarm is locked	VT_BOOL
OS_EVENTID*	Indicates the WinCC alarm number	VT_I4
BIG_COUNTER*	Alarm counter	VT_CY
UNIQUE EVENT ID*	corresponds to the "BIG_COUNTER"	VT_CY
*: Are supported only by "Mode 3".		

See also

Alarm classes and alarm methods (Page 53)

Mapping rule "Mode 1": (Page 50)

Mapping rules "Mode 2": (Page 51)

Mapping rules "Mode 3": (Page 51)

Mapping rule "Mode 1":

"Mode 1" displays the whole WinCC alarm system under a "OPC Source" with the name "localhost::" The "TEXT01" WinCC attribute is written in the OPC alarm text.

The WinCC alarm number is displayed on the "Condition" of the WinCC OPC A&E server.

"Mode 1" supports the "Simple Events" and "Condition Related Events" types of event. Events can be filtered according to the "Event Type", "Category" and "Severity" criteria.

See also

Restrictions for Mode 1 and Mode 2 (Page 52)

Attributes (Page 49)

OPC A&E types of event (Page 46)

Configure WinCC OPC A&E server (Page 54)

Alarm classes and alarm methods (Page 53)

Priorities (Page 53)

Mapping rules "Mode 2":

"Mode 2" is implemented as of version 3.5.2. of the WinCC OPC A&E server. The WinCC alarm number is displayed on the "Condition" of the WinCC OPC A&E server.

The text attribute of the WinCC alarm ("TEXT01" to "TEXT10") you have configured, is written in the OPC alarm text of "OPC Source".

"Mode 2" supports the "Simple Events" and "Condition Related Events" types of event. Events can be filtered according to the "Event Type", "Category" and "Severity" criteria.

See also

Restrictions for Mode 1 and Mode 2 (Page 52)

Attributes (Page 49)

OPC A&E types of event (Page 46)

Configure WinCC OPC A&E server (Page 54)

Alarm classes and alarm methods (Page 53)

Priorities (Page 53)

Mapping rules "Mode 3":

"Mode 3" is implemented as of version 3.6 of the WinCC OPC A&E server. "Mode 3" supports the hierarchical mapping of user-defined alarms. The hierarchy of "OPC Areas" is set by the user-defined alarm groups.

With "Mode 3" you query the address space of the WinCC OPC A&E server according to "Areas" and "Sources".

"Mode 3" supports the "Simple Events" and "Condition Related Events" types of event. Events can be additionally filtered according to the "Area" and "Source" criteria. "Mode 3" supports the "System Message" OPC category.

See also

Attributes (Page 49)

OPC A&E types of event (Page 46)

Configure WinCC OPC A&E server (Page 54)

Priorities (Page 53)

Restrictions for Mode 1 and Mode 2

Notes on mapping rules "Mode 1" and "Mode 2":

The mapping rules for "Mode 1" and "Mode 2" contain significant restrictions. The mapping rules for "Mode 1" and "Mode 2" are therefore only because of compatibility reasons supported by the previous version and are not developed further.

Note

In order to ensure compatibility with the previous version, the WinCC OPC A&E server cannot use multiplexer.

The restrictions are solved in the mapping rule "Mode 3".

Some of the restrictions of the mapping rules "Mode 1" and "Mode 2" are listed below:

- The WinCC alarm attributes are passed on via the OPC interface.
  The IDs value range of WinCC alarm attributes is defined by the OPC specification by "Property sets":
  - ID Set 1 contains the "OPC-specific Properties". Range: 1 ... 99
  - ID Set 2 contains the "Recommended Properties". Value range 100 to 199
  - IDs 300...399 are especially reserved for OPC A&E
  - IDs 400 to 4999 are reserved by OPC.
  - ID Set 3 contains the "Vendor-specific Properties". Range: 5000 and higher
    The IDs of the WinCC attributes must have values from 5000 and higher.

  Mode 1 and Mode 2 pass on the IDs of the WinCC alarm attributes via the OPC interface. These IDs are in the range of 0 to 61.
  If an OPC client, for example, requires the attribute with the IDs 2 ("Item Value") and 3 ("Item Quality") from the WinCC OPC A&E server, the OPC client receives instead the corresponding WinCC alarm atrocities. In this case the attributes are "ForeColor" and "BackColor".
- Restrictions for transfer of WinCC alarm numbers
  Alarm numbers are only transferred as text in the "Condition" when "Condition Related Events". When "Simple Events", the alarm numbers are completely lost.
- Some WinCC attributes are not supported, for example "UNIQUE_EVENT_ID"
  You can use the "UNIQUE_EVENT_ID" attribute for alarm synchronization of redundant server couples. During switching between redundant servers, the following could happen:
  - Alarms can be double- transferred
  - Alarms can be lost

You can find an overview of all attributes supported by the different modes at "Attributes (Page 49)".

See also

Mapping rule "Mode 1": (Page 50)

Mapping rules "Mode 2": (Page 51)

Attributes (Page 49)

Alarm classes and alarm methods

Introduction

The WinCC message system informs the user of disturbances and operating conditions in the process. A WinCC alarm always belongs to a specific alarm class and alarm method that is related to the "Event Category".

Event Category

Note

"Event Category" depends on the set mapping rules

The characteristics of the mapping rules "Mode 1" and "Mode 2" are described below:

An "Event Category" is displayed on the WinCC OPC A&E Server for every combination of an alarm class and alarm method.

An "Event Category" is determined by a "CategoryID" and a descriptive "Category Description". The "CategoryID" consists of the WinCC-internal ID of an alarm class and alarm method. The "Category Description" comes from the names of the alarm class and alarm method.

The names of the alarm classes and alarm methods can be ascertained explicitly via the alarm attributes "CLASSNAME" and "TYPENAME".

See also

Mapping rules "Mode 2": (Page 51)

Mapping rule "Mode 1": (Page 50)

Priorities

Introduction

The priority of WinCC alarms is displayed by the WinCC OPC A&E server on the "Severity" attribute.

You can configure up to 17 priority levels in WinCC. The "Severity" OPC attribute has a value range from "1" to "1000":

- 1: Lowest priority
- 1000: Highest priority

Depending on the number of the configured WinCC priority levels, the value in the "Severity" attribute is automatically retrieved by linear scaling.

Exception: For the mapping rules "Mode 1", the priority levels configured in WinCC are displayed directly on the "Severity" attribute.

Example

If you do not change the default settings in the configuration settings of the WinCC OPC A&E server, the 17 priority levels are going to be displayed in the WinCC OPC A&E server as follows:

Priority level in WinCC	Assigned value in the "Severity" attribute
0	1
1	64
2	127
3	190
...	...
15	937
16	1000

See also

Mapping rule "Mode 1": (Page 50)

Mapping rules "Mode 2": (Page 51)

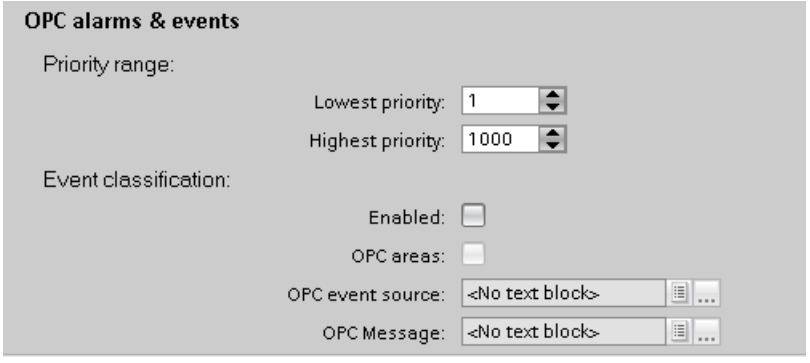
Mapping rules "Mode 3": (Page 51)

Configure WinCC OPC A&E server

Introduction

Define the display of the WinCC alarm system in the configuration settings of the WinCC OPC A&E server.

The following screen shows the configuration settings in the "Runtime settings" of the HMI device:



Requirement

Alarms are configured.

Procedure

To configure WinCC OPC A&E server, proceed as follows:

1. Open the "Runtime settings" of the HMI device in the project tree.
2. Configure the server settings at "Alarms > OPC Alarms & Events":
   - Define the low and high limit for the display of the priority range.
   - To activate the WinCC alarms with the mapping rule "Mode 3" in the WinCC OPC A&E, activate "Activated" and "OPC areas".
     Use primarily this setting.
   - Assign to the "OPC event source" and the "OPC event alarm" respectively an alarm quality, for example, " User text block 3".

Result

The WinCC alarms are displayed completely in the WinCC OPC A&E. If you have configured user-defined alarms, areas and sources of the user-defined alarms are displayed in the WinCC OPC A&E server.

Alternative procedure

You can display alternatively because of compatibility reasons the WinCC alarm also according to the "Mode 2" and "Mode 1" mapping rules in the WinCC OPC A&E server:

1. To display the WinCC alarms with the mapping rule "Mode 2" in the WinCC OPC A&E server, activate "Activated" and deactivate"OPC areas".
2. To display the WinCC alarms with the mapping rule "Mode 1" in the WinCC OPC A&E server, deactivate "Activated" and "OPC areas".

See also

OPC A&E types of event (Page 46)

Priorities (Page 53)

Basic settings for OPC (Page 32)

Mapping rule "Mode 1": (Page 50)

Mapping rules "Mode 2": (Page 51)

Mapping rules "Mode 3": (Page 51)

Accessing logged alarms

Access to logged events

Introduction

You can access the archived messages via the OPC A&E Server using an OPC client. Two methods are supported for accessing logged alarms:

- Output archived messages from a time period in the past
- Output archived messages from a time period in the past without mentioning end of period. After the output of archived messages, all other newly generated messages are automatically sent to the OPC client.

Note

After reading tagged logs, you are not allowed to use the returned "ActiveTime" of an alarm to acknowledge the alarm or to track any transitions in the alarm. To ensure this does not happen, the OPC A&E Client must check the "EventType" of an alarm for the extra flag "OPC_HAE_HISTORICAL_EVENTFLAG". The "ActiveTime" is not correct on logged alarms. You can find information on the additional flag under "Identifying archived messages".

Polling the "Historic alarms and events" functionality

The table below lists the filters offered by the WinCC's extended OPC A&E Server, in addition to the standard filters:

Filter	Filter Values	Description
OPC_HAE_FILTER_BY_TIMEFRAME	0x80000000	Corresponds to the "ReadRaw" function for OPC Historical Data Access
OPC_HAE_FILTER_BY_STARTTIME	0x40000000	Corresponds to the "AdviseRaw" function for OPC Historical Data Access

Source filter and historical alarm request

In order for you to be able to request logged alarms, the OPC client must support the "SetFilter" functionality on a Subscription. The OPC server will also send logged alarms if you insert the keyword "OPCHAEServer" into the array of the "Source Filter" in a Subscription. In addition to this keyword, you can use other parameters to define which messages are to be read:

- Method
- Time period
- With or without limits

The lists of sources that are assigned in the filter can include other source names besides the "OPCHAEServer" source. In this a case, the Subscription delivers only the logged events of the given sources. The sequence of the source names is inconsequential.

After configuring the source filter, you can retrieve the selected time period from the client with a "Refresh" call.

See also

Identifying logged alarms (Page 60)

Syntax for accessing logged alarms using OPC (Page 57)

Read methods for logged alarms (Page 58)

Syntax for accessing logged alarms using OPC

Syntax

```
OPCHAEServer hMode=(read|advise) htStartTime=szTime
[hEndTime=szTime] [bBounds=(TRUE|FALSE)]
```

Parameter

hMode = [read|advise]

Required Defines how the archived messages and events are to be read.

- Read
  Outputs logged alarms and events of a defined period from the past (comparable to ReadRaw for OPC Historical Data Access).
  The following example shows how a filter is set to read over the past 30 minutes:
  `OPCHAEServer hMode=read htStartTime=NOW-30M bBounds=TRUE`
- Advise
  Outputs archived messages and events from a definite period, When all logged alarms are received, new alarms are sent in the same way as for an active subscription (comparable to AdviseRaw for OPC Historical Data Accesss).
  The following example shows how alarms are read from 30 minutes ago until now (subscription must be active):
  `OPCHAEServer hMode=advise htStartTime=NOW-30M`

Note

The following notation is supported for the parameters "htStartTime" and "htEndTime":

- Relative notations such as "NOW"
- Symbolic values such as "NOW", "YEAR", "MONTH"
- Specification of absolute UTC data/time values according to XML notation: 2006-09-01T10:00:00.000Z

The use of symbolic notation corresponds to the syntax of OPC Historical Data Access.

htStartTime =

Required Defines the time from when the messages and events are to be read from the archive.

htEndTime =

Optional Defines the time up to which the messages and events are to be read from the archive. For "`hMode = read`", the default setting "NOW" is used.

bBounds = [TRUE|FALSE]

Optional Defines how messages close to the start and end time are to be handled. The function is identical to OPC historical data access.

- `bBounds=FALSE:`
  - Time stamp of the first transferred alarm >= htStartTime
  - Time stamp of the last transferred alarm < htEndTime
- `bBounds=TRUE:`
  - Time stamp of the first transferred alarm <= htStartTime
  - Time stamp of the last transferred alarm >= htEndTime

The default setting is FALSE.

See also

Read methods for logged alarms (Page 58)

Identifying logged alarms (Page 60)

Read methods for logged alarms

Introduction

You can use one of the two read modes to read archived messages:

- read
- advise

"read" mode

"read" mode is used to read logged alarms from a defined period in the past. The sequence of read messages is always read from the alarms in chronological sequence in reference to each OS server. By setting the start and end time, you can specify whether the last message is to be read first or last. If the start time is earlier than the end time, the last message is last in the output.

If you want to use "read" mode, run the following functions on the subscription:

1. SetFilter
2. Refresh

A "SetFilter" during the "Refresh" will be rejected. Activating the subscription during "Refresh" does not affect how the Refresh is carried out.

The historic events will continue to be transferred with the Refresh flag.

The newly generated events are transferred according to the standard reaction of an active subscription:

- Taking into account the set filter values with the exception of "historic" source "OPCHAEServer"
- Without the Refresh flag

Enables the client to distinguish between the received events based on the Refresh flag. An event package never contains historic and new events at the same time.

- Event packets with the Refresh flag contain only historic events. These events can also be in queue.
- Event packets without the Refresh flag contain only newly generated events.

"advise" mode

"advise" mode is used to read logged alarms starting from a defined period in the past. After reading all archived messages, new messages are sent in the same way as for an active subscription. The archived messages are transferred in chronological sequence in reference to each OS server: The archived messages from a start time onwards are transmitted. Thereafter, the newly archived messages transferred.

Note that you are not permitted to define an end time for "advise".

An active subscription is used for "advise" mode. If you run the "SetFilter" function on an active subscription, the historical alarms are transferred immediately.

If you run the "SetFilter" function on an inactive subscription, the logged alarms are sent only after the subscription is activated. If you want to use "advise" mode with an inactive subscription, follow these steps:

1. SetFilter.
2. Set the subscription to "Active" using "SetState".

The transmission is interrupted if you deactivate the subscription.

The transmission is ended if you set the subscription to "inactive". A "SetFilter" is rejected when the subscription is active.

A "Refresh" on an active "historic" subscription in "advise" mode functions in the same way as it does for a standard subscription:

All queued condition related events are transferred to packages with Refresh flag. The last packet also contains an additional "Last Refresh" flag.

A "Refresh" call has no influence on the reading of historical alarms in "advise" mode.

See also

Syntax for accessing logged alarms using OPC (Page 57)

Identifying logged alarms

Principle

Logged alarms are distinguished using an additional flag in EventType. This flag is ORed with the real EventType.

Name	EventType	EventType (archived message)
OPC_SIMPLE_EVENT	0x01	0x81
OPC_CONDITION_EVENT	0x04	0x84
OPC_TRACKING_EVENT	0x02	0x82
OPC_HAE_HISTORICAL_EVENTFLAG		0x80

Example 1

The following source filter is used to output logged alarms and events from the past 30 minutes in "read" mode. The oldest message for each OS server is output as the first one. The low limit value is also sent.

```
OPCHAEServer hMode=read htStartTime=NOW-30M bBounds=TRUE
```

Example 2

The following source filter is used to output events logged on September 1, 2006, from 10:00 a.m. to 12:00 p.m. in "read" mode. The newest message for each OS server is output as the first one. The limits for this time period are also sent.

```
OPCHAEServer hMode=read htStartTime=2006-09-01T12:00:00.000Z
htEndTime=2006-09-01T10:00:00.000Z bBounds=TRUE
```

Example 3

The following source filter is used to output logged alarms and events from the past 30 minutes in "advise" mode. After reading the archived messages, newly generated messages are sent in the same way as for an active subscription.

```
OPCHAEServer hmode=advise htStartTime=NOW-30M
```

See also

Access to logged events (Page 56)

Syntax for accessing logged alarms using OPC (Page 57)

1.2.3.6 WinCC OPC UA Server

Principle of operation the WinCC OPC UA Server

Principle of operation

The WinCC OPC UA Server provides the following values:

- Process values
- Values from tag logs
- WinCC alarms

The WinCC OPC UA server is installed as Windows service and started automatically. The WinCC OPC UA server supports only the "UA-TCP UA-SC UA Binary" communication profile. The used port number is adjustable.

Supported specifications

OPC Unified Architecture is a specification for the transmission of process values and log data, as well as alarms and events. The WinCC OPC UA server supports the transfer of process values, log data and alarms. The WinCC OPC UA server supports the OPC UA specification 1.02. You can find detailed information under "Supported OPC UA Services".

URL of the WinCC OPC UA server

You access the WinCC OPC UA server via the following URL: "opc.tcp://[NodeName]:[Port]".

[NodeName]: Placeholder for the computer name. Is set automatically.

[Port]: Port number. The default is "4861".

Discovery Server

The "Discovery Server" is available by the OPC foundation. The "Discovery Server" is by default installed on the HMI device as Windows service. On the "Discovery Server" via OPC UA server UA clients information is available that is registered on the "Discovery Server".

The WinCC OPC UA server registers for Runtime-Start, depending on the configuration, on no, one or all available "Discovery Servers" If you end the Runtime, the WinCC OPC UA server is automatically logged off from the "Discovery Server".

Supported languages in the WinCC address area

The WinCC OPC A&E Server supports the WinCC address area in the following languages:

- German
- English
- French

- Italian
- Spanish

See also

Using OPC in WinCC (Page 30)

Accessibility of WinCC OPC servers (Page 32)

Security concept of OPC UA (Page 62)

OPC UA Services support (Page 64)

Name area of the WinCC OPC UA server (Page 66)

OPC UA Data Access (Page 68)

OPC UA Log Access (Page 68)

Configure WinCC OPC HDA Server (Page 43)

Security concept of OPC UA

Introduction

The WinCC OPC UA server uses the TCP/IP protocol for data exchange. For authorization between WinCC OPC UA server and OPC UA client certificates are exchanged. In addition, you can encode the data transfer.

Security concept

The WinCC OPC UA server and each OPC UA client authorize themselves mutually by exchanging certificates.

By default, the WinCC OPC UA server creates during installation a self signed instance certificate. You can alternatively replace this instance certificate with a project-specific instance certificate.

Note

Private key and own certificates

If you have an own certification center, you can create your own certificates and make them available for all communication partners. In this case, delete the instance certificate created by WinCC OPC UA server.

The instance certificate is stored in the certificate memory. Depending on the configuration of the WinCC OPC UA server, one of the following certificate memories is used:

- Certificate memory of the WinCC OPC UA server
- Certificate memory of the operating system in "UA Applications" folder

In order for the WinCC OPC UA server and an OPC UA client to communicate with each other, the certificates must be known to each other:

- You use the common certificate of the operating system
  Or:
- You copy the certificates in the certificate memory of the participating communication partners:
  - WinCC OPC UA Server
  - OPC UA Client
  - Discovery server (optional)

Security settings

The following table lists the security settings supported by the WinCC OPC UA server:

Security guidelines	Alarm security mode		
None[1]	None		
Basic128Rsa15[2]	None[4]	Signing[5]	Signing and encryption[6]
Basic256[3]	None	Signing	Signing and encryption
Basic256Sha256[3]	None	Signing	Signing and encryption
1: The certificate exchange is switched off. Every OPC UA client can log on to the WinCC OPC UA server.			
2: Certificate exchange with depth of encryption of 128 bit.			
3: Certificate exchange with depth of encryption of 256 bit.			
4: The data packages are exchanged after certificate check unsecured between client and server.			
5: The data packages are signed with the certificates, but not encoded			
6: The data packages are signed with the certificates and encoded			

Note

Unsecured communication between client and server possible

Use the "None" setting only for test and diagnostics purposes.

For a secure communication between client and server, use in operating mode at least the following settings:

- Security guideline: Basic128Rsa15
- Alarm security mode: Signing

You can deactivate the security guideline "None" in the configuration settings of the WinCC OPC UA server.

User identification

For user account identification of an OPC UA client, the WinCC OPC UA server supports the methods "Anonymous" and "User name / password". Each user account must be known in the user administration of the operating system of the WinCC OPC UA server.

The user identification is only used for setting up a communication session. Different access rights are not supported.

You can deactivate the support of anonymous users in the configuration settings of the WinCC OPC UA server.

See also

Structure of the configuration file (Page 75)

Configure WinCC OPC UA Server (Page 77)

Principle of operation the WinCC OPC UA Server (Page 61)

Setting up authentication via certificates. (Page 82)

OPC UA Services support

Introduction

The WinCC OPC A&E Server supports the following described functionality.

OPC UA Service Sets

The following table shows the supported OPC UA Service Sets:

OPC UA Service Sets	Services	Comment
Discovery Service Set	FindServers GetEndpoints	-
Secure Channel Service Session Service Set	All	-
View Service Set	Browse BrowseNext RegisterNodes UnregisterNodes	Detecting the displayed WinCC data: Process values and archived data
Attribute Service Set	Read Write HistoryRead HistoryUpdate*)	only WinCC tags only WinCC tags only logging tags only logging tags
Subscription Service Set	CreateSubscription SetPublishingMode Publish RePublish DeleteSubscription	-
MonitoredItem Service Set	CreateMonitoredItems SetMonitoringMode DeleteMonitoredItems	only "Value" attribute of WinCC tags .EventNotifier during access to WinCC alarms

OPC UA Service Sets	Services	Comment
Method Service Set	Call	Acknowledge ConditionRefresh
*: With restrictions, see "Supported Write-Accesses (Page 37)"		

OPC UA profile and Conformance Units

Note

The OPC UA profiles for "Historical Access" are not yet released, that is why they are not listed in the following:

The WinCC OPC UA server supports the following OPC UA 1.02 profiles without restrictions:

- 6.5.3 Base Server Behaviour Facet
- 6.5.12 Standard Event Subscription Server Facet
- 6.5.14 A & C Base Condition Server Facet
- 6.5.24 Method Server Facet
- 6.5.30 Historical Raw Data Server Facet
- 6.5.36 Historical Data Update Server Facet
- 6.5.37 Historical Data Insert Server Facet
- 6.5.38 Historical Data Delete Server Facet
- 6.5.107 UA-TCP UA-SC UA Binary
- 6.5.125 SecurityPolicy - Basic256
- 6.5.124 SecurityPolicy - Basic128Rsa15
- 6.5.123 SecurityPolicy - None

The WinCC OPC A&E Server supports the following OPC UA profiles shown in the following table, however with restrictions:

Profile	"Group"	Not supported "Conformance Unit"
6.5.8 Standard DataChange Subscription Server Facet	Monitored Item Services	DeadBand Filter
6.5.9 Enhanced DataChange Subscription Server Facet	Monitored Item Services	
6.5.25 Core Server Facet	Attribute Services	Attribute Write Index
6.5.26 Data Access Server Facet	Data Access	Data Access Analog Data Access Multistate Data Access PercentDeadBand Data Access Semantic Changes Data Access Two State

Profile	"Group"	Not supported "Conformance Unit"
6.5.35 Standard UA Server	Attribute Services	Attribute Write StatusCode & TimeStamp
6.5.47 Standard UA Server Profile	Attribute Services	Attribute Write StatusCode & Timestamp

See also

Principle of operation the WinCC OPC UA Server (Page 61)

Name area of the WinCC OPC UA server

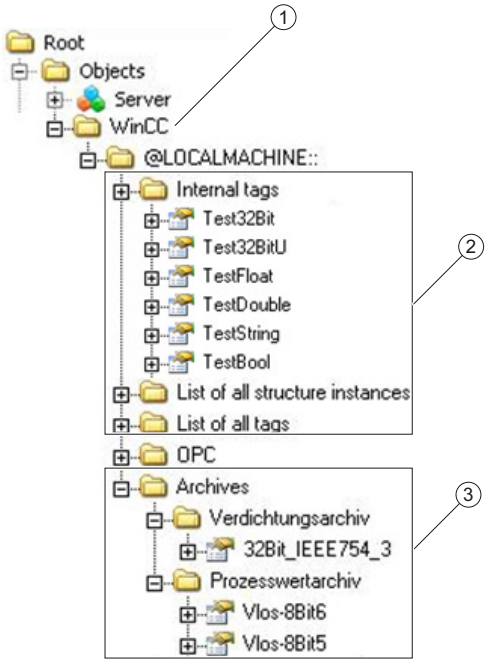
Introduction

The WinCC OPC UA server provides OPC UA clients with an hierarchical name area to display the following Runtime data:

- Process values (WinCC tags and WinCC tag groups)
- Data log inclusive logging tags
- WinCC alarms

The name area of the WinCC OPC UA server is attached in the "Objects" default folder.

The following screen shows the name area of the WinCC OPC UA server of an active WinCC project on the local PC ("@LOCALMACHINE::"):



① Start node of the specific name area of WinCC.
② Display of the WinCC tags; the structure corresponds to the structure of the tags in WinCC.
③ Display of the data log

Display of the WinCC tags

Tag groups, communication drivers and connections are displayed by OPC UA objects of the "FolderType" type. Each of these folders has references of the "Organizes" type to the subordinate objects and tags.

Internal and external WinCC tags are displayed by OPC UA tags of the "DataItemType" type. If a WinCC tag is additionally logged, the displayed OPC UA tag has additionally a reference of the "HasHistoricalConfiguration" type for a log configuration. The "Historizing" and "AccessLevel" attributes are respectively set.

The following table shows the most important attributes of the OPC UA tags that represent a WinCC tag. You can find the complete list of attributes in the "OPC UA Part 3 - Address Space Model 1.02 Specification" document under "5.6":

Attribute	Description	Comment
NodeId	Unique designation of the WinCC tag	
BrowseName	WinCC tag name	
DisplayName	WinCC tag name	
Value	Tag value and status	
DataType	OPC UA data type that corresponds to the WinCC tag type, for example: • Int32; signed 32 bit value • UInt32; unsigned 32 bit value	-
AccessLevel	"CurrentRead" / "CurrentWrite" "HistoryRead" / "HistoryWrite"	correspondingly to the WinCC tag configuration
ValueRank	Always "Scalar"	-

Display of the logging tags

Process values and compressed logs are displayed by OPC UA objects of the "FolderType" type. Each of these folders has references of the "Organizes" type to the related logging tags.

Logging tags from process value or compressed logs are displayed by OPC UA tags of the "BaseDateVariableType" type. A logging tag always has a reference of the "HasHistoricalConfiguration" type for a log configuration.

The following table shows the most important attributes of the OPC UA tags that represent a WinCC logging tag. You can find the complete list of attributes in the "OPC UA Part 3 - Address Space Model 1.02 Specification" document under "§5.6":

Attribute	Description	Comment
NodeId	Unique designation of a logging tag	
BrowseName	Name of the logging tag	
DisplayName	Name of the logging tag	
Description	Node description	
Value	Not available	For a logging tag, this attribute cannot be read nor changed.

Attribute	Description	Comment
DataType	OPC UA data type that corresponds to the WinCC tag type, for example: • Double; 64-bit floating point number • UInt32; unsigned 32 bit value	-
AccessLevel	"HistoryRead" / "HistoryWrite"	-
ValueRank	Always "Scalar"	-

Access to WinCC alarms

The starting node of the WinCC namespace is an Event Notifier with which the OPC UA clients can receive the status changes of WinCC alarms via Subscriptions in Runtime.

OPC UA Data Access

Internal and external WinCC tags are displayed by OPC UA tags of the "DataItemType" type. Other DataAccess tag types as "AnalogItem" or "DiscreteType" are not supported.

The WinCC OPC A&E Server supports the reading access on the OPC UA tag attributes as "DataType" or "AccessLevel". Writing access and subscriptions are only supported for the "Value" attribute.

OPC UA Log Access

Introduction

"OPC Historical Access" enables the access to logs and includes the "Historical Data" and "Alarms & Events" services. The WinCC OPC UA server supports only the "Historical Data" service.

The WinCC OPC UA Server offers the OPC clients access to the raw data of tag logs via "Services".

- HistoryRead (READRAW)
- HistoryUpdate (INSERTDATA, REPLACEDATA, UPDATEDATA, DELETE_RAW)

You can read and limitedly write with an OPC UA client the values of logging tags in the tag logs. Depending on the configuration of the tag log, the logging tag can contain either raw or already processed process values.

Characteristics of logging tags

According to the OPC "OPC UA Part 11 - Historical Access 1.00 Specification" document, the tags to be logged must have a precise reference to a tag configuration ("HistoricalConfiguration"). A process tag can be contained in WinCC and also in several data logs. In this case the process tag is linked to one of the corresponding logging tags.

Properties of log configurations

The following table shows the Properties of an OPC UA tag configuration of the "HistoricalConfigurationType" type: In the "Description" property, the logging tag comment configured in WinCC is displayed. You can find the complete list of properties in the "OPC UA Part 11 - Historical Access 1.02 Specification" document under "§5.2.2":

Property	Description / Value	Comment
Definition	WinCC process tag name	For a process value log
Stepped	True	-

The following optional Properties are not supported:

- MaxTimeInterval
- MinTimeInterval
- ExceptionDeviation
- ExceptionDeviationFormat

Limitations for Service "HistoryUpdate"

You can use the Service "HistoryUpdate" only on process value logs.

The following table lists the functions supported by the WinCC OPC UA server: Which functions are supported depends on the configuration of the WinCC OPC UA server as well as the process value log configuration. You will find additional information in the "OPC UA Part 11 - Historical Access 1.00 Specification" document under "§5.5":

Service	Function	Description
HistoryUpdate	INSERTDATA	Insert new log values
	REPLACEDATA	Replace existing log values
	UPDATEDATA	Replace of insert log values
	DELETE_RAW	Delete log values

See also

OPC UA Services support (Page 64)

Name area of the WinCC OPC UA server (Page 66)

Supported Write-Accesses (Page 37)

Configure WinCC OPC HDA Server (Page 43)

OPC UA Alarms & Conditions

Introduction

The OPC UA server allows access to alarms of the WinCC message system.

The OPC UA server relays status changes of messages to OPC UA clients with WinCC-Event-Notifications via Subscriptions and Monitored Event Items , but manages no Condition

instance in its namespace. The Event Notifier node to be used is the start node of the WinCC name area. The UA client can filter the messages and define the list of message attributes returned.

The OPC UA server supports the "OPC UA Alarms & Conditions 1.02" specification.

Mapping of the message system to OPC UA event types

The messages are mapped to the following OPC UA event types:

WinCCEventType

This type is based on "BaseEventType" and maps "simple" WinCC messages with the following acknowledgment theory:

- "Message without status went out" is activated
- "Acknowledgment came in" is not activated

Examples of this type of message are starting and stopping motors.

WinCCAlarmConditionType

This type is based on "AlarmConditionType" and maps all messages which cannot be mapped on WinCCEventType, for example acknowledgeable messages and messages with the status "came in" and "went out".

With an alarm of the type "WinCCAlarmConditionType", the event is linked to a condition. For example, WinCC generates an alarm as soon as a limit violation of a tag occurs. This message in OPC UA is equivalent to an Alarm Condition.

WinCC message attributes

The two Event types add WinCC-specific message attributes to the basic type. The attributes are mapped 1:1 as UA Event Properties and are described in more detail in "Attributes of the WinCC message system".

Message class and message type

The message system informs the user of disturbances and operating conditions in the process. A message always belongs to a specific message class and message type, which are specified in the attributes "CLASSID", "TYPEID", "CLASSNAME" and "TYPENAME" of the corresponding UA Events.

Priority

When configuring messages in the message system, you can configure a priority of "0" to "16". The OPC UA specification defines a value range of "1" to "1000" for the Severity. "1" stands for the lowest and "1000" for the highest Severity.

The values of the priority must therefore be suitably mapped to the OPC severity. In standard mapping, a priority of "0" is assigned to OPC-Severity "1" and a priority of "16" to OPC-Severity "1000". All other values are interpolated linearly between "0" and "1000".

OPC UA mapping rules

During the configuration of the WinCC message system, settings are made to determine which process events generate a message. This message is generally shown as an Event in OPC UA.

The following table shows the most important Properties of an Events and how the WinCC message system provides the information.

OPC UA property	Mapping in the WinCC message system
For all event types:	
EventID	Unique message designation
EventType	Event type: Node ID of the WinCCAlarmConditionType node or WinCCEventType node
SourceNode	Irrelevant
SourceName	Indicates the source of the message. Mapping is described in more detail below.
Message	Message text for the corresponding message number.
Time	Time of the event. The time stamp is given in UTC
Severity	Priority of the WinCC message
Only with WinCCAlarmConditionType:	
ConditionName	Set text that is output as well as the message. The text output depends on the mapping rule set: • "Mode 1" and "Mode 2": Message number • "Mode 3": Message class, for example "Process control message"
Quality	Returns the quality of the message
ConditionClassId	Node ID of the "ProcessConditionClassType" node
ConditionClassName	"ProcessConditionClassType"
Retain	"TRUE" for pending alarms
NodeId	ConditionId: Designates an UA-Condition uniquely, for example, an alarm. Required for acknowledgment, even if no Condition instances are supported
EnabledState	"TRUE" when the alarm has been enabled
ActiveState/Id	"TRUE" when the alarm has arrived
AckedState/Id	"TRUE" when the alarm has been acknowledged
ClientUserId	Indicates the user that is logged on

Note

The following OPC UA Condition and Alarm Properties are not supported by the OPC UA server:

- BranchId
- LastSeverity
- InputNode
- ConfirmedState
- SuppressedState
- ShelvingState
- SuppressedOrShelved
- MaxTimeShelved

Message statuses / acknowledgment statuses

The following table shows WinCC message status mapping to the corresponding WinCCAlarmConditionType - Properties:

Message status	EnabledState/Id	ActiveState/Id	AckedState/Id
Locked message	FALSE	-	-
Enabled message	TRUE		
Received message	TRUE	TRUE	FALSE
Sent message with acknowledgment	TRUE	FALSE	TRUE
Sent message without acknowledgment	TRUE	FALSE	FALSE
Acknowledged messages (message pending)	TRUE	TRUE	TRUE
Acknowledged messages (message no longer pending)	TRUE	FALSE	TRUE
Received, acknowledged message	TRUE	TRUE	TRUE
Received, sent message with acknowledgment	TRUE	FALSE	TRUE
Received, sent message without acknowledgment	TRUE	FALSE	FALSE
Message acknowledged by the system (message pending)	TRUE	TRUE	TRUE
Message acknowledged by the system (message no longer pending)	TRUE	FALSE	TRUE
Emergency-acknowledged message (message pending)	TRUE	TRUE	TRUE
Emergency-acknowledged message (message no longer pending)	TRUE	FALSE	TRUE

Settings for mapping the WinCC message system

The configuration of the OPC UA server also applies to the OPC UA server as regards the mapping of the Properties "SourceName" and "Message" of a message.

- With OPC A&E server with hierarchical access:

SourceName	Indicates the source of a message. The Source has the format "<Server prefix>::Area\UserTextBlock 2". The server prefix of the local computer is "@LOCALMACHINE".
Message	Returns the message text of the corresponding message number

- With OPC A&E server without hierarchical access:

SourceName	Indicates the source of a message. The Source has the format "<Server prefix>::localhost::". The server prefix of the local computer is "@LOCALMACHINE".
Message	Returns the message text of the corresponding message number

Alarm groups

In WinCC, the alarm groups are not displayed in the namespace.

Supported event methods

Acknowledgment

A WinCC message is acknowledged using the "Acknowledge" method of the "AcknowledgeableConditionType" node in the standard OPC UA info model.

Only messages of the "WinCCAlarmConditionType" type can be acknowledged.

ConditionRefresh

Messages still pending are established using the "ConditionRefresh" method of the "ConditionType" node in the standard OPC UA info model.

Filters

The OPC UA client can defined a filter for Monitored Event Items .

The following operators are, however, not supported by the OPC UA server:

- FilterOperator_Cast
- FilterOperator_BitwiseAnd
- FilterOperator_BitwiseOr
- FilterOperator_RelatedTo
- FilterOperator_InView

Attributes of the WinCC message system

Overview

The following table lists the configurable attributes of the WinCC message system. The attributes are mapped 1:1 as UA Event Properties .

WinCC message attribute	Meaning	Data type
CLASSNAME	Name of the message class	String
TYPENAME	Name of message type	String
FORECOLOR	Foreground color for incoming, outgoing and acknowledged messages.	Int32
BACKCOLOR	Background color for incoming, outgoing and acknowledged messages.	Int32
FLASHCOLOR	Flash color	Int32
FLAGS	Indicates mandatory message acknowledgment	Int32
TEXT01...TEXT10	Content of user text block #1....#10	String
PROCESSVALUE01...PROCESSVALUE10	Content of process value block #1....#10	
STATETEXT	Status message	String
INFOTEXT	Information text for the message	String
LOOPINALARM	Indicates whether LoopInAlarm was configured	Int32
CLASSID	Message class ID	Int32
TYPEID	Message type ID	Int32
MODIFYSTATE	Value of message status tag	Int32
AGNR	Outputs the number of the automation system that generated the message	Int16
CPUNR	Outputs the number of the CPU that generated the message	Int16
DURATION	Outputs the time period between the incoming state, outgoing state and acknowledgment of a message	Int32
COUNTER	Number of messages after the start of runtime	Int32
QUITSTATETEXT	Indicates whether the message has been acknowledged	String
QUITCOUNT	Number of open, unacknowledged messages	Int32
PARAMETER	Configuration parameter of the message	Int32
BLOCKINFO	Current content of the message block	String
ALARMCOUNT	Number of pending messages	Int32
LOCKCOUNT	Number of locked messages	Int32
PRIORITY	Priority of the message	Int32
APPLICATION	Outputs the application which triggered the message	String
COMPUTER	Outputs the name of the computer which processed the message	String

WinCC message attribute	Meaning	Data type
USER	Outputs the name of the user who processed the message	String
COMMENT	Comment of the alarm	String
HIDDEN-COUNT	Number of hidden messages	Int32
OS-HIDDEN	Indicates that the message is hidden	Boolean
OS_EVENTID	WinCC alarm number	Int32
BIG_COUNTER	Message counter	Int64

Structure of the configuration file

Introduction

You configure the WinCC OPC UA server in the configuration file "OPCUASERVERWINCCPRO.XML".

File location

When you create a project in the TIA Portal, the project-specific configuration file "OPCUASERVERWINCCPRO.XML" is stored in the WinCC project folder under:

"<WinCC project folder>\OPC\UASERVER"

`<SecuredApplication>` section

In this section the OPC UA application security is set.

Section	Description
`<Secured Application>`	

`<BaseAddresses>`
`<...></...>`
`</BaseAddresses>`
	Address and port number
	The parameter [`HostName`] is the placeholder for the computer name and is determined during runtime.
	Example:
	`<BaseAddresses>`
	`<ua:String>opc.tcp://`
	`[HostName]:5210</ua:String>`
	`</BaseAddresses>`

Section	Description
<SecurityProfileUris> <SecurityProfile> <...></...> </SecurityProfile> ... </SecurityProfileUris>	Security guidelines • You enable the setting with "true". • You disable the setting with "false". All active OPC clients with this certificate are thus disabled. Example: <SecurityProfile> <ProfileUri>http://opcfoundation.org/ UA/SecurityPolicy#Basic128Rsa15</ProfileUri> <Enabled>false</Enabled> </SecurityProfile>
</Secured Application>	

<ServerConfiguration> section

In this section you set the parameters for data transmission, authentication and optimized WinCC archive write access.

<ServerConfiguration>	
<SecurityPolicies> <SecurityPolicy> <...></...> </SecurityPolicy> ... </SecurityPolicies>	Alarm security mode To deactivate a security setting, delete the entire entry. Example: <SecurityPolicy> <ProfileUri>http://opcfoundation.org/ UA/SecurityPolicy#Basic128Rsa15 </ProfileUri> <MessageSecurityModes>SignAndEncrypt </MessageSecurityModes> </SecurityPolicy>
<UserTokenPolicies> <UserTokenPolicy> <...></...> </UserTokenPolicy> ... </UserTokenPolicies>	User authentication To deactivate a setting, delete the entire entry. Example <UserTokenPolicy> <TokenType> <!--[User]--> </TokenType> </UserTokenPolicy> Use the "Anonymous" setting only for test and diagnostics purposes.

<FastInsert> <Users> <...></...> </Users> <Clients> <...></...> </Clients> </FastInsert>	Optimized WinCC archive write access • You use "true" to activate the optimized WinCC archive write access for all OPC UA clients. • You use "false" to specify whether specific Windows users or OPC UA clients may use the optimized WinCC archive write access. You specify the Windows user under <Users>. You specify the OPC UA clients under <Clients>. As ClientName, use the "Common Name" that is entered in the client certificate. Example: <EnabledByDefault>false</EnabledByDefault> <Users> <User>domain\user1</User> </Users> <Clients> <Client>ClientName1</Client> </Clients>
</ServerConfiguration>	

See also

Security concept of OPC UA (Page 62)

Configure WinCC OPC UA Server (Page 77)

Setting up authentication via certificates. (Page 82)

Configure WinCC OPC UA Server

Introduction

You configure the WinCC OPC UA server in the runtime settings and in the configuration file "OPCUASERVERWINCCPRO.XML".

Configuring the WinCC OPC UA server in runtime settings

To configure the WinCC OPC UA server in the runtime settings, proceed as follows:

1. Open the "Runtime settings" of the HMI device in the project tree.
2. If necessary, configure the write access validation under "OPC settings > Configuration of the OPC Historical Access".

   Note

   The "Configuration of the OPC Historical Access" applies for the WinCC OPC HDA server as well as for the OPC UA server.

3. Configure the server settings under "OPC settings > Configuration of the OPC Unified Architecture Server":
   - Change the "Port number", if required
   - Activate at least one of the "Security guidelines" and the respective "Alarm security mode".

     Note

     Unsecured communication between client and server possible

     Use the "None" setting only for test and diagnostics purposes.

Configuring the WinCC OPC UA server in the configuration file

To configure the WinCC OPC UA server in the configuration file, proceed as follows:

1. Open the Windows Explorer.
2. Navigate to the directory "<WinCC project folder>\OPC\UASERVER".
3. Open the configuration file "OPCUASERVERWINCCPRO.XML".
4. If necessary, change the port number under `<BaseAddresses>`.
   Do not use a port number that is already assigned to another application.
5. Configure the OPC UA application security in the section `<SecuredApplication>`..
6. Configure the data transmission, authentication and optimized WinCC archive write access in the section `<ServerConfiguration>`.

Result

The WinCC OPC UA Server is configured. If you activate the project on the HMI device, the WinCC OPC UA server is accessible.

See also

Security concept of OPC UA (Page 62)

Structure of the configuration file (Page 75)

OPC UA Log Access (Page 68)

Supported Write-Accesses (Page 37)

Basic settings for OPC (Page 32)

1.2.4 WinCC OPC connection

1.2.4.1 Creating a connection to an OPC server

Introduction

If to an OPC DA server an external control is available, you can access its process values from WinCC with the help of the WinCC OPC connection. You configure for this in the engineering system an OPC connection to the desired OPC server. You configure afterwards WinCC tags for the process values of the OPC server.

Requirement

- The OPC server to be addressed is ready-to-operate and in the "running" status
- The OPC server and the HMI device with WinCC Runtime Professional are located in a network

Procedure

To create a connection to an OPC server, follow these steps:

1. On the HMI device, open the "Connections" editor.
2. Create a new connection and enter a meaningful name.
3. Choose "OPC" as the "Communication driver".
4. In the work area under "Parameters > OPC settings", enter the communication peer.
   - Select the "Type of OPC server".
   - Select the OPC server based on the OPC server type you chose, or enter the IP address or name of the remote computer.

Result

The OPC connection is configured. To access data from the OPC server, you create tags.

See also

Accessing process values of an OPC server (Page 80)

Using OPC in WinCC (Page 30)

1.2.4.2 Accessing process values of an OPC server

Requirement

- The OPC server to be addressed is ready-to-operate and in the "running" status
- A connection to the OPC-Server is created

Procedure

To access process values of an OPC-Server via the OPC connection, follow these steps:

1. On the configuration PC in the project navigation, open the "HMI tags" editor under the HMI device that you use as an OPC client.
2. Create a tag with the same data type as the tag on the OPC server.
3. Select the OPC connection for "Connection".
4. Enter the "Address", or select the desired tag on the OPC server via the object list.

Result

If you launch Runtime on the HMI device, the process value from the OPC server will be written to the tag on the HMI device via the OPC connection.

See also

Creating a connection to an OPC server (Page 79)

1.2.5 WinCC OPC UA connection

1.2.5.1 Creating a connection to an OPC UA server

Introduction

The OPC UA client accesses process values in the hierarchical name space of an OPC UA server. For this access to be carried out, an OPC UA server and the OPC UA client authorize each other through the exchange of certificates. In addition, you can encode the data transfer.

TIA Portal has its own independent certificate store. The certificates from the certificate store are not automatically transferred in Runtime or to the simulation. Therefore, the certificates for any Runtime or simulation PC must be configured manually.

The OPC UA client of the TIA Portal by default classifies each certificate of an OPC UA server as "trustworthy". The certificates must be configured manually for the OPC UA clients in Runtime or simulation. How an OPC UA server responds to a connection request of the OPC UA client depends on the configuration of the OPC UA server.

To establish communication to an OPC UA server, request the following information from the operator of the OPC UA server:

- URL of the OPC UA server
- Security settings
- Required certificates

You can protect the OPC UA connections through authentication. By default, each new OPC UA connection is created as "Anonymous". If you want to protect the connection, enter your user name and password.

Requirement

URL and security settings of the OPC UA server are known.

Procedure

To create a connection to an OPC UA server, proceed as follows:

1. Open the "Connections" editor on the HMI device.
2. Create a new connection and enter a meaningful name.
3. Select the entry "OPC UA" as the "Communication driver".
4. Configure the "OPC Server" under "Parameters" in the work area:
   - Enter the URL of the OPC UA server in the "UA server discovery URL" text box, or select the OPC UA server from the list.
   - In "Security policy", select one of the supported security policy options.
   - In "Message security mode", select one of the supported security mode options.
5. Enter your user name and password to protect the OPC UA connection.

Note

Migration

For security reasons, the password is not migrated during migration.

Result

The OPC UA connection is configured. You create tags or arrays to access data from the OPC UA server.

Additional information on OPC UA connections

- To make a device of the device version V14.0.0.0 available for the OPC UA connection, you need to change the device version to V14.0.1.0 or higher.
- The OPC UA connection is available for a WinCC Runtime Professional device as of version V14.0.1.0. The OPC UA connection protected through authentication is available as of version V15.0.
- In addition, you can protect the assigned password through password protection in runtime (password for transport).

See also

Setting up authentication via certificates. (Page 82)

Creating OPC UA tags (Page 83)

Overview of the supported data types (Page 86)

1.2.5.2 Setting up authentication via certificates.

Server certificates and client certificates

Distinguish between client and server certificates when configuring. Secure communication is only possible when client and server recognize each other's certificates.

Certificates are linked to the respective PC. If the Runtime project is loaded on another PC, the process of mutual exchange of trust certificates must be repeated.

For communication, the OPC UA server must recognize the following client certificates as trustworthy:

- "Siemens OPC UA Client for WinCC"
  Without a valid certificate of the TIA Portal, the tag browser aborts the connection attempt.
- "Siemens OPC UA Client for WinCC Runtime"
  Without a valid Runtime certificate, no current values are displayed in runtime.

Use the WinCC channel diagnostics for the analysis.

You can find additional information under Security Concept of OPC UA.

Setting up valid certificates

A self-signed certificate for the tag browser of the TIA Portal ""Siemens OPC UA Client for WinCC" and a certificate for the OPC UA channel of WinCC Runtime "Siemens OPC UA Client for WinCC Runtime" are created during installation. The OPC UA client can only connect to the OPC UA server if the server identifies this client certificate as trustworthy.

When the tag browser opens, the OPC UA server checks the client certificate of the TIA Portal. If the server does not recognize the client certificate as trustworthy, the connection is rejected. The error message "Connection failed" appears. The tag browser automatically accepts the server certificate.

Each WinCC Runtime or simulation has its own certificate and a project-independent certificate store. Certificates are not transferred when downloading the project. This means you must trust the certificate of the OPC UA server on each HMI device. The OPC UA server must trust each HMI device certificate.

For the OPC UA client of Runtime, the certificates are located below the installation path: c:\Program Files (x86)\Siemens\Automation\SCADA-RT_V1x\WinCC\OPC\UAClient\PKI\.

A rejected certificate is stored in the "rejected\certs" folder. To trust a certificate, move it to the "trusted\certs" folder.

See also

Creating a connection to an OPC UA server (Page 80)

Security concept of OPC UA (Page 62)

Creating OPC UA tags (Page 83)

Structure of the configuration file (Page 75)

1.2.5.3 Creating OPC UA tags

Introduction

You create the tags for the OPC UA connection in the "HMI Tags" editor. You can view these tags in the tag browser.

Requirements

- The project is open.
- The connection to the OPC UA server has been set up.

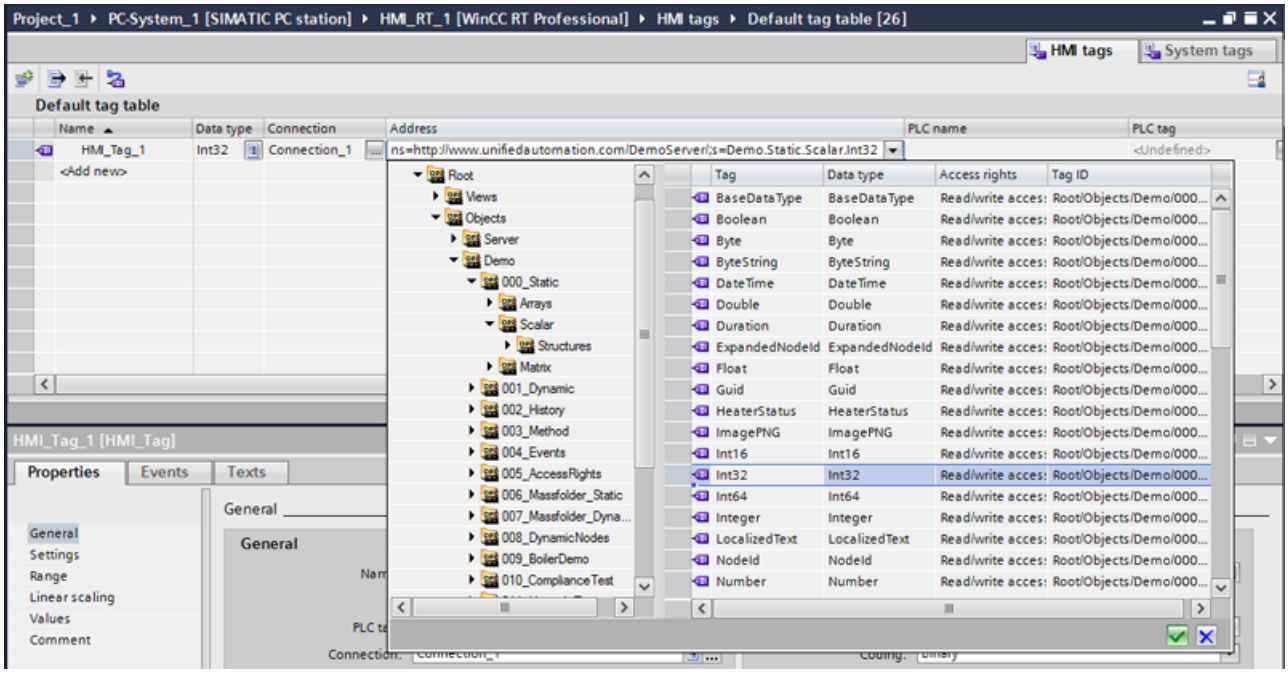
Procedure

You create OPC UA tags exactly as you would internal HMI tags. You can find additional information under "Creating internal tags".

Make the following deviating settings for the OPC UA tags:

1. In the "Connection" field, select the connection to the OPC UA server that you have set up beforehand.
2. In the "Data type" field, select a permissible data type for OPC UA tags.

3. In the "Address" column, click on the drop-down list and select the address in the tag browser. You can also enter the address manually.



4. If required, make settings under "Area", "Linear scaling", "Start value" and "Substitute value".

Result

You have created OPC UA tags that you can use for the OPC UA connection in your project.

See also

Overview of the supported data types (Page 86)

Creating a connection to an OPC UA server (Page 80)

Setting up authentication via certificates. (Page 82)

1.2.5.4 Creating array tags for OPC UA connection

Introduction

You create the array tags for the OPC UA connection in the "HMI Tags" editor. You can view these array tags in the tag browser.

You can select the arrays but not the individual array elements in the tag browser.

It is possible to configure one-dimensional arrays for all supported OPC UA data types, with the exception of ByteString.

Requirement

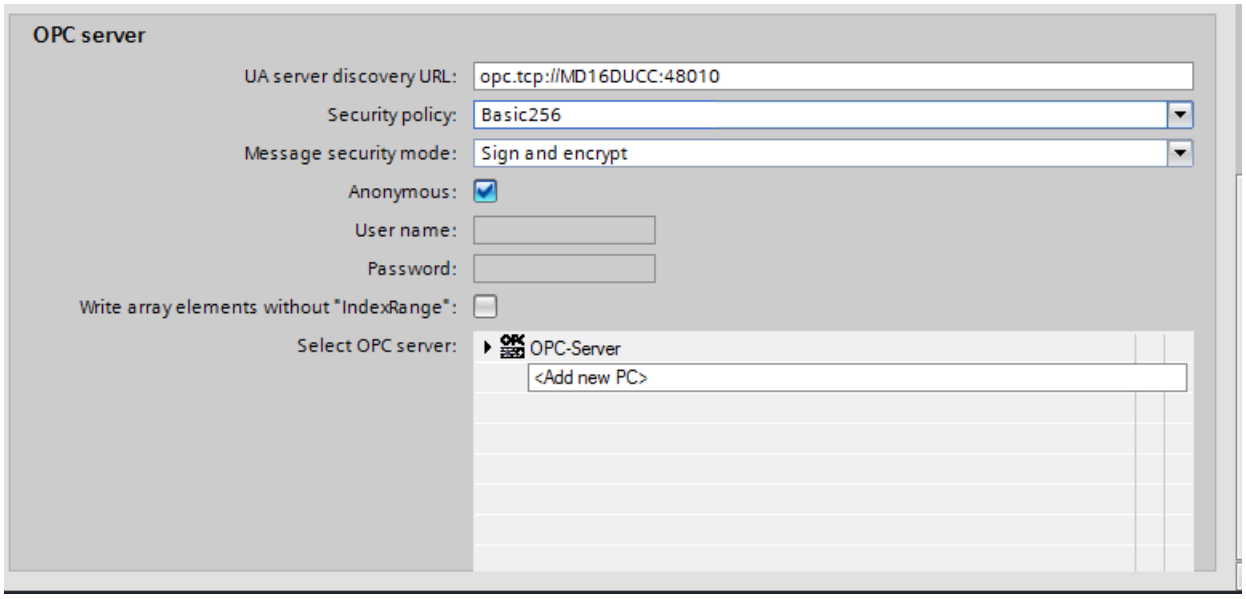
- The project is open.
- The connection to the OPC UA server has been set up.

Procedure

You create the array tags for OPC UA connections exactly as you would normal array tags. You can find additional information under "Create array tag".

If needed, make the following deviating settings for the created OPC UA connection under "Parameters > OPC Server":

1. To use arrays on older OPC UA servers that do not support writing single array elements, use the "Write array elements without IndexRange" setting.



If the "Write array elements without IndexRange" setting is selected, the complete array is read and the corresponding elements of the array modified. The entire array is then written back to the server.

NOTICE
Loss of data If the "Write array elements without IndexRange" setting is selected, data loss may occur if the arrays are changed between the read and write access on the server.

For additional information on arrays, refer to "Array basics".

1.2.5.5 Overview of the supported data types

Introduction

For the data exchange, configure access to the tags of the OPC UA server for the OPC UA client. To do so, import an OPC UA node as HMI tag in the tag management.

Supported data types

The OPC UA client supports the following data types:

- Boolean
- Byte (unsigned byte)
- Date Time
- Double
- Float
- Int16
- Int32
- Raw (ByteString)
- SByte
- String
- UInt16
- UInt32
- Int32
- UDT

Note

The elements of the UDT can be interconnected to OPC UA tags.

Note

OPC UA structures are not supported.

If the server supports OPC UA structures, however, you can access elements of an OPC UA structure.

See also

Creating OPC UA tags (Page 83)

Creating a connection to an OPC UA server (Page 80)

1.2.5.6 Diagnostics options of the "OPC UA" channel

The following options are available for diagnosing and troubleshooting the "OPC UA" channel and a tag of this channel:

Checking the configuration of connection and tags

Possible errors can lie in the configured system parameters and connection parameters. An incorrect addressing of the tag can also be responsible for faulty tag values.

Diagnosis of the channel using "Channel Diagnosis"

The status of the channel and the connection can be queried in Runtime using "Channel Diagnosis". Errors that occur are displayed as so-called "Error Codes".

Diagnosis of the channel tags

In the tag management, the current value, the current quality code and the last change time of the tags can be queried in Runtime.

1.2.5.7 Supported OPC UA services of the OPC UA client

OPC UA Services support

The OPC UA client supports the following OPC UA services:

- SecurityPolicy - Basic256Sha256
- SecurityPolicy - Basic128Rsa15
- SecurityPolicy - Basic256
- SecurityPolicy - None
- DataAccess ClientFacet

Explanation of the security settings

The following table lists the security settings supported by the OPC UA client:

SecurityPolicy	Message Security Mode		
None[1]	None		
Basic128Rsa15[2]	None[4]	Sign[5]	SignAndEncrypt[6]
Basic256[3]	None	Sign	SignAndEncrypt
Basic256Sha256[3]	None	Sign	SignAndEncrypt
1: The certificate exchange is disabled. Every OPC UA client can log on to the WinCC OPC UA server. This setting can be disabled on each OPC UA server.			
2: Certificate exchange with depth of encryption of 128 bit.			
3: Certificate exchange with depth of encryption of 256 bit.			
4: The data packages are exchanged after certificate check unsecured between client and server.			

SecurityPolicy	Message Security Mode
5: The data packages are signed with the certificates, but not encoded	
6: The data packages are signed with the certificates and encoded	

Note

Unsecured communication between client and server possible

Use the "none" setting only for test and diagnostics purposes.

For a secure communication between client and server, use in operating mode at least the following settings:

- SecurityPolicy: Basic128Rsa15
- Message Security Mode: Sign

The SecurityPolicy - Basic256Sha256 setting is recommended.

1.2.5.8 Description of Log File Entries

Introduction

Important changes of state and errors are entered in the log file through the channel. Only the most important entries are described in the following sections. These entries can be used to further analyze a communication problem.

There are two type of such problems:

- INFO
- ERROR

Logging must be enabled in the channel diagnosis. The file name and flags (trace level) are also configured there.

The log files are stored here: c:\Program Files (x86)\Siemens\Automation\SCADA-RT_V1x\WinCC\diagnose\.

Systematic structure of an entry

Date/Time Stamp	Flag Name	Description

Example of entries in the logbook

2016-03-24 10:43:18.756 INFO Log starting ...

2016-03-24 10:43:18.756 INFO | LogFileName : C:\Program Files (x86)\Siemens\Automation\SCADA-RT_V1x\WinCC\diagnose

2016-03-24 10:43:18.756 INFO | LogFileCount : 3

2016-03-24 10:43:18.756 INFO | LogFileSize : 1400000

2016-03-24 10:43:18.756 INFO | TraceFlags : fa000007

2016-03-24 10:43:18.756 INFO Process attached at 2016-03-24 09:43:18.746 UTC

2016-03-23 10:46:18.756 INFO Process detached at 2016-03-2410:46:18.746UTC

2016-03-27 13:22:43.390 ERROR ..FOPCData::InitOPC CoCreateInstanceEx- ERROR 800706ba

2016-03-27 13:22:43.390 ERROR - ChannelUnit::SysMessage("[OPC Groups (OPCHN Unit #1)]![OPC_No_Machine]: CoCreateInstance for server "OPCServer.WinCC" on machine OPC_No_Machine failed, Error=800706ba (HRESULT = 800706ba - RPC_S_SERVER_UNAVAILABLE (The RPC Server is unavailable.))")

1.2.6 Commissioning OPC

1.2.6.1 Configuring DCOM user permissions in Windows

Introduction

The OPC DA client and OPC DA server are DCOM applications, whose security settings must be set in compliance with the DCOM security mechanisms:

- The OPC DA client needs launch/activation rights and access rights for the OPC DA server.
- The OPC DA server only needs access rights for the OPC DA client.

The following must be known on the PCs of the OPC DA server and the OPC DA client respectively:

- The user account for which the OPC DA client is executed
- The "Simatic HMI" user group

The "Simatic HMI" user group is already created on an HMI device during installation. The security settings of WinCC OPC DA server and the WinCC OPC channel are also preconfigured during installation.

In order to create an additional user for the operation of the OPC DA client, add the user as a member to the "Simatic HMI" user group.

Requirement

You have administrator rights.

Setting up a user account on PCs with an external OPC DA server or external client

On PCs without a WinCC installation, you configure the access rights manually.

1. Open Windows user administration.
2. Create the "SIMATIC HMI" user group.

3. Under "DCOM > My Computer > COM Security > Access Permissions > Edit Default" enter the permissions "Local Access" and "Remote Access".
4. Assign the user accounts for this user group.

For more information on assigning user rights, see the documentation for Windows XP / 7.

1.2.6.2 Adjusting Windows firewall settings

Introduction

After installation of WinCC, the Windows firewall settings of the WinCC OPC servers are correctly configured.

If OPC clients access OPC servers in different subnets, you must adapt the configuration of the permitted network areas to the OPC servers.

Communicating with controllers

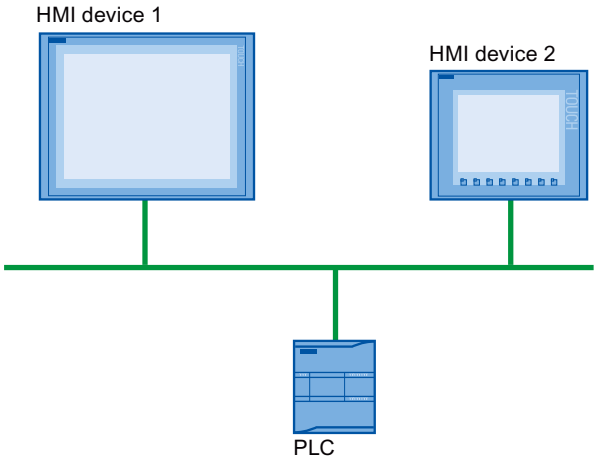
2

2.1 Basics of communication

2.1.1 Communication between devices

Communication

The data exchange between two 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
- Archive process values and alarms
- Document process values and 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 93)

Data exchange using tags (Page 96)

Data exchange using area pointers (Page 97)

Communication drivers (Page 98)

2.1.2 Devices and networks in the automation system

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.

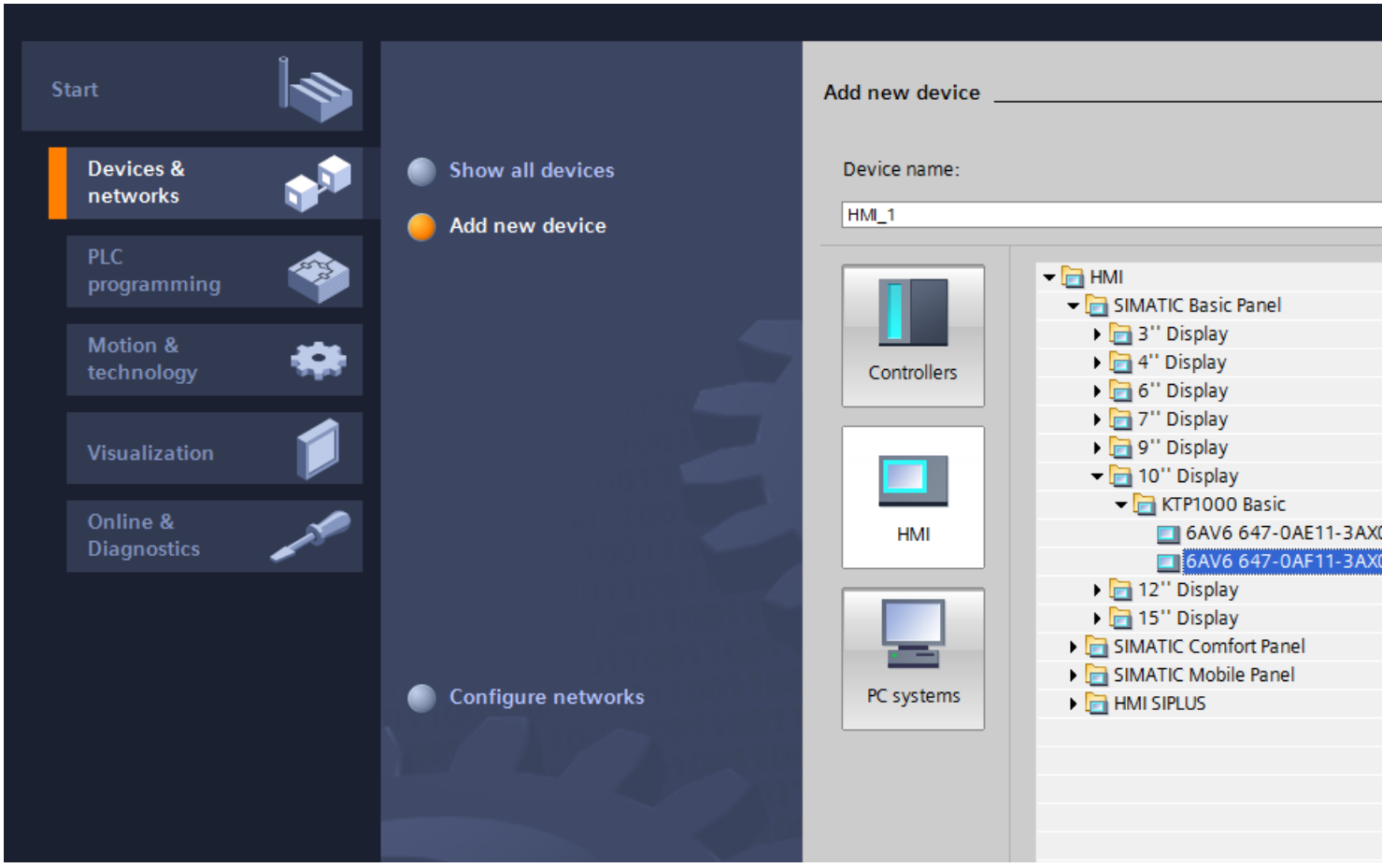
Automation system setup:

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

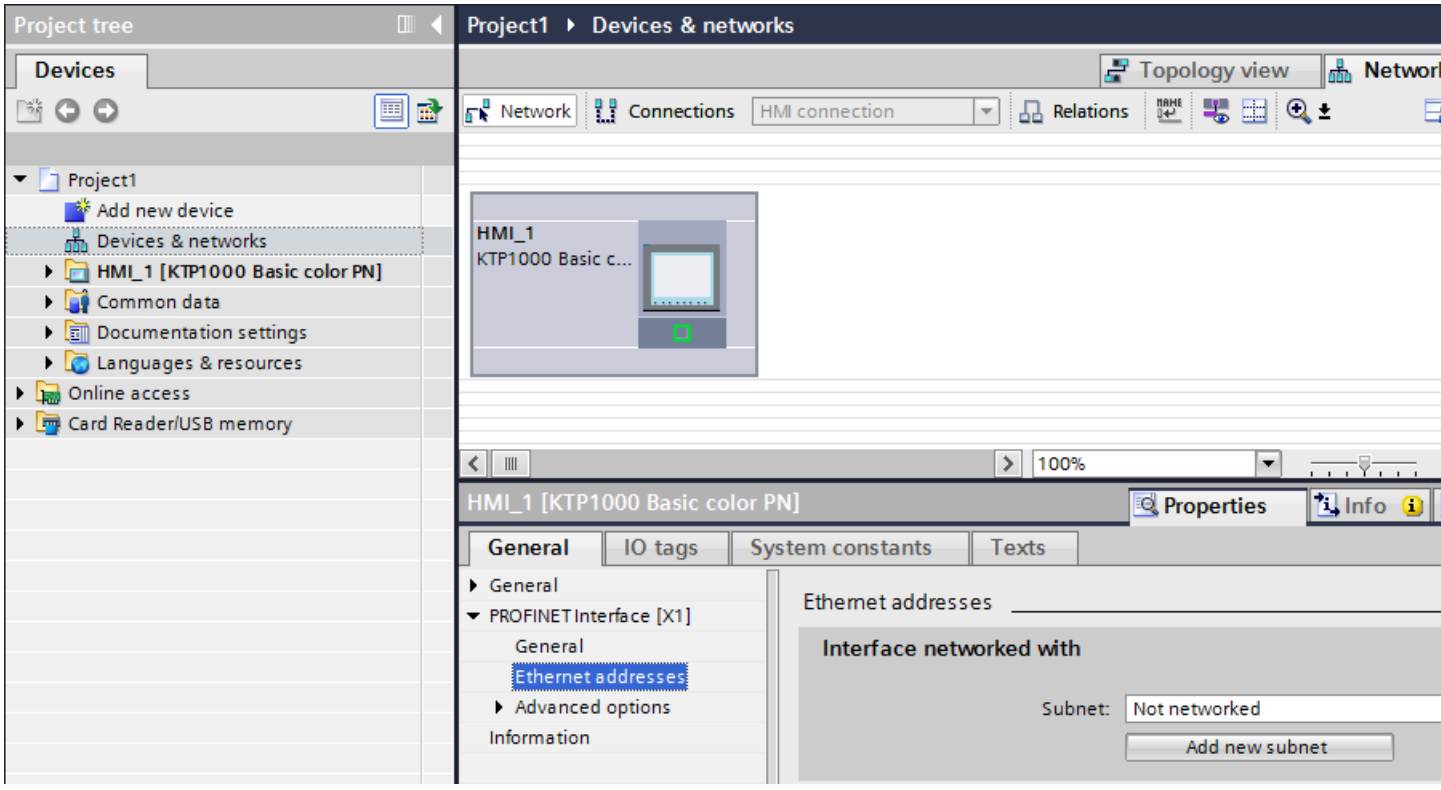
Inserting devices

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

- Portal view

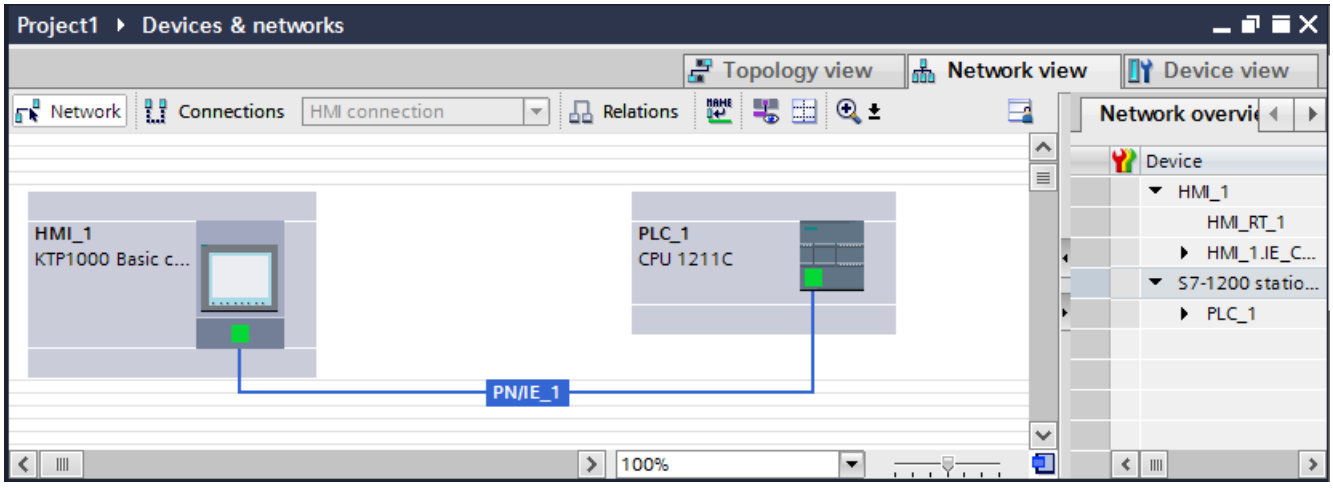


- Project view



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.

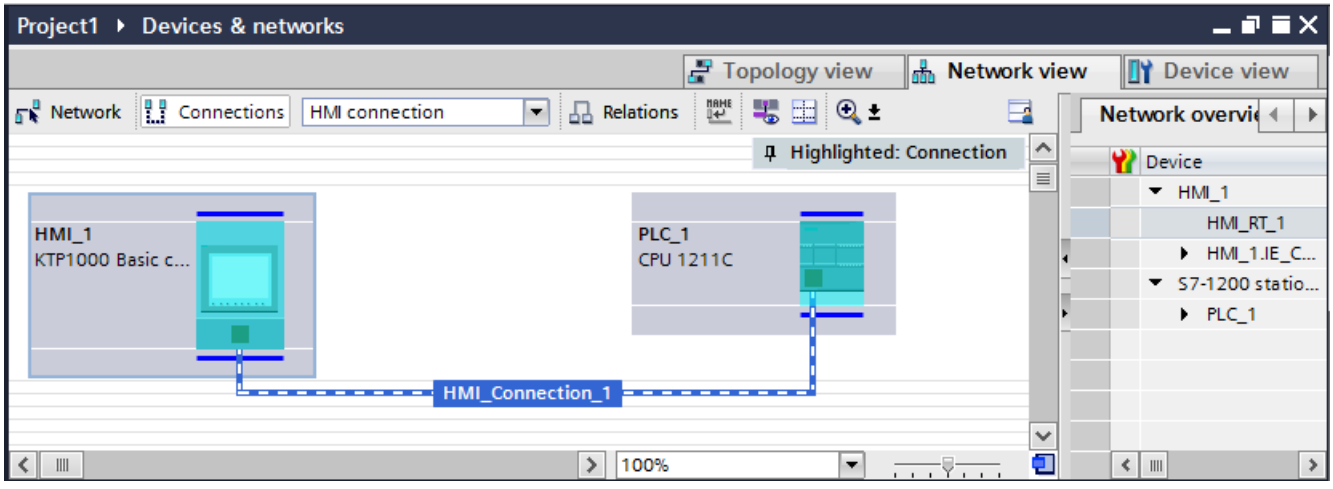


The tabular network overview supplements the graphical network view with the following additional functions:

- 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 91)

Data exchange using tags (Page 96)

Data exchange using area pointers (Page 97)

Communication drivers (Page 98)

2.1.3 Data exchange using tags

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 91)

Devices and networks in the automation system (Page 93)

Data exchange using area pointers (Page 97)

Communication drivers (Page 98)

2.1.4 Data exchange using area pointers

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 91)
Devices and networks in the automation system (Page 93)
Data exchange using tags (Page 96)
Communication drivers (Page 98)

2.1.5 Communication drivers

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 91)
Devices and networks in the automation system (Page 93)
Data exchange using tags (Page 96)
Data exchange using area pointers (Page 97)

2.1.6 Secure communication

Runtime Professional / Runtime Advanced supports secure communication to PLCs of the S7 series, e.g. S7-1200 and S7-1500.

Each connection is encrypted by the TLS protocol.

In TIA Portal you configure the secure communication to the PLCs by means of a PLC certificate.

Connection establishment

Establishing an unencrypted connection

Establishing connections to PLCs such as S7-1200 and S7-1500 with older firmware or establishing connections without PLC certificates is still possible.

Establishing an encrypted connection with secure communication

Current firmware allows connection with secure communication via TLS protocol.

Certificates from a certification authority, end entity certificates (self-signed or trusted), and certificate revocation lists, if applicable, are passed to create an encrypted session.

The session is created if the certificates match. If the certificates do not match, the establishment of the session is rejected and a system event is generated.

Connection with secure communication from Runtime Professional to the PLC

Runtime Advanced certificates are automatically handled by the engineering system and sent to the device.

The engineering system does not provide certificates for connections with a device proxy.

However, if certificates are modified in the PLC and the HMI device is not compiled again, the following scenarios may occur:

Connecting Runtime Professional without certificate to a PLC with certificate

- The connection will be established the next time you connect.

Connecting Runtime Professional to a PLC with the same certificate

- The connection will be established the next time you connect.

Connecting Runtime Professional without certificate to a PLC with certificate

- If the PLC is configured to allow HMI connections with and without secure communication, the connection is established.
- If the PLC only allows connections with secure communication, the connection is denied.
- The PLC certificate is saved in "%PROGRAMDATA%/Siemens/Automation/device-certificate-store/**untrusted**/certs". . The certificate file contains the IP address of the PLC and a fingerprint of the certificate.
- Every 30 seconds the HMI device tries to establish the connection again. If the certificate is copied from "%PROGRAMDATA%/Siemens/Automation/device-certificate-store/**untrusted**/certs" to "%PROGRAMDATA%/Siemens/Automation/device-certificate-store/**trusted**/certs" , the PLC accepts the certificate at the next attempt and the connection is established.

The certificate must also be copied manually if the connection is to be established to a non-integrated PLC.

Connecting Runtime Professional to a PLC with different certificates

- If the HMI connection has a certificate, an attempt is made to establish the connection with this certificate. If the certificate of the HMI connection does not match that of the PLC, the connection is denied. The PLC certificate is saved in "%PROGRAMDATA%/Siemens/Automation/device-certificate-store/**untrusted**/certs". . The certificate file contains the IP address of the PLC and a fingerprint of the certificate.
- Every 30 seconds the HMI device tries to establish the connection again. If the certificate is copied from "%PROGRAMDATA%/Siemens/Automation/device-certificate-store/**untrusted**/certs" to "%PROGRAMDATA%/Siemens/Automation/device-certificate-store/**trusted**/certs" , the PLC accepts the certificate at the next attempt and the connection is established.

Connection with secure communication from Runtime Advanced to the PLC

Runtime Advanced certificates are automatically handled by the engineering system and sent to the device.

However, if certificates are modified in the PLC and the HMI device is not compiled again, the following scenarios may occur:

Connecting Runtime Advanced without certificates to a PLC without certificate

- The connection will be established the next time you connect.

Connecting Runtime Advanced to a PLC with the same certificate

- The connection will be established the next time you connect.

Connecting Runtime Advanced without certificate to a PLC with certificate

- If the PLC is configured to allow HMI connections with and without secure communication, the connection is established.
- If the PLC only allows connections with secure communication, the connection is denied. The PLC certificate is saved in "C/ProgramData/Siemens/CoRThmiRTm/OMS/Untrusted". . The certificate file contains the IP address of the PLC and a fingerprint of the certificate.
- Every 30 seconds the HMI device tries to establish the connection again. The PLC accepts the certificate on the next attempt and the connection is established.

Connecting Runtime Advanced to a PLC with the different certificates

- If the HMI connection has a certificate, an attempt is made to establish the connection with this certificate. If the certificate of the HMI connection does not match that of the PLC, the connection is denied. The PLC certificate is saved in "C/ProgramData/Siemens/CoRThmiRTm/OMS/Untrusted". . The certificate file contains the IP address of the PLC and a fingerprint of the certificate.
- Every 30 seconds the HMI device tries to establish the connection again. The PLC accepts the certificate on the next attempt and the connection is established.

Certificate Revocation List (CRL)

A certificate revocation list contains certificates that have been revoked before the end of the validity period and are therefore no longer trusted. A certificate revocation list is provided as a CERT file (encoded in PEM or DER format). More than one certificate revocation list can exist.

Certificate revocation lists are stored under the path "{ProgramData}//Siemens//Automation//device-certificate-store//trusted//crl/" .

While the HMI connection to a PLC is being established, the PLC certificate is checked against the certificate revocation lists. If the certificate is listed there, the connection is denied. An alarm is output.

2.2 Editors for communication

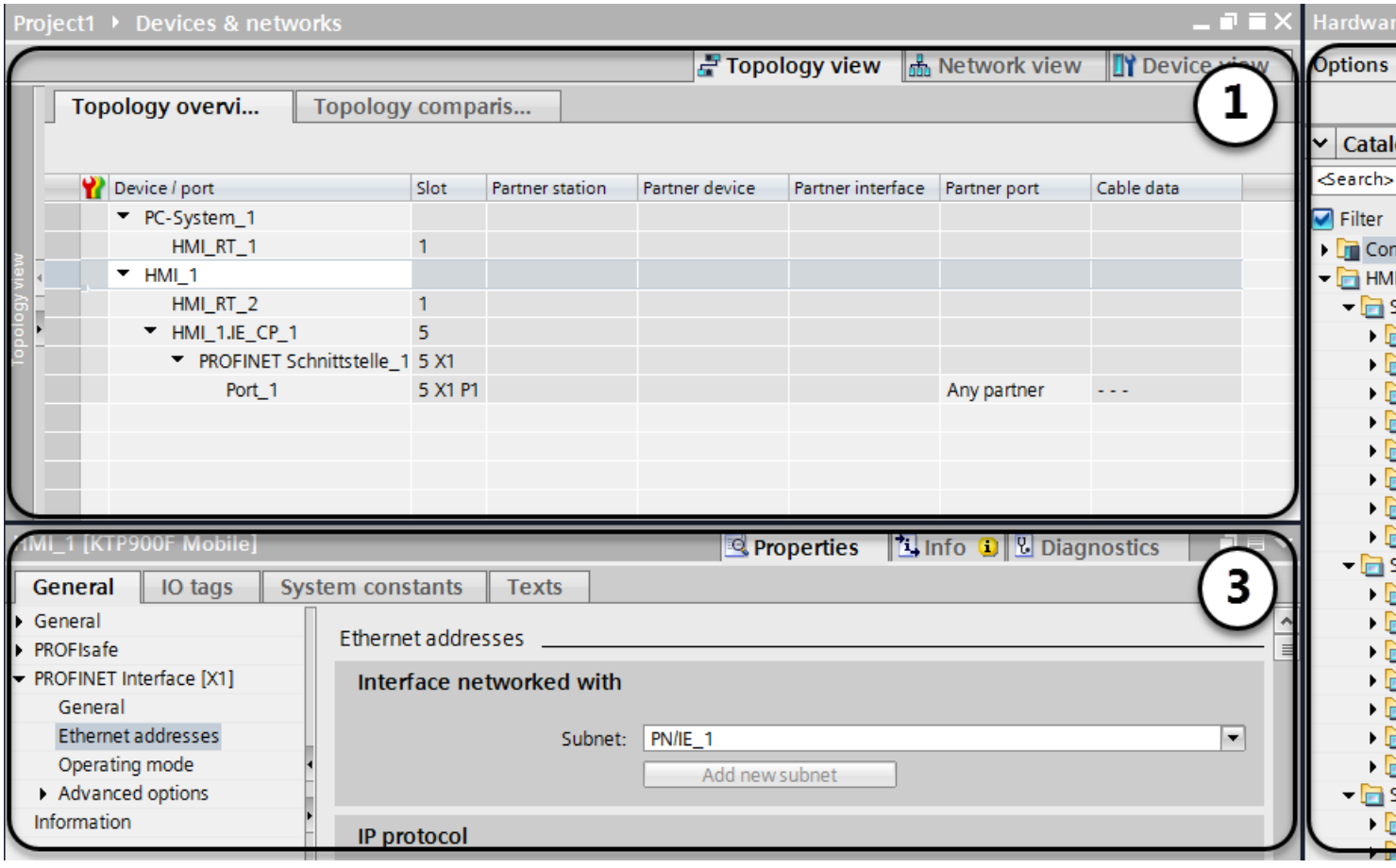
2.2.1 "Devices & networks" editor

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 103)
Network data (Page 106)
Diagnostics of online connections (Page 109)
Device view (Page 110)
Topology view (Page 113)
Inspector window (Page 116)
Hardware catalog (Page 117)
Information on hardware components (Page 120)

2.2.2 Network view

Introduction

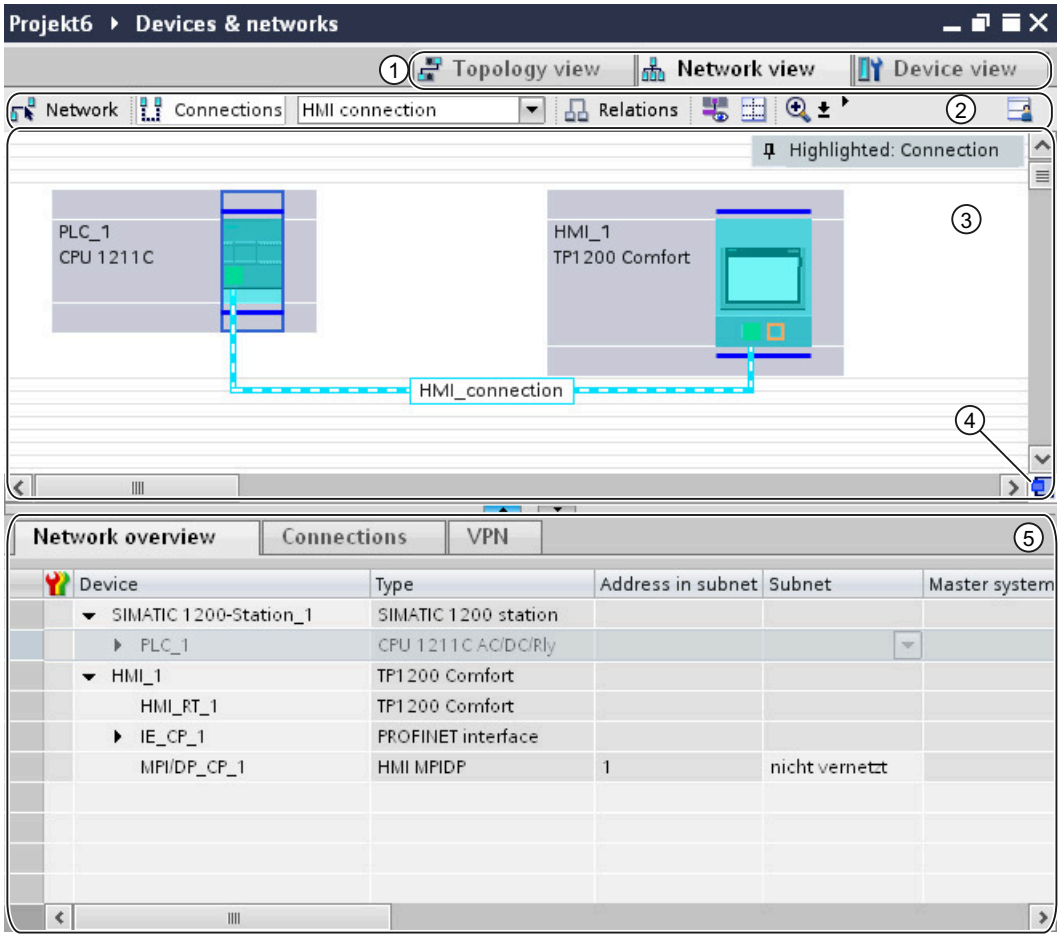
The network view is a working area of the hardware and network editor.

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 101)

Network data (Page 106)

Diagnostics of online connections (Page 109)
Device view (Page 110)
Topology view (Page 113)
Inspector window (Page 116)
Hardware catalog (Page 117)
Information on hardware components (Page 120)

2.2.3 Network data

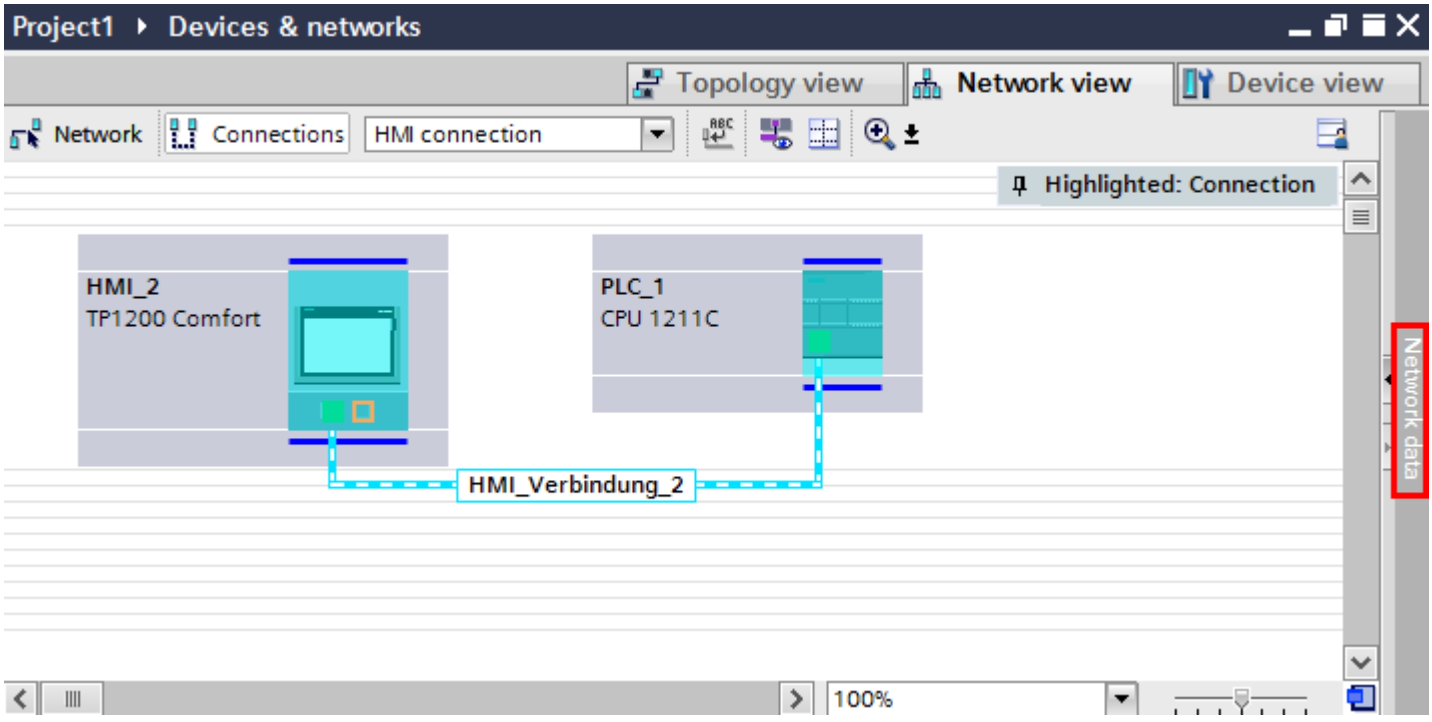
Introduction

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:

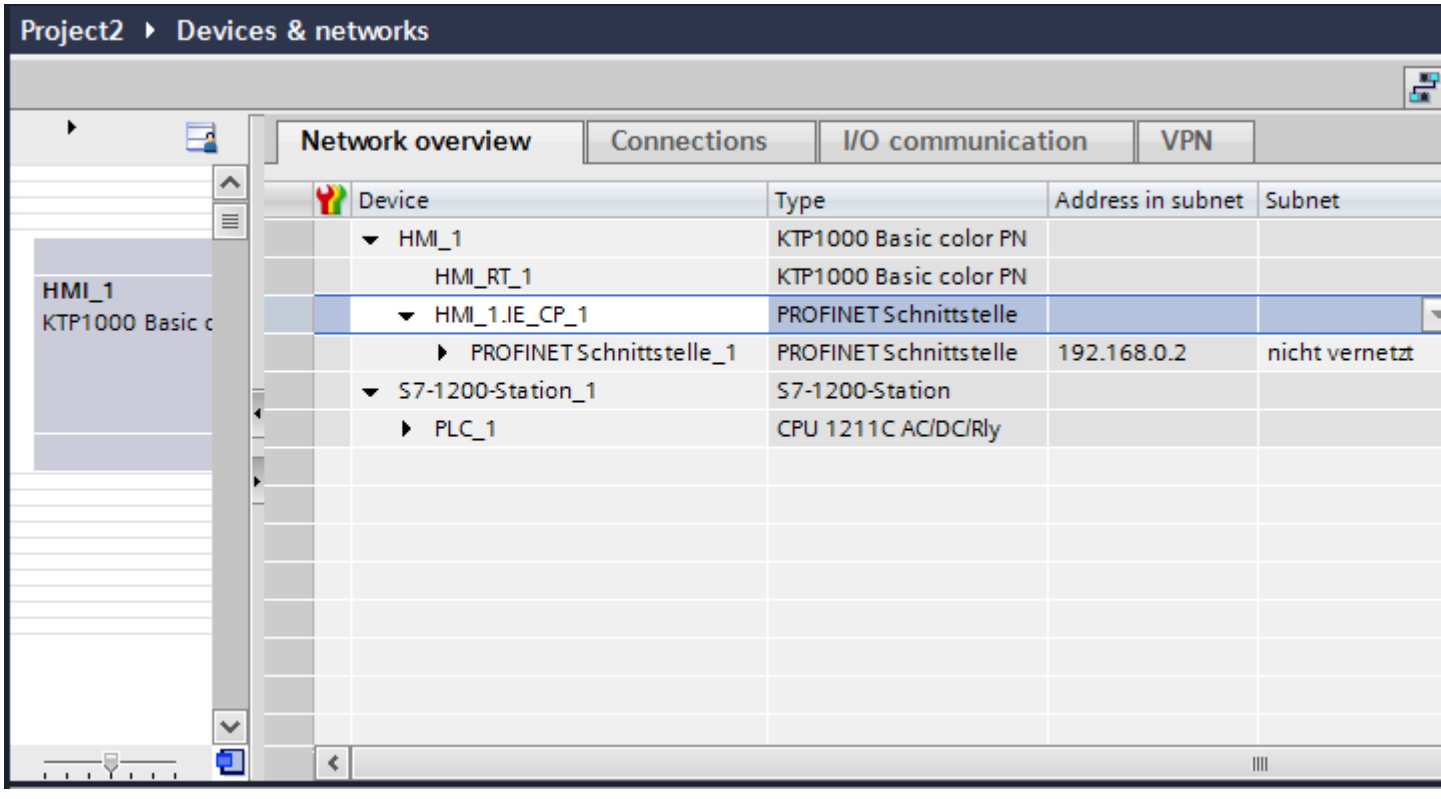
- Displaying and hiding table columns
  Note: The columns of relevance to configuration cannot be hidden.
- Optimizing column width
- Sorting table
- Displaying the meaning of a column, a row or cell using tooltips.

Network overview

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

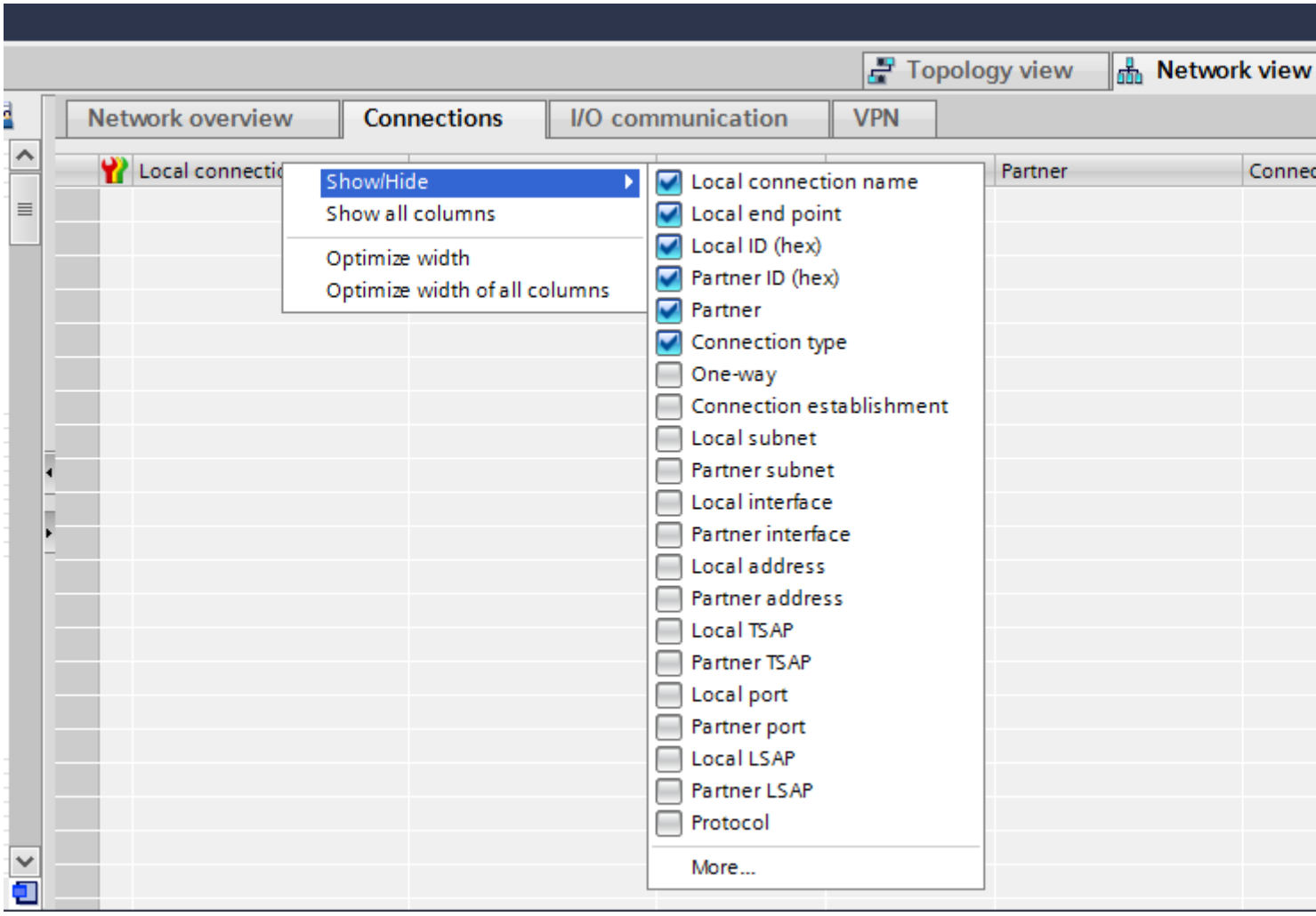
- You obtain detailed information on the structure and parameter settings of the devices.
- Using the "Subnet" column, you can connect components capable of communication with created subnets.

The



Connections

You will find additional network data under "Connections".



See also

"Devices & networks" editor (Page 101)

Network view (Page 103)

Diagnostics of online connections (Page 109)

Device view (Page 110)

Topology view (Page 113)

Inspector window (Page 116)

Hardware catalog (Page 117)

Information on hardware components (Page 120)

2.2.4 Diagnostics of online connections

Diagnostics of online connections

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

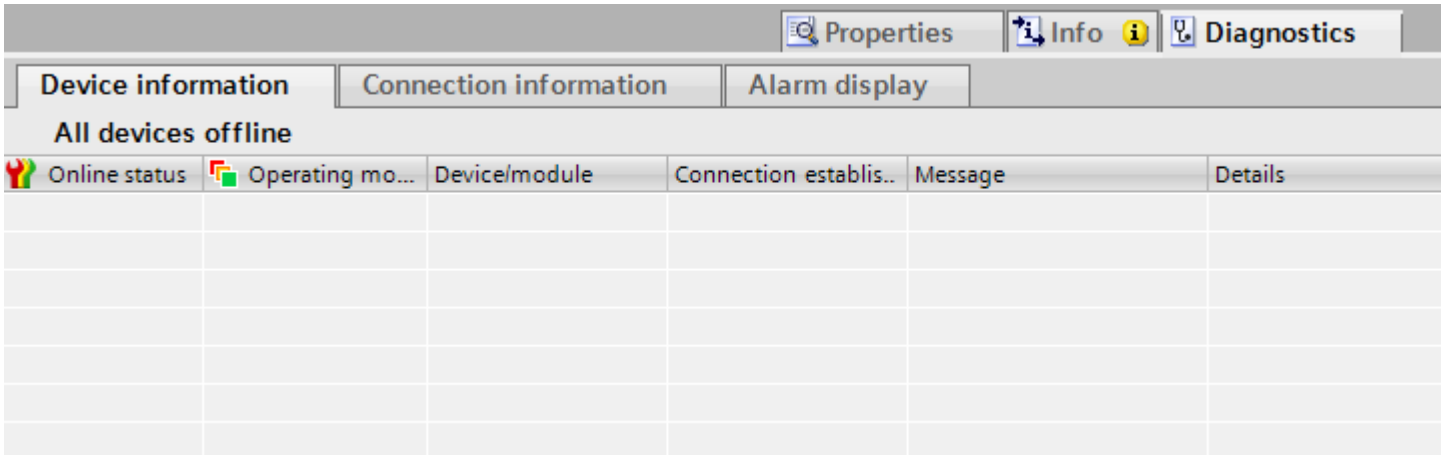
The "Diagnostics" function shows the connection data in tabular form in the Inspector window.

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.



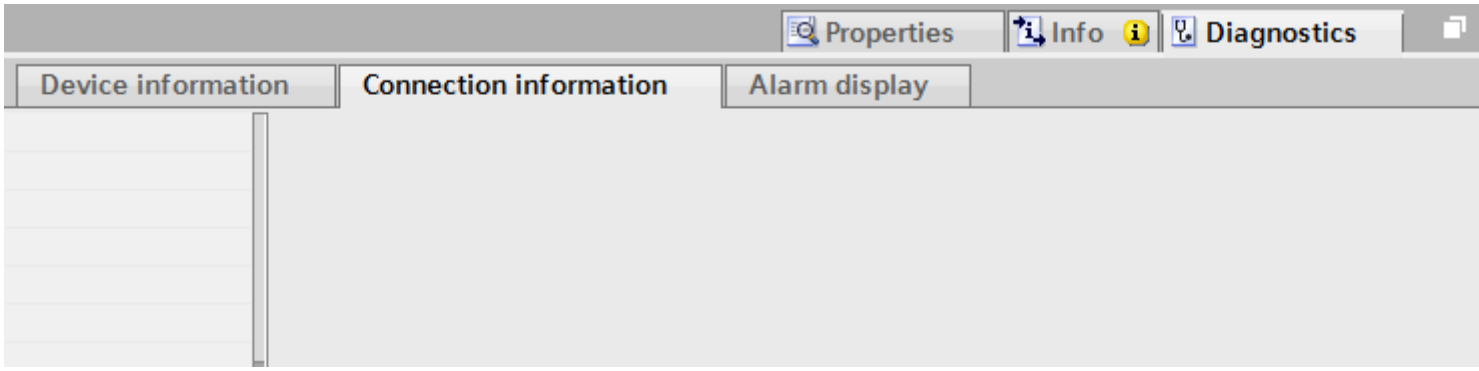
The following data is displayed:

- 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 101)
Network view (Page 103)
Network data (Page 106)
Device view (Page 110)
Topology view (Page 113)
Inspector window (Page 116)
Hardware catalog (Page 117)
Information on hardware components (Page 120)

2.2.5 Device view

Introduction

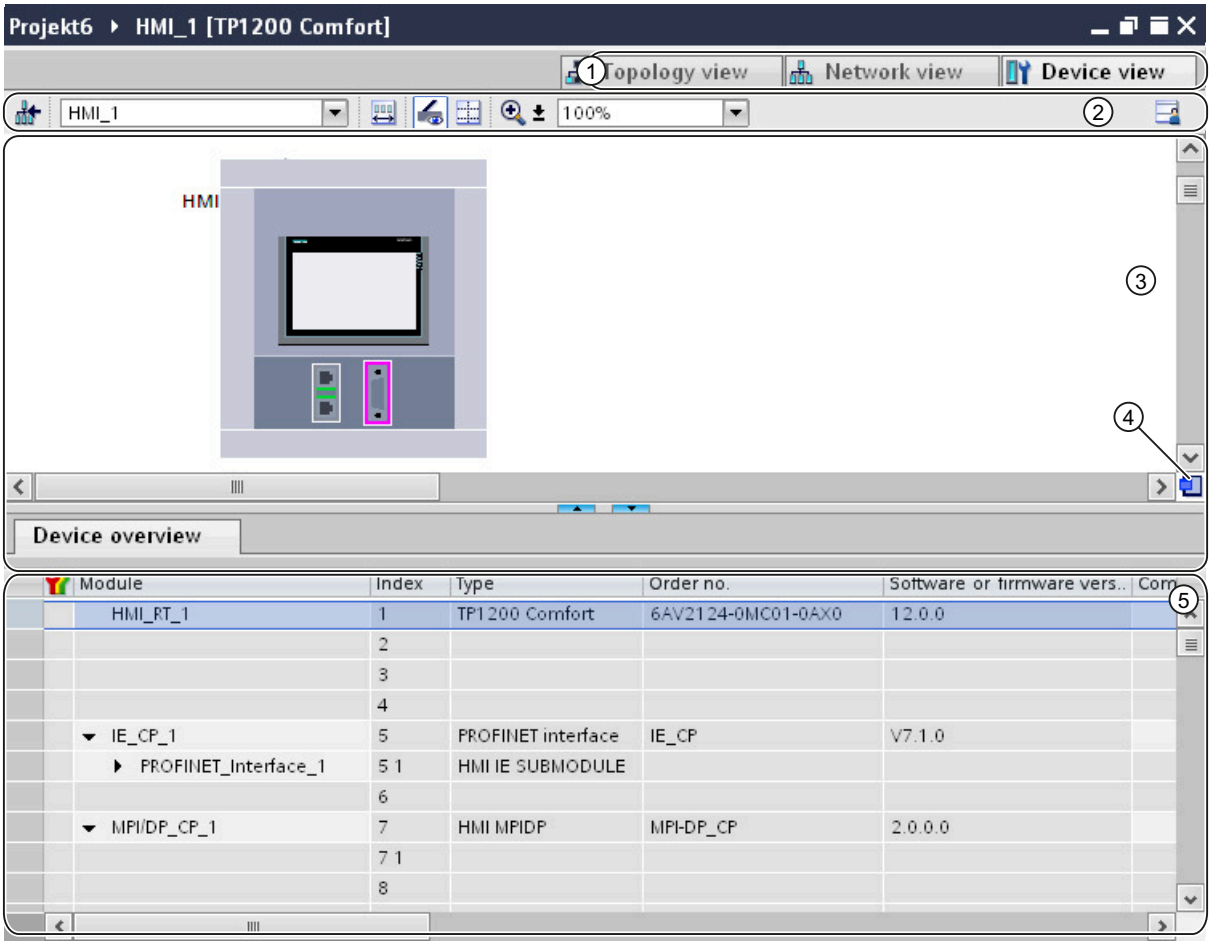
The device view is a working area of the hardware and network editor.

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 101)

Network view (Page 103)

Network data (Page 106)

Diagnostics of online connections (Page 109)

Topology view (Page 113)
Inspector window (Page 116)
Hardware catalog (Page 117)
Information on hardware components (Page 120)

2.2.6 Topology view

Introduction

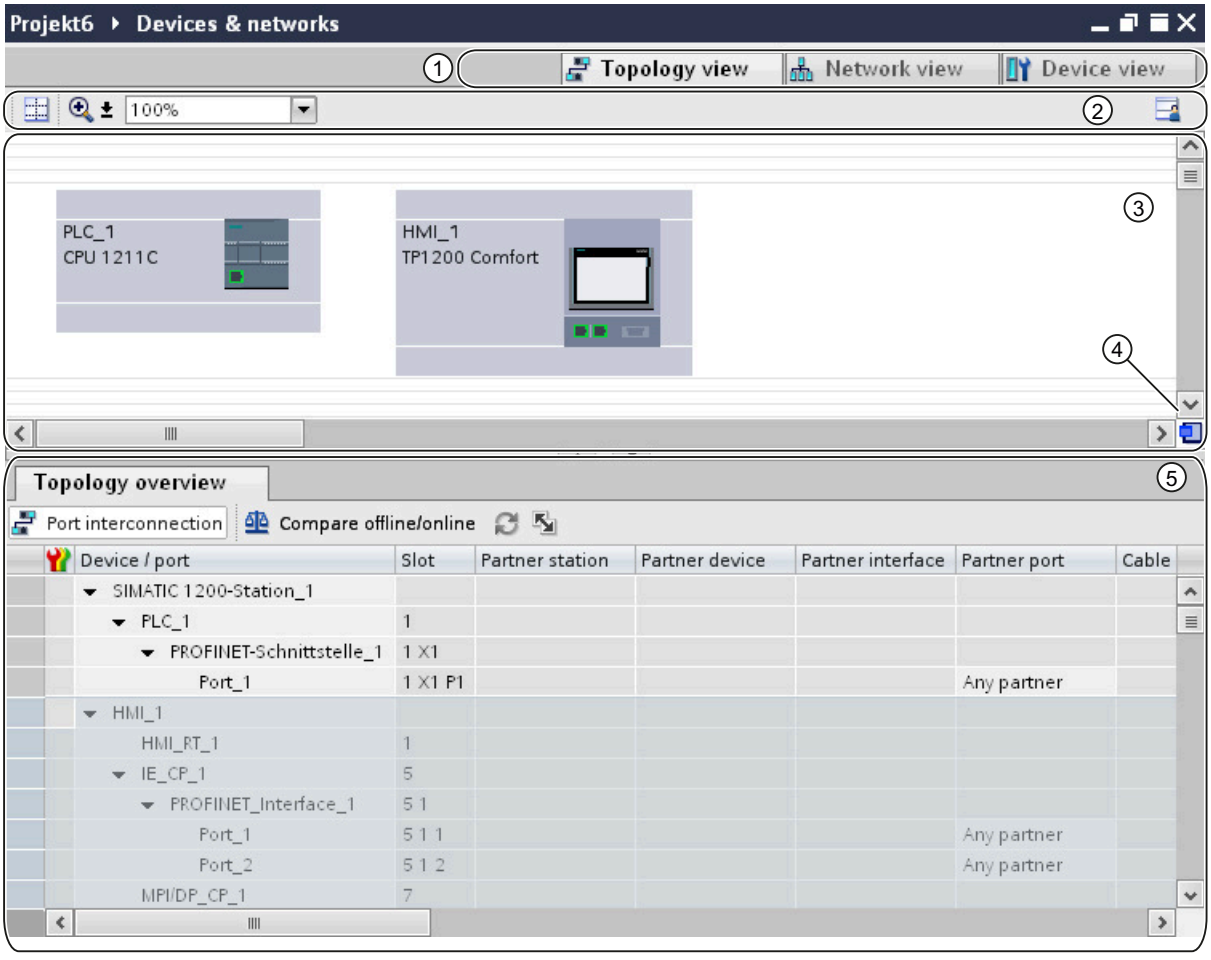
The topology view is a working area of the hardware and network editor.

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 101)

Network view (Page 103)

Network data (Page 106)

Diagnostics of online connections (Page 109)

Device view (Page 110)

Inspector window (Page 116)

Hardware catalog (Page 117)

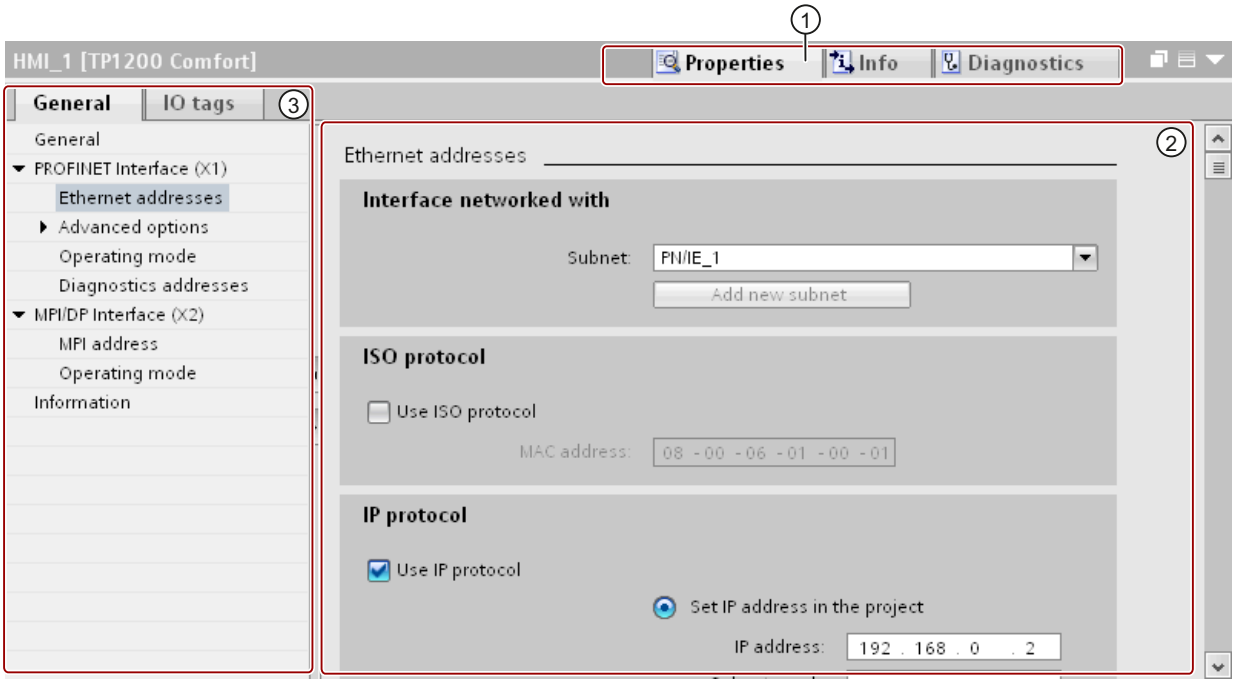
Information on hardware components (Page 120)

2.2.7 Inspector window

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 101)

Network view (Page 103)

Network data (Page 106)

Diagnostics of online connections (Page 109)

Device view (Page 110)

Topology view (Page 113)

Hardware catalog (Page 117)

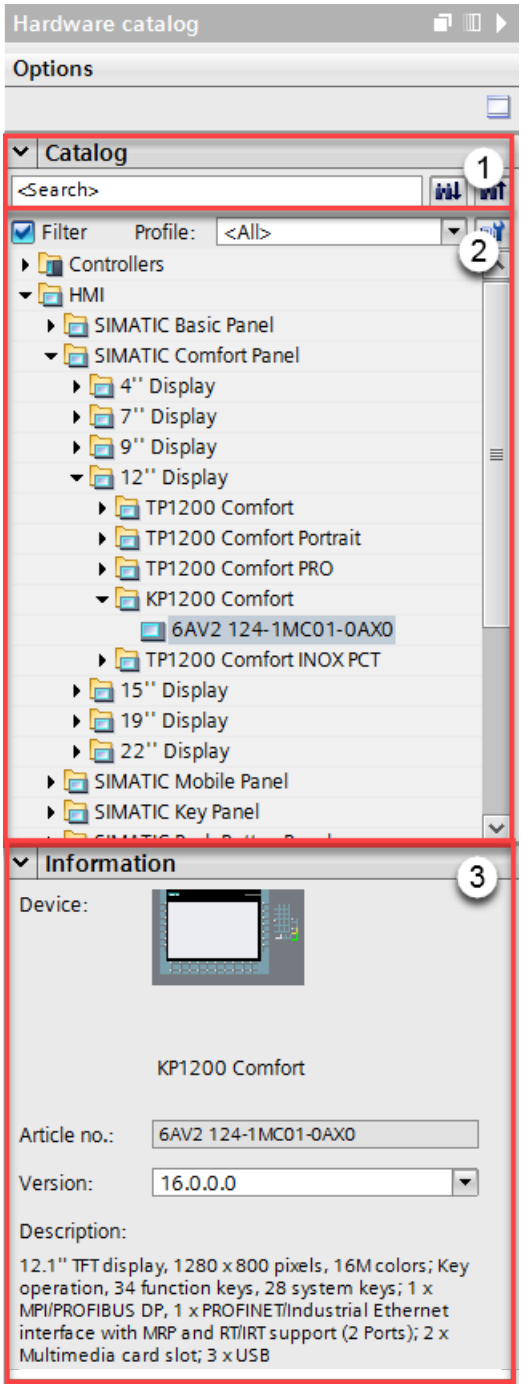
Information on hardware components (Page 120)

2.2.8 Hardware catalog

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 101)

Network view (Page 103)

Network data (Page 106)

Diagnostics of online connections (Page 109)

Device view (Page 110)

Topology view (Page 113)

Inspector window (Page 116)

Information on hardware components (Page 120)

2.2.9 Information on hardware components

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 101)

Network view (Page 103)

Network data (Page 106)

Diagnostics of online connections (Page 109)
Device view (Page 110)
Topology view (Page 113)
Inspector window (Page 116)
Hardware catalog (Page 117)

2.3 Networks and connections

2.3.1 SIMATIC communication networks

2.3.1.1 Communication networks

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:

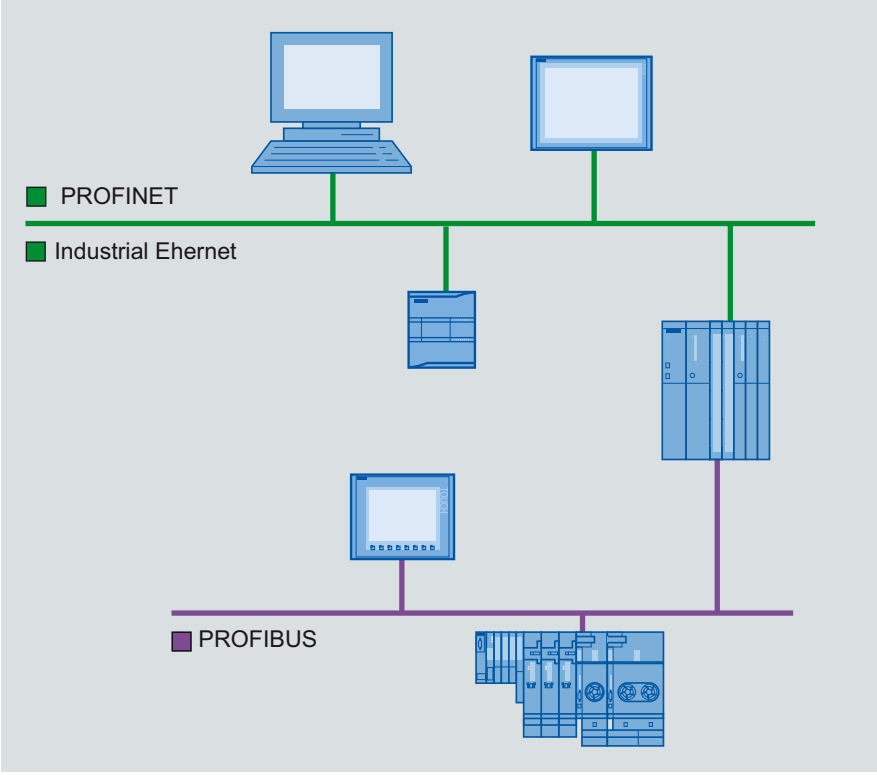
- Process and field communication of the automation systems including sensors and actuators
- Data communication between automation systems
- IT communication for integrating the modern information technology

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 122)

PROFIBUS (Page 123)

MPI (Page 124)

PPI (Page 125)

2.3.1.2 PROFINET and Ethernet

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 121)

PROFIBUS (Page 123)

MPI (Page 124)

PPI (Page 125)

2.3.1.3 PROFIBUS

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 121)

PROFINET and Ethernet (Page 122)

MPI (Page 124)

PPI (Page 125)

2.3.1.4 MPI

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 121)

PROFINET and Ethernet (Page 122)

PROFIBUS (Page 123)

PPI (Page 125)

2.3.1.5 PPI

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 121)
PROFINET and Ethernet (Page 122)
PROFIBUS (Page 123)
MPI (Page 124)

2.3.2 Configuring networks and connections

2.3.2.1 Networking devices

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

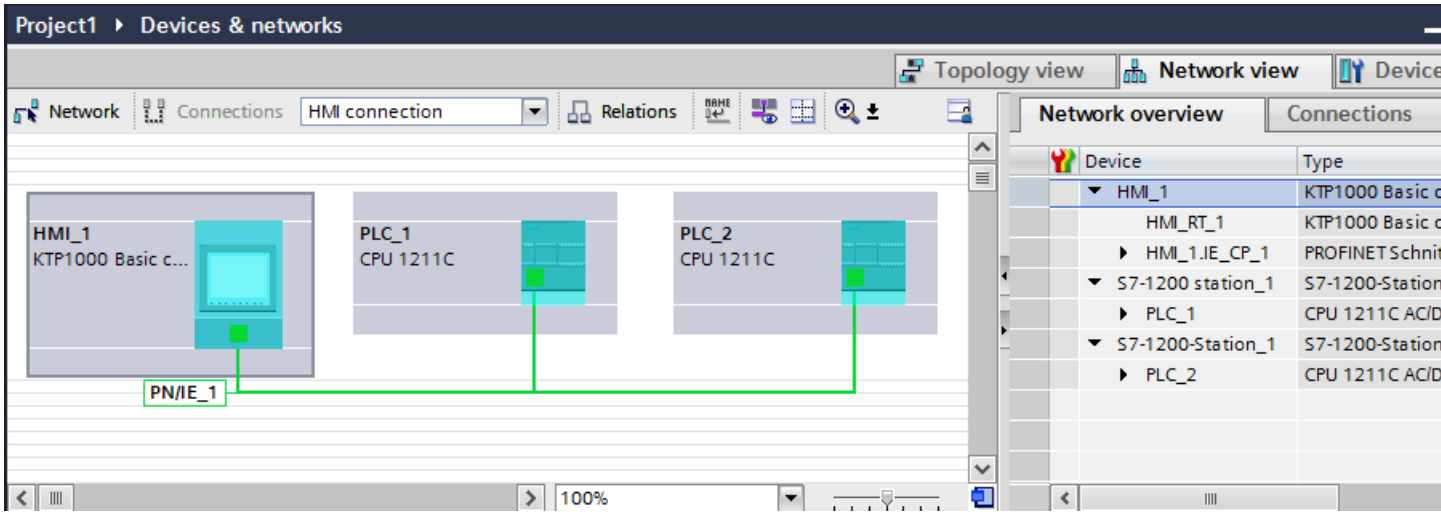
The network view of the "Devices & Networks" editor includes a graphical area and a tabular area. You can use the graphical area to network the devices in the project with drag-and-drop. The tabular area provides an overview of the devices and their components.

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 127)

Special considerations of the "Devices & Networks" editor (Page 129)

Configuring a non-integrated connection in the "Connections" editor (Page 131)

Configuring a routed connection in the "Devices & Networks" editor (Page 135)

2.3.2.2 Configuring an integrated connection in the "Devices & Networks" editor

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

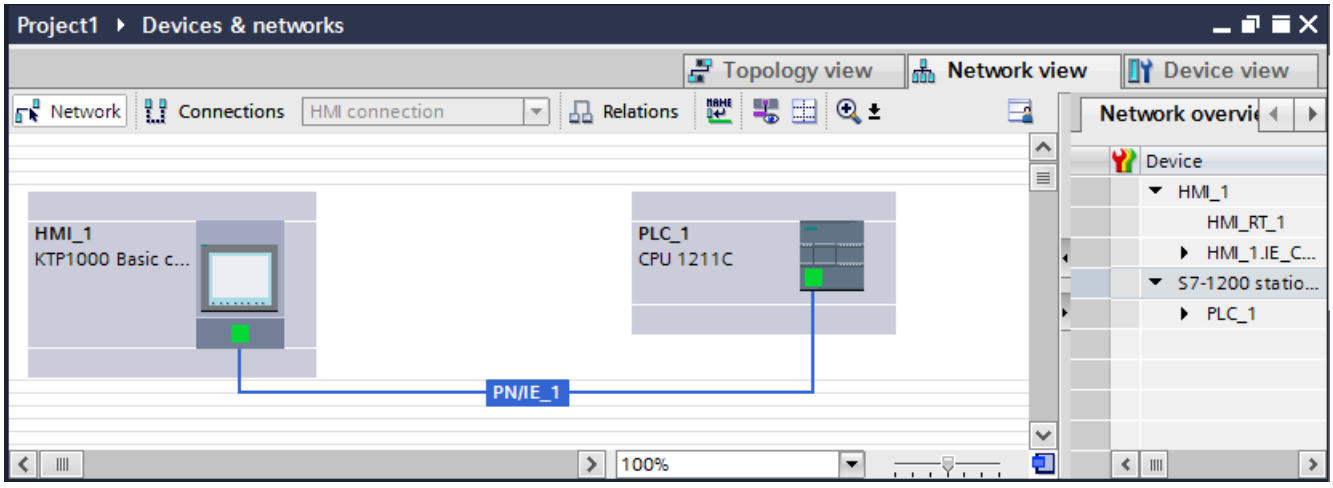
An HMI connection can be configured in the "Devices & Networks" editor for the following PLCs only:

- 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.

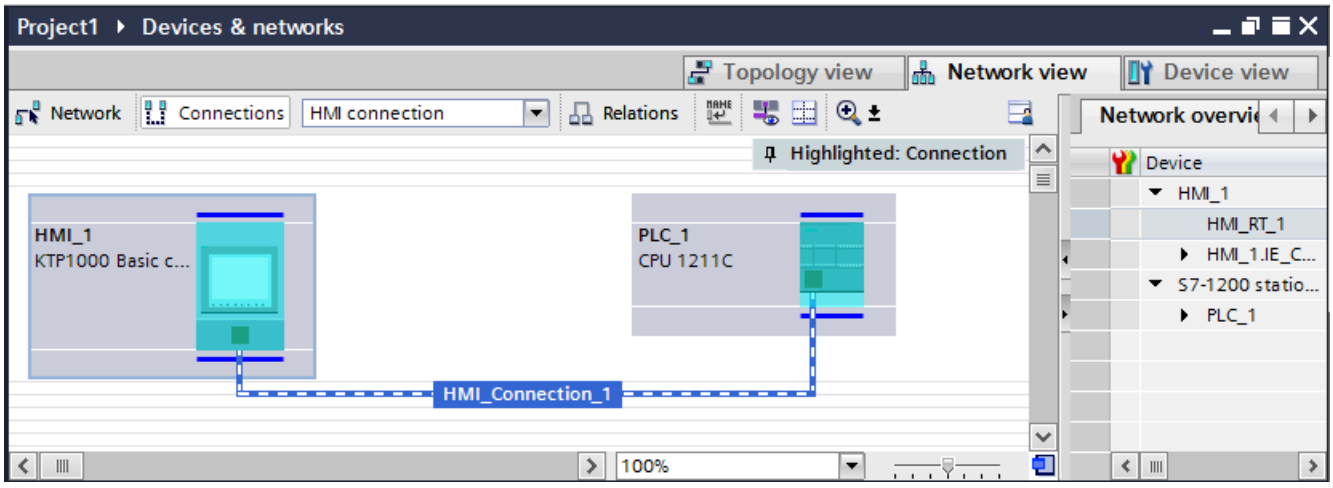
Configuring an HMI connection in the "Devices & Networks" editor

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 126)

Special considerations of the "Devices & Networks" editor (Page 129)

Configuring a non-integrated connection in the "Connections" editor (Page 131)

Configuring a routed connection in the "Devices & Networks" editor (Page 135)

2.3.2.3 Special considerations of the "Devices & Networks" editor

Introduction

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

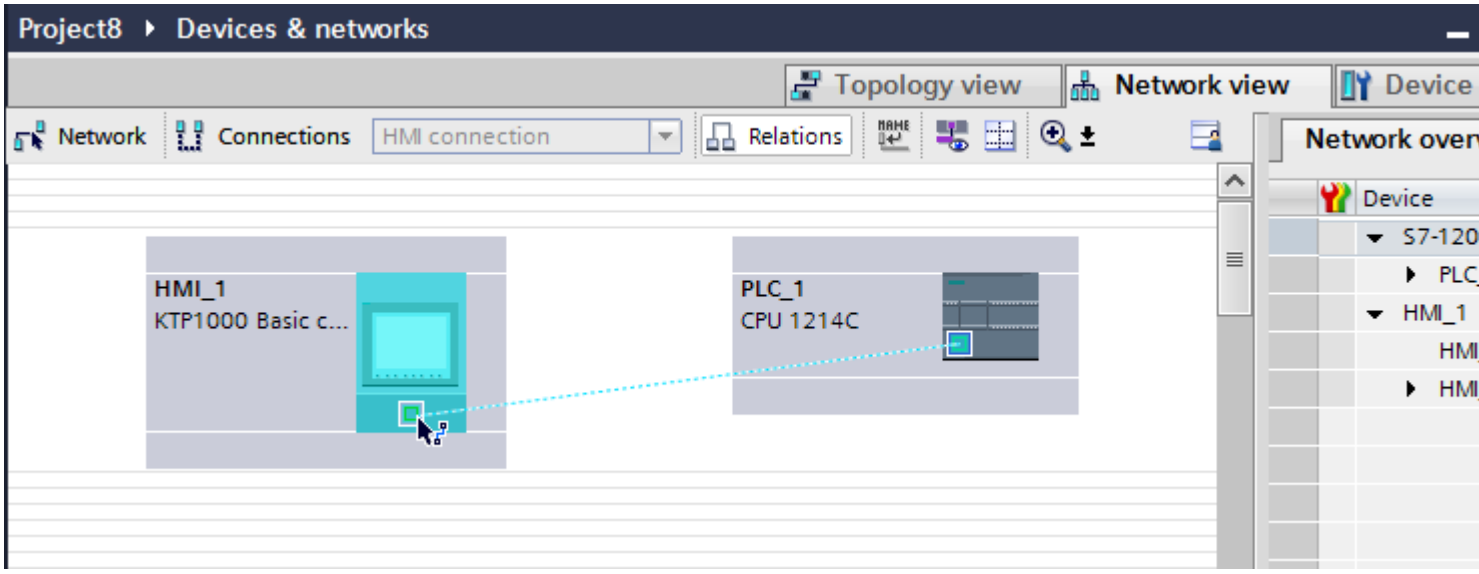
- Highlighting of communication partners
- Highlighting of HMI connections
- Automatic creation of subnets

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.



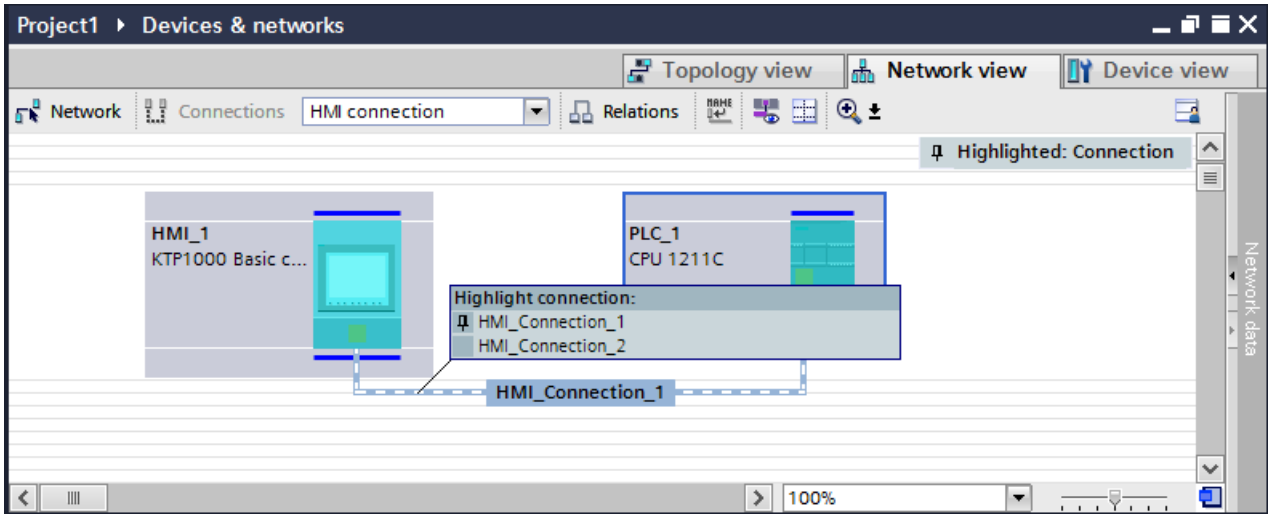
When the mouse pointer is moved over the interface of a device, the following icons indicate whether a connection is possible:

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 126)

Configuring an integrated connection in the "Devices & Networks" editor (Page 127)

Configuring a non-integrated connection in the "Connections" editor (Page 131)

Configuring a routed connection in the "Devices & Networks" editor (Page 135)

2.3.2.4 Configuring a non-integrated connection in the "Connections" editor

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.

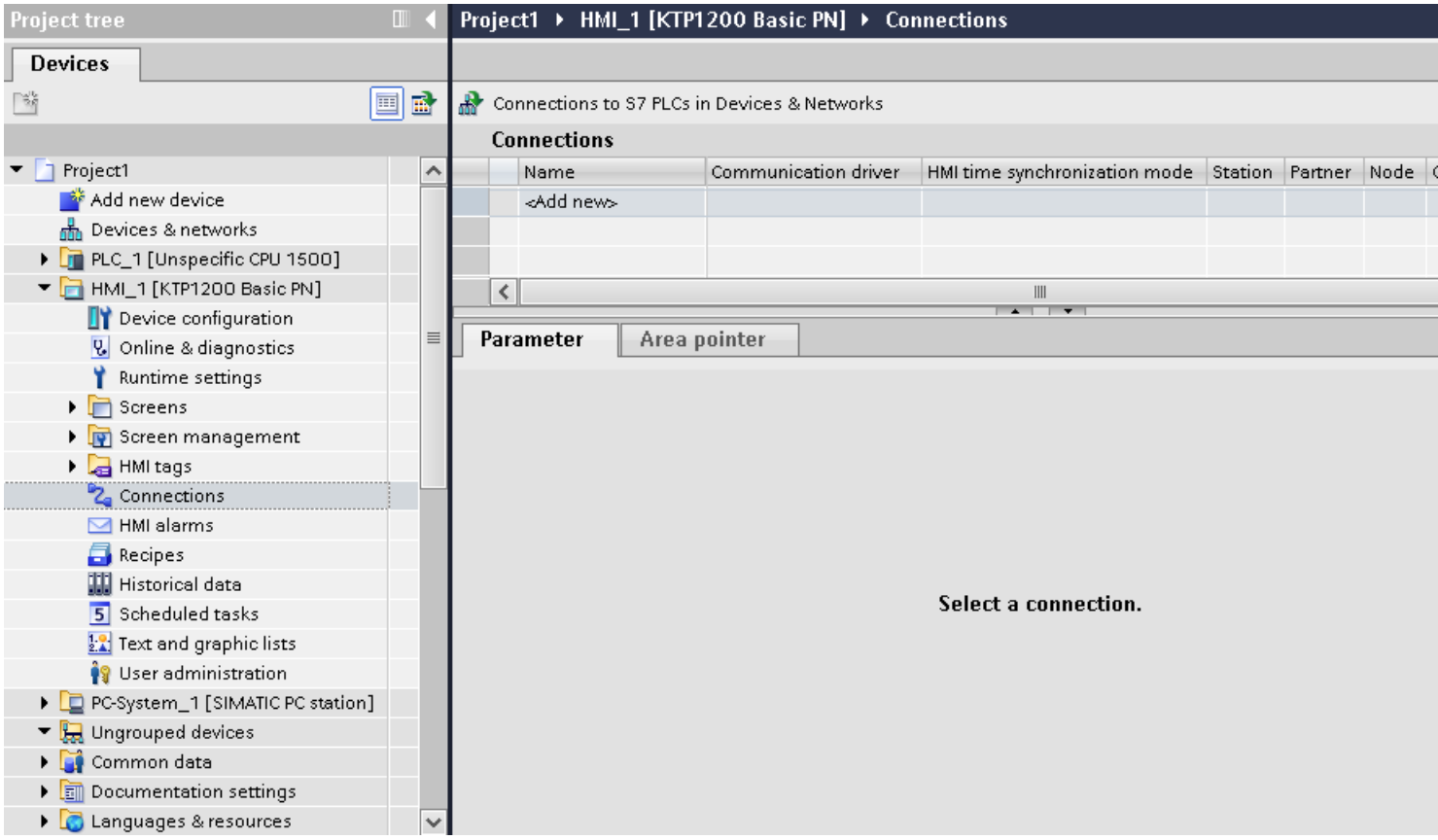
These connections are referred to as non-integrated connections.

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.

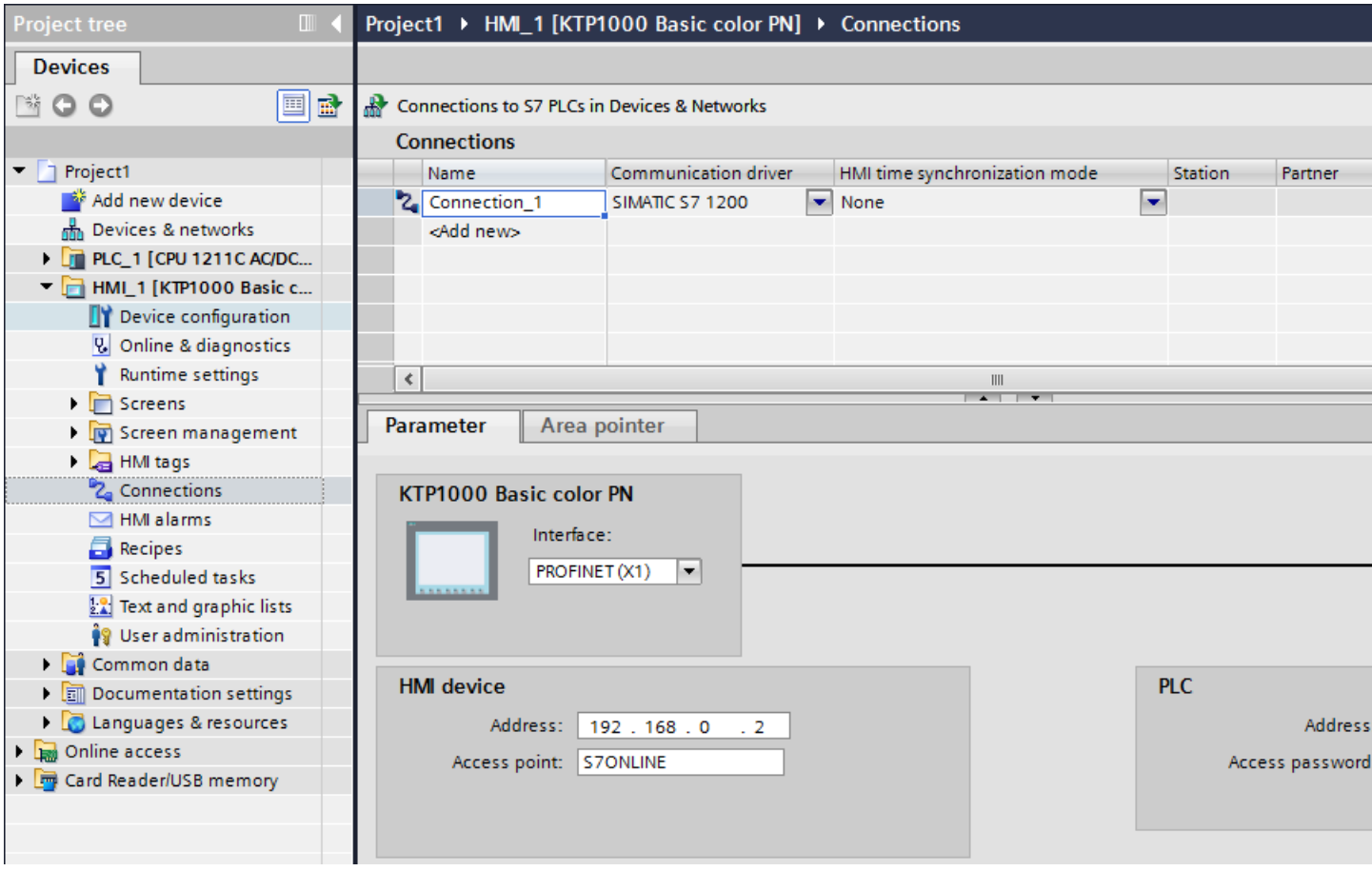


3. Select the communication driver.

Note

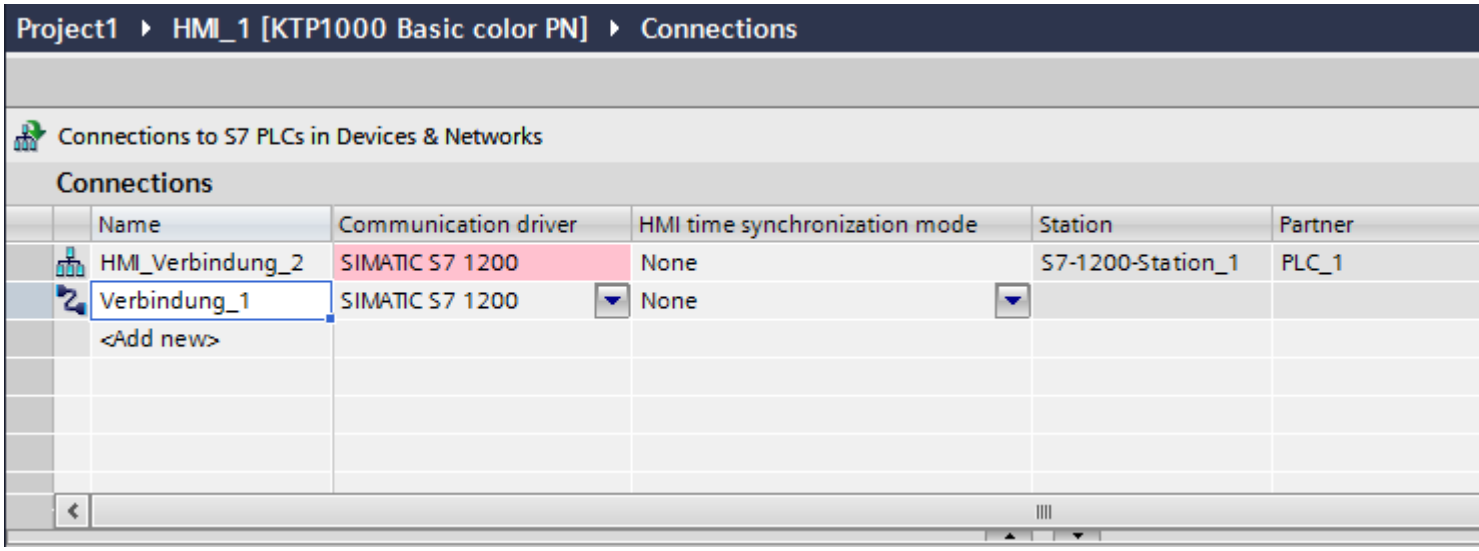
A non-integrated connection between an HMI device and a SIMATIC S7-1500 software controller is not supported in WinCC.

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 126)

Configuring an integrated connection in the "Devices & Networks" editor (Page 127)

Special considerations of the "Devices & Networks" editor (Page 129)

Configuring a routed connection in the "Devices & Networks" editor (Page 135)

2.3.2.5 Configuring a routed connection in the "Devices & Networks" editor

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

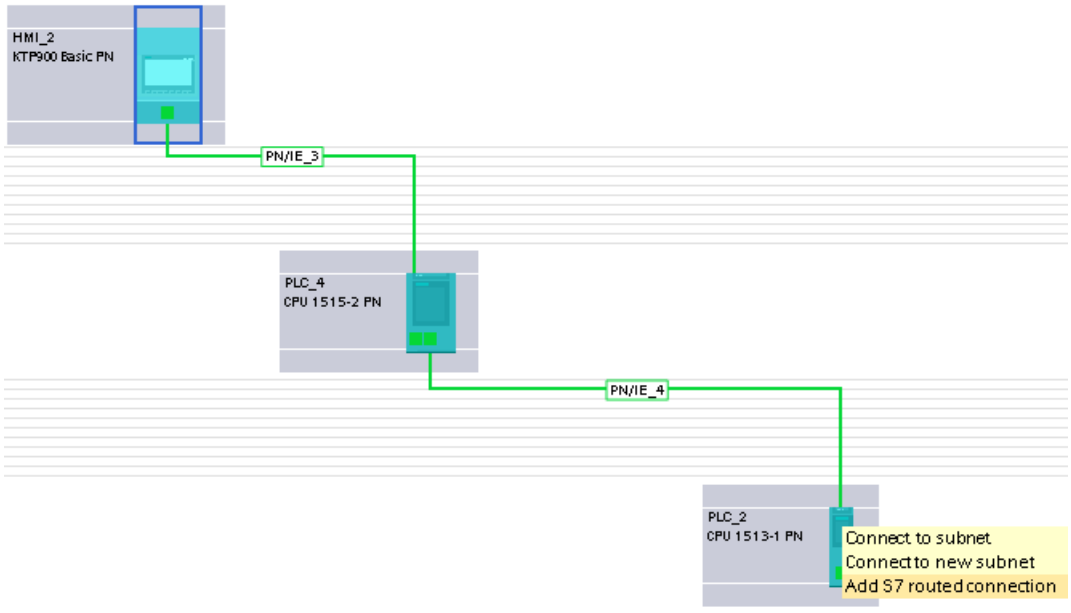
Requirement

- 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.

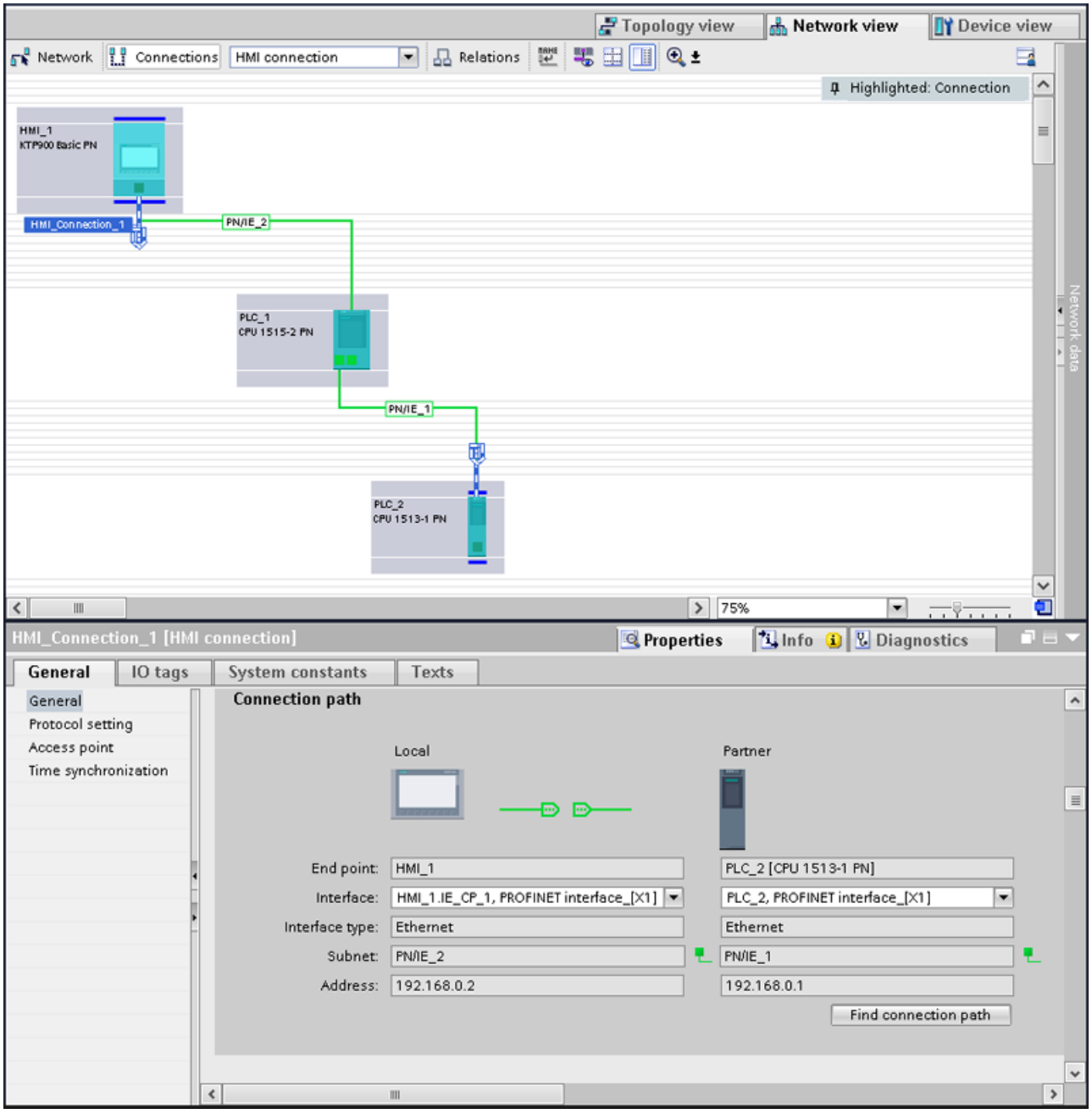
3. Drag-and-drop the HMI device to the PLC in the other subnet.
   A dialog opens.



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 126)

Configuring an integrated connection in the "Devices & Networks" editor (Page 127)

Special considerations of the "Devices & Networks" editor (Page 129)

Configuring a non-integrated connection in the "Connections" editor (Page 131)

2.3.2.6 Setting up switch on/switch off of a connection in Runtime

Introduction

If you do not require a connection between the HMI device and the controller, you can terminate the connection in Runtime and establish it again when necessary. This reduces the load on the communication channel.

Note

Redundant systems

Please note that it is also possible to switch to a redundant system when the connection is switched.

Note

Alarm system and system diagnostics

After switching off the connection to a PLC, the alarms from this PLC are still being displayed. The system diagnostics for this PLC is also available.

Overview

Switchover in Runtime takes place by means of connection-specific system tags. The state of the connection at the start of Runtime is defined by the start value of the tag.

The state of the connection is saved in an HMI tag. You use this tag for displaying screens, in scripts, and for diagnostics. You create the HMI tags for controlling the connection in Runtime in your project according to fixed specifications.

Note

Tags for controlling the connection

To use a tag to control the connection, it needs to have defined properties. You create these tags without system support. Errors are not displayed.

To avoid connection errors in Runtime, create the tags exactly according to the specifications.

Diagnostics: Performance of a connection with Runtime Professional

Use the performance tags to evaluate the time response of a connection.

Requirement

- The "HMI tags" editor is open.

Procedure

To create tags for controlling a connection in Runtime, follow these steps:

1. Create the tags for connection control in the following text format:
   - "@<Connection name>@ConnectionState", for example @HMI_Connection_1@ConnectionState
     This tag contains the current state of the connection. You only have read access to this tag.
     "@<Connection name>@ForceConnectionState", for example @HMI_connection_1@ForceConnectionState
     You set the new status of the connection in Runtime using this tag. The start value for @HMI_Verbindung_1@ForceConnectionState is 0 (not connected) or 1 (connected).
2. Configure the tag according to the following specifications:
   - Data type: "DWord"
   - PLC tag: None
   - Connection: HMI_Connection_1
   - PLC name: Not relevant
   - Address: %DB40000.DBD65000 (any unused address)
   - Access type: Absolute access

2.3.2.7 Using local ID of HMI connections for communication

Introduction

You need the local ID of an HMI connection, for example, for communication over the S7 function block AR_SEND.

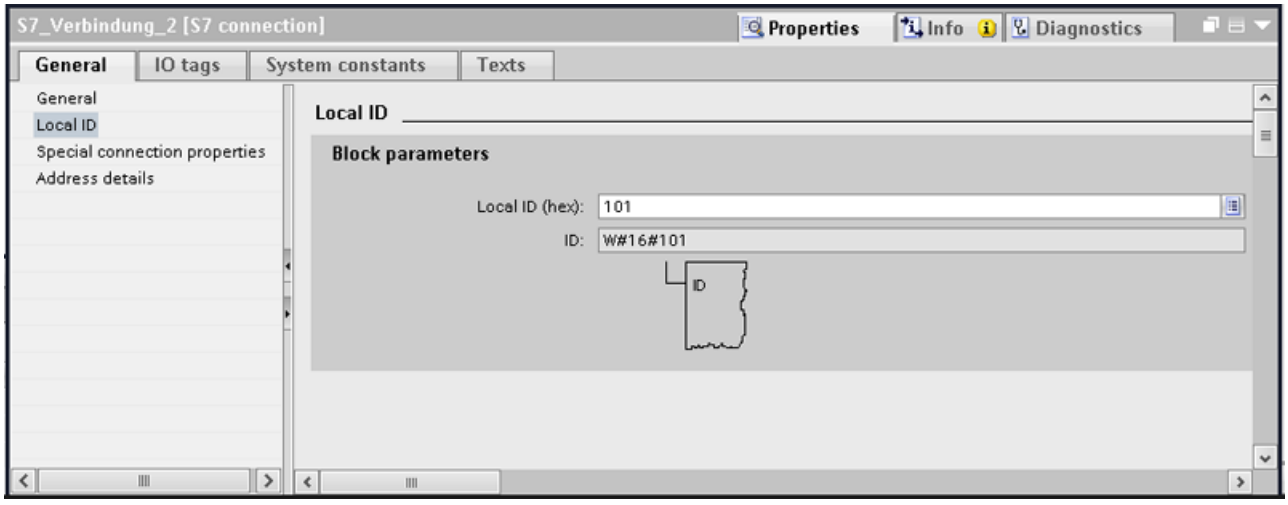
You cannot edit the "Local ID" value in the HMI connection properties.

Procedure

To enable usage of the "Local ID" for communication, follow these steps:

1. Open the network view in the "Devices & Networks" editor.
2. Click "Connections".
3. Select an S7 connection.
4. Select the "Add new connection" command in the shortcut menu of the PLC.
5. Click on the interface.

6. Specify the "Local ID (hex)".



7. Click "Add" and then "Close".
8. Select "Properties > General" from the partner area of the Inspector window and enter the IP address of the HMI device for the new connection.
9. Configure the necessary raw data tags for communication in the HMI device.

2.4 Data exchange

2.4.1 Data exchange using tags

2.4.1.1 Basics of tags

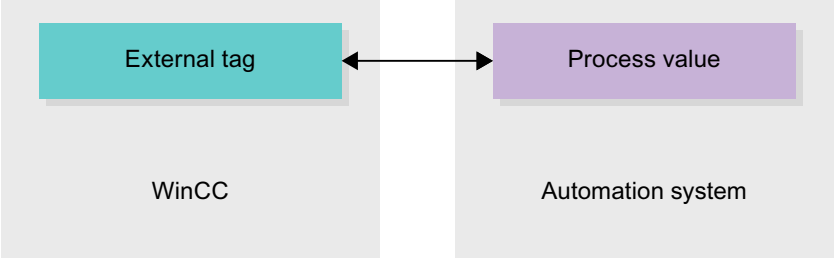
Introduction

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 141)

External tags (Page 143)

Addressing external tags (Page 145)

Internal Tags (Page 148)

2.4.1.2 Overview of HMI tag tables

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 140)

External tags (Page 143)

Addressing external tags (Page 145)

Internal Tags (Page 148)

2.4.1.3 External tags

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 91)" 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

Communication between devices (Page 91)

Basics of tags (Page 140)

Overview of HMI tag tables (Page 141)

Addressing external tags (Page 145)

Internal Tags (Page 148)

2.4.1.4 Addressing external tags

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 91)".

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 140)

Overview of HMI tag tables (Page 141)

External tags (Page 143)

Internal Tags (Page 148)

Basics of communication (Page 91)

2.4.1.5 Internal Tags

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
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 140)
Overview of HMI tag tables (Page 141)
External tags (Page 143)
Addressing external tags (Page 145)

2.4.2 Data exchange using area pointers

2.4.2.1 Basic information on area pointers

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 150)

Use of area pointers (Page 151)

Accessing data areas (Page 153)

Configuration of area pointers (Page 153)

Parameterizing a global data block (Page 154)

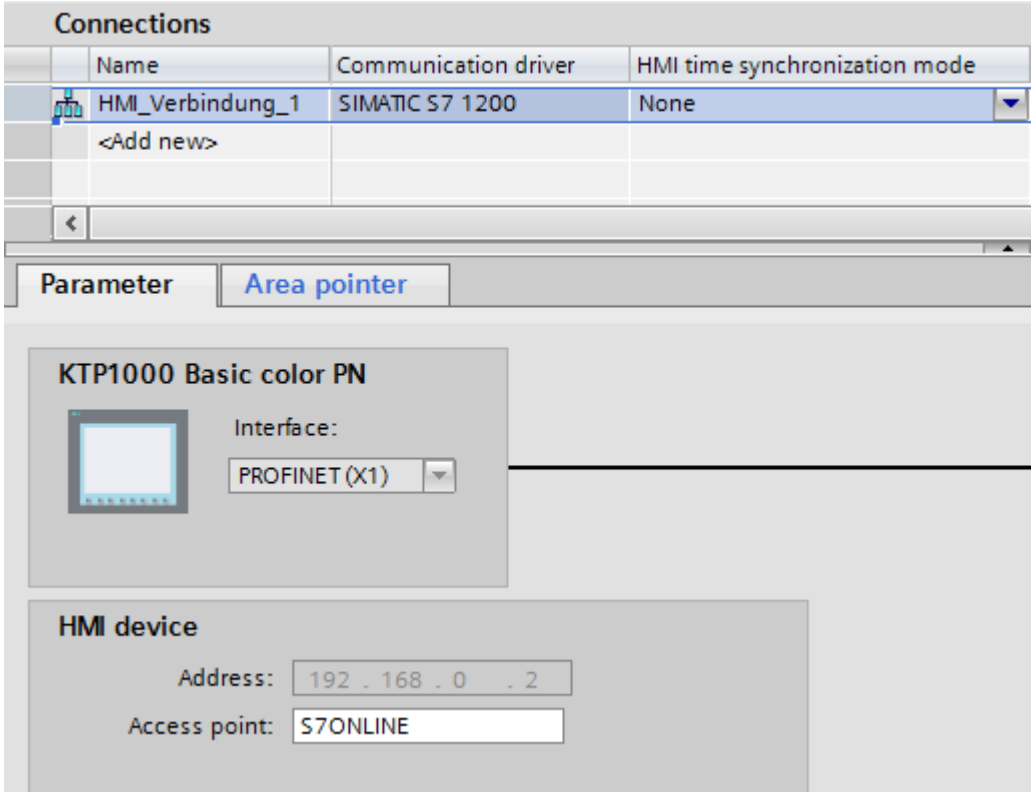
Configuring an area pointer for a connection (Page 156)

2.4.2.2 Area pointers for connections

Introduction

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

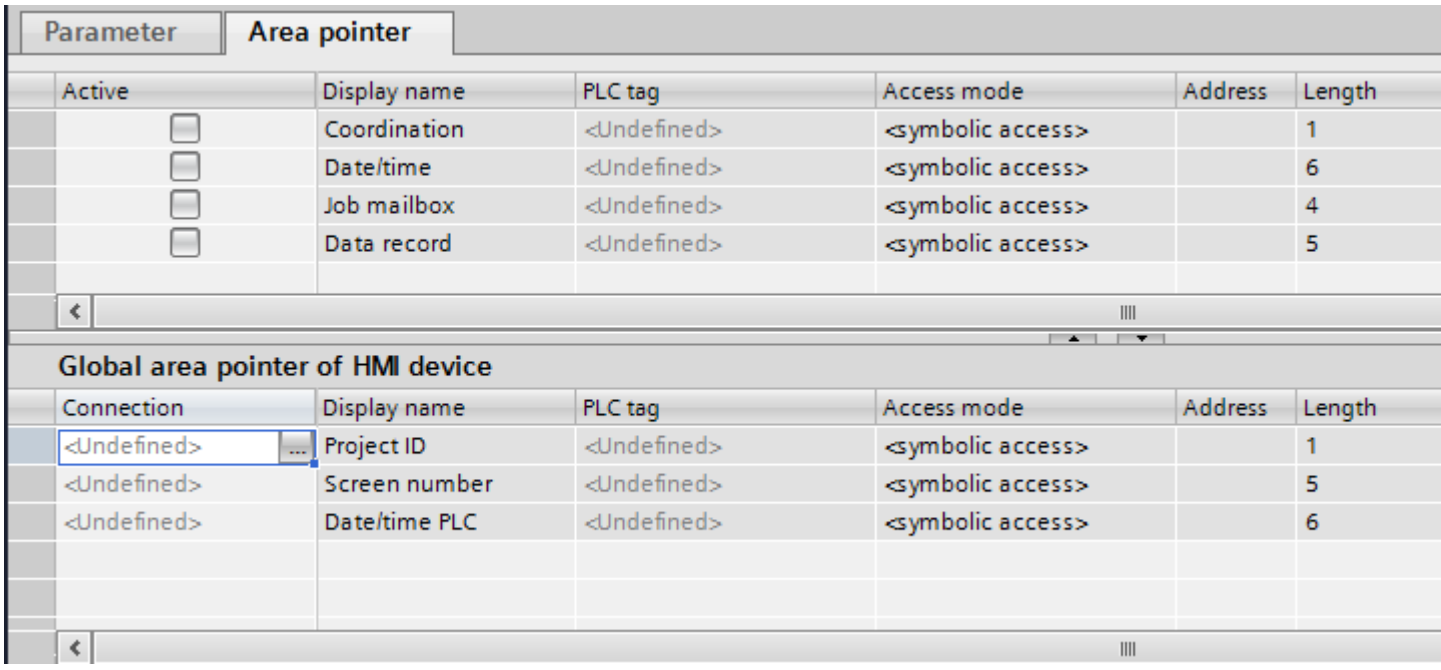
To configure the area pointers, open the "Connections" editor and open the "Area pointer" tab.



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 149)

Use of area pointers (Page 151)

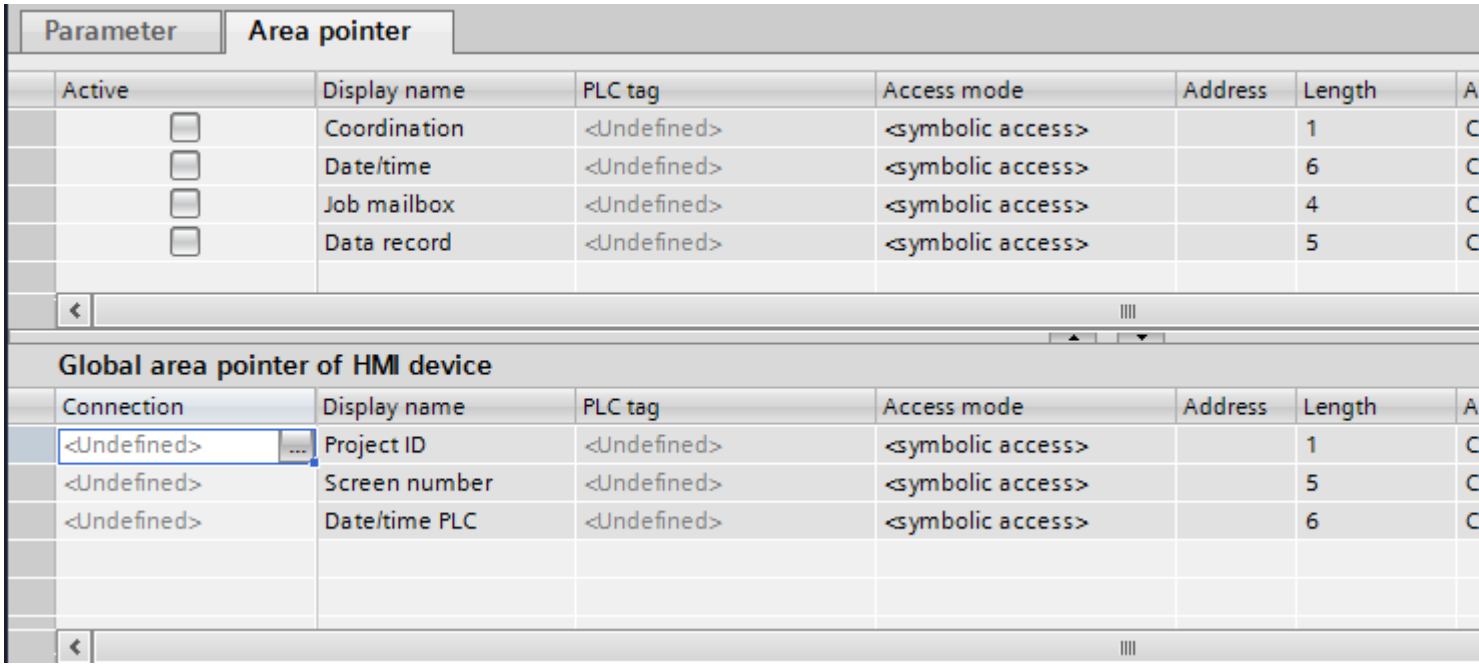
Accessing data areas (Page 153)

2.4.2.3 Use of area pointers

"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 149)

Area pointers for connections (Page 150)

Accessing data areas (Page 153)

2.4.2.4 Accessing data areas

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 149)

Area pointers for connections (Page 150)

Use of area pointers (Page 151)

2.4.2.5 Configuring area pointers

Configuration of area pointers

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 149)

Parameterizing a global data block (Page 154)

Configuring an area pointer for a connection (Page 156)

Parameterizing a global data block

Introduction

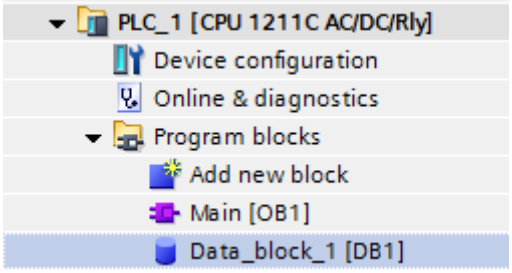
To access the data area in the PLC, a global data block for the area pointer must be parameterized in the PLC program.

Requirements

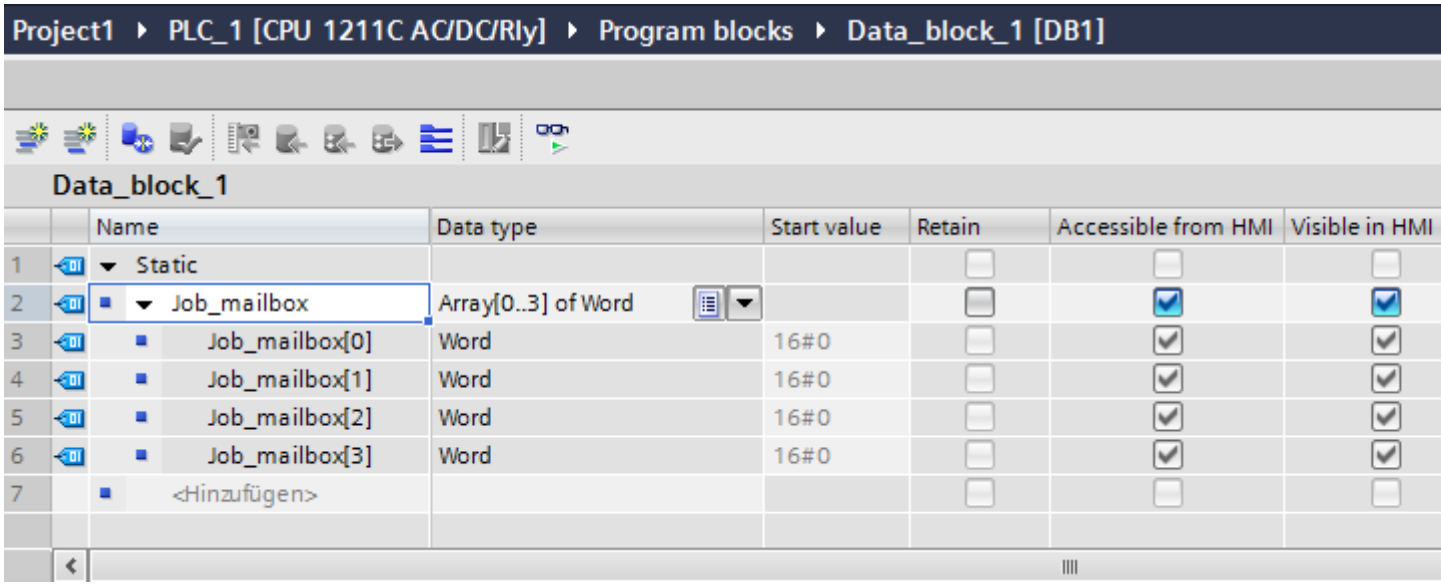
- A PLC is created in the project.
- A connection is configured between the PLC and the HMI device.
- The PLC program contains a global data block.

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 149)

Configuration of area pointers (Page 153)

Configuring an area pointer for a connection (Page 156)

Configuring an area pointer for a connection

Introduction

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

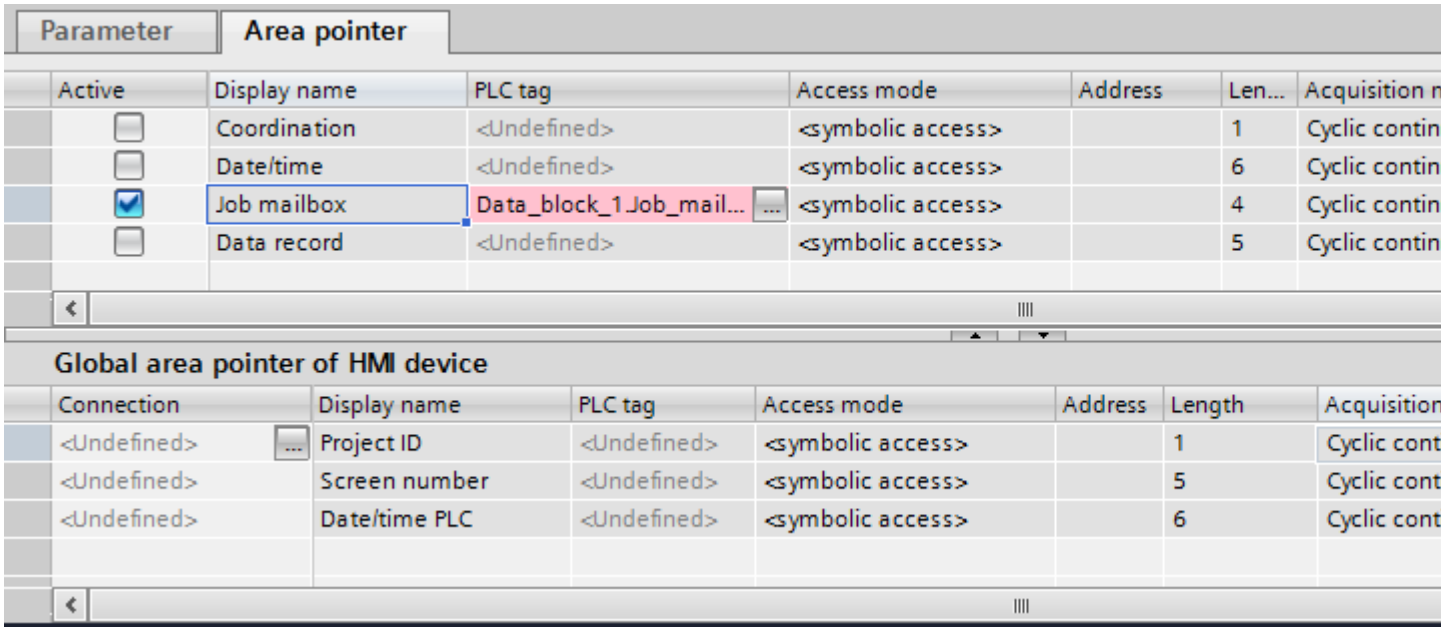
Requirements

- 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 149)
Configuration of area pointers (Page 153)
Parameterizing a global data block (Page 154)

2.5 Device dependency

2.5.1 Basic Panel

2.5.1.1 Communication drivers for Basic Panels

SIMATIC communication drivers

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 (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
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

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 (V14.0.1)

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
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 (V15.0)

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 161)

Other communication drivers

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

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
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 158)

2.5.1.2 Interfaces of the Basic Panels

Device dependency of the Basic Panels

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

Table 2-1 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 164)

2.5.1.3 Area pointers for 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
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 163)

2.5.1.4 Restrictions

Communication drivers

Basic Panels support several connections, but only one communication driver.

A combination of connections with SIMOTION and SIMATIC S7 is not possible.

2.5.2 Comfort Panel

2.5.2.1 Communication drivers for Comfort Panels

SIMATIC communication drivers

Device dependency of the Comfort 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 (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
KP400 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
KTP400 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
KTP400 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
KP700 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
TP700 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
TP700 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
KP900 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
TP900 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
TP900 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
KP1200 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
TP1200 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No

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
TP1200 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
KP1500 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
TP1500 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
TP1500 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
TP1900 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
TP1900 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
TP2200 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
TP2200 Comfort Portrait	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
KP400 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
KTP400 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
KTP400 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
KP700 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
TP700 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
TP700 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
KP900 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
TP900 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
TP900 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
KP1200 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
TP1200 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
TP1200 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
KP1500 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
TP1500 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
TP1500 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No

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
TP1900 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
TP1900 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
TP2200 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
TP2200 Comfort Portrait	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
KP400 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KTP400 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KTP400 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KP700 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP700 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP700 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KP900 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP900 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP900 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KP1200 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1200 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1200 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KP1500 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1500 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1500 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1900 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1900 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP2200 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP2200 Comfort Portrait	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
KP400 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KTP400 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KTP400 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KP700 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP700 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP700 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KP900 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP900 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP900 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KP1200 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1200 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1200 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KP1500 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1500 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1500 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1900 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1900 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP2200 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP2200 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Communication drivers for integrated communication (V14.0.1)

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
KP400 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KTP400 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KTP400 Comfort Portrait	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
KP700 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP700 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP700 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KP900 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP900 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP900 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KP1200 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1200 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1200 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KP1500 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1500 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1500 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1900 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1900 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP2200 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP2200 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Communication drivers for integrated communication (V15.0)

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
KP400 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KTP400 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KTP400 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KP700 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP700 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP700 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KP900 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP900 Comfort	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
TP900 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KP1200 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1200 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1200 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KP1500 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1500 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1500 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1900 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1900 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP2200 Comfort	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP2200 Comfort Portrait	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Communication drivers for non-integrated communication

HMI devices	SIMATIC S7-1500	SIMATIC S7-1200	SIMATIC S7-300/400	SIMATIC S7-200	SIMATIC LOGO!	SIMATIC HTTP protocol	SIMATIC ET 200 CPU	SIMATIC S7-1500 Software controller
KP400 Comfort	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KTP400 Comfort	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KTP400 Comfort Portrait	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KP700 Comfort	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP700 Comfort	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP700 Comfort Portrait	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KP900 Comfort	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP900 Comfort	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP900 Comfort Portrait	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KP1200 Comfort	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1200 Comfort	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

HMI devices	SIMATIC S7-1500	SIMATIC S7-1200	SIMATIC S7-300/400	SIMATIC S7-200	SIMATIC LOGO!	SIMATIC HTTP protocol	SIMATIC ET 200 CPU	SIMATIC S7-1500 Software controller
TP1200 Comfort Portrait	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KP1500 Comfort	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1500 Comfort	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1500 Comfort Portrait	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1900 Comfort	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP1900 Comfort Portrait	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP2200 Comfort	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TP2200 Comfort Portrait	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

See also

Other communication drivers (Page 171)

Other communication drivers

Device dependency of the Comfort Panels

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

Communication drivers

HMI devices	OPC [1]	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
KP400 Comfort	Yes	Yes	Yes	Yes [2] [3]	Yes	Yes [3]	Yes	Yes [3] [4]	Yes [3]
KTP400 Comfort	Yes	Yes	Yes	Yes [2] [3]	Yes	Yes [3]	Yes	Yes [3] [4]	Yes [3]
KTP400 Comfort Portrait	Yes	Yes	Yes	Yes [2] [3]	Yes	Yes [3]	Yes	Yes [3] [4]	Yes [3]
KP700 Comfort	Yes	Yes	Yes	Yes [2] [3]	Yes	Yes [3]	Yes	Yes [3] [4]	Yes [3]
TP700 Comfort	Yes	Yes	Yes	Yes [2] [3]	Yes	Yes [3]	Yes	Yes [3] [4]	Yes [3]
TP700 Comfort Portrait	Yes	Yes	Yes	Yes [2] [3]	Yes	Yes [3]	Yes	Yes [3] [4]	Yes [3]

HMI devices	OPC [1]	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
TP700 Comfort Outdoor	Yes	Yes	Yes	Yes [2] [3]	Yes	Yes [3]	Yes	Yes [3] [4]	Yes [3]
TP700 Comfort Outdoor Portrait	Yes	Yes	Yes	Yes [2] [3]	Yes	Yes [3]	Yes	Yes [3] [4]	Yes [3]
KP900 Comfort	Yes	Yes	Yes	Yes [2] [3]	Yes	Yes [3]	Yes	Yes [3] [4]	Yes [3]
TP900 Comfort	Yes	Yes	Yes	Yes [2] [3]	Yes	Yes [3]	Yes	Yes [3] [4]	Yes [3]
TP900 Comfort Portrait	Yes	Yes	Yes	Yes [2] [3]	Yes	Yes [3]	Yes	Yes [3] [4]	Yes [3]
KP1200 Comfort	Yes	Yes	Yes	Yes [2] [3]	Yes	Yes [3]	Yes	Yes [3] [4]	Yes [3]
TP1200 Comfort	Yes	Yes	Yes	Yes [2] [3]	Yes	Yes [3]	Yes	Yes [3] [4]	Yes [3]
TP1200 Comfort Portrait	Yes	Yes	Yes	Yes [2] [3]	Yes	Yes [3]	Yes	Yes [3] [4]	Yes [3]
KP1500 Comfort	Yes	Yes	Yes	Yes [2] [3]	Yes	Yes [3]	Yes	Yes [3] [4]	Yes [3]
TP1500 Comfort	Yes	Yes	Yes	Yes [2] [3]	Yes	Yes [3]	Yes	Yes [3] [4]	Yes [3]
TP1500 Comfort Portrait	Yes	Yes	Yes	Yes [2] [3]	Yes	Yes [3]	Yes	Yes [3] [4]	Yes [3]
TP1500 Comfort Outdoor	Yes	Yes	Yes	Yes [2] [3]	Yes	Yes [3]	Yes	Yes [3] [4]	Yes [3]
TP1500 Comfort Outdoor Portrait	Yes	Yes	Yes	Yes [2] [3]	Yes	Yes [3]	Yes	Yes [3] [4]	Yes [3]
TP1900 Comfort	Yes	Yes	Yes	Yes [2] [3]	Yes	Yes [3]	Yes	Yes [3] [4]	Yes [3]
TP1900 Comfort Portrait	Yes	Yes	Yes	Yes [2] [3]	Yes	Yes [3]	Yes	Yes [3] [4]	Yes [3]
TP2200 Comfort	Yes	Yes	Yes	Yes [2] [3]	Yes	Yes [3]	Yes	Yes [3] [4]	Yes [3]
TP2200 Comfort Portrait	Yes	Yes	Yes	Yes [2] [3]	Yes	Yes [3]	Yes	Yes [3] [4]	Yes [3]

[1] OPC UA client (DA only)
OPC UA server (DA only)

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

3) "PROFINET IO Enabled" must be disabled.

4) Only approved with RS 422-RS232 converter
Order number: 6AV6 671-8XE00-0AX0

See also

SIMATIC communication drivers (Page 165)

2.5.2.2 Interfaces of Comfort Panels

Device dependency of Comfort Panels

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

Table 2-2 Comfort Panels

	KP400 Comfort	KTP400 Comfort	KP700 Comfort	TP700 Comfort	KP900 Comfort	TP900 Comfort	KP1200 Comfort	TP1200 Comfort	KP1500 Comfort	TP1500 Comfort	TP1900 Comfort	TP2200 Comfort
SIMATIC S7 - PPI 1)	IF1B	IF1B	IF1B	IF1B	IF1B	IF1B	IF1B	IF1B	IF1B	IF1B	IF1B	IF1B
SIMATIC S7 - MPI	IF1B	IF1B	IF1B	IF1B	IF1B	IF1B	IF1B	IF1B	IF1B	IF1B	IF1B	IF1B
SIMATIC S7 - PROFIBUS	IF1B	IF1B	IF1B	IF1B	IF1B	IF1B	IF1B	IF1B	IF1B	IF1B	IF1B	IF1B
SIMATIC S7 - PROFINET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET
SIMATIC HMI HTTP protocol	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET
OPC	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET
Allen-Bradley EtherNet/IP	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET
Allen-Bradley DF1	IF1B 1)3)4)	IF1B 1)3)4)	IF1B 1)3)4)	IF1B 1)3)4)	IF1B 1)3)4)	IF1B 1)3)4)	IF1B 1)3)4)	IF1B 1)3)4)	IF1B 1)3)4)	IF1B 1)3)4)	IF1B 1)3)4)	IF1B 1)3)4)
Mitsubishi MC TCP/IP	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET
Mitsubishi FX	IF1B 3)4)	IF1B 3)4)	IF1B 3)4)	IF1B 3)4)	IF1B 3)4)	IF1B 3)4)	IF1B 3)4)	IF1B 3)4)	IF1B 3)4)	IF1B 3)4)	IF1B 3)4)	IF1B 3)4)

	KP400 Comfort	KTP400 Comfort	KP700 Comfort	TP700 Comfort	KP900 Comfort	TP900 Comfort	KP1200 Comfort	TP1200 Comfort	KP1500 Comfort	TP1500 Comfort	TP1900 Comfort	TP2200 Comfort
Modicon Modbus TCP/	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET	ETHERNET
Modicon Modbus RTU	IF1B 2)3)4)	IF1B 2)3)4)	IF1B 2)3)4)	IF1B 2)3)4)	IF1B 2)3)4)	IF1B 2)3)4)	IF1B 2)3)4)	IF1B 2)3)4)	IF1B 2)3)4)	IF1B 2)3)4)	IF1B 2)3)4)	IF1B 2)3)4)
Omron Host Link	IF1B 3)4)	IF1B 3)4)	IF1B 3)4)	IF1B 3)4)	IF1B 3)4)	IF1B 3)4)	IF1B 3)4)	IF1B 3)4)	IF1B 3)4)	IF1B 3)4)	IF1B 3)4)	IF1B 3)4)

1) Direct communication with PLC5 or KF2 module, otherwise only with RS422-RS232 converter.
   Order number: 6AV6 671-8XE00-0AX0
2) only with RS422-RS232 converter (option)
   Order number: 6AV6 671-8XE00-0AX0
3) For serial communication, deselect "Remote Control" of "Channel 1" in the "File > Transfer >Options" menu.
4) "PROFINET IO Enabled" must be disabled.

See also

Area pointers for Comfort Panel (Page 174)

2.5.2.3 Area pointers for Comfort Panel

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

	KP400 Comfort	KTP400 Comfort	KP700 Comfort	TP700 Comfort	KP900 Comfort	TP900 Comfort	KP1200 Comfort	TP1200 Comfort	KP1500 Comfort	TP1500 Comfort	TP1900 Comfort	TP2200 Comfort
Screen number	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Data record	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Date/time	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Date/time PLC	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Coordination	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Project ID	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Job mailbox	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

See also

Interfaces of Comfort Panels (Page 173)

2.5.3 Mobile Panel

2.5.3.1 Communication drivers for Mobile Panels

SIMATIC communication drivers

Device dependency of the Mobile 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 (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
KTP700 Mobile	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KTP700F Mobile	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KTP900 Mobile	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KTP900F Mobile	Yes	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
KTP400F Mobile	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Communication drivers for integrated communication (V14.0.1)

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
KTP400F Mobile	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Communication drivers for integrated communication (V15.0)

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
KTP400F Mobile	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Communication drivers for non-integrated communication

HMI devices	SIMATIC S7-1500	SIMATIC S7-1200	SIMATIC S7-300/400	SIMATIC S7-200	SIMATIC LOGO!	SIMATIC HTTP protocol	SIMATIC ET 200 CPU	SIMATIC S7-1500 software controller
Mobile Panel 177 6" DP	Yes [2]	Yes	Yes	Yes	No	No	Yes	Yes
Mobile Panel 177 6" PN	Yes [2]	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Mobile Panel 277 8"	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Mobile Panel 277 8" IWLAN V2	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Mobile Panel 277F 8" IWLAN V2	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Mobile Panel 277F 8" IWLAN (RFID tag)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Mobile Panel 277 10"	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KTP400F Mobile	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KTP700 Mobile	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KTP700F Mobile	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KTP900 Mobile	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
KTP900F Mobile	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

[1] "PROFINET IO Enabled" must be disabled.

[2] You must activate "Access via PUT/GET communication" for communication with the SIMATIC S7-1500 PLC.

Activate PUT/GET communication in the PLC properties at:

"General > Security > Enable PUT/GET communication with remote partner (PLC, HMI, OPC, etc.)"

See also

Other communication drivers (Page 178)

Other communication drivers

Device dependency of the Mobile Panels

The following table shows which communication drivers you can configure with the various Mobile 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
KTP400F Mobile	Yes [1]	Yes	Yes	No	Yes	No	Yes	No	No
KTP700 Mobile	Yes [1]	Yes	Yes	No	Yes	No	Yes	No	No
KTP700F Mobile	Yes [1]	Yes	Yes	No	Yes	No	Yes	No	No
KTP900 Mobile	Yes [1]	Yes	Yes	No	Yes	No	Yes	No	No
KTP900F Mobile	Yes [1]	Yes	Yes	No	Yes	No	Yes	No	No

[1] OPC UA client (DA only)
OPC UA server (DA only)

See also

SIMATIC communication drivers (Page 175)

2.5.3.2 Interfaces of Mobile Panels

Device dependency of the Mobile Panels

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

	KTP400F Mobile	KTP700 Mobile KTP700F Mobile	KTP900 Mobile KTP900F Mobile
SIMATIC S7 - PPI [1]	—	—	—
SIMATIC S7 - MPI	—	—	—
SIMATIC S7 - PROFIBUS	—	—	—
SIMATIC S7 - PROFINET	ETHERNET	ETHERNET	ETHERNET
SIMATIC HMI HTTP protocol	ETHERNET	ETHERNET	ETHERNET

	KTP400F Mobile	KTP700 Mobile KTP700F Mobile	KTP900 Mobile KTP900F Mobile
OPC	ETHERNET	ETHERNET	ETHERNET
Allen-Bradley DF1	—	—	—
Allen-Bradley EtherNet/IP	ETHERNET	ETHERNET	ETHERNET
Mitsubishi MC TCP/IP	ETHERNET	ETHERNET	ETHERNET
Mitsubishi FX	—	—	—
Modicon Modbus RTU	—	—	—
Modicon Modbus TCP/IP	ETHERNET	ETHERNET	ETHERNET
Omron Host Link	—	—	—

See also

Area pointers for Mobile Panel (Page 179)

2.5.3.3 Area pointers for Mobile Panel

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

	KTP400F Mobile	KTP700 Mobile KTP700F Mobile	KTP900 Mobile KTP900F Mobile
Screen number	Yes	Yes	Yes
Data record	Yes	Yes	Yes
Date/time	Yes	Yes	Yes
Date/time PLC	Yes	Yes	Yes
Coordination	Yes	Yes	Yes
Project ID	Yes	Yes	Yes
Job mailbox	Yes	Yes	Yes

See also

Interfaces of Mobile Panels (Page 178)

2.5.4 PC systems with WinCC Runtime

2.5.4.1 Communication drivers for WinCC Runtime

SIMATIC communication drivers

Device dependency

The following table shows which communication drivers you can configure with the various Runtimes on a PC.

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
WinCC RT Professional	No	Yes	Yes	Yes	No	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
WinCC RT Advanced	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
WinCC RT Professional	No	Yes	Yes	Yes	Yes	Yes	Yes	No	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
WinCC RT Advanced	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
WinCC RT Professional	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
WinCC RT Advanced	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
WinCC RT Professional	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No

Communication drivers for non-integrated communication

HMI devices	SIMATIC S7-1500	SIMATIC S7-1200	SIMATIC S7-300/400	SIMATIC S7-200	SIMATIC LOGO!	SIMATIC HTTP protocol	SIMATIC ET 200 CPU	SIMATIC S7-1500 software controller
WinCC RT Advanced	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
WinCC RT Professional	Yes	Yes	Yes	No	No	No	Yes	No

1) OPC DA server
OPC DA client
OPC XML DA client
OPC UA server (DA only)
OPC UA client (DA only)

2) OPC DA server
OPC DA client
OPC XML DA server
OPC XML DA client
OPC UA server (DA only)

See also

Other communication drivers (Page 182)

Other communication drivers

Device dependency

The following table shows which communication drivers you can configure with the various Runtimes on a PC.

Communication drivers

HMI devices	OPC	Allen-Bradley EtherNet/IP	Allen-Bradley DF1	Mitsubishi MC TCP/IP	Mitsubishi FX	Modicon Modbus TCP/IP	Modicon Modbus RTU	Omron Host Link
WinCC RT Advanced	Yes[1)]	Yes	Yes	Yes	Yes	Yes	Yes	Yes
WinCC RT Professional	Yes[2)]	Yes	No	Yes	No	Yes	No	No

[1] OPC DA server
OPC DA client
OPC XML DA client
OPC UA server (DA only)
OPC UA client (DA only)

[2] OPC DA server
OPC DA client
OPC XML DA server
OPC XML DA client
OPC UA server (DA only)

See also

SIMATIC communication drivers (Page 180)

2.5.4.2 Interfaces for PC systems

Device dependency

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

	WinCC RT Advanced	WinCC RT Professional
SIMATIC S7 - PPI	MPI/DP [2]	--
SIMATIC S7 - MPI	MPI/DP [2]	MPI/DP [2]
SIMATIC S7 - PROFIBUS	MPI/DP [2]	MPI/DP [2]
SIMATIC S7 - PROFINET	ETHERNET	ETHERNET
SIMATIC HMI HTTP protocol	ETHERNET	ETHERNET
OPC	ETHERNET	ETHERNET
OPC UA	ETHERNET	--
Allen-Bradley EtherNet/IP	ETHERNET	ETHERNET
Allen-Bradley DF1	COM1 to COM4 [1]	--
Mitsubishi MC TCPI/IP	ETHERNET	ETHERNET
Mitsubishi FX	COM1 to COM4 [1]	--
Modicon Modbus TCP/IP	ETHERNET	ETHERNET
Modicon Modbus RTU	COM1 to COM4 [1]	--
Omron Host Link	COM1 to COM4 [1]	--

[1] COM2 is blocked for Panel PC 477

[2] for SIMATIC PC, via integrated interface; for standard PC, e.g., via SIMATIC NET CP 5611 A2

2.5.5 Parallel communication

Parallel communication of communication drivers

The following table shows an overview of which communication drivers you can use simultaneously on one HMI device.

Note

Note the parallel communication has not been released 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.

The following table shows the special features of the parallel communication for WinCC Runtime Advanced. For WinCC Runtime Professional there are no restrictions regarding parallel communication between different 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
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

	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
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	Yes	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.

2.6 Communicating with SIMATIC S7 1500

2.6.1 Communication with SIMATIC S7-1500

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.

Overwriting data from the HMI system or the web server

During operation, data from the HMI system or the web server can be overwritten in the S7-1500. The two processes (PLC system and HMI system) run independently of each other in parallel.

If both systems have write access to the same tags, the system can overwrite the data.

Additional information can be found on the Internet:

- FAQ with the entry ID 109478253: "How can it happen that data of the HMI system or of the Web server is overwritten in the S7-1500?"

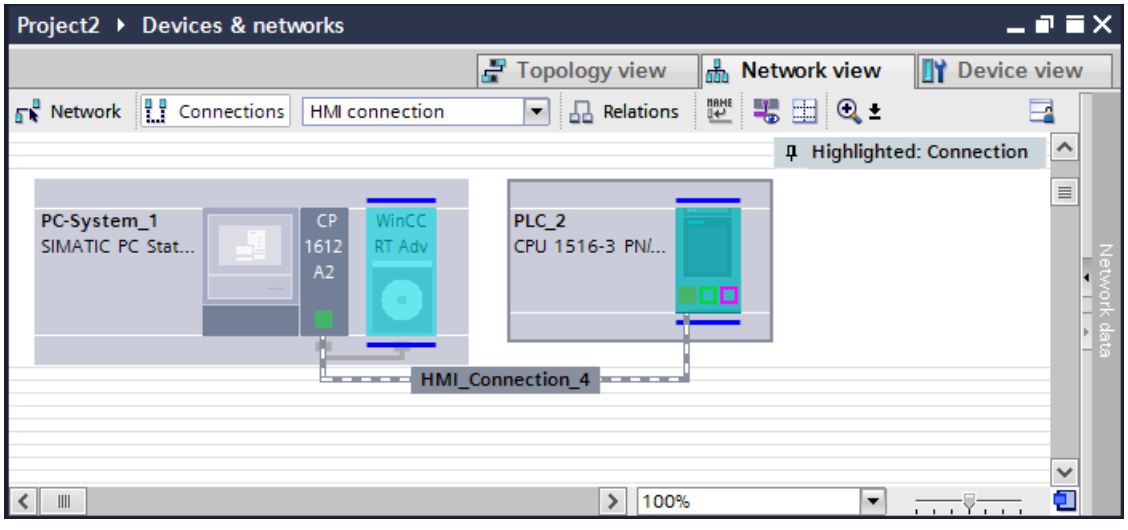
2.6.2 Communication via PROFINET

2.6.2.1 Configuring an HMI connection

Communication via PROFINET

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 185)

Configuring an HMI connection via PROFINET (Page 187)

Configuring an HMI connection via PROFINET

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.

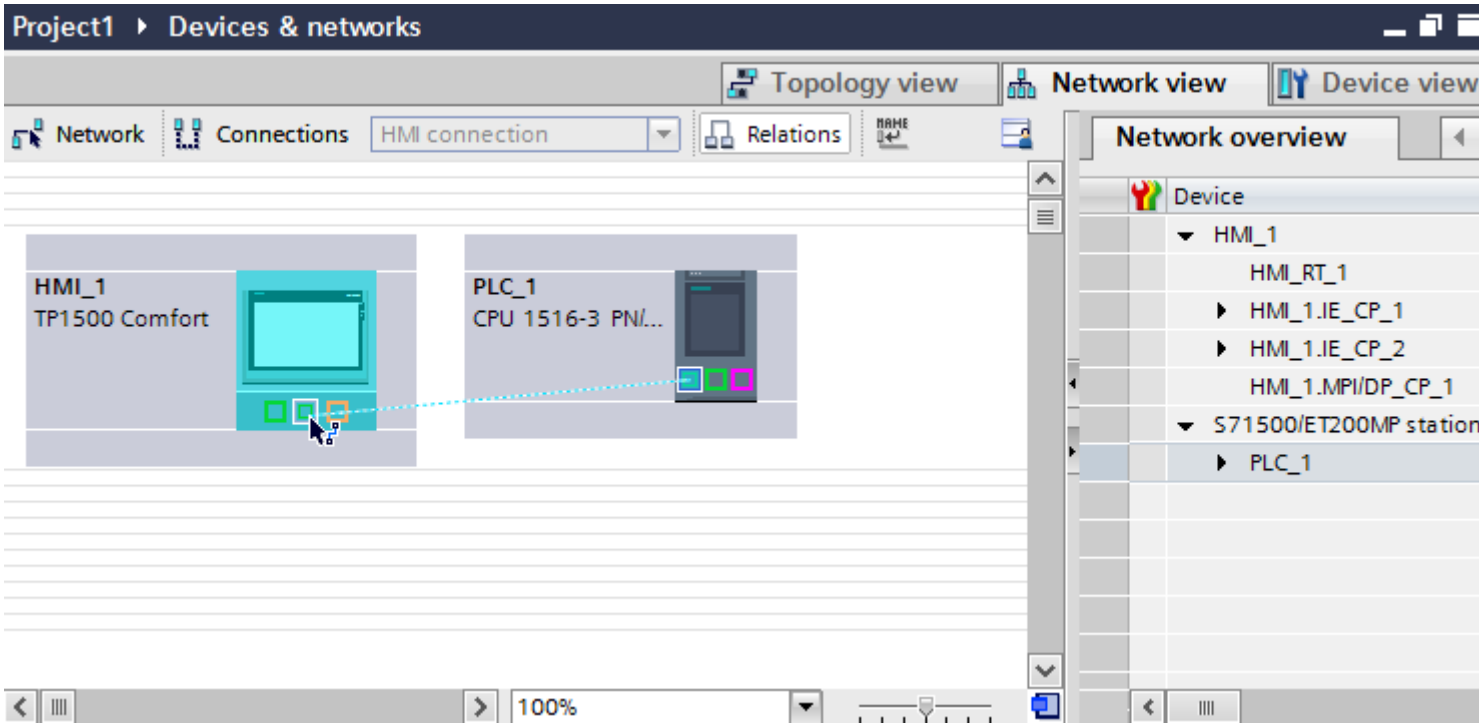
Requirements

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 197)".

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 185)
Communication via PROFINET (Page 186)
PROFINET parameters (Page 197)

2.6.2.2 Configuring an HMI connection

Communication via PROFINET

Communication via PROFINET

This chapter describes communication over PROFINET between a WinCC Runtime and the SIMATIC S7-1500 controller.

You can use the following WinCC Runtimes as an HMI device:

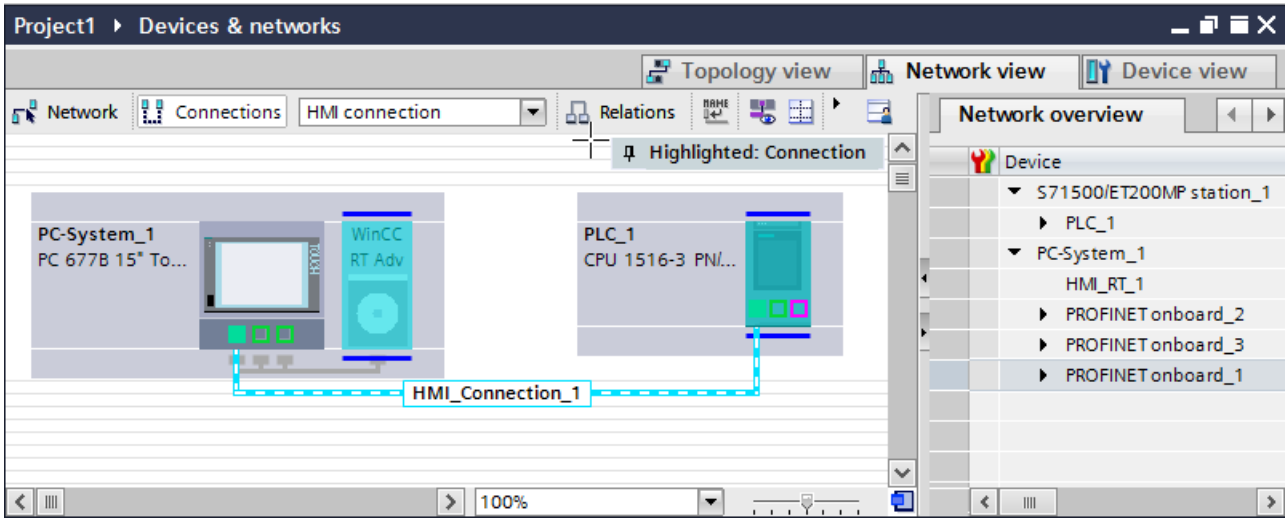
- WinCC RT Advanced
- WinCC RT Professional

WinCC Runtime as HMI device

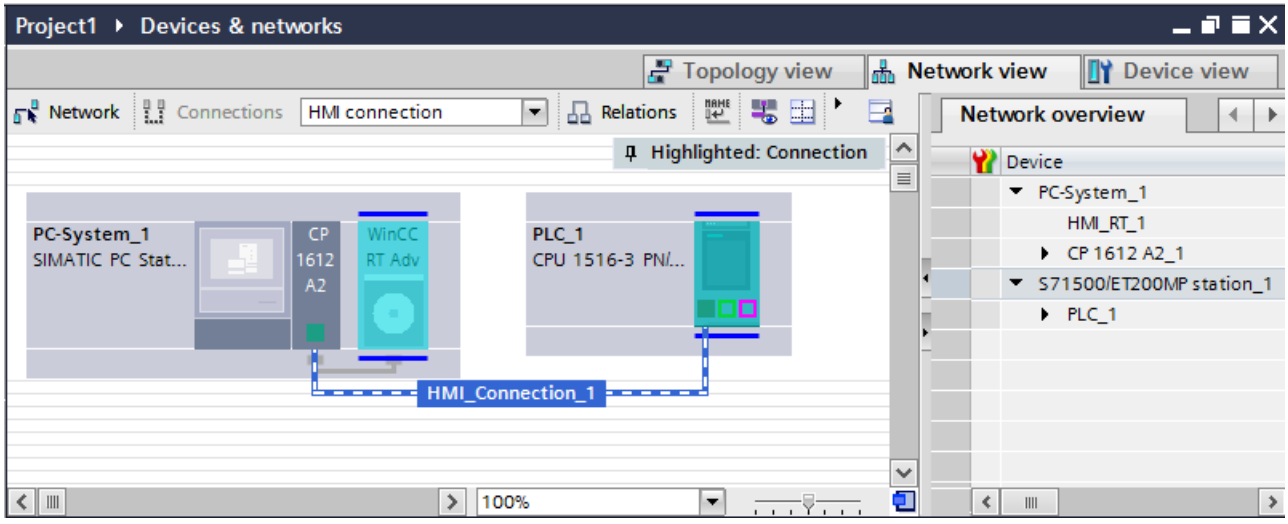
Configure the HMI connections between a WinCC Runtime and SIMATIC S7-1500 in the "Devices & Networks" editor.

If you want to use a WinCC Runtime as an HMI device, you have the following options:

1. You create a SIMATIC PC and configure a WinCC Runtime on this SIMATIC PC.
   In this way you use the SIMATIC PC with a WinCC Runtime as an HMI device.



2. You create a WinCC Runtime and configure a communication processor to the Runtime.
   In this way you use your configuration PC with a WinCC Runtime as an HMI device.



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 185)

Configuring an HMI connection via PROFINET with a PC (Page 191)

Configuring an HMI connection via PROFINET with a SIMATIC PC (Page 194)

Configuring an HMI connection via PROFINET with a PC

Introduction

You configure an HMI connection between an HMI device 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.

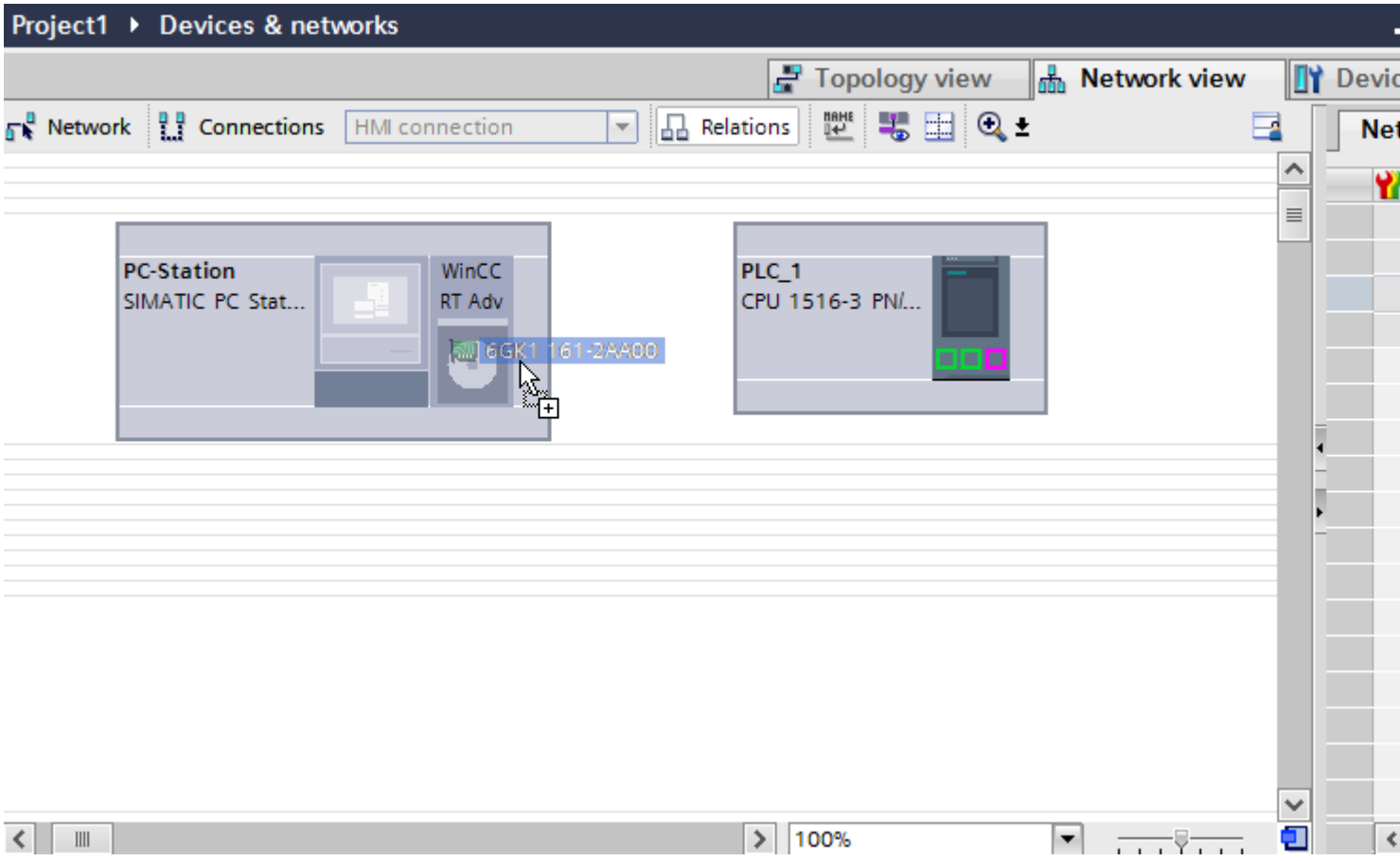
Requirements

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

- SIMATIC S7 1500 with PROFINET interface
- PC station with WinCC RT Advanced or WinCC RT Professional

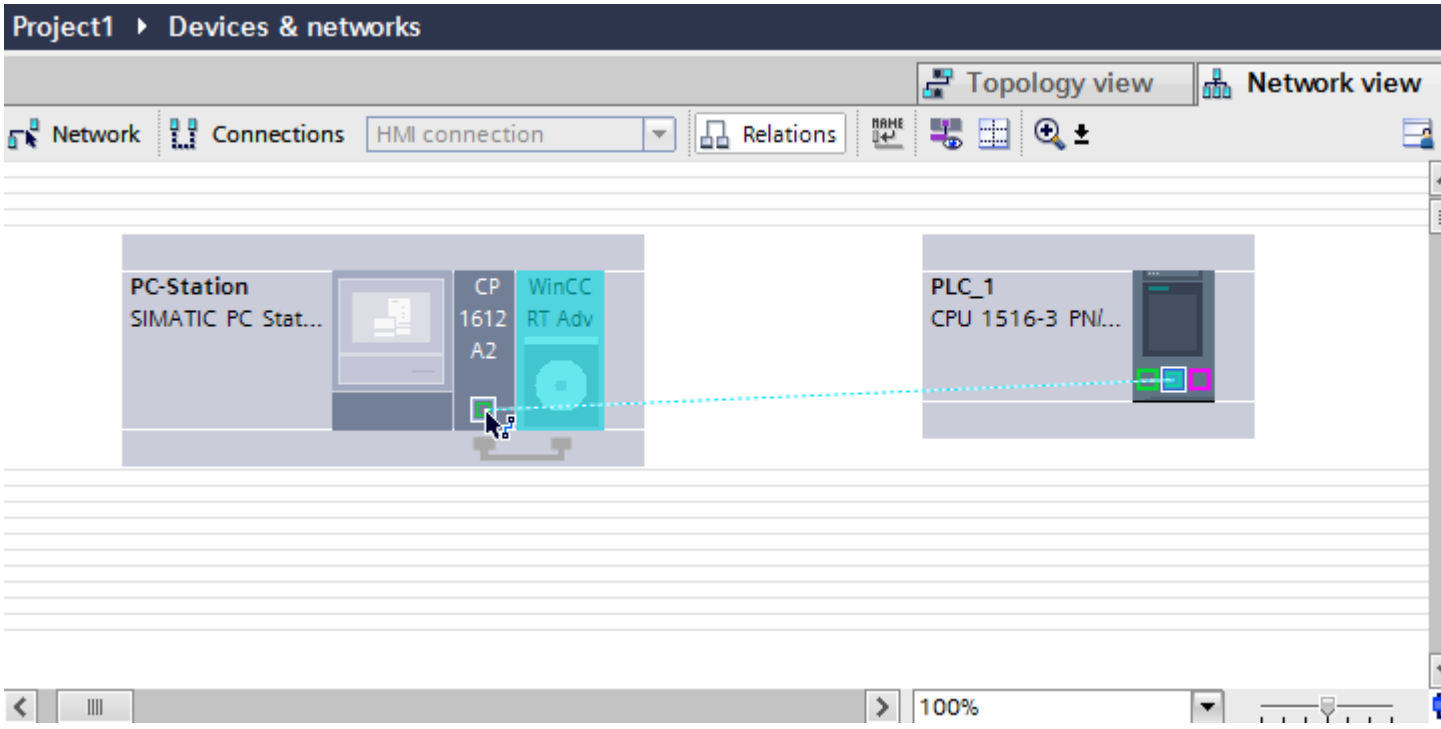
Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed in the network view.
2. Use a drag-and-drop operation to move a PROFINET-capable communication processor from the hardware catalog to the WinCC Runtime.



3. Click the "Connections" button and select "HMI connection" for the connection type.
   The devices available for connection are highlighted in color.

4. Click the PROFINET interface of the PLC and use a drag-and-drop operation to draw a connection to the PROFINET interface of the communication processor.



5. Click the connecting line.
   The connection is displayed graphically in the Inspector window.
6. Click "Highlight HMI connection" and select the HMI connection.
   The connection is displayed graphically in the Inspector window.
7. 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 197)".

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 185)

Communication via PROFINET (Page 189)

Configuring an HMI connection via PROFINET with a SIMATIC PC (Page 194)
PROFINET parameters (Page 197)

Configuring an HMI connection via PROFINET with a SIMATIC PC

Introduction

You configure an HMI connection between an HMI device and 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.

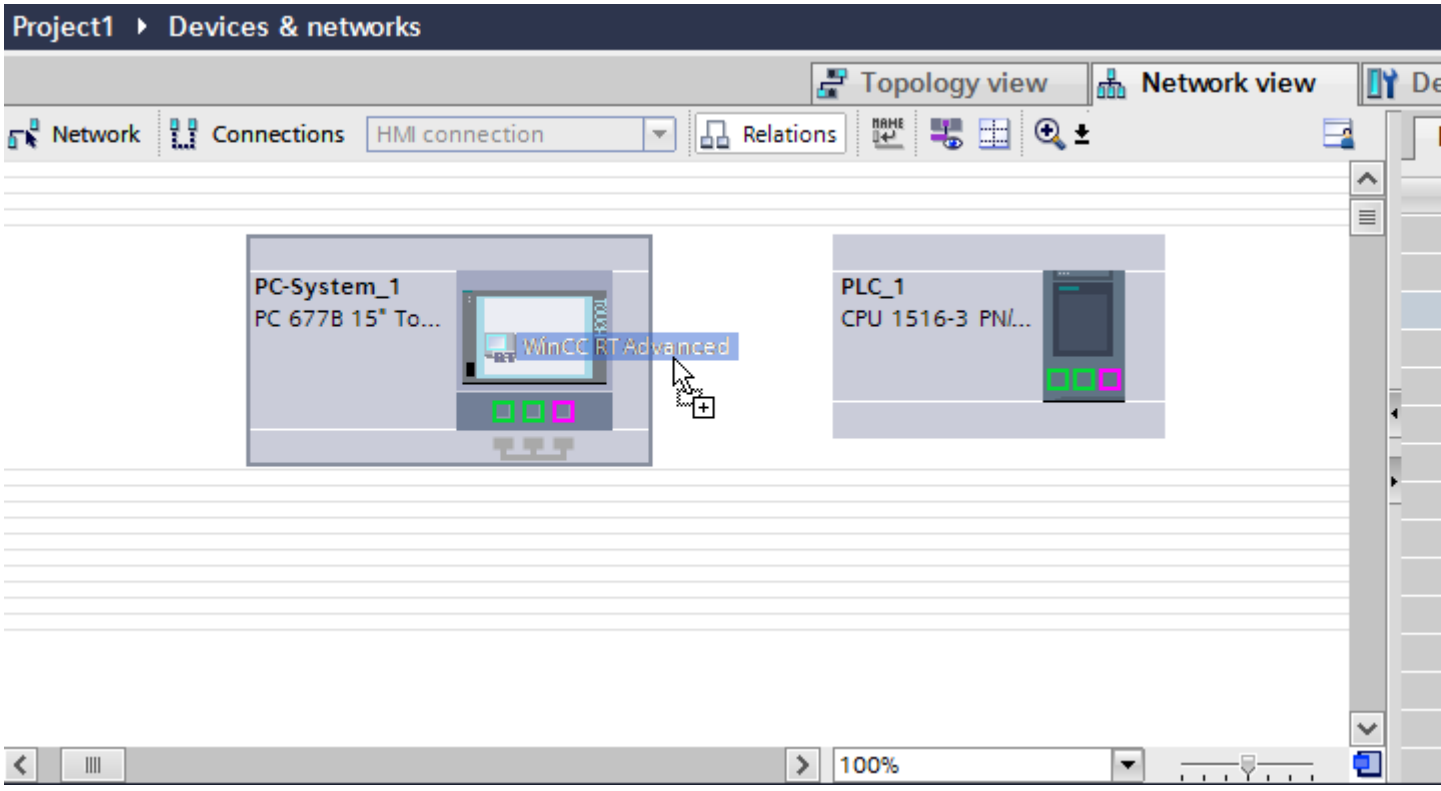
Requirements

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

- SIMATIC S7 1500 with PROFINET interface
- SIMATIC PC with PROFINET interface

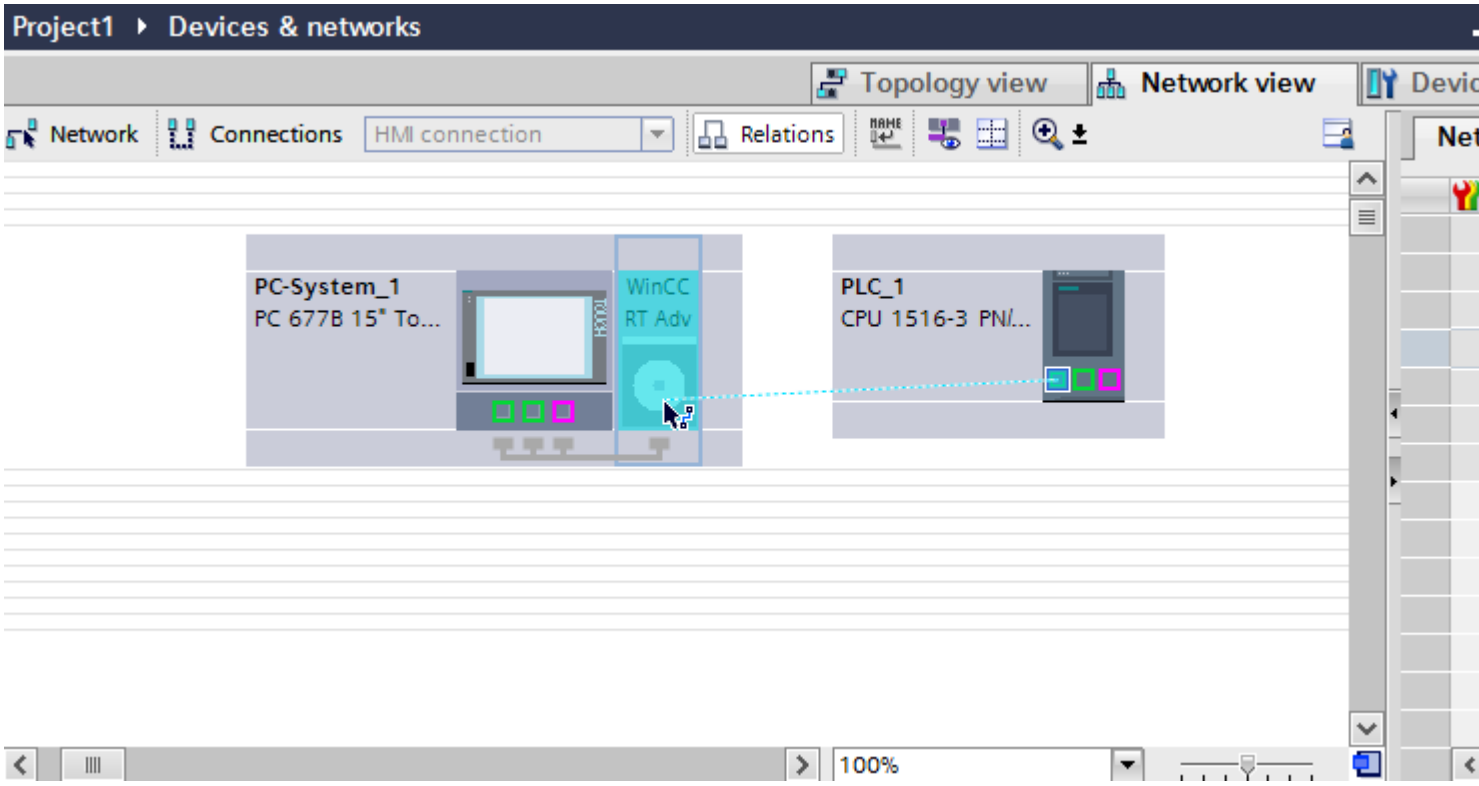
Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed in the network view.
2. Use a drag-and-drop operation to move a WinCC Runtime from the hardware catalog to the PC.



3. Click the "Connections" button and select "HMI connection" for the connection type.
   The devices available for connection are highlighted in color.

4. 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 PC.



5. Click the connecting line.
6. Click "Highlight HMI connection" and select the HMI connection.
   The connection is displayed graphically in the Inspector window.
7. 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 197)".

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 185)

Communication via PROFINET (Page 189)

Configuring an HMI connection via PROFINET with a PC (Page 191)

PROFINET parameters (Page 197)

2.6.2.3 PROFINET parameters

PROFINET parameters for the HMI connection

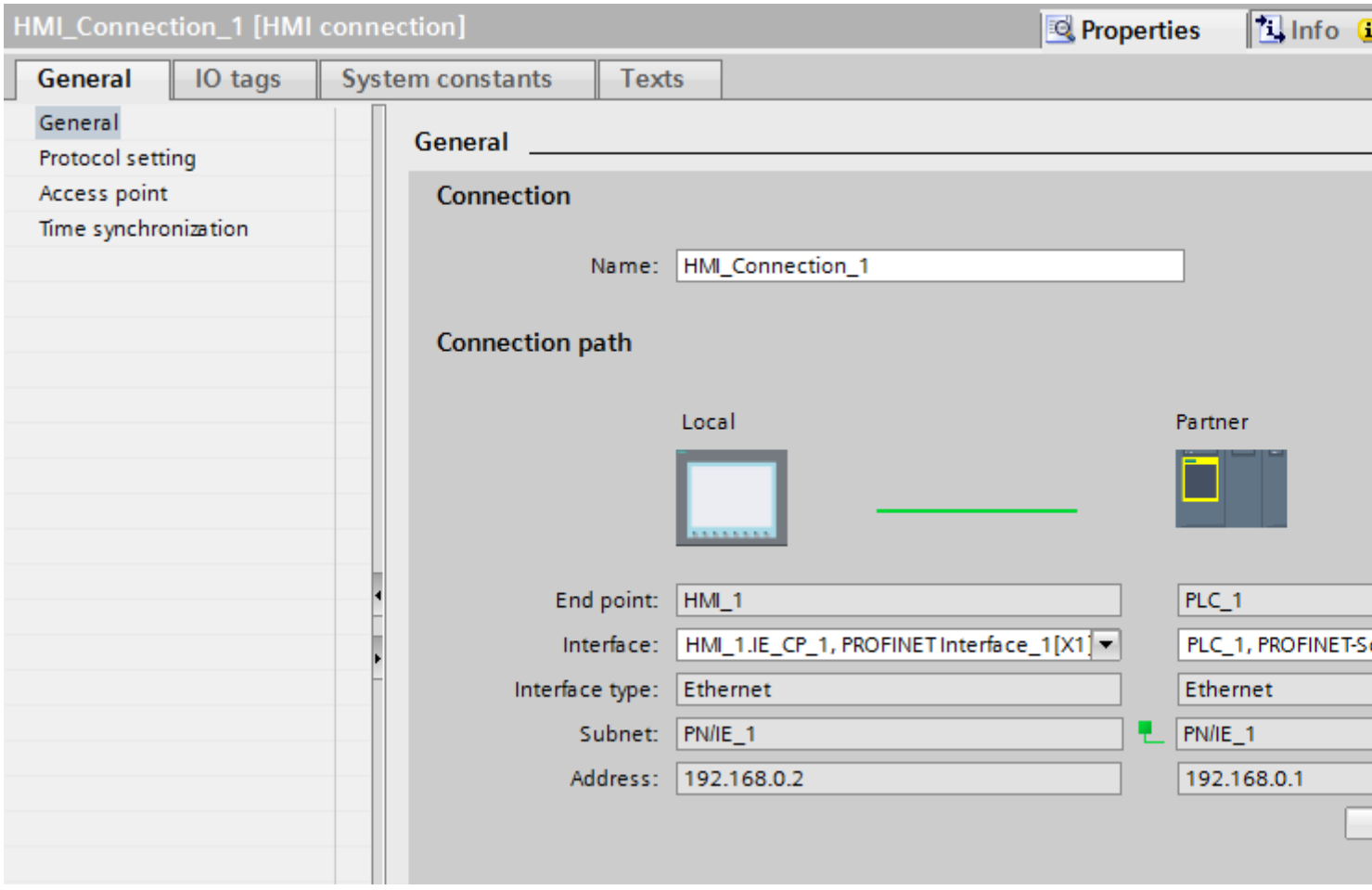
PROFINET parameters for the HMI connection

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 185)

PROFINET parameters for the HMI device (Page 199)

PROFINET parameters for the PLC (Page 201)

Configuring Industrial Ethernet (Page 203)

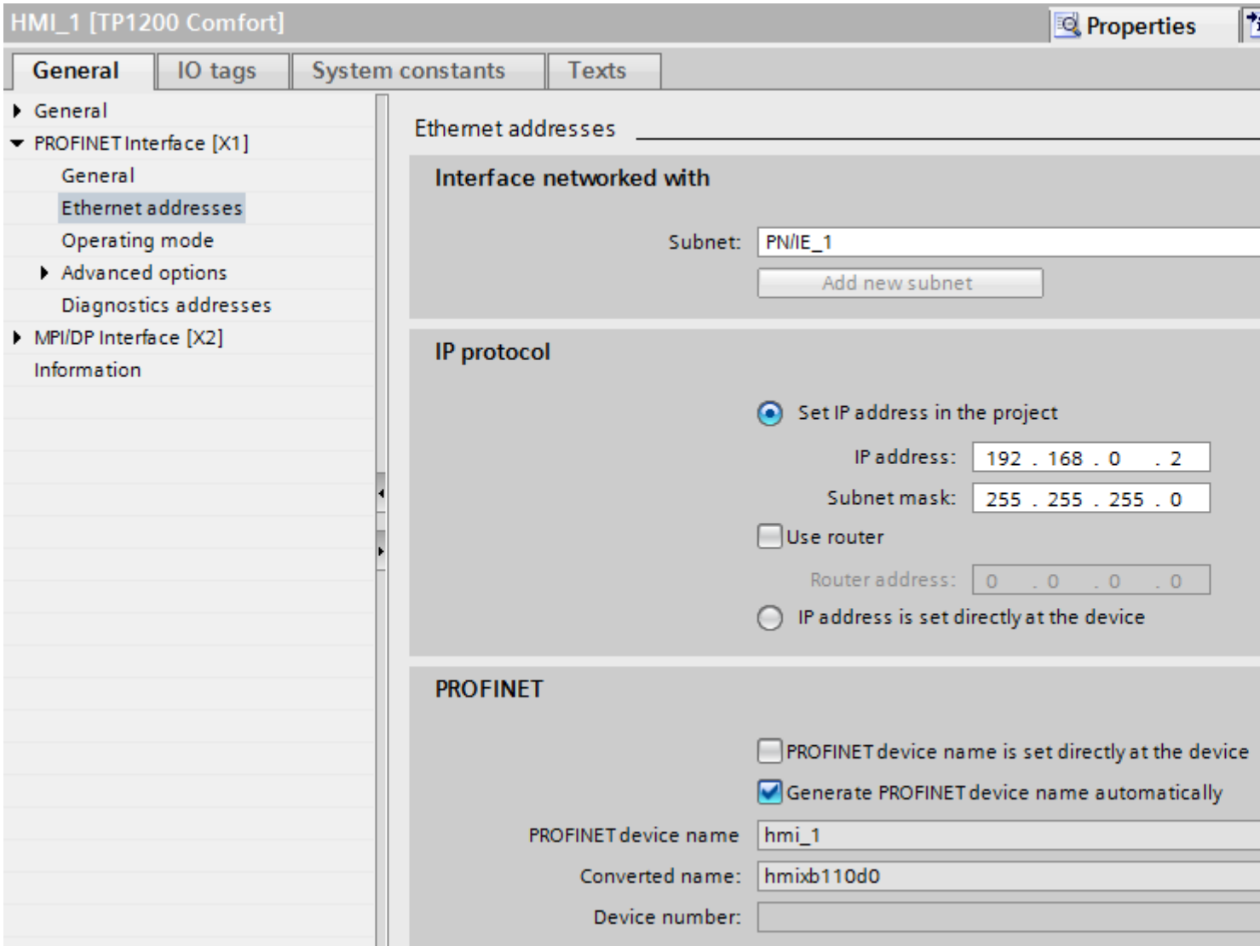
PROFINET parameters for the HMI device

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:
  - Mobile Panel 177 PN
  - Mobile Panel 177 DP

- "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.
- "Permit setting of the IP address directly on the device"
  If the "Permit setting of the IP address on the device" function is enabled, 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 185)

PROFINET parameters for the HMI connection (Page 197)

PROFINET parameters for the PLC (Page 201)

Configuring Industrial Ethernet (Page 203)

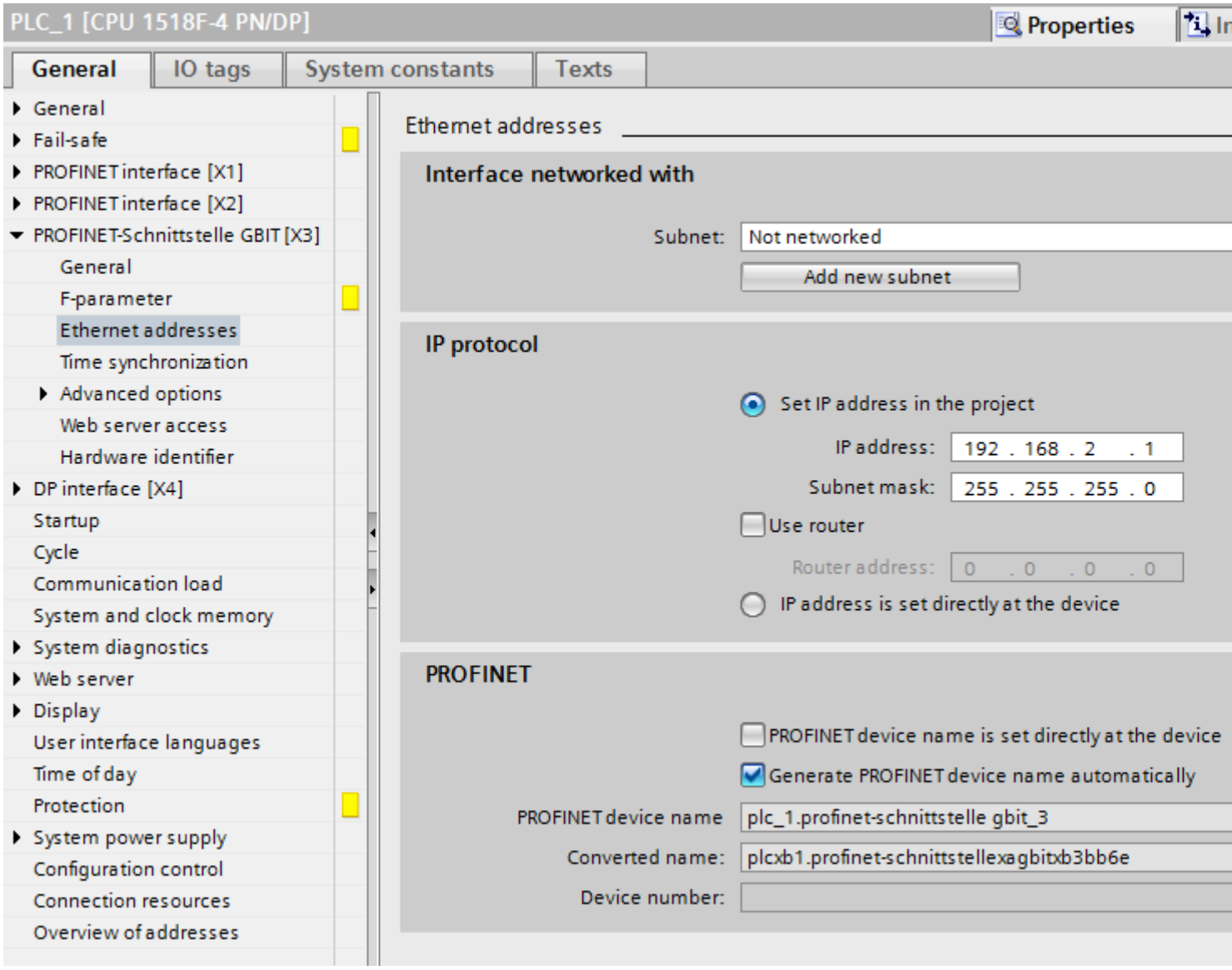
PROFINET parameters for the PLC

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 185)

PROFINET parameters for the HMI connection (Page 197)

PROFINET parameters for the HMI device (Page 199)

Configuring Industrial Ethernet (Page 203)

Configuring Industrial Ethernet

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 with SIMATIC S7-1500 (Page 185)

PROFINET parameters for the HMI connection (Page 197)

PROFINET parameters for the HMI device (Page 199)

PROFINET parameters for the PLC (Page 201)

Communication security

Security levels

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 206)

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 206)

Communication with SIMATIC S7-1500 (Page 185)

PROFINET parameters for the HMI connection (Page 197)

Access password for the HMI connection (Page 210)

Setting options for the protection

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 185)

PROFINET parameters for the HMI connection (Page 197)

Security levels (Page 205)

Access password for the HMI connection (Page 210)

Access password for the HMI connection

Introduction

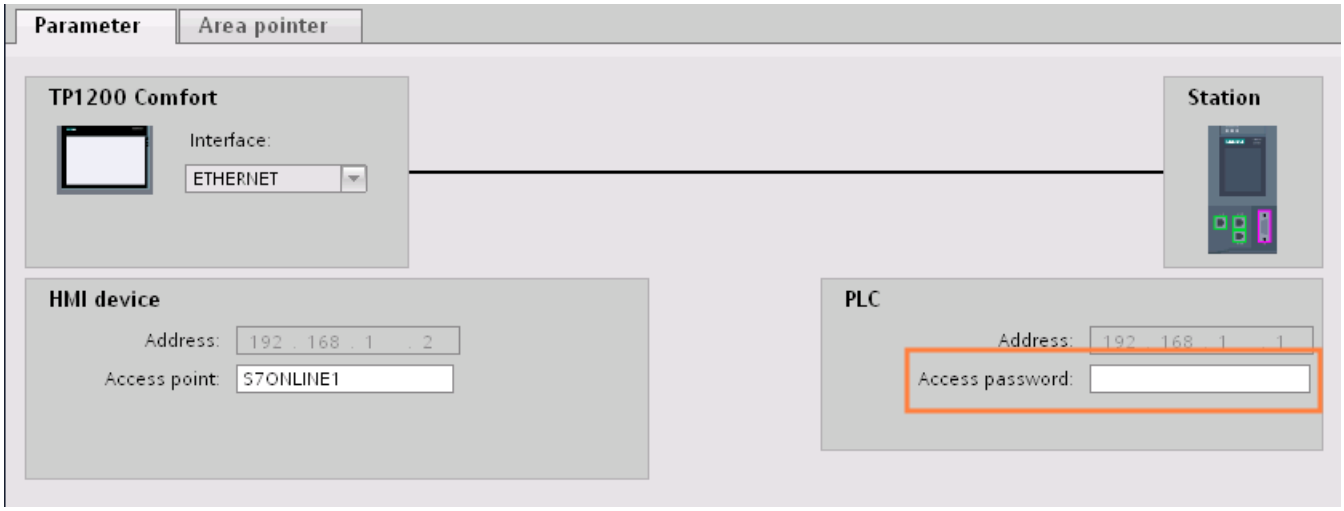
Communication with a PLC with the"Complete protection" protection level is protected by a password. The password is stored in the properties of the PLC.

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

Communication with SIMATIC S7-1500 (Page 185)

PROFINET parameters for the HMI connection (Page 197)

Security levels (Page 205)

Setting options for the protection (Page 206)

2.6.2.4 Setting port options

Setting the port options

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

Communication with SIMATIC S7-1500 (Page 185)

Wiring rules for disabled autonegotiation (Page 212)

Boundaries at the port (Page 213)

Wiring rules for disabled autonegotiation

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:

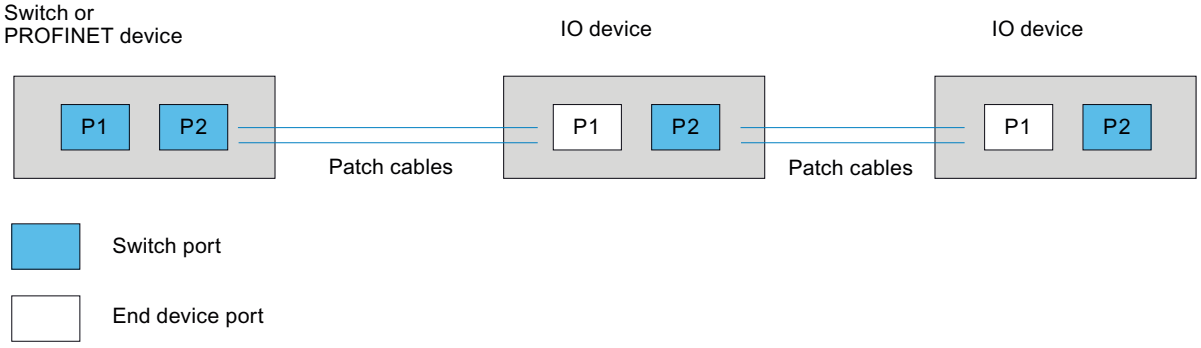
Type of port	PROFINET devices	Note
Switch port with crossed pin assignment	For IO devices: Port 2 For S7 CPUs with 2 ports: Ports 1 and 2	Crossed pin assignment means that the pin assignment for the ports for sending and receiving between the respective PROFINET devices is exchanged internally.
End device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

Communication with SIMATIC S7-1500 (Page 185)

Setting the port options (Page 211)

Boundaries at the port (Page 213)

Boundaries at the port

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

Communication with SIMATIC S7-1500 (Page 185)

Setting the port options (Page 211)

Wiring rules for disabled autonegotiation (Page 212)

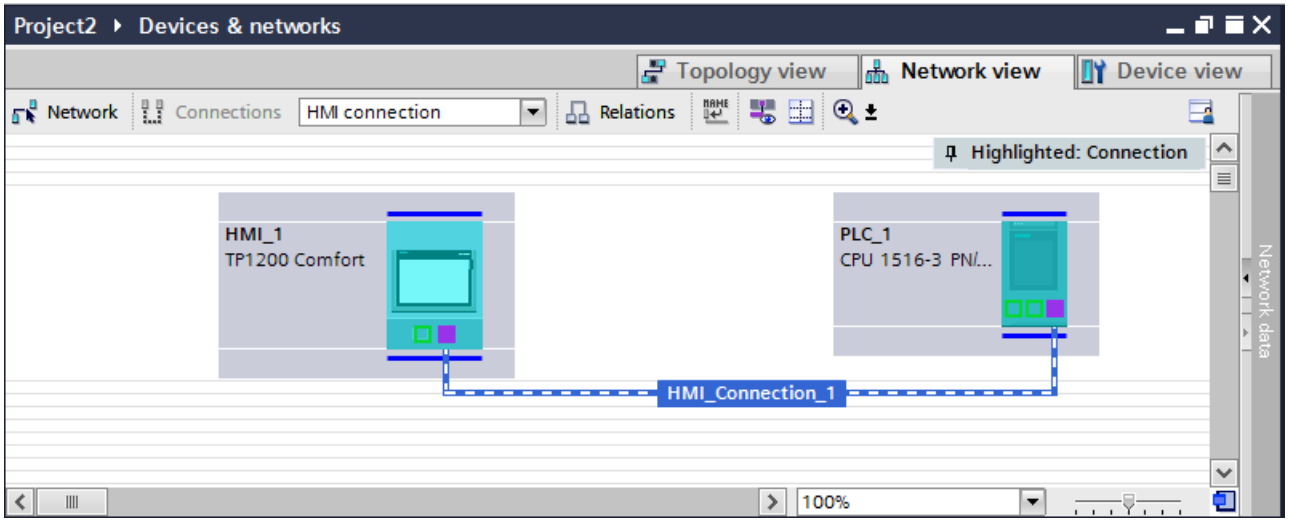
2.6.3 Communication via PROFIBUS

2.6.3.1 Configuring an HMI connection

Communication via PROFIBUS

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 185)

Configuring an HMI connection via PROFIBUS (Page 216)

Configuring an HMI connection via PROFIBUS

Introduction

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

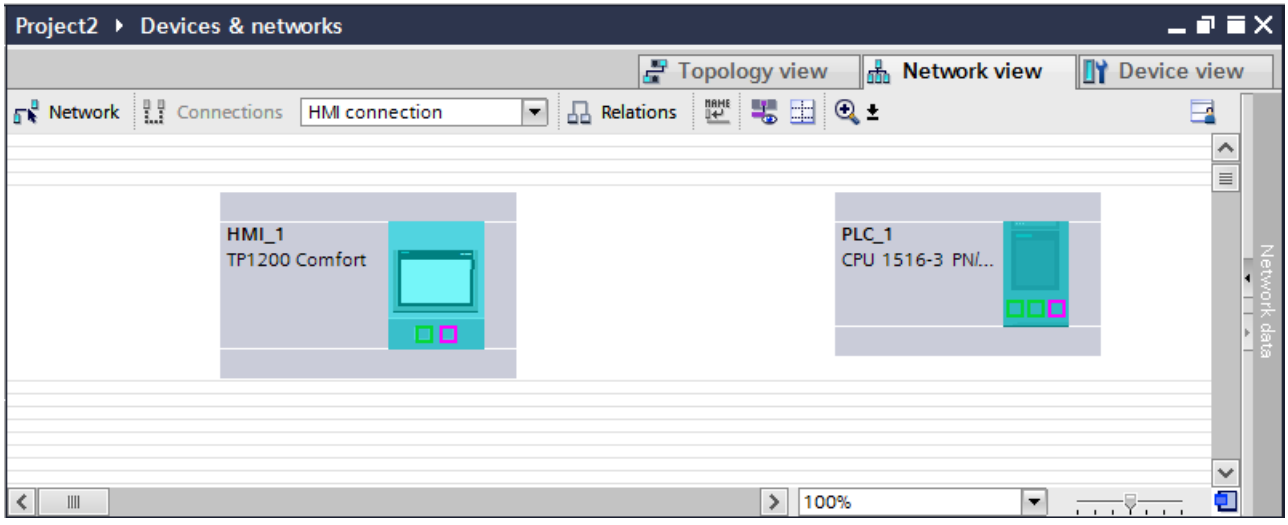
Requirements

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

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

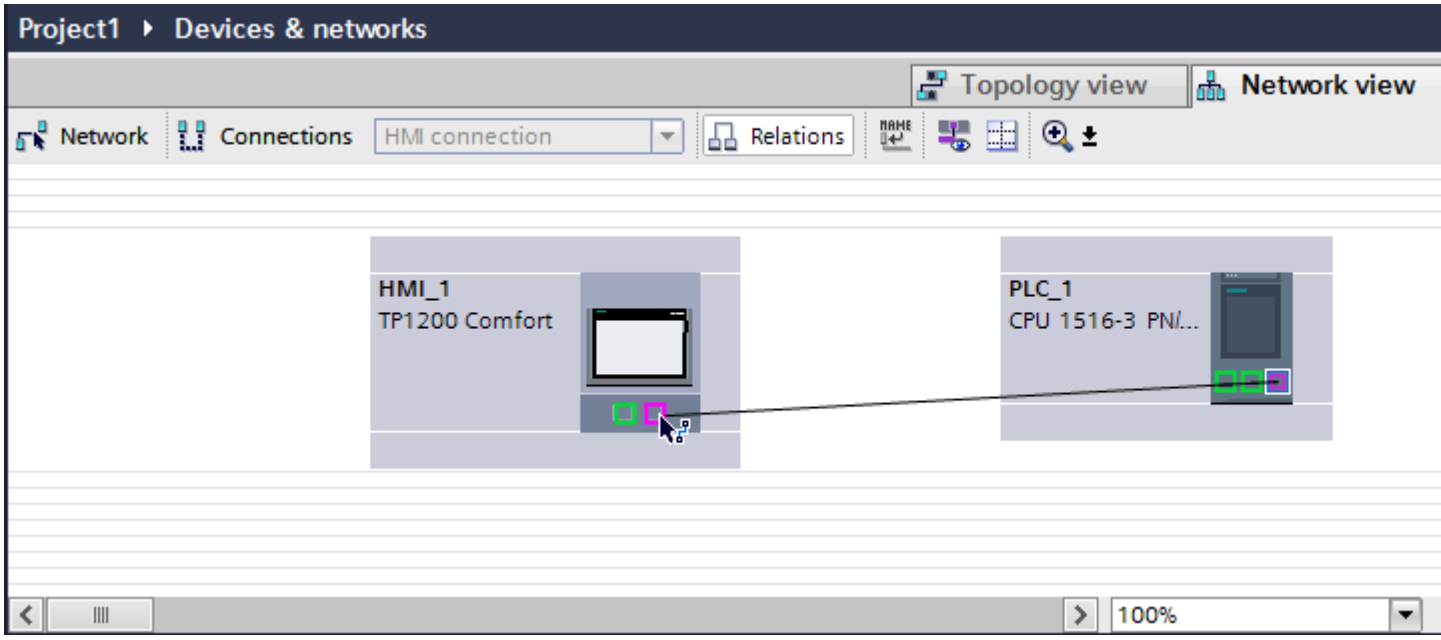
Procedure

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 PROFINET parameters in the Inspector window according to the requirements of your project.
   See the "PROFIBUS parameters (Page 225)" 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 over PROFIBUS between an HMI device and a SIMATIC S7 1500.

See also

Communication with SIMATIC S7-1500 (Page 185)

Communication via PROFIBUS (Page 215)

PROFIBUS parameters (Page 225)

2.6.3.2 Configuring an HMI connection

Communication via PROFIBUS

Communication via PROFIBUS

This section describes the communication between a WinCC Runtime and the SIMATIC S7 1500 controller via PROFIBUS.

You can use the following WinCC Runtimes as an HMI device:

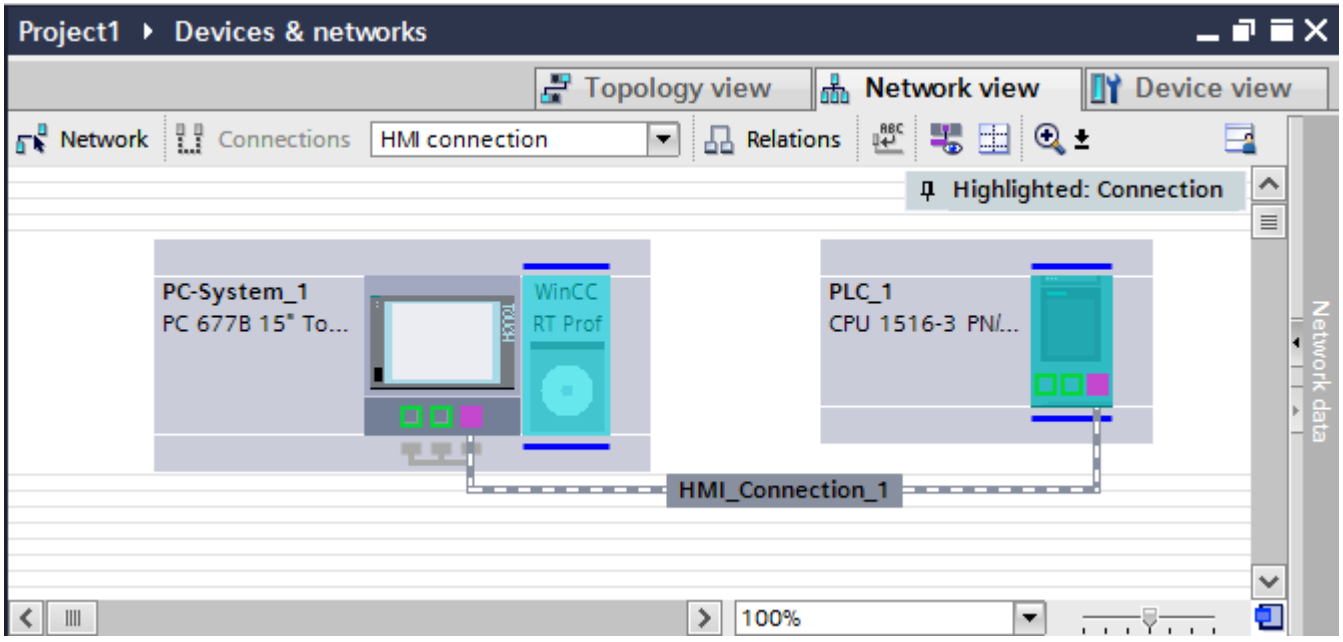
- WinCC RT Advanced
- WinCC RT Professional

WinCC Runtime as HMI device

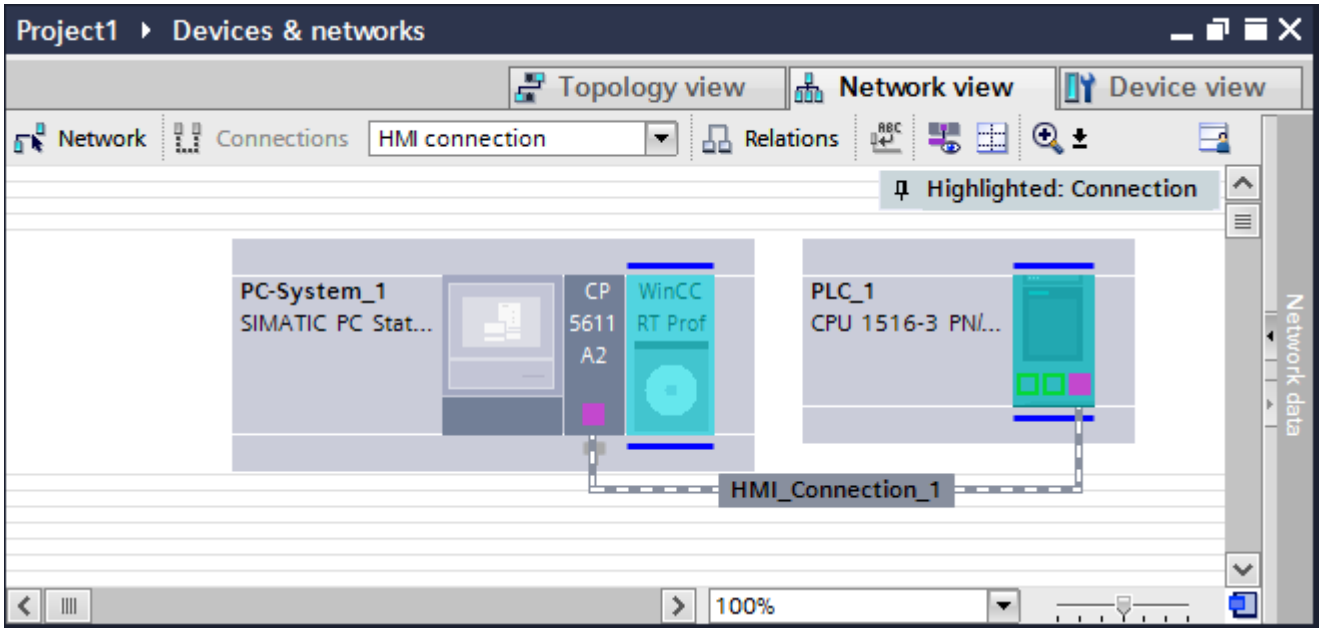
You configure the HMI connections between WinCC Runtime and SIMATIC S7 1500 in the "Devices & Networks" editor.

If you want to use a WinCC Runtime as an HMI device, you have the following options:

1. You create a SIMATIC PC and configure a WinCC Runtime on this SIMATIC PC.
   In this way you use the SIMATIC PC with a WinCC Runtime as an HMI device.



2. You create a WinCC Runtime and configure a communication processor to the Runtime.
   In this way you use your configuration PC with a WinCC Runtime as an HMI device.



You can also connect multiple HMI devices to one SIMATIC S7 1500 and multiple SIMATIC S7 1500 controllers 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 PROFIBUS in the "Devices & Networks" editor.

Connection in the "Connections" editor

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

See also

Communication with SIMATIC S7-1500 (Page 185)

Configuring an HMI connection via PROFIBUS with a PC (Page 220)

Configuring an HMI connection via PROFIBUS with a SIMATIC PC (Page 223)

Configuring an HMI connection via PROFIBUS with a PC

Introduction

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

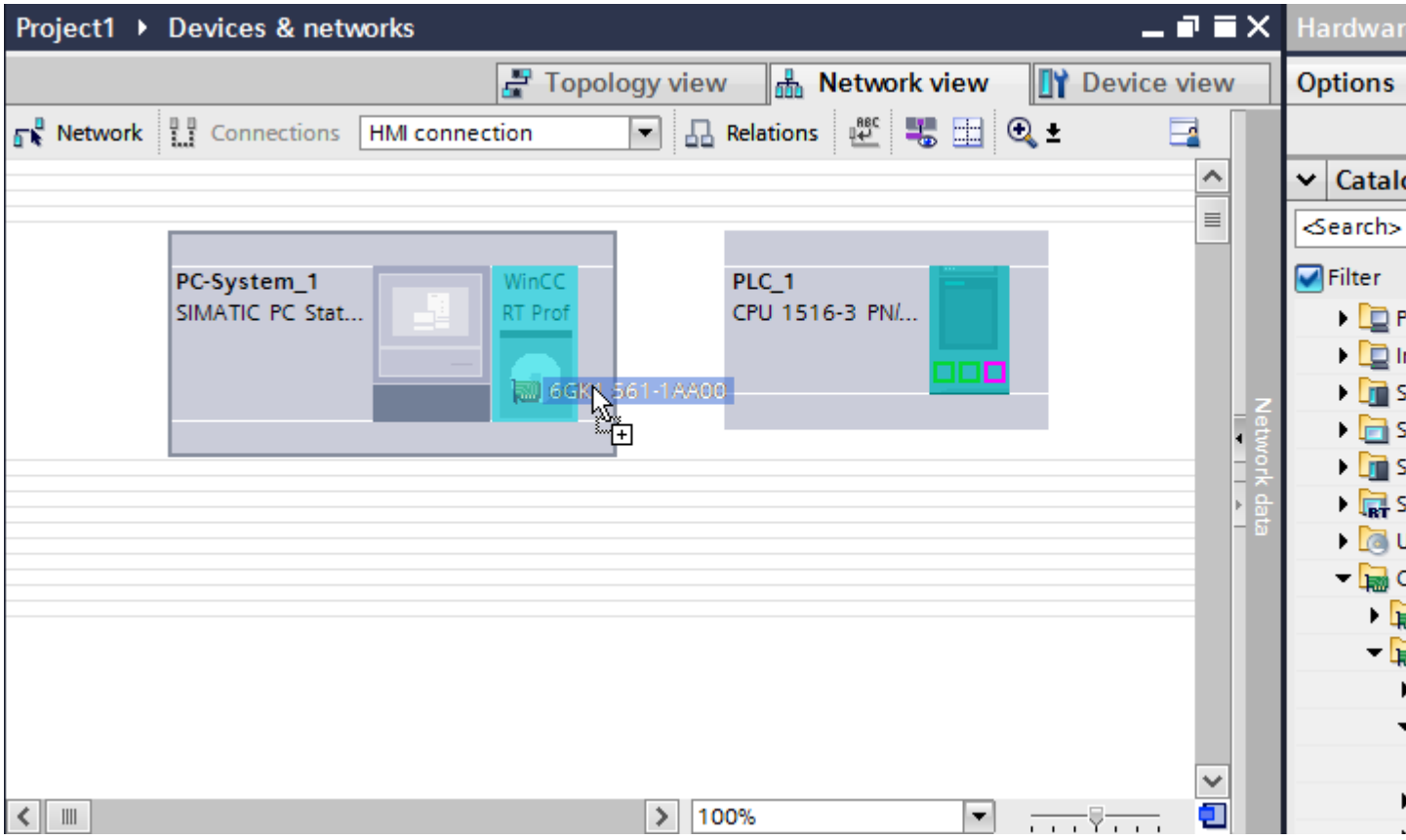
Requirements

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

- SIMATIC S7 1500 with PROFIBUS interface
- PC station with WinCC RT Advanced or WinCC RT Professional

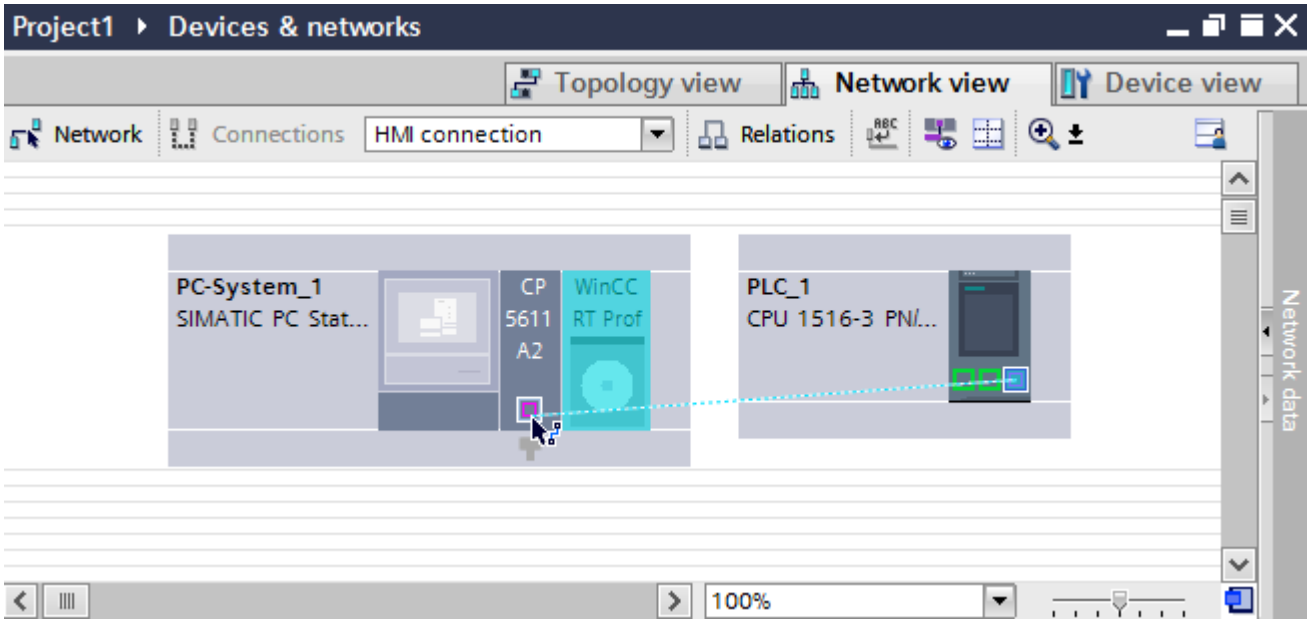
Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed in the network view.
2. Use a drag-and-drop operation to move a PROFIBUS-capable communication processor from the hardware catalog to the WinCC Runtime.



3. Click the "Connections" button and select "HMI connection" for the connection type.
   The devices available for connection are highlighted in color.

4. Click the PROFIBUS interface of the PLC and use a drag-and-drop operation to draw a connection to the PROFIBUS interface of the communication processor.



5. Click the connecting line.
6. Click "Highlight HMI connection" and select the HMI connection.
   The connection is displayed graphically in the Inspector window.
7. 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 225)" 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 1500 via PROFIBUS.

See also

Communication with SIMATIC S7-1500 (Page 185)

Communication via PROFIBUS (Page 218)

Configuring an HMI connection via PROFIBUS with a SIMATIC PC (Page 223)

PROFIBUS parameters (Page 225)

Configuring an HMI connection via PROFIBUS with a SIMATIC PC

Introduction

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

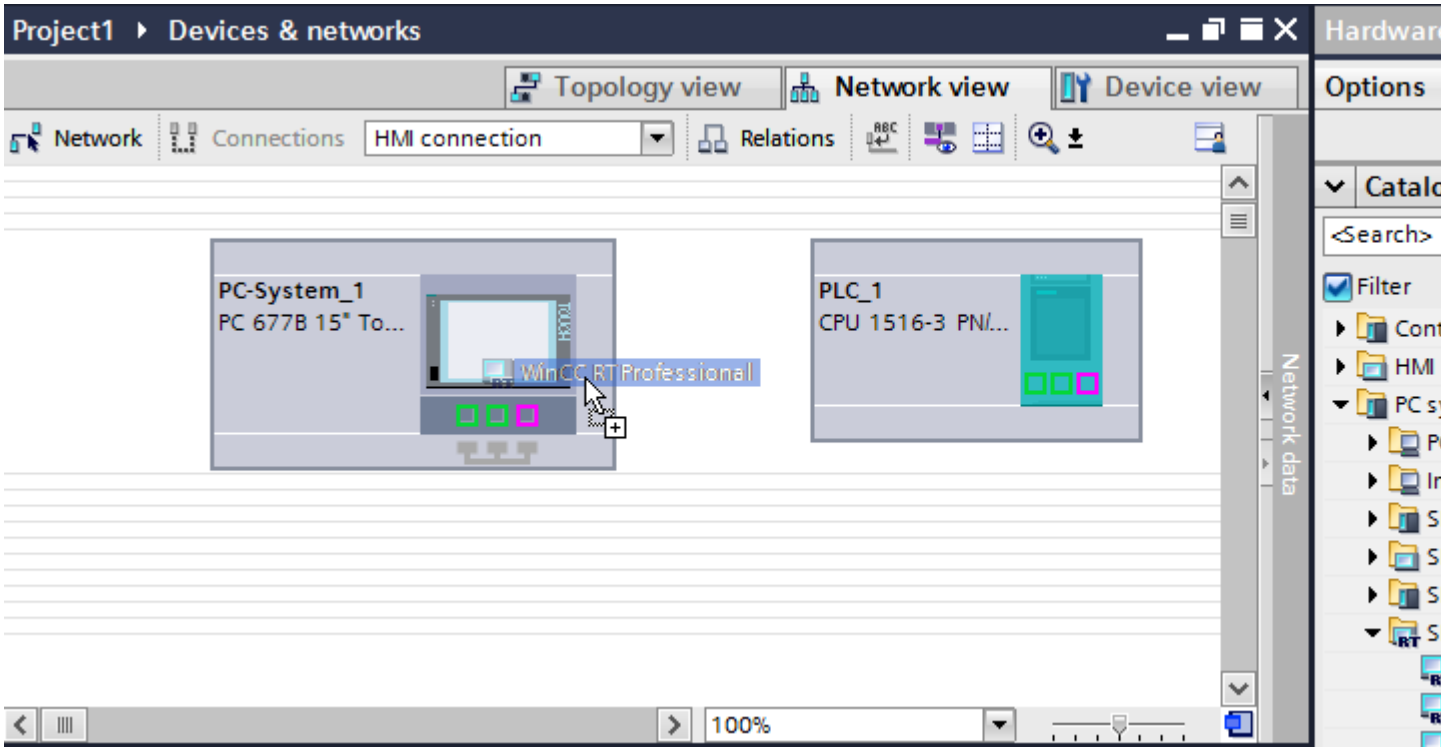
Requirements

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

- SIMATIC S7 1500 with PROFIBUS interface
- SIMATIC PC with PROFIBUS interface

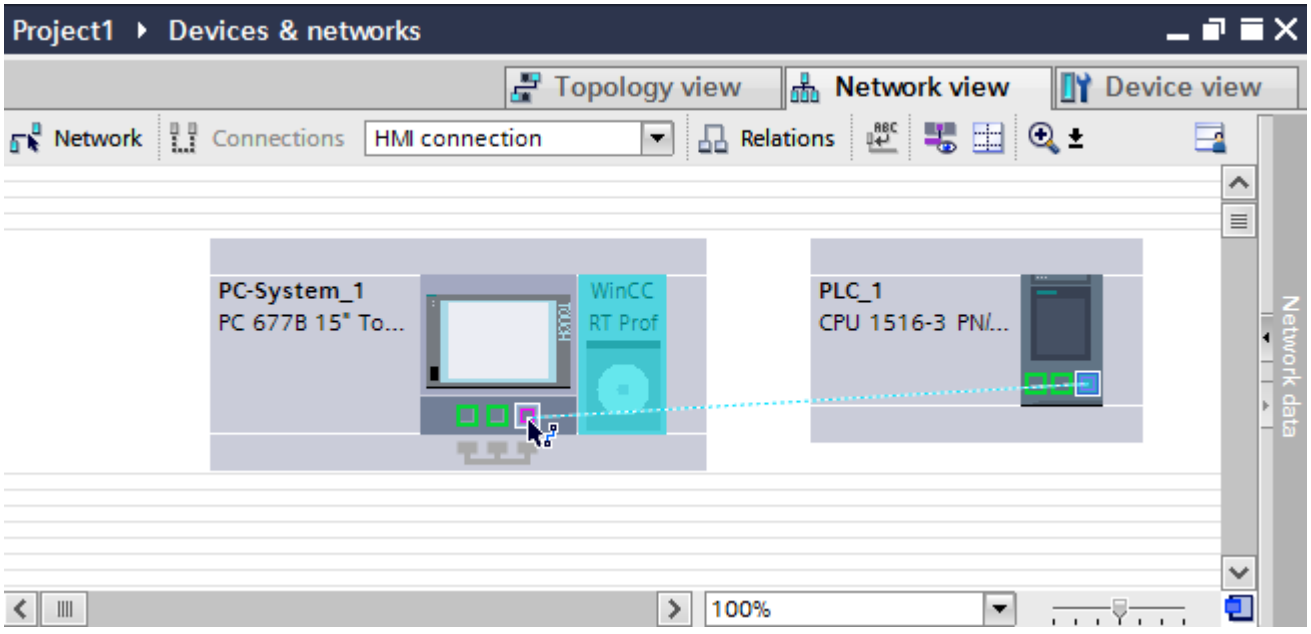
Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed in the network view.
2. Click the MPI interface of the PC and select "PROFIBUS" for the interface type in the Inspector window.
3. Use a drag-and-drop operation to move a WinCC Runtime from the hardware catalog to the PC.



4. Click the "Connections" button and select "HMI connection" for the connection type.
   The devices available for connection are highlighted in color.

5. Click the PROFIBUS interface of the PLC and use a drag-and-drop operation to draw a connection to the PROFIBUS interface of the PC.



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 225)".

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 1500 via PROFIBUS.

See also

Communication with SIMATIC S7-1500 (Page 185)

Communication via PROFIBUS (Page 218)

Configuring an HMI connection via PROFIBUS with a PC (Page 220)

PROFIBUS parameters (Page 225)

2.6.3.3 PROFIBUS parameters

PROFIBUS parameters for the HMI connection

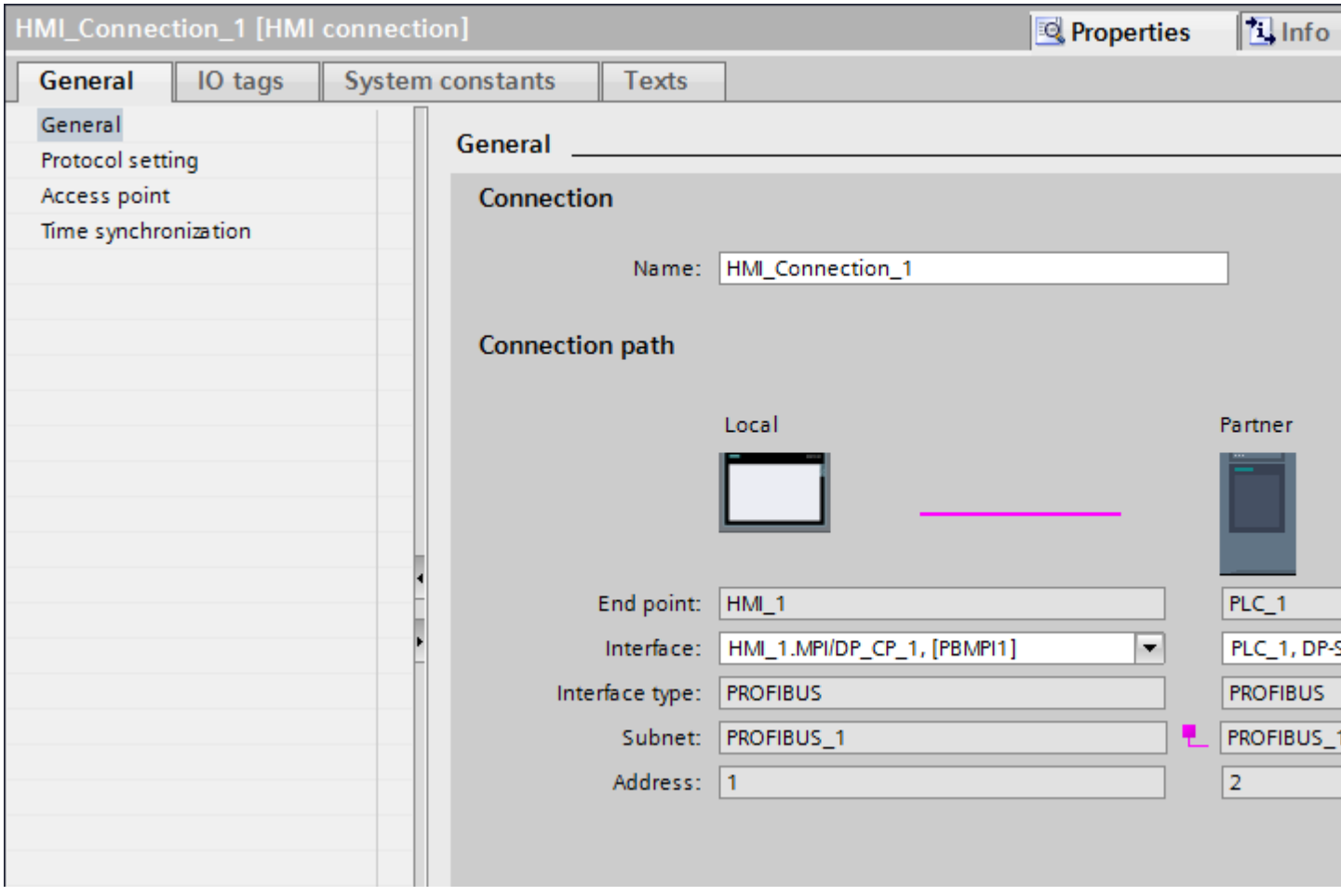
PROFIBUS parameters for the HMI connection

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 with SIMATIC S7-1500 (Page 185)

Access password for the HMI connection (Page 226)

PROFIBUS parameters for the HMI device (Page 227)

PROFIBUS parameters for the PLC (Page 229)

Bus profiles with PROFIBUS (Page 231)

Access password for the HMI connection

Introduction

Secure access to a PLC by assigning a password.

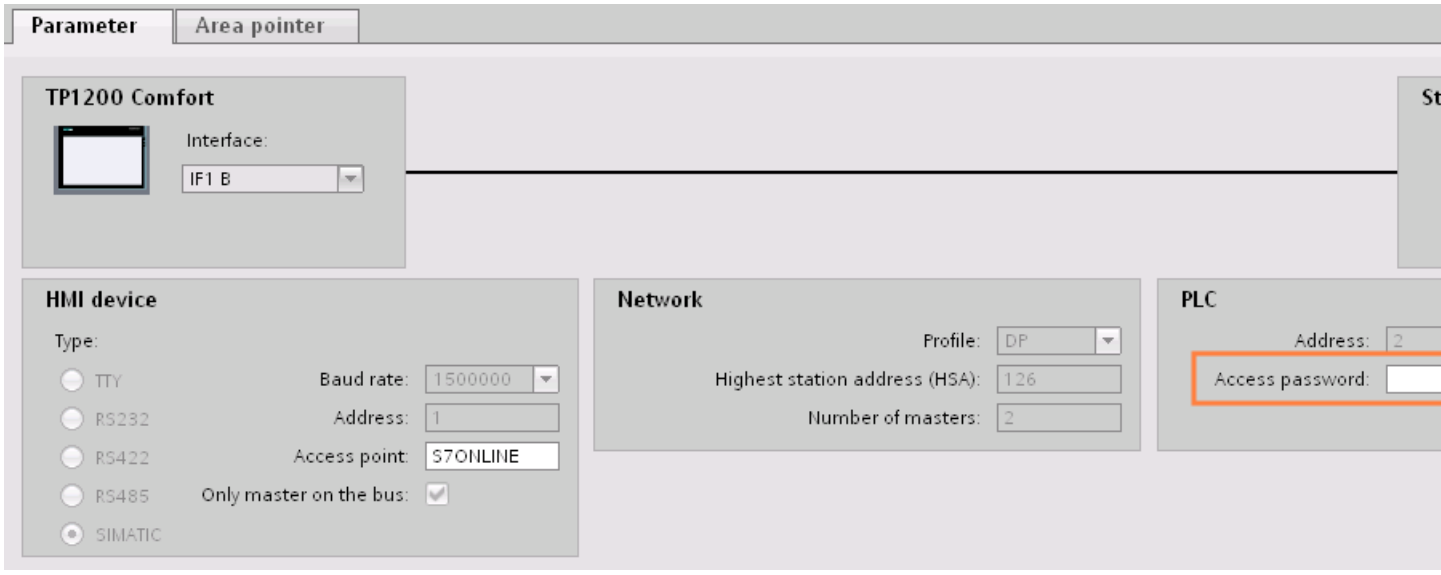
Assign the password when configuring the connection.

Input of the password of the PLC is mandatory as of "Complete protection" security level.

Communication to the PLC is denied if an incorrect password or no password is entered.

Assign password

Enter the "Access password" for the PLC in the "Connections" editor.



See also

Communication with SIMATIC S7-1500 (Page 185)

PROFIBUS parameters for the HMI connection (Page 225)

PROFIBUS parameters for the HMI device (Page 227)

PROFIBUS parameters for the PLC (Page 229)

Bus profiles with PROFIBUS (Page 231)

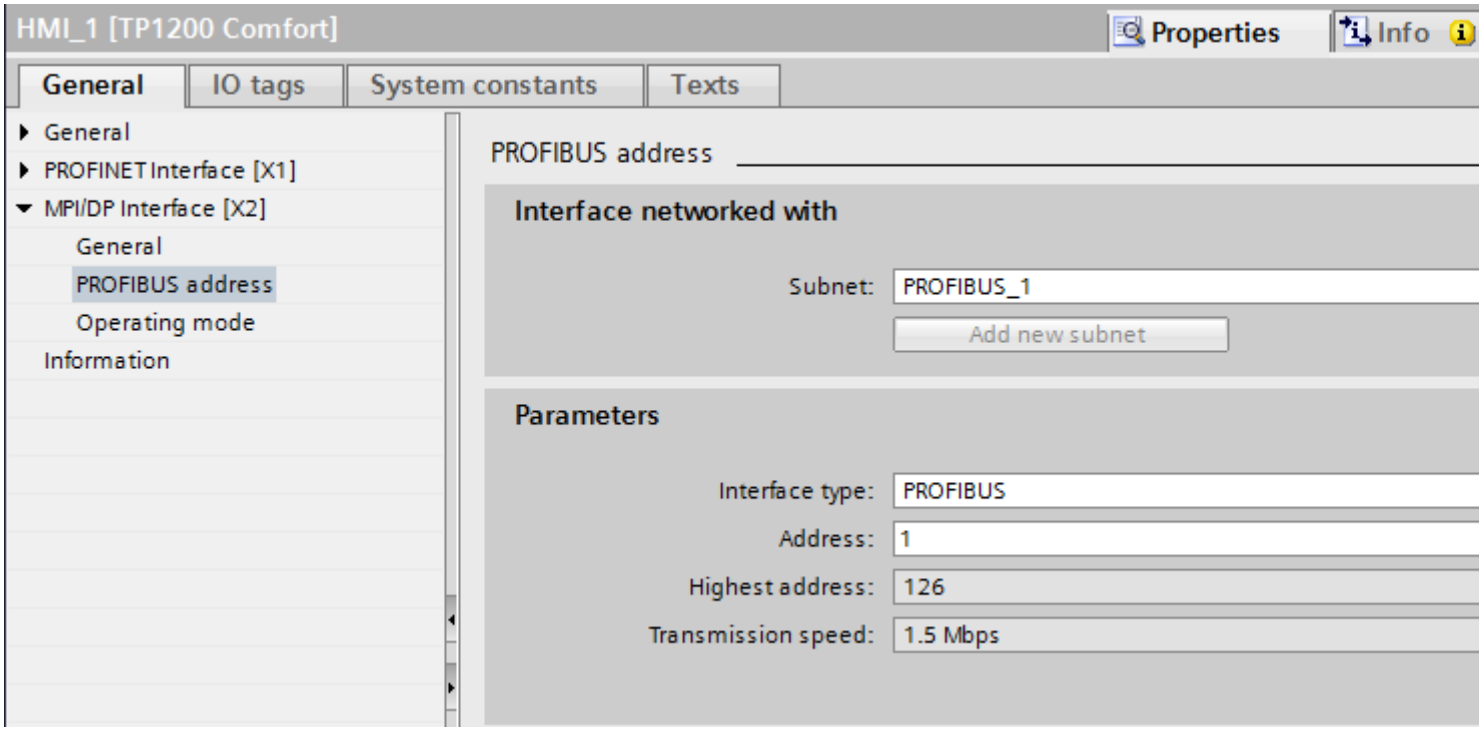
PROFIBUS parameters for the HMI device

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 185)
PROFIBUS parameters for the HMI connection (Page 225)
Access password for the HMI connection (Page 226)
PROFIBUS parameters for the PLC (Page 229)
Bus profiles with PROFIBUS (Page 231)

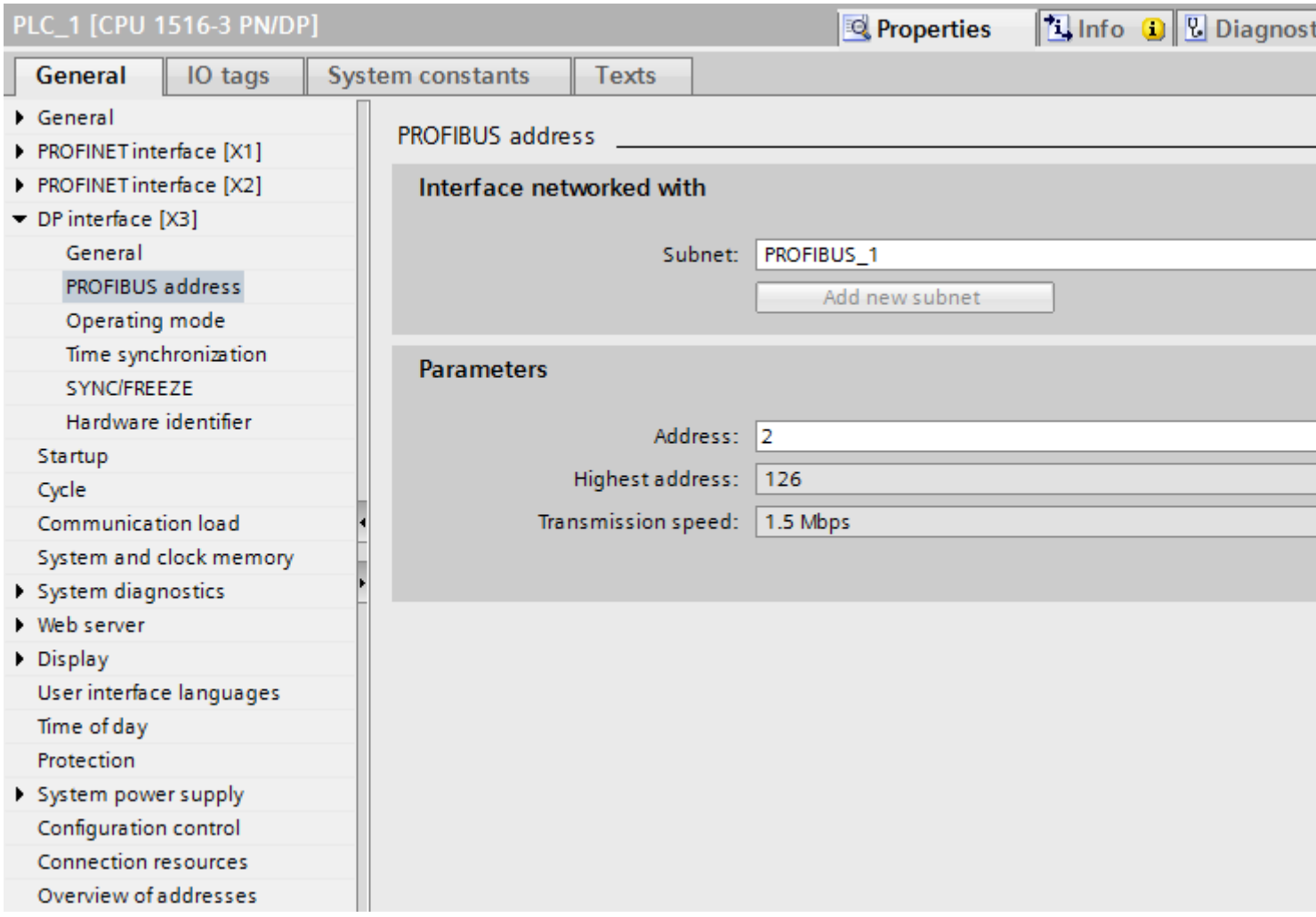
PROFIBUS parameters for the PLC

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 185)

PROFIBUS parameters for the HMI connection (Page 225)

Access password for the HMI connection (Page 226)

PROFIBUS parameters for the HMI device (Page 227)

Bus profiles with PROFIBUS (Page 231)

Bus profiles with PROFIBUS

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 transfer speeds in Kbps
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 with SIMATIC S7-1500 (Page 185)

PROFIBUS parameters for the HMI connection (Page 225)

Access password for the HMI connection (Page 226)

PROFIBUS parameters for the HMI device (Page 227)

PROFIBUS parameters for the PLC (Page 229)

Communication security

Security levels

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 233)

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

Communication with SIMATIC S7-1500 (Page 185)

PROFIBUS parameters for the HMI connection (Page 225)

Access password for the HMI connection (Page 237)

Setting options for the protection (Page 233)

Setting options for the protection

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

PROFIBUS parameters for the HMI connection (Page 225)

Security levels (Page 232)

Access password for the HMI connection (Page 237)

Access password for the HMI connection

Introduction

Secure access to a PLC by assigning a password.

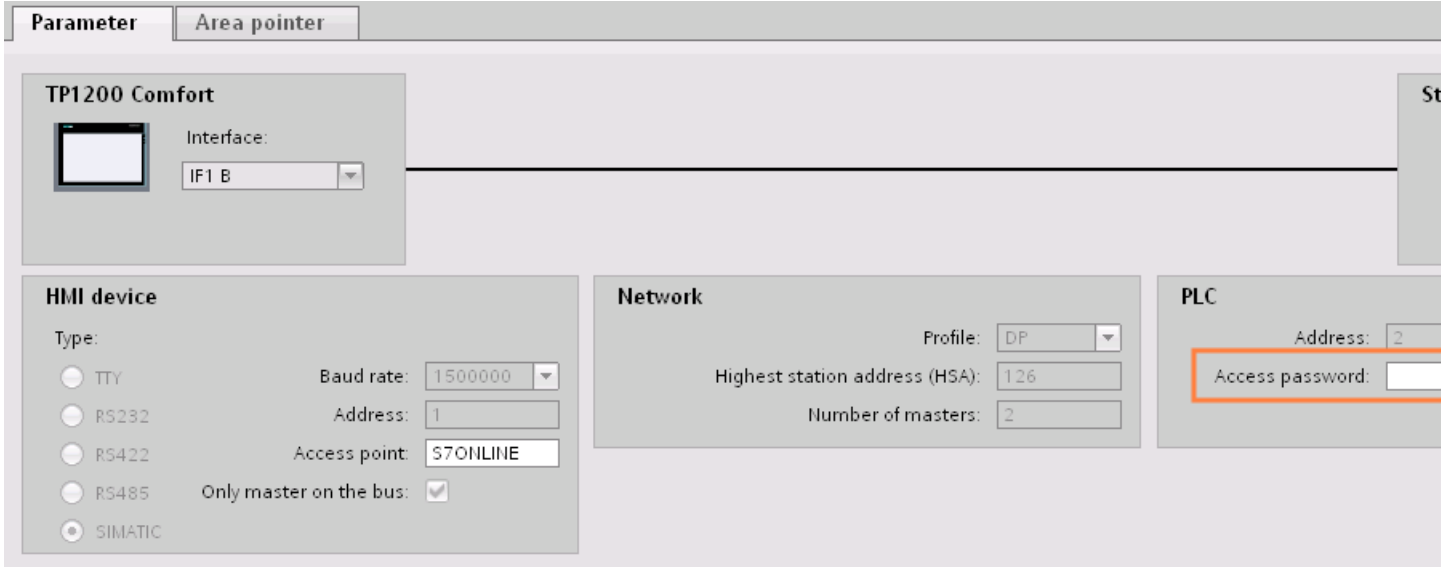
Assign the password when configuring the connection.

Input of the password of the PLC is mandatory as of "Complete protection" security level.

Communication to the PLC is denied if an incorrect password or no password is entered.

Assign password

Enter the "Access password" for the PLC in the "Connections" editor.



See also

Communication with SIMATIC S7-1500 (Page 185)

PROFIBUS parameters for the HMI connection (Page 225)

Security levels (Page 232)

2.6.4 Data exchange

2.6.4.1 Data exchange using area pointers

General information on area pointers

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 153)

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

Communication with SIMATIC S7-1500 (Page 185)

Configuring area pointers (Page 153)

"Screen number" area pointer (Page 239)

"Date/time" area pointer (Page 240)

"Date/time PLC" area pointer (Page 241)

"Coordination" area pointer (Page 243)

"Project ID" area pointer (Page 244)

"Job mailbox" area pointer (Page 246)

"Screen number" area pointer

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

Communication with SIMATIC S7-1500 (Page 185)

General information on area pointers (Page 238)

"Date/time" area pointer (Page 240)

"Date/time PLC" area pointer (Page 241)
"Coordination" area pointer (Page 243)
"Project ID" area pointer (Page 244)
"Job mailbox" area pointer (Page 246)

"Date/time" area pointer

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

Communication with SIMATIC S7-1500 (Page 185)

General information on area pointers (Page 238)

"Screen number" area pointer (Page 239)

"Date/time PLC" area pointer (Page 241)

"Coordination" area pointer (Page 243)

"Project ID" area pointer (Page 244)

"Job mailbox" area pointer (Page 246)

"Date/time PLC" area pointer

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

Communication with SIMATIC S7-1500 (Page 185)

General information on area pointers (Page 238)

"Screen number" area pointer (Page 239)

"Date/time" area pointer (Page 240)

"Coordination" area pointer (Page 243)

"Project ID" area pointer (Page 244)

"Job mailbox" area pointer (Page 246)

"Coordination" area pointer

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

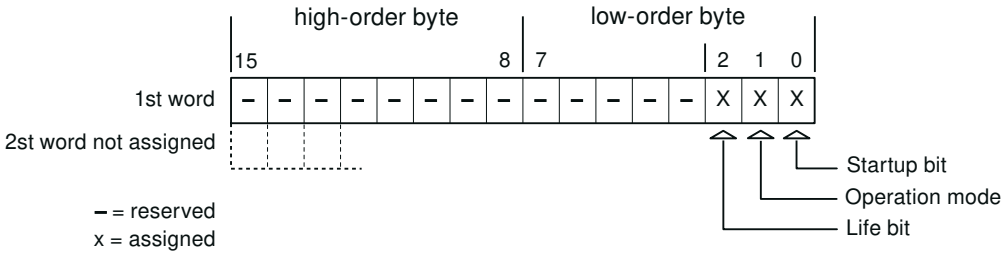
By default, the "Coordination" area pointer has the length of one word and cannot be changed.

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

Communication with SIMATIC S7-1500 (Page 185)

General information on area pointers (Page 238)

"Screen number" area pointer (Page 239)

"Date/time" area pointer (Page 240)

"Date/time PLC" area pointer (Page 241)

"Project ID" area pointer (Page 244)

"Job mailbox" area pointer (Page 246)

"Project ID" area pointer

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

Communication with SIMATIC S7-1500 (Page 185)

General information on area pointers (Page 238)

"Screen number" area pointer (Page 239)

"Date/time" area pointer (Page 240)

"Date/time PLC" area pointer (Page 241)

"Coordination" area pointer (Page 243)

"Job mailbox" area pointer (Page 246)

"Job mailbox" area pointer

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 to 23)
	Parameter 2	Left byte: Minutes (0-59) Right byte: Seconds (0 to 59)
	Parameter 3	-
15	**Setting the date (BCD coded)** [2) 3)]	
	Parameter 1	Left byte: - Right byte: Weekday (1-7: Sunday-Saturday)
	Parameter 2	Left byte: Day (1-31) Right byte: Month (1 to 12)
	Parameter 3	Left byte: Year

No.	Function	
14	**Set time (BCD-coded)**	
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**	
	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	**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**	
	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)	The weekday is ignored on HMI device KTP 600 BASIC PN.
3)	The weekday is ignored when you configure the "Date/Time PLC" area pointer.

See also

Communication with SIMATIC S7-1500 (Page 185)

General information on area pointers (Page 238)

"Screen number" area pointer (Page 239)

"Date/time" area pointer (Page 240)

"Date/time PLC" area pointer (Page 241)

"Coordination" area pointer (Page 243)

"Project ID" area pointer (Page 244)

"Data record" area pointer

"Data record" area pointer

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

Communication with SIMATIC S7-1500 (Page 185)

General information on area pointers (Page 238)

Transfer without synchronization (Page 249)

Transfer with synchronization (Page 250)

Sequence of a transfer started by the operator in the recipe view (Page 251)

Sequence of a transfer triggered by a job mailbox (Page 253)

Sequence of the transfer when triggered by a configured function (Page 254)

Possible causes of error when transferring data records (Page 256)

Transfer without synchronization

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

Communication with SIMATIC S7-1500 (Page 185)

General information on area pointers (Page 238)

"Data record" area pointer (Page 248)

Transfer with synchronization (Page 250)

Sequence of a transfer started by the operator in the recipe view (Page 251)

Sequence of a transfer triggered by a job mailbox (Page 253)

Sequence of the transfer when triggered by a configured function (Page 254)

Possible causes of error when transferring data records (Page 256)

Transfer with synchronization

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

Communication with SIMATIC S7-1500 (Page 185)
General information on area pointers (Page 238)
"Data record" area pointer (Page 248)
Transfer without synchronization (Page 249)
Sequence of a transfer started by the operator in the recipe view (Page 251)
Sequence of a transfer triggered by a job mailbox (Page 253)
Sequence of the transfer when triggered by a configured function (Page 254)
Possible causes of error when transferring data records (Page 256)

Sequence of a transfer started by the operator in the recipe view

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.

Step	Action	
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 must only ever be set by the HMI device. The PLC can only reset the status word to zero.

For Comfort Panel and Runtime Advanced, the data records are transferred even if the status word is set to 2.

Note

The PLC can only evaluate the recipe and data record number 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

Communication with SIMATIC S7-1500 (Page 185)

General information on area pointers (Page 238)

"Data record" area pointer (Page 248)

Transfer without synchronization (Page 249)

Transfer with synchronization (Page 250)

Sequence of a transfer triggered by a job mailbox (Page 253)

Sequence of the transfer when triggered by a configured function (Page 254)

Possible causes of error when transferring data records (Page 256)

Sequence of a transfer triggered by a job mailbox

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

Communication with SIMATIC S7-1500 (Page 185)
General information on area pointers (Page 238)
"Data record" area pointer (Page 248)
Transfer without synchronization (Page 249)
Transfer with synchronization (Page 250)
Sequence of a transfer started by the operator in the recipe view (Page 251)
Sequence of the transfer when triggered by a configured function (Page 254)
Possible causes of error when transferring data records (Page 256)

Sequence of the transfer when triggered by a configured function

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 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

Communication with SIMATIC S7-1500 (Page 185)
General information on area pointers (Page 238)
"Data record" area pointer (Page 248)
Transfer without synchronization (Page 249)
Transfer with synchronization (Page 250)
Sequence of a transfer started by the operator in the recipe view (Page 251)
Sequence of a transfer triggered by a job mailbox (Page 253)
Possible causes of error when transferring data records (Page 256)

Possible causes of error when transferring data records

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

Communication with SIMATIC S7-1500 (Page 185)

General information on area pointers (Page 238)

"Data record" area pointer (Page 248)

Transfer without synchronization (Page 249)

Transfer with synchronization (Page 250)

Sequence of a transfer started by the operator in the recipe view (Page 251)

Sequence of a transfer triggered by a job mailbox (Page 253)

Sequence of the transfer when triggered by a configured function (Page 254)

2.6.4.2 Trends

General information on trends

Trends

A trend is the graphical representation of one or more values from the PLC. The value is read out either time- or bit-triggered, depending on the configuration.

For additional information see:

AUTOHOTSPOT

Note

The value is read out time-triggered for Basic Panels.

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.

Bit-triggered trends

Through a trigger bit set in the trend request tag, the HMI device either reads in a trend value or an entire trend buffer. This setting is defined in the configuration. Bit-triggered trends are normally used to represent fast 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 "HMI tags" editor and connected to trend areas during configuration. The HMI device and PLC then communicate with each other via 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)

See also

Communication with SIMATIC S7-1500 (Page 185)

Trend request and trend transfer (Page 258)

Permitted data types for trends (Page 259)

Trend request and trend transfer

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 closing the screen, the HMI device resets the relevant bits in the trend request area.

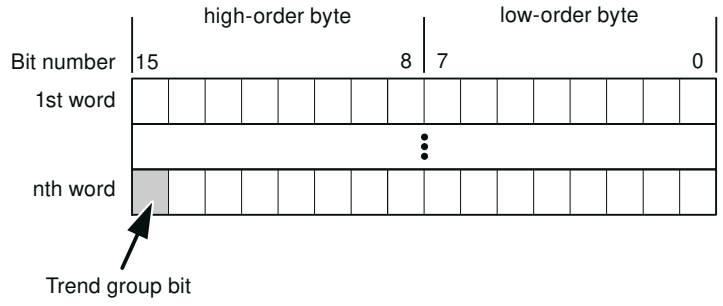
The trend request area can be used for evaluation purposes in the PLC to determine which trend is currently being displayed on the HMI device. Trends can also be triggered without evaluation of the trend request area.

Trend transfer area 1

This area is used to trigger trends. You must set the bit assigned to the trend in the trend transfer area and set the trend group bit in your control program. 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.

The PLC writes to Buffer 2 while the HMI device reads values from Buffer 1, and writes to Buffer 1 when the HMI device is reading Buffer 2. This prevents the PLC from overwriting trend values while the trend is being read by the HMI device.

See also

Communication with SIMATIC S7-1500 (Page 185)

General information on trends (Page 257)

Permitted data types for trends (Page 259)

Permitted data types for trends

For SIMATIC S7

You assign one bit to each trend during configuration. Tags and array tags of the "Word" or "Int" data type are permitted.

See also

Communication with SIMATIC S7-1500 (Page 185)

General information on trends (Page 257)

Trend request and trend transfer (Page 258)

2.6.4.3 Alarms

Configuring alarms

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

Communication with SIMATIC S7-1500 (Page 185)

Acknowledgment of alarms (Page 260)

Acknowledgment of alarms

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

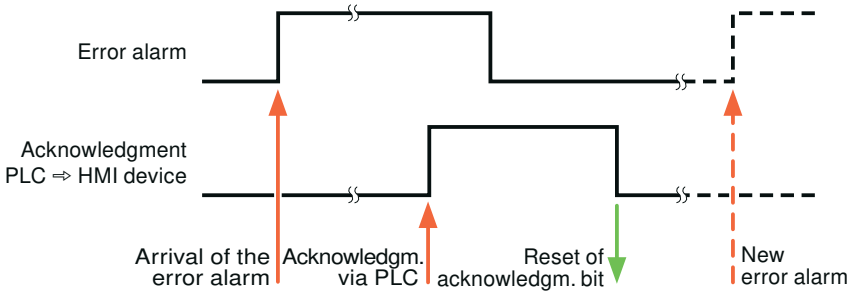
Acknowledgment by the PLC

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.

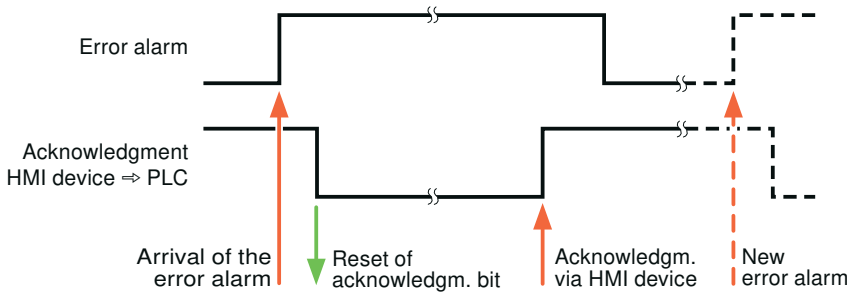
If a discrete alarm subject to acknowledgment is acknowledged by the HMI device, then the corresponding bit in the assigned acknowledgment tag will be set. The entire acknowledgment tag is then written to the PLC by the HMI device. This allows the PLC to recognize that a certain alarm message has been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

Communication with SIMATIC S7-1500 (Page 185)

Configuring alarms (Page 259)

2.6.4.4 LED mapping

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 Comfort Panels	Panel PCs
0	0	Off	Off
0	1	Rapid flashing	Flashing

Bit n+ 1	Bit n	LED function	
		all Mobile Panels, all Comfort Panels	**Panel PCs**
1	0	Slow flashing	Flashing
1	1	On permanently	On permanently

See also

Communication with SIMATIC S7-1500 (Page 185)

2.6.5 Performance features of communication

2.6.5.1 S7-1500 device dependency

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 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

Basic Panels V14.0.1

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 V15.0

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

Mobile Panels V12.0

HMI devices	SIMATIC S7-1500
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 S7-1500
KTP700 Mobile	Yes
KTP700F Mobile	Yes
KTP900 Mobile	Yes
KTP900F Mobile	Yes

Mobile Panels V14.0

HMI devices	SIMATIC S7-1500
KTP700 Mobile	Yes
KTP700F Mobile	Yes
KTP900 Mobile	Yes
KTP900F Mobile	Yes
KTP400F Mobile	Yes

Mobile Panels V14.0.1

HMI devices	SIMATIC S7-1500
KTP700 Mobile	Yes
KTP700F Mobile	Yes
KTP900 Mobile	Yes
KTP900F Mobile	Yes
KTP400F Mobile	Yes

Mobile Panels V15.0

HMI devices	SIMATIC S7-1500
KTP700 Mobile	Yes
KTP700F Mobile	Yes

HMI devices	SIMATIC S7-1500
KTP900 Mobile	Yes
KTP900F Mobile	Yes
KTP400F Mobile	Yes

Comfort Panels V12.0

HMI devices	SIMATIC S7-1500
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

HMI devices	SIMATIC S7-1500
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 S7-1500
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 S7-1500
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 V14.0

HMI devices	SIMATIC S7-1500
KP400 Comfort	Yes
KTP400 Comfort	Yes
KTP400 Comfort Portrait	Yes
KP700 Comfort	Yes
TP700 Comfort	Yes
TP700 Comfort Portrait	Yes

HMI devices	SIMATIC S7-1500
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

Comfort Panels V14.0.1

HMI devices	SIMATIC S7-1500
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

HMI devices	SIMATIC S7-1500
TP1900 Comfort Portrait	Yes
TP2200 Comfort	Yes
TP2200 Comfort Portrait	Yes

Comfort Panels V15.0

HMI devices	SIMATIC S7-1500
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 V12.0

HMI devices	SIMATIC S7-1500
WinCC RT Advanced	Yes

Runtime V13.0

HMI devices	SIMATIC S7-1500
WinCC RT Advanced	Yes

Runtime V13.0.1

HMI devices	SIMATIC S7-1500
WinCC RT Advanced	Yes

Runtime V14.0

HMI devices	SIMATIC S7-1500
WinCC RT Advanced	Yes
WinCC RT Professional	Yes

Runtime V14.0.1

HMI devices	SIMATIC S7-1500
WinCC RT Advanced	Yes
WinCC RT Professional	Yes

Runtime V15.0

HMI devices	SIMATIC S7-1500
WinCC RT Advanced	Yes
WinCC RT Professional	Yes

See also

Communication with SIMATIC S7-1500 (Page 185)
Valid data types for SIMATIC S7 1500 (Page 270)
Addressing (Page 272)

2.6.5.2 Valid data types for SIMATIC S7 1500

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

Data type	Length	
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
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 185)

S7-1500 device dependency (Page 263)

Addressing (Page 272)

2.6.5.3 Addressing

Addressing of tags with SIMATIC S7 1500

WinCC supports the following types of addressing in communication of HMI devices with the SIMATIC S7 1500 controller:

- Symbolic addressing
- Absolute addressing

See also

Communication with SIMATIC S7-1500 (Page 185)

S7-1500 device dependency (Page 263)

Valid data types for SIMATIC S7 1500 (Page 270)

2.6.6 Configuring connections in the "Connections" editor

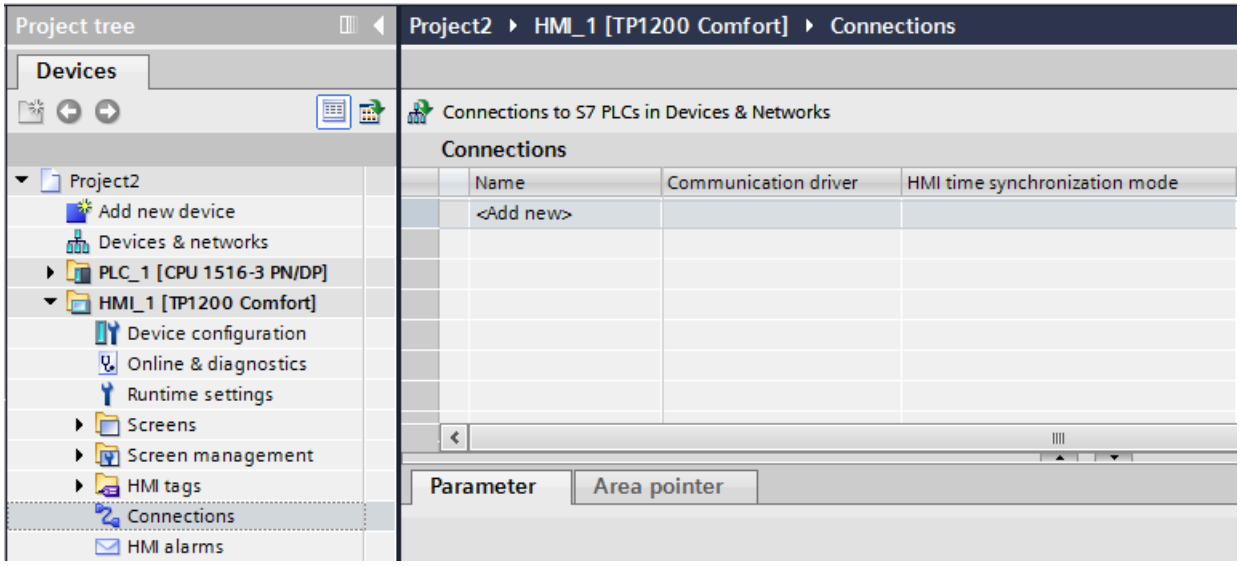
2.6.6.1 Creating a PROFINET connection

Requirements

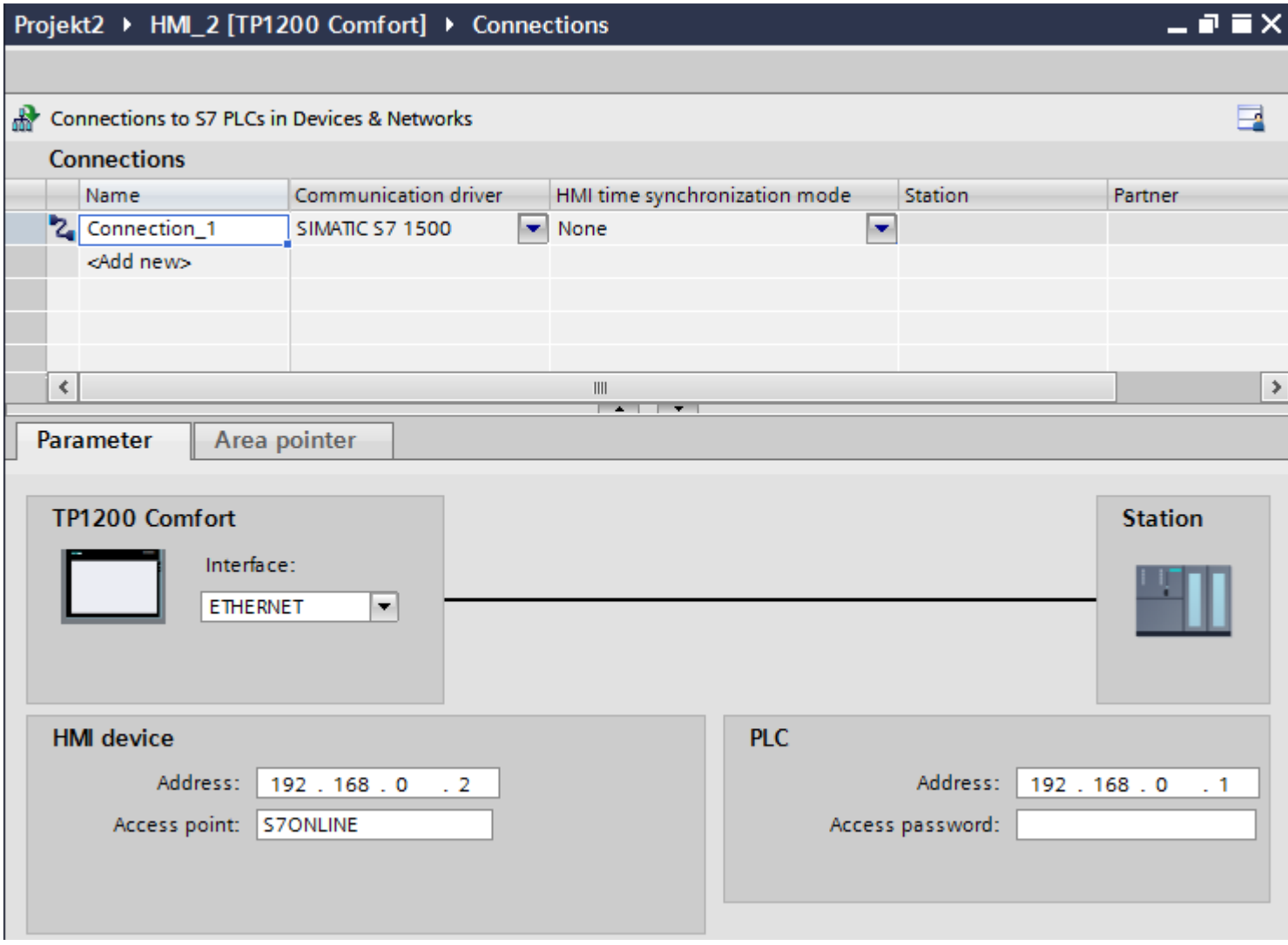
- A project is open.
- An HMI device with a PROFINET interface has been created.

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 with SIMATIC S7-1500 (Page 185)

Creating a PROFIBUS connection (Page 275)

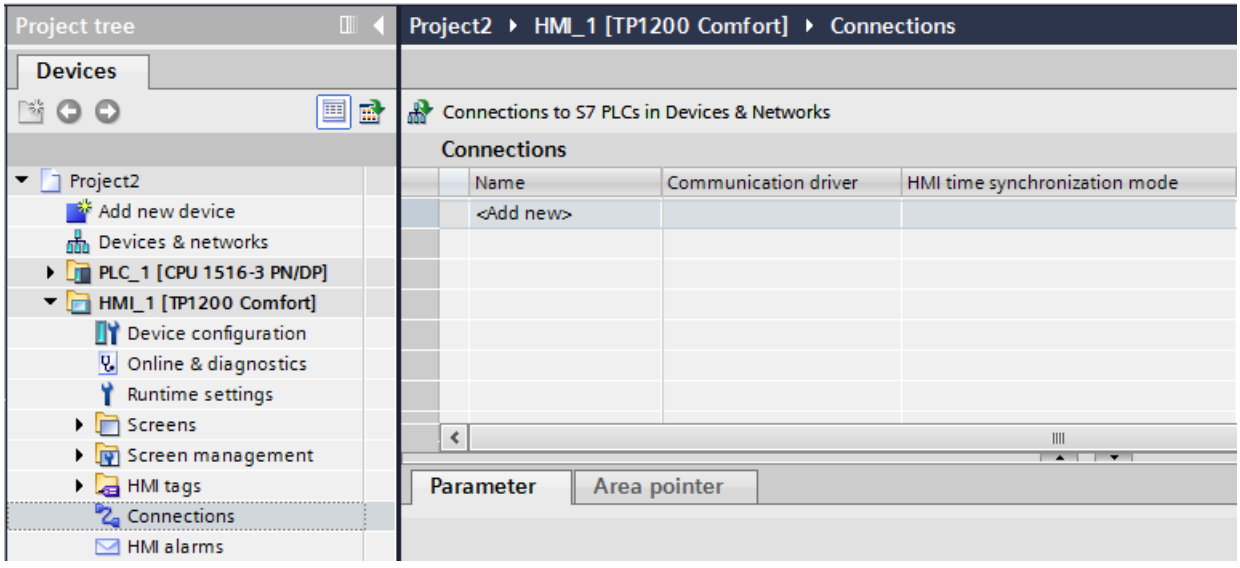
2.6.6.2 Creating a PROFIBUS connection

Requirements

- A project is open.
- An HMI device with a PROFIBUS interface has been created.

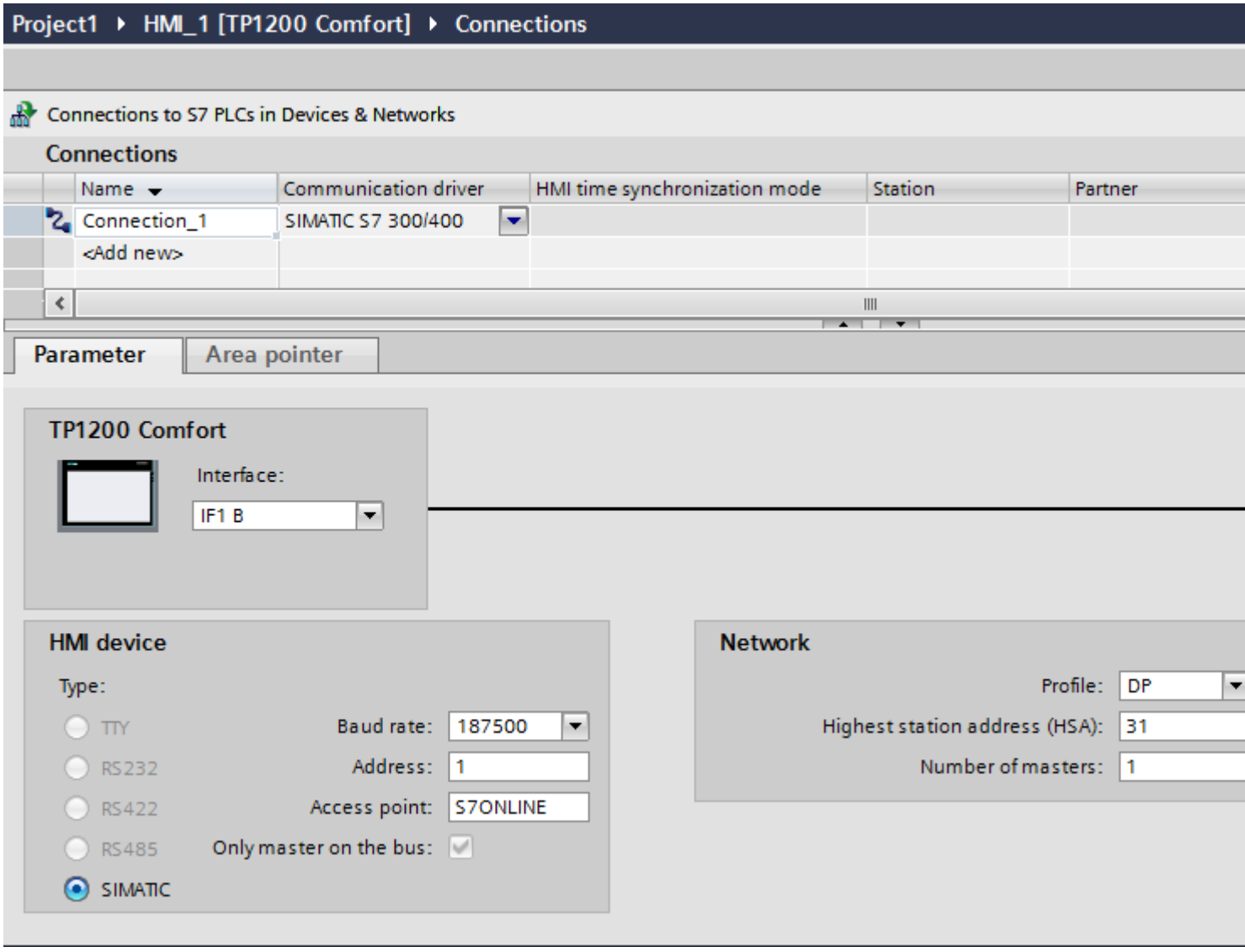
Procedure

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 "IF 1 B" interface from "Parameters > interface" in the Inspector window.

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 S7-1500 (Page 185)

Creating a PROFINET connection (Page 272)

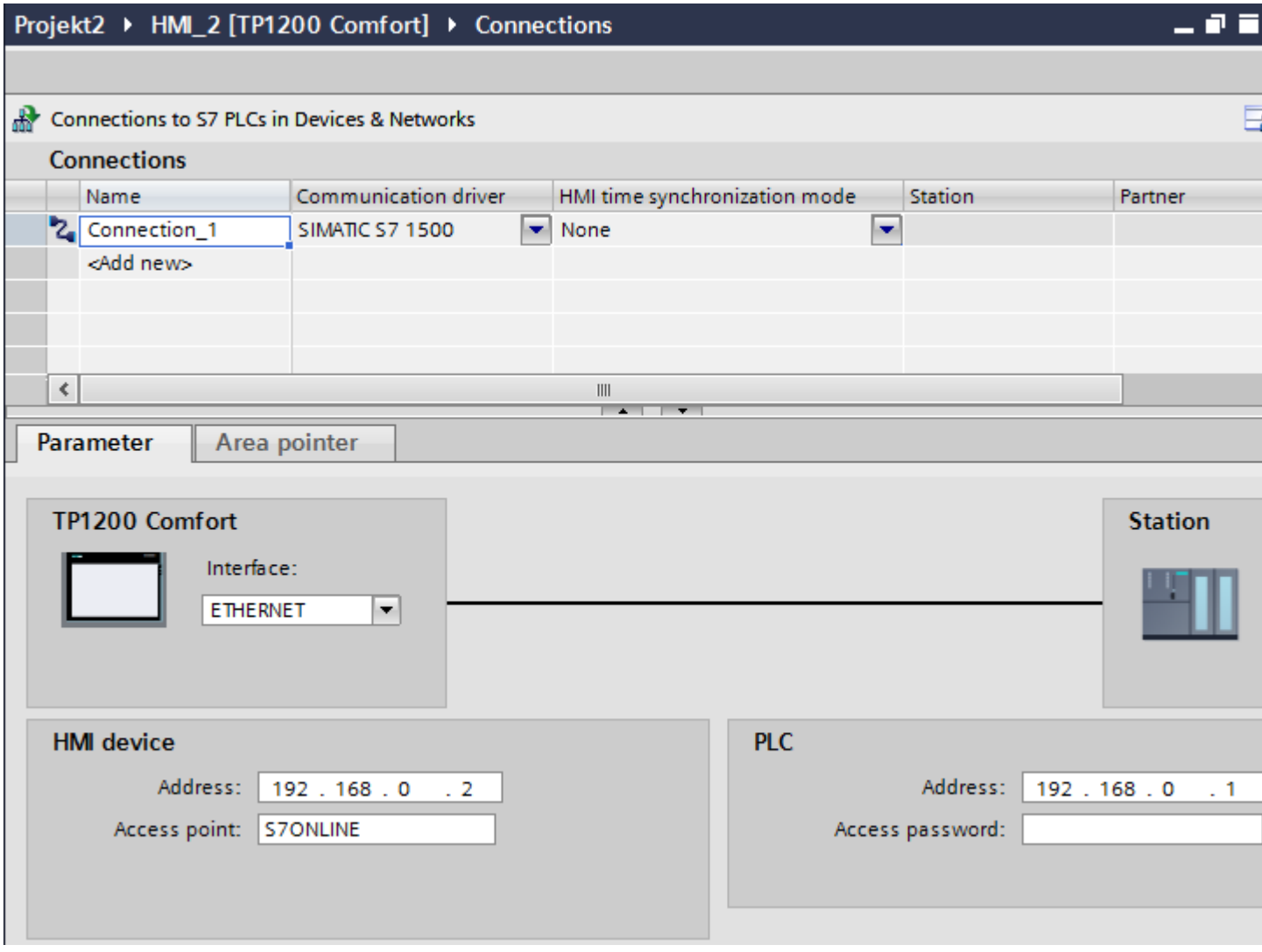
2.6.6.3 Parameters for the connection

Parameters for the connection (SIMATIC S7 1500)

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 185)

PROFIBUS parameters (Page 278)

Ethernet parameters (Page 280)

PROFIBUS parameters

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.
- "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.
- "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 185)

Parameters for the connection (SIMATIC S7 1500) (Page 277)

Ethernet parameters (Page 280)

Ethernet parameters

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 185)

Parameters for the connection (SIMATIC S7 1500) (Page 277)

PROFIBUS parameters (Page 278)

2.6.7 Configuring connections in the "Connections" editor

2.6.7.1 Creating a connection

Requirements

- A project is open.
- WinCC RT Professional is 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 an interface of the HMI device in the Inspector window under "Parameters > Interface".
6. Set the parameters for the connection in the Inspector window.

Interfaces

Select one of the following interfaces in the Inspector window under "Parameters > WinCC RT Professional > Interfaces":

- TCP/IP
- PROFIBUS

For additional information on the parameters of the interfaces see:

Parameters for the connection (SIMATIC S7 1500) (Page 282)

See also

Communication with SIMATIC S7-1500 (Page 185)

Parameters for the connection (SIMATIC S7 1500) (Page 282)

2.6.7.2 Parameters for the connection (SIMATIC S7 1500)

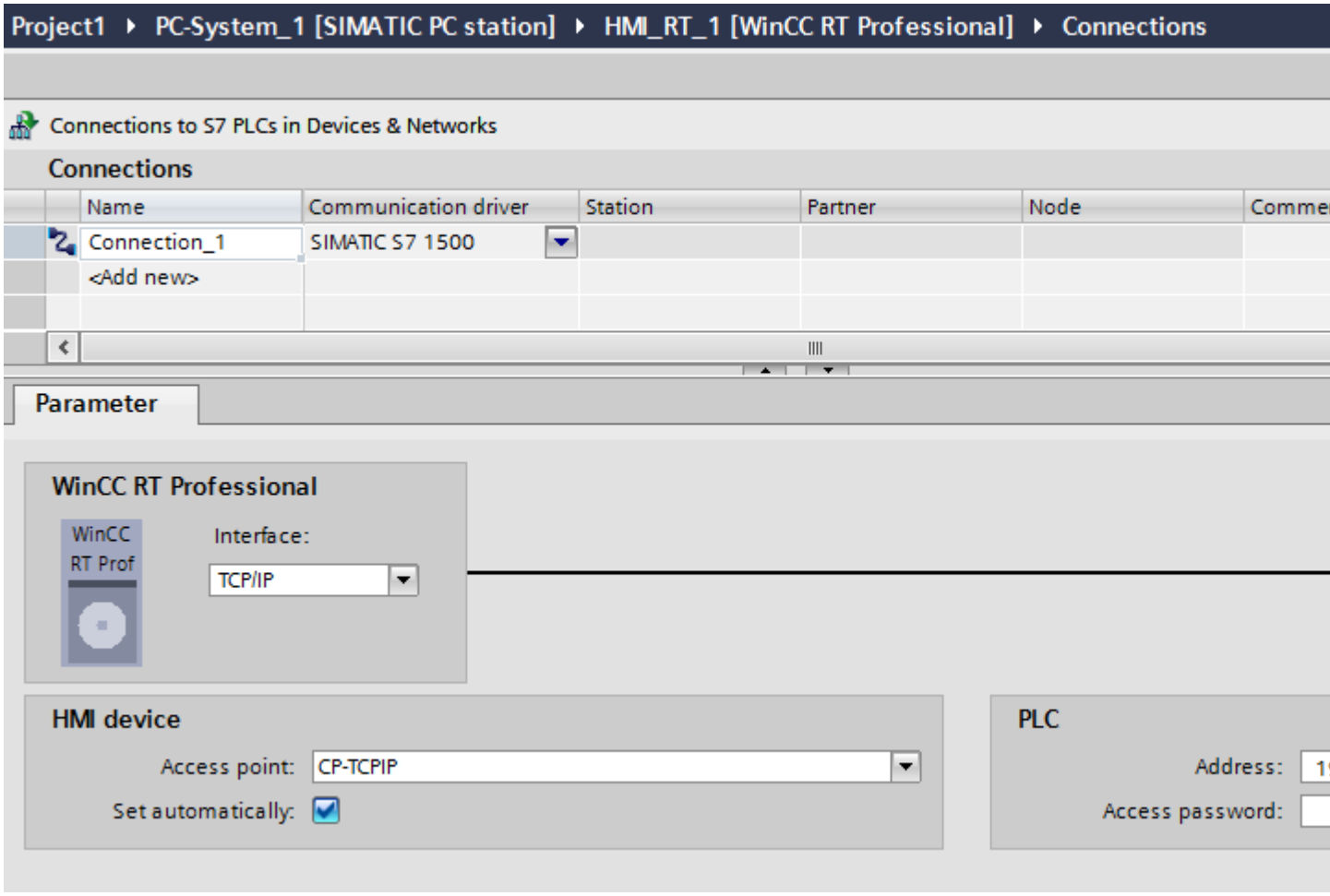
TCP/IP

Introduction

You assign the parameters for the HMI device and the PLC in the network under "Parameters".

The parameters described in the following apply to the following interfaces:

- ETHERNET



HMI device

- "Access point"
  Specifies the access point for the PG/PC interface that can be used to reach the communication partner.
  If only one communications processor is installed for a communication type, select the option "Set automatically". The logical device name is then set automatically when you start Runtime.

PLC

- "Address"
  Enter the station address of the HMI device under "Address".
- "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 185)

Creating a connection (Page 281)

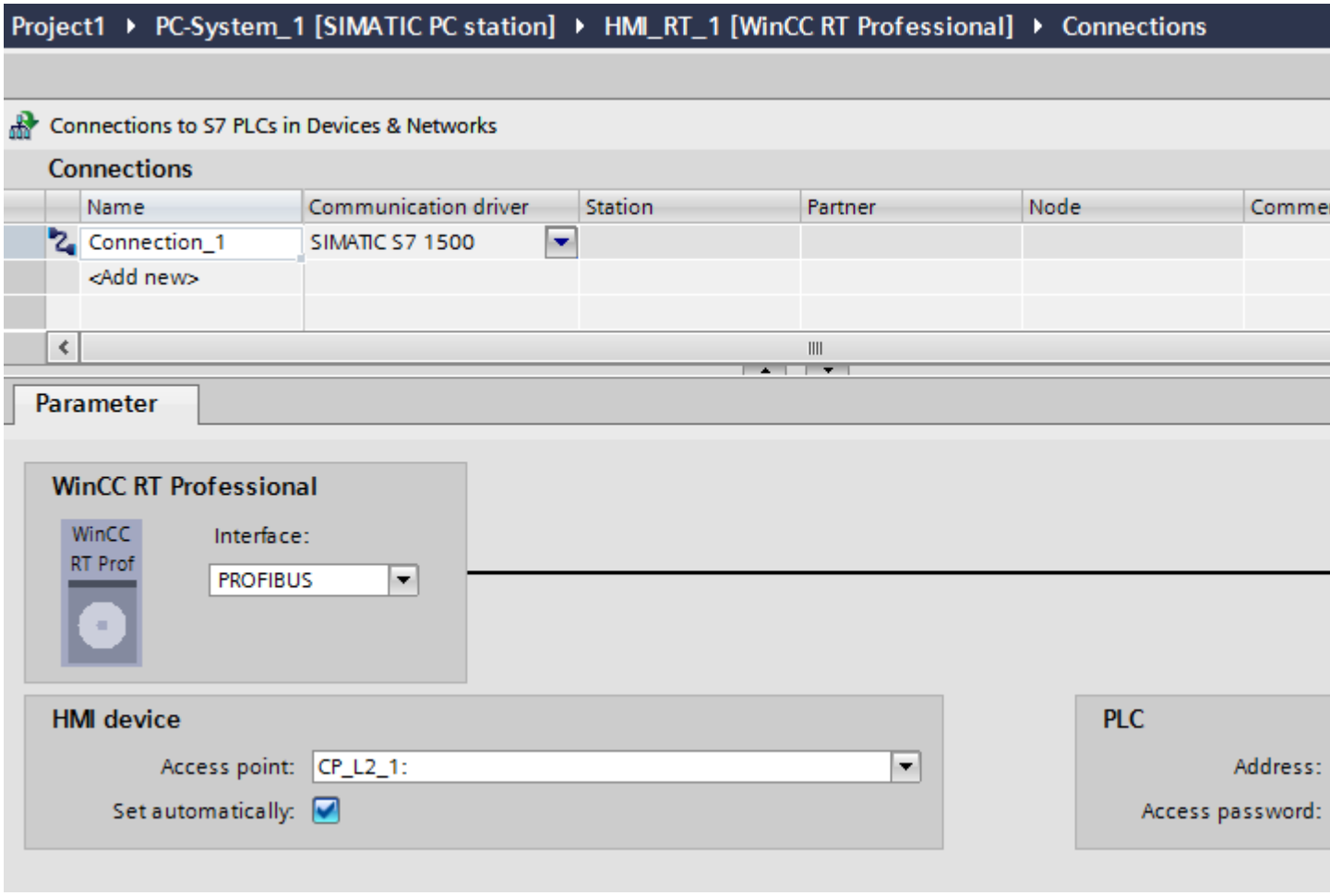
PROFIBUS

Introduction

You assign the parameters for the HMI device and the PLC in the network under "Parameters".

The parameters described in the following apply to the following interfaces:

- PROFIBUS



HMI device

- "Access point"
  Enter the logical device name under "Access point".

PLC

- "Address"
  Enter the station address of the CPU under "Address".
- "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 185)

Creating a connection (Page 281)

2.6.8 Configuring time synchronization

2.6.8.1 Time synchronization

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

Communication with SIMATIC S7-1500 (Page 185)

Time synchronization restrictions (Page 287)

Configuring time synchronization for integrated connections (Page 288)

Configuring time synchronization for non-integrated connections (Page 289)

2.6.8.2 Time synchronization restrictions

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	-
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.

See also

Communication with SIMATIC S7-1500 (Page 185)

Time synchronization (Page 286)

Configuring time synchronization for integrated connections (Page 288)

Configuring time synchronization for non-integrated connections (Page 289)

2.6.8.3 Configuring time synchronization for integrated connections

Introduction

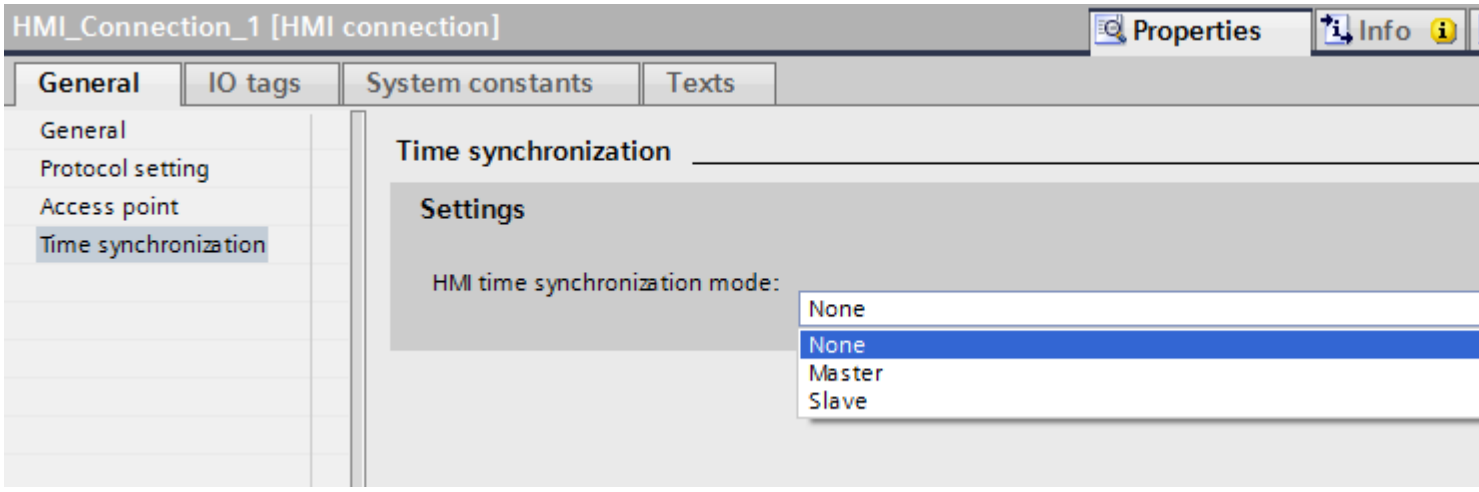
You configure time synchronization for an integrated connection in the "Devices & Networks" editor.

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

Communication with SIMATIC S7-1500 (Page 185)

Time synchronization (Page 286)

Time synchronization restrictions (Page 287)

Configuring time synchronization for non-integrated connections (Page 289)

2.6.8.4 Configuring time synchronization for non-integrated connections

Introduction

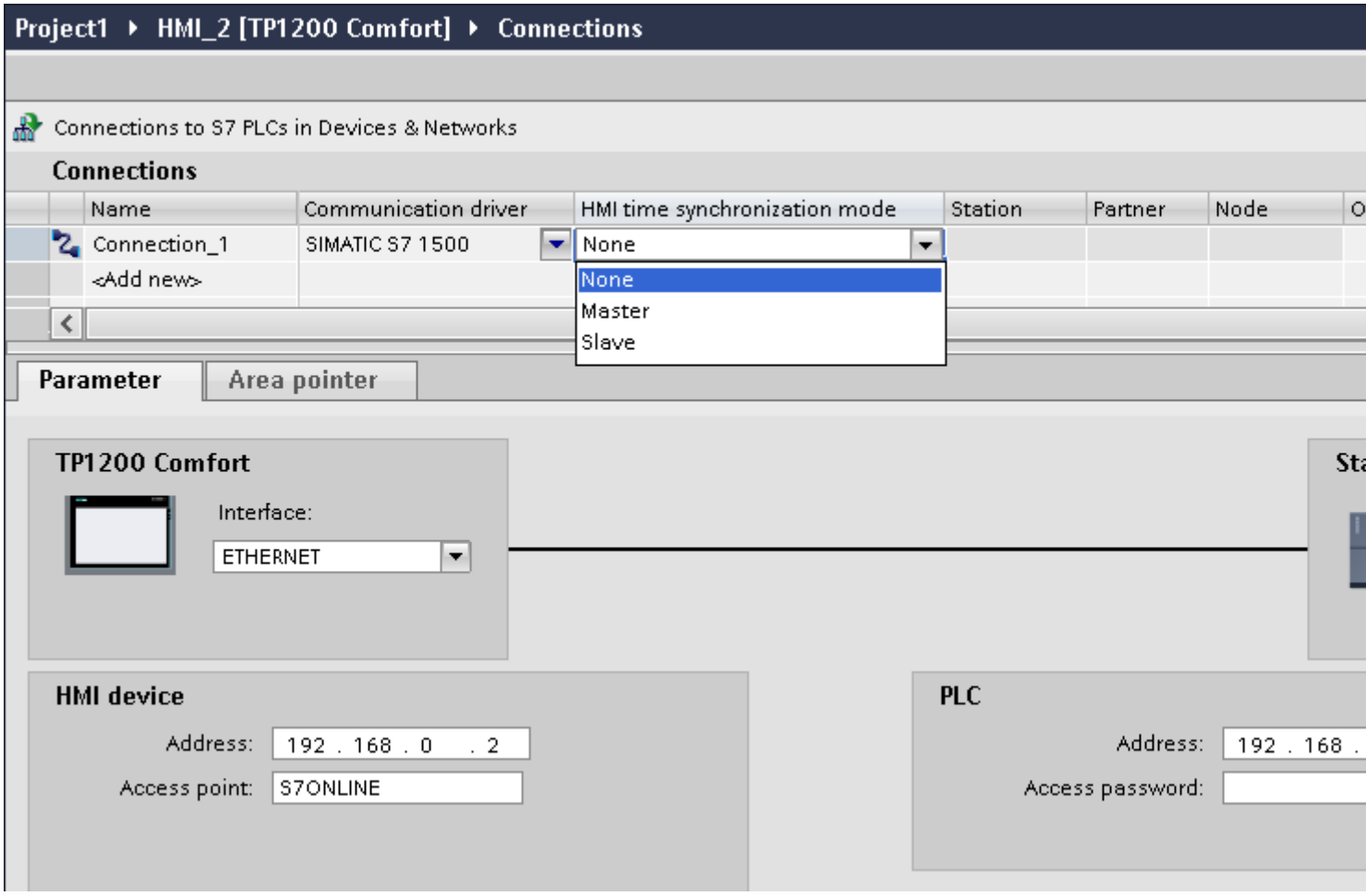
You configure time synchronization for a non-integrated connection in the "Connections" editor.

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

Communication with SIMATIC S7-1500 (Page 185)

Time synchronization (Page 286)

Time synchronization restrictions (Page 287)

Configuring time synchronization for integrated connections (Page 288)

2.6.9 Querying the PLC status

Introduction

In an S7 connection, you can query the status of the PLC using a special tag whose name corresponds to the schema "@<connection name>@OpStateConfiguredAddress".

In Runtime this tag is automatically updated every 5 seconds. The values correspond to the state of the PLC.

- 0: Not connected
- 4: Stop
- 8: Run

Querying the PLC status

To query the status of a PLC, proceed as follows:

1. Establish a connection with the communication driver "SIMATIC S7 1500" or "SIMATIC S7 1200". Assign the name "MyConnection" to this connection.
2. Create the tag "@MyConnection@OpStateConfiguredAddress" and assign the data type "UDInt" and the connection "MyConnection" to it.
3. Select "<Absolute access>" for the access mode.

Note

The configured address has no relevance in Runtime. However, it must correspond to a valid value so as not to cause an error when compiling the project.

2.6.10 Raw data tags of the "SIMATIC S7-1500" communication driver

2.6.10.1 Raw Data Tag

Definition

In WinCC, you can create external and internal tags of the type "Raw data type". The format and the length of a raw data tag are not fixed. Its length can be in the range from 1 and 65535 bytes. It is either defined by the user or results from a specific application.

The content of the raw data tags is not defined. Only senders and receivers can interpret the contents of a raw data tag. WinCC does not interpret them.

Note

A raw data tag cannot be displayed in screens.

Potential Applications within WinCC

Raw data tags can be used in the following areas in WinCC:

- For data exchange with the PLC alarm blocks for processing and acknowledgment of alarms
- In scripts for data exchange with the functions "Get/SetTagRaw"
- In tag logs
- To transfer jobs, data, processing acknowledgments between the HMI device and the PLC

Note

When the raw data tag is displayed in an I/O field, you must observe the conventions of the string with a "\0" character at the end.

"Properties address"

For external raw data tags, the "Address properties" dialog is not the same for all communication drivers because the parameters of the tag address and the supported raw data tag types depend on the type of communication being used.

Format adaptation

There is no format adaptation in WinCC for the "raw data type".

2.6.10.2 Raw data tags of the "SIMATIC S7-1500" communication driver

Introduction

A tag of the "Raw data type" is a data telegram of the ISO/OSI layer model. The contents of the raw data tags are not defined and therefore only the sender and the receiver can interpret the transmitted data. There is no format adaptation in WinCC for this data type. The maximum length is 65535 bytes.

WinCC distinguishes between two types of raw data tags:

- Raw data tag for free use
- Raw data tag for handling S7 functions

Raw data tag for free use

Raw data tags for free use are used for transferring user data blocks between the HMI device and the PLC and only contain user data.

The raw data tag is configured as byte array.

Raw data tag for handling S7 functions

These raw data tags do not have any connection-specific header and are normally used by the alarm system and for process data entry in WinCC.

No further description is needed here as these are tags and functions internal to the connection.

2.6.10.3 Configuring raw data tags for SIMATIC S7-1500

Introduction

The "SIMATIC S7-1500" communication driver supports the "Raw data tag" data type.

Raw data tags in the "SIMATIC S7-1500" communication driver

Raw data tags as byte array are used for transferring user data blocks between the HMI device and the PLC and only contain user data.

Only the acyclic read service of the controller is supported for raw data tags, e.g. the tag request via C scripts.

The "SIMATIC S7-1500" communication driver does not support cyclic read services for raw data tags.

Addressing the raw data tag

Raw data tags as byte array are handled in the same way as process tags in the communication driver; these are addressed based on the address and length of the data range (e.g. DB 1, DBB10, length 100 bytes).

Only "Byte" is possible for the "Raw data tag" data type of the WinCC tag. Except for the length of the raw data range, the parameters are preset and cannot be changed.

Length of the data blocks

Observe the maximum length of data blocks that can be sent by the communication driver:

- S7-1500: Data block length max. 8000 bytes

Exchanging large data volumes

The following application example shows you how to use raw data to transfer large amounts of data from the PLC to WinCC:

- Example - Data transfer (https://support.industry.siemens.com/cs/us/en/view/37873547)

2.6.10.4 Raw data tag as byte array

Introduction

Raw data tags as byte array are used for transferring user data blocks between the HMI device and the PLC and only contain user data.

Raw data tags as byte array are handled in the same way as process tags in the communication driver; these are addressed based on the address and length of the data range (e.g. DB 1, DBB10, length 100 bytes).

Note

The raw data length is limited to one transferable data block and must be fully transferable using a PDU (Protocol Data Unit). The PDU length is negotiated during connection establishment. The maximum length of the data block that can be transferred is based on the PDU length minus the header and additional information. The PDU lengths normally used in SIMATIC S7 result in the following maximum length:

- S7 1500: Data block length max. 8000 bytes

An incorrectly specified length leads to rejection of the read/write job with a display.

Data must be blocked if larger data blocks are to be transferred. This blocking is implemented by the S7 software in the PLC and by scripts in WinCC.

Configuring a raw data tag as byte array

The raw data tags for transferring data blocks are configured as raw data of the "Send/receive block" type with one address and one length detail.

The following figures show an example of a configuration for a data range that is 100 bytes long in the data block 100 as of data word 20:

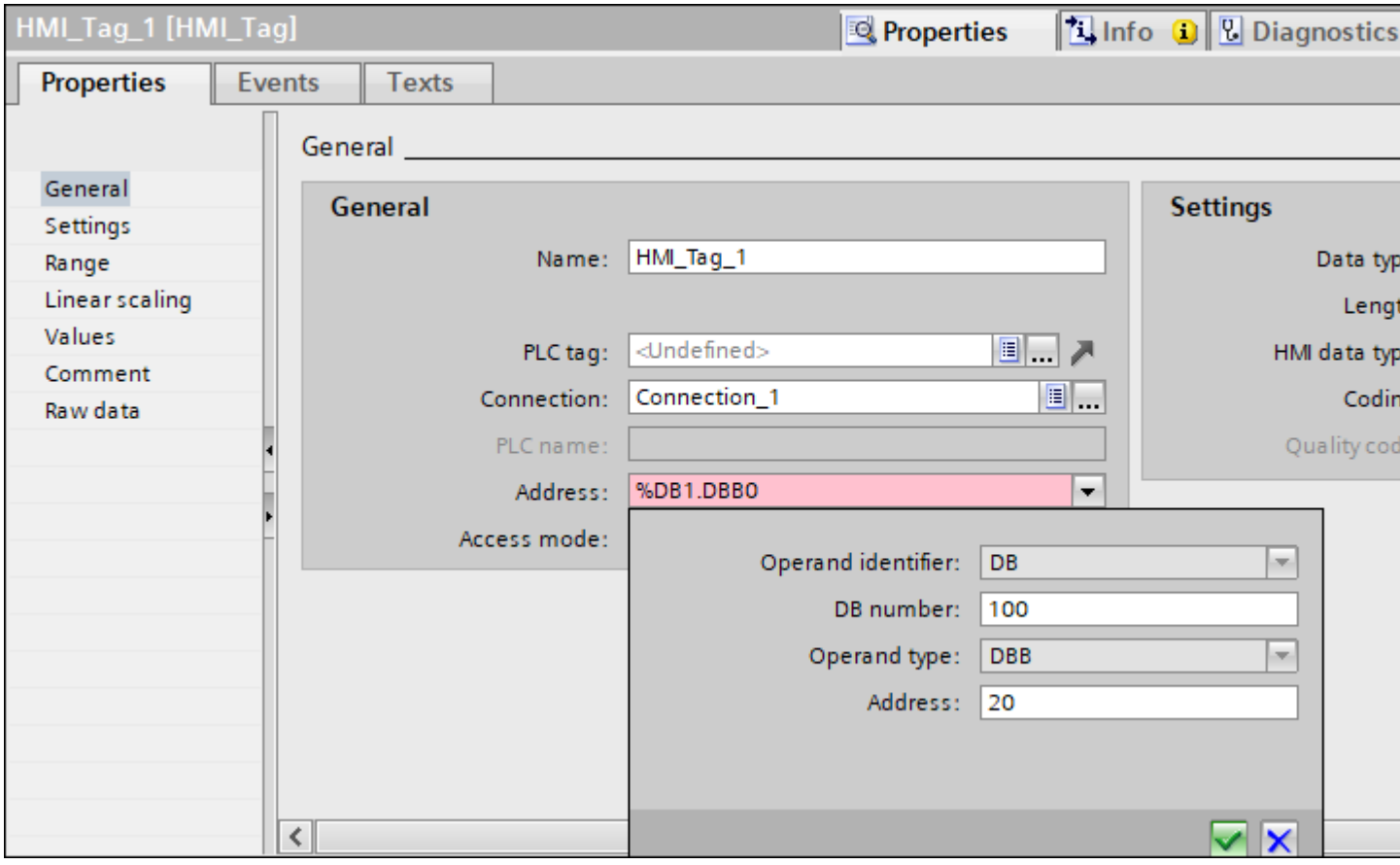


Figure 2-1 Raw data address_set

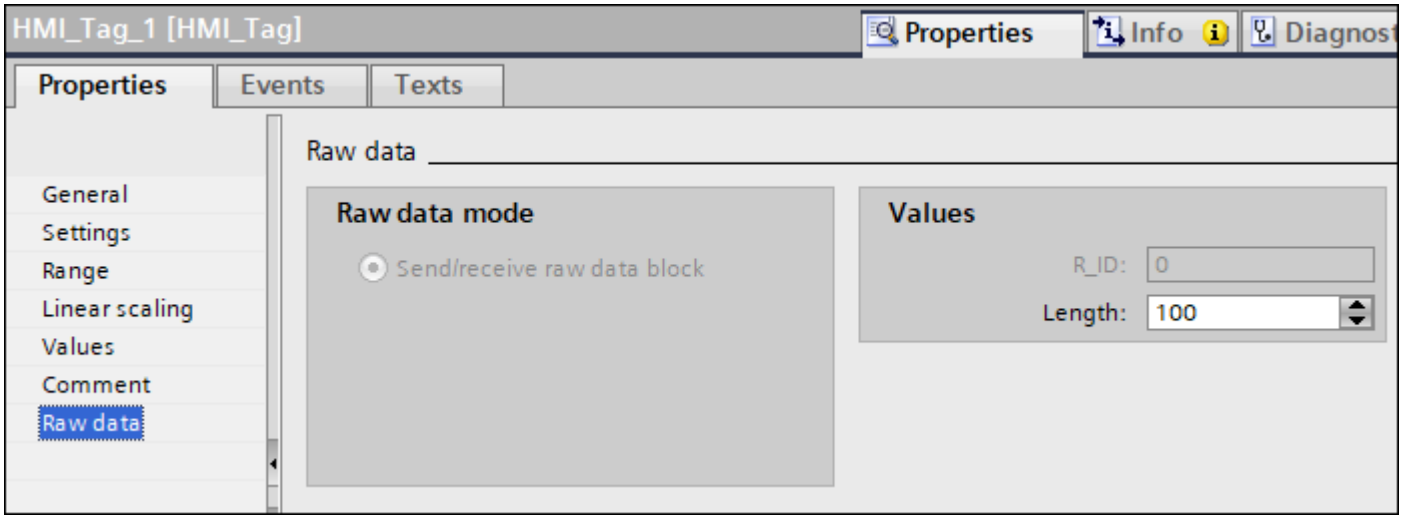


Figure 2-2 Raw data mode "Send/receive raw data block"

Read a Raw data tag as Byte Array

This raw data tag is read in the same way as a process tag. The corresponding data block is requested from the PLC and transferred to the HMI device after the data has been received.

Data is transferred always at the initiative of WinCC. Sporadic or event-driven data receipt initiated by the PLC is not possible using this raw data tag.

Write a Raw data tag as Byte Array

This raw data tag is written in the same way as a process tag. After sending the data block and receiving a positive acknowledgement from the PLC, the data block is transferred to the tag image in the HMI device.

2.6.10.5 How to configure a raw data tag as byte array

Introduction

In this section, you will learn based on an example how a raw data tag of the "SIMATIC S7-1500" communication driver is configured as a byte array.

The configuration is the same for all interfaces of the communication driver.

Procedure

1. Double-click "Connections" in the project navigation and create a connection.
2. Select the "SIMATIC S7-1500" communication driver and the "PROFIBUS" or "TCP/IP" interface.

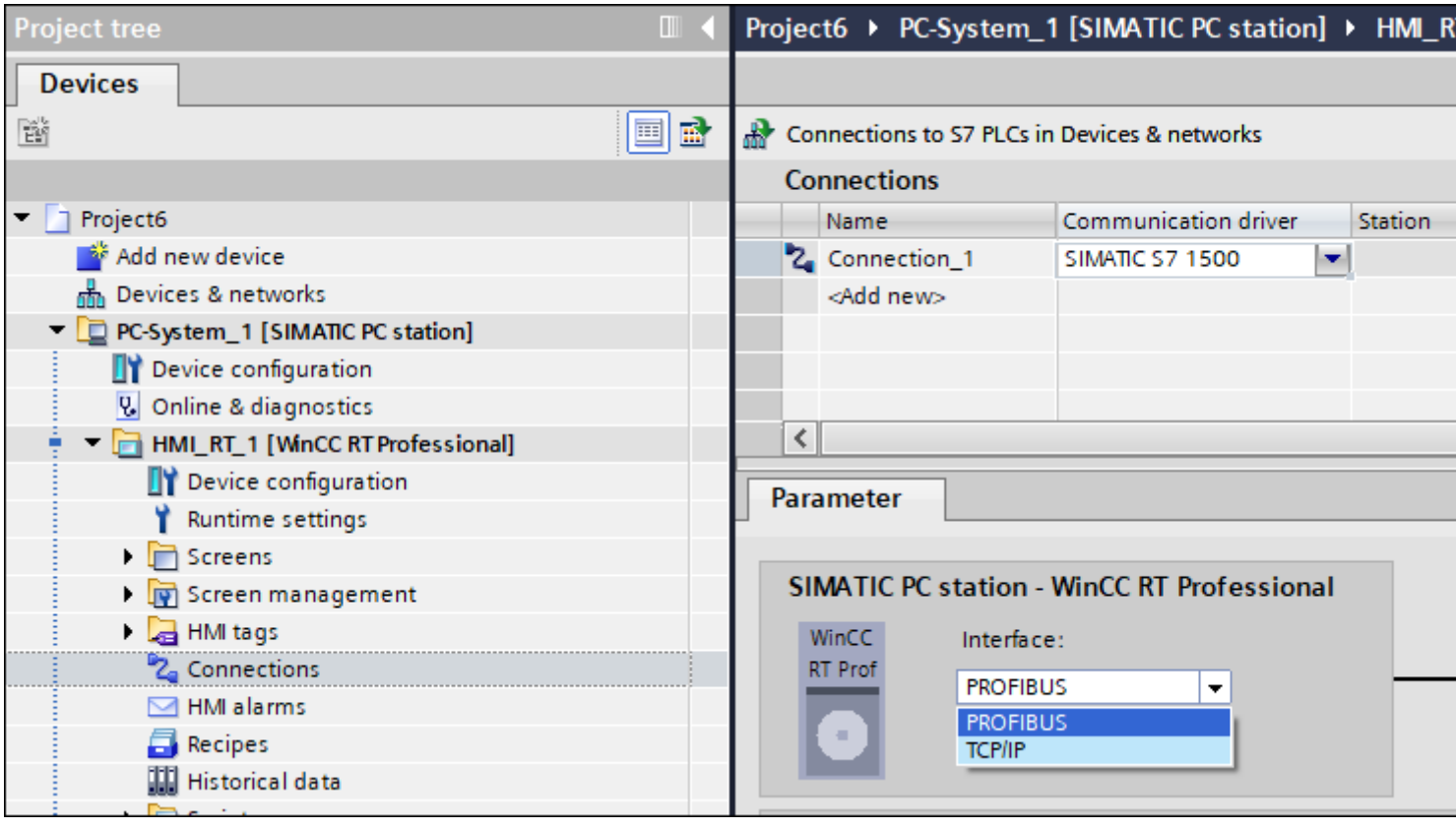


Figure 2-3 Set communication driver and interface

3. Double-click "HMI tags > Show all tags" in the project navigation and create an HMI tag under "HMI tags".
4. Select the data type "Raw" in the "Data type" field.
5. Select "Connection_1" in the "Connection" field and then confirm the selection.
6. Select "Properties > Properties > General" in the Inspector window.

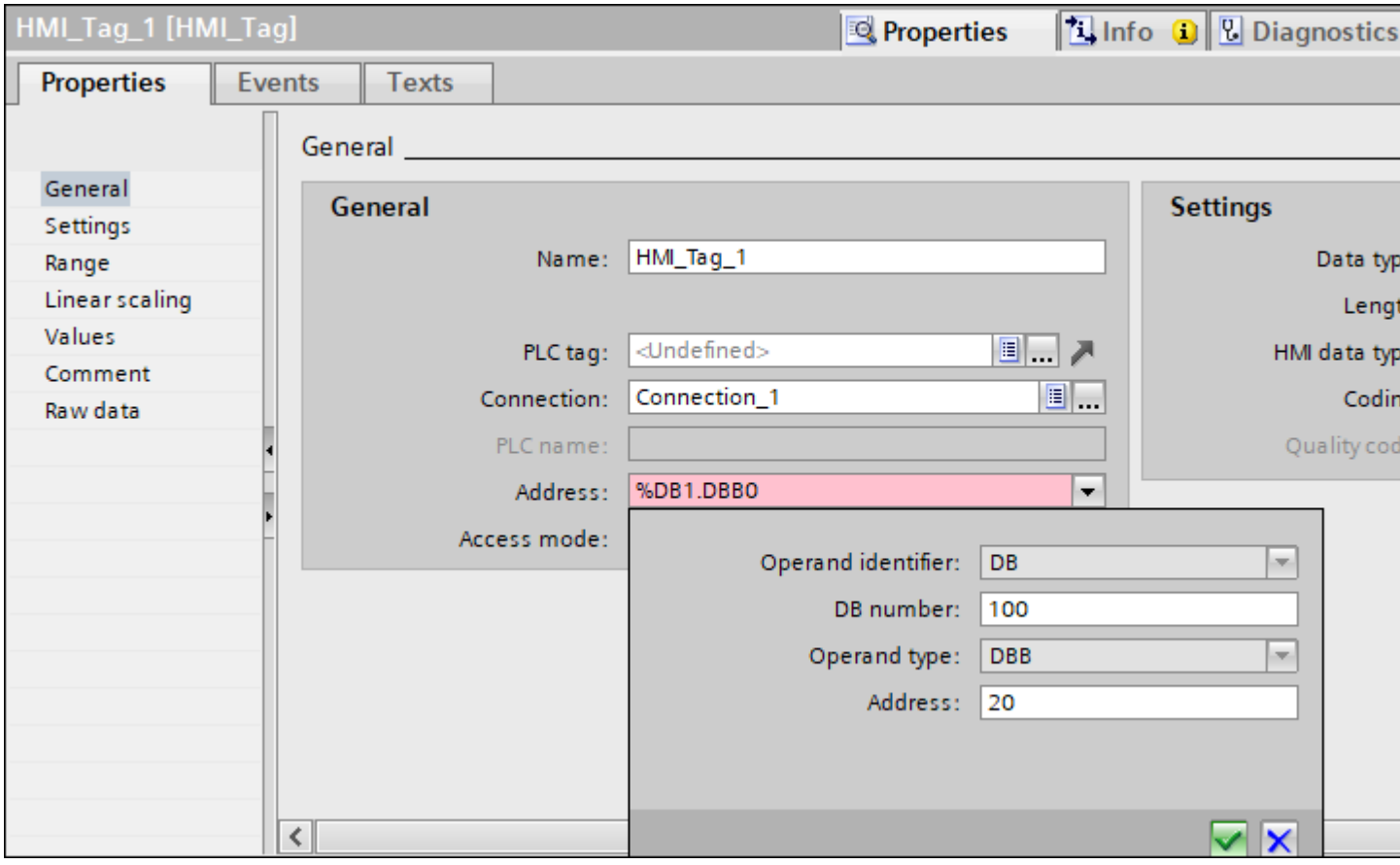


Figure 2-4 Raw data address_set

7. Enter the length of the raw data block (in bytes) in the Length field.
8. Enter the number of the data block.

9. Enter the value of the start address.
10. The type "Send/receive raw data block" in the "Raw data mode" is automatically set (in the Inspector window under "Properties > Properties > Raw data").

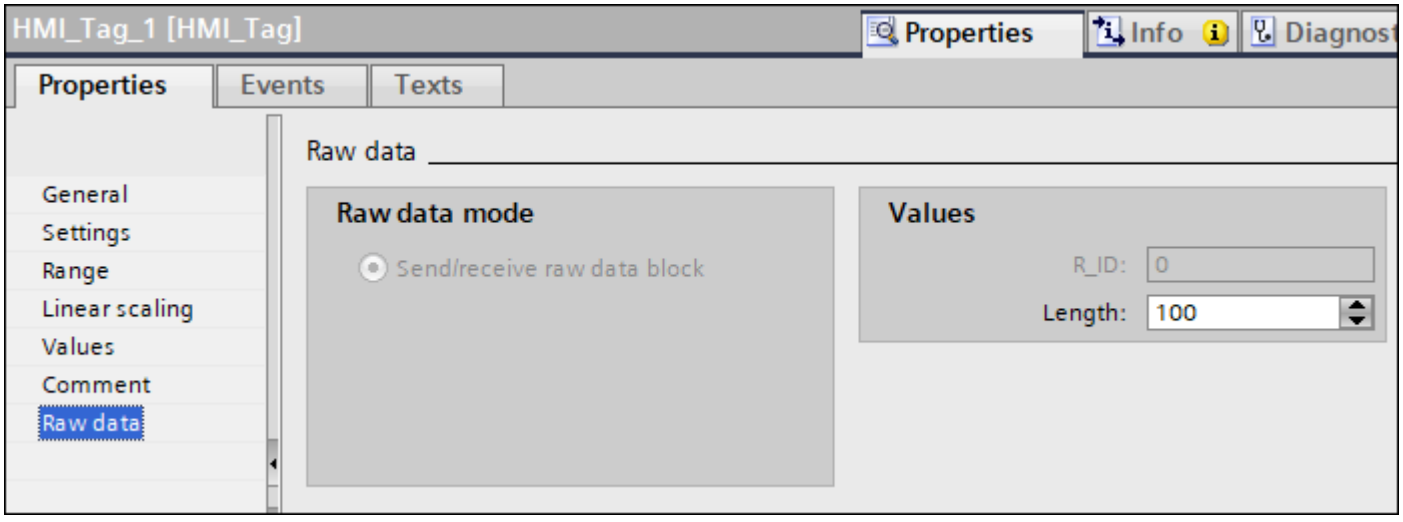


Figure 2-5 Raw data mode

Note

The raw data length is limited to one transferable data block and must be fully transferable using a PDU (Protocol Data Unit). The PDU length is negotiated during connection establishment. The maximum length of the data block that can be transferred is based on the PDU length minus the header and additional information. The PDU lengths normally used in SIMATIC S7 result in the following maximum length:

- S7-1500: Data block length max. 8000 bytes

An incorrectly specified length leads to rejection of the read/write job with a display.

2.7 Communicating with SIMATIC S7 1200

2.7.1 Communication with SIMATIC S7-1200

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.

Note

HMI connections to SIMATIC S7-1200 PLCs with firmware versions earlier than V2.0 are not possible.

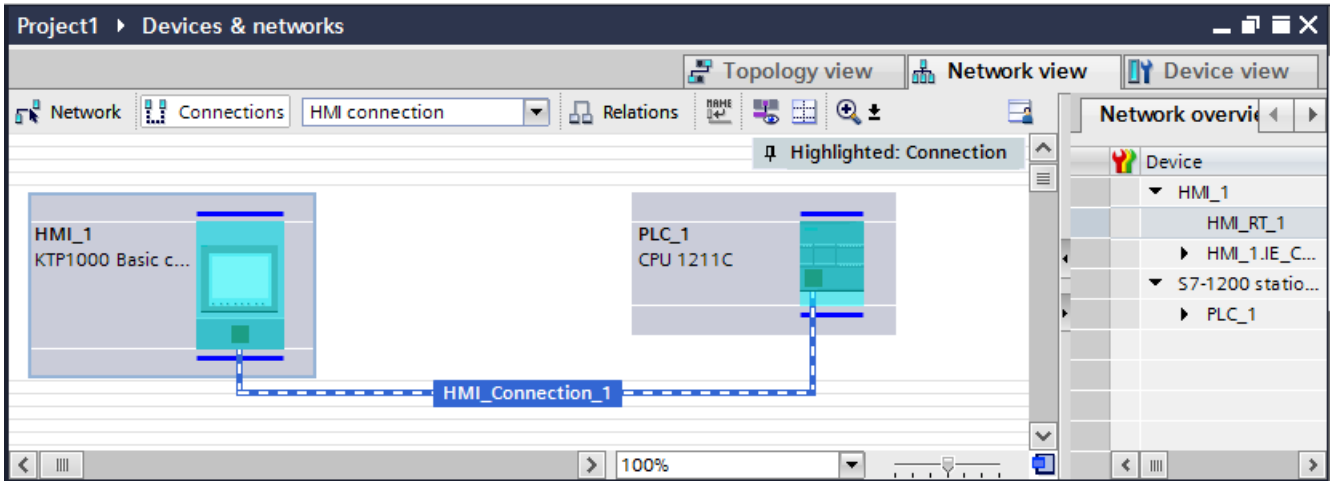
2.7.2 Communication via PROFINET

2.7.2.1 Configuring an HMI connection

Communication via PROFINET

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 298)

Configuring an HMI connection via PROFINET (Page 300)

Configuring an HMI connection via PROFINET

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

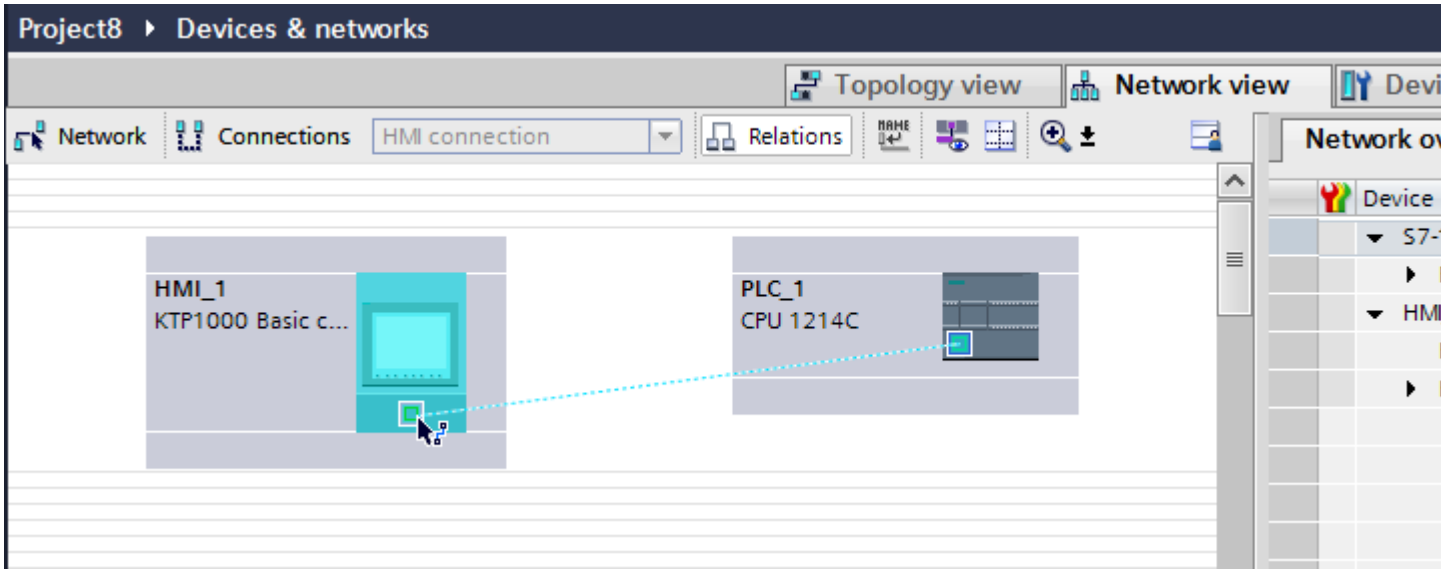
The following communication partners are created in the "Devices & Networks" editor:

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

Procedure

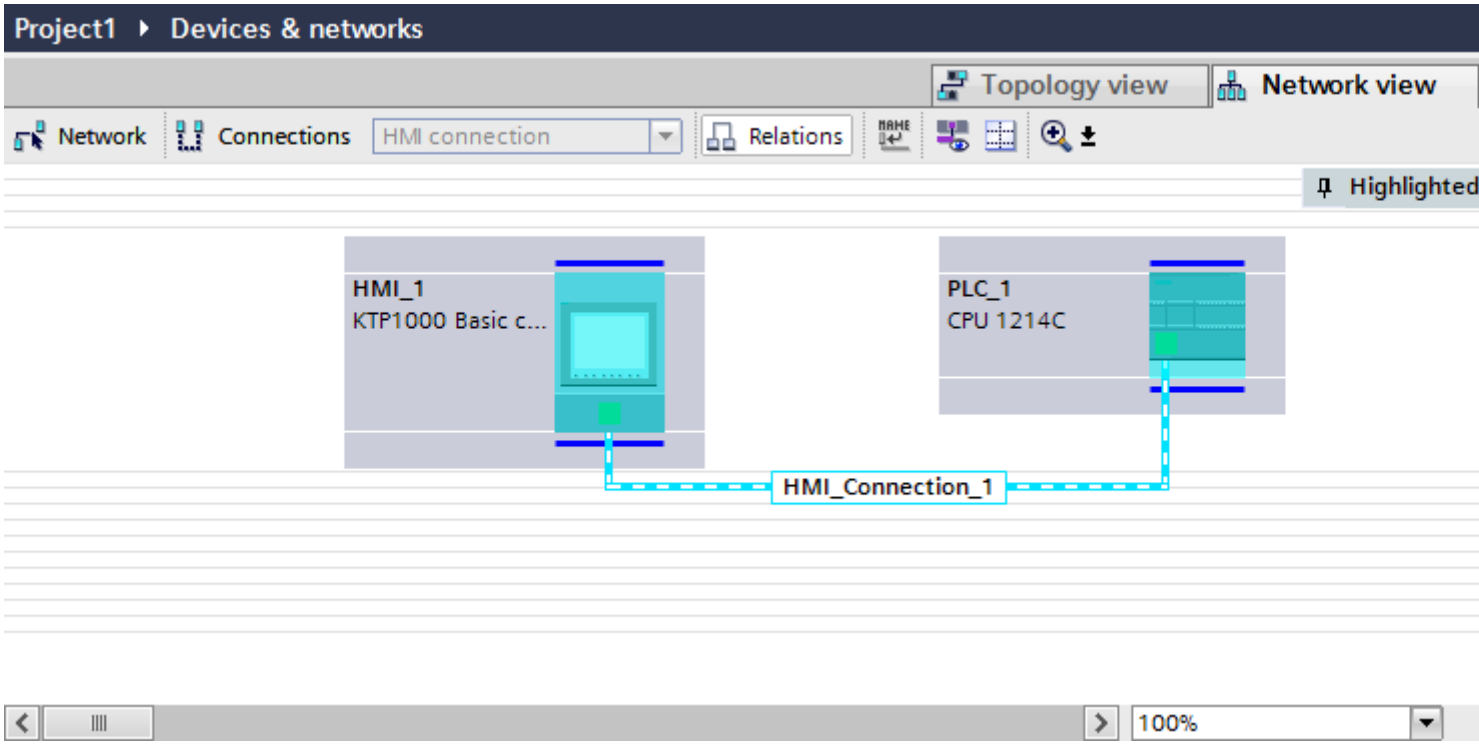
1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed graphically in the network view.
2. Click the "Connections" button and select "HMI connection" for the connection type.
   The devices available for connection are highlighted 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 310)".

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 298)

Communication via PROFINET (Page 299)

PROFINET parameters (Page 310)

2.7.2.2 Configuring an HMI connection

Communication via PROFINET

Communication via PROFINET

This section describes the communication between a WinCC Runtime and the SIMATIC S7 1200 PLC via PROFINET.

You can use the following WinCC Runtimes as an HMI device:

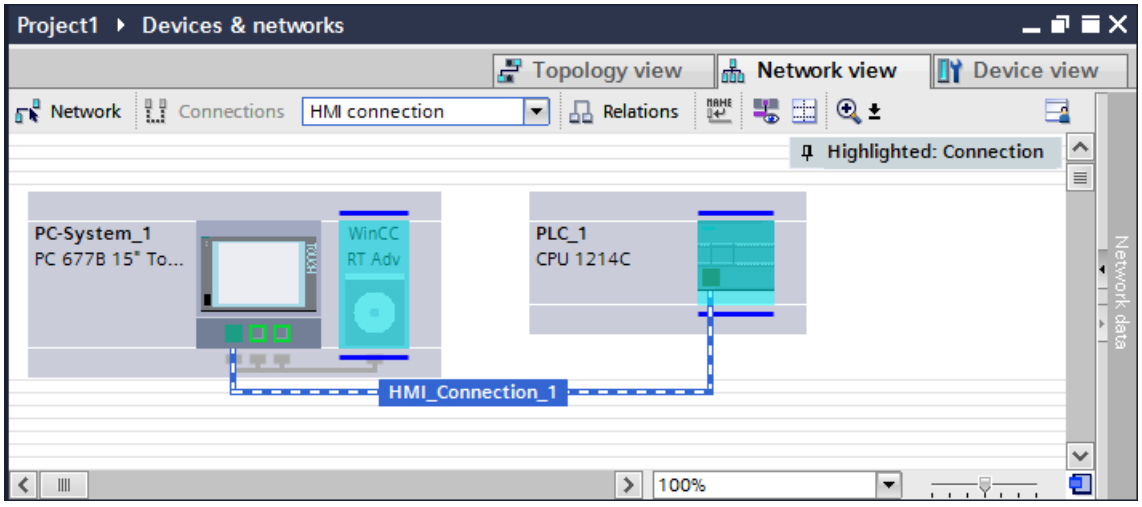
- WinCC RT Advanced
- WinCC RT Professional

WinCC Runtime as HMI device

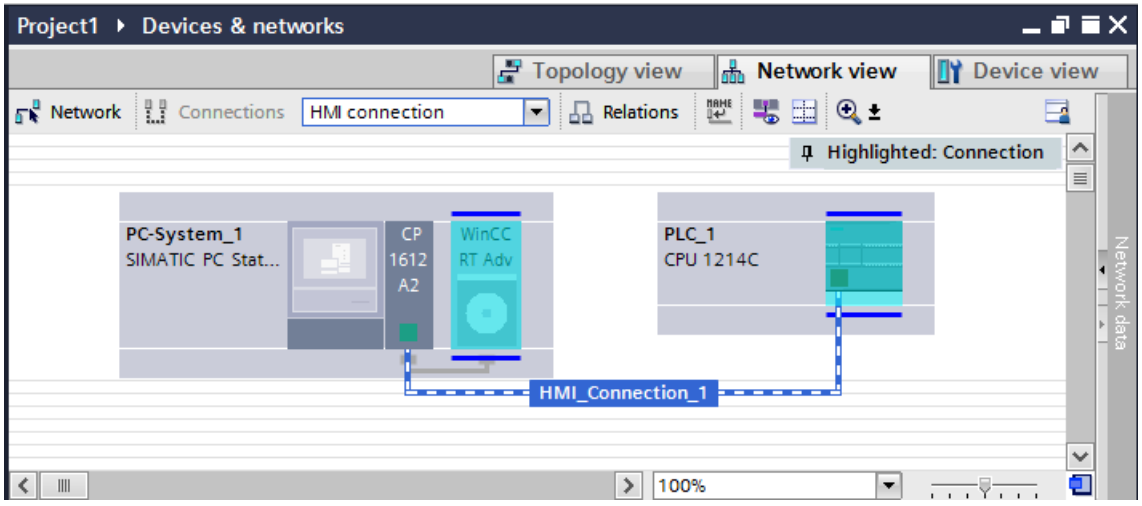
You configure the HMI connections between WinCC Runtime and SIMATIC S7 1200 in the "Devices & Networks" editor.

If you want to use a WinCC Runtime as an HMI device, you have the following options:

1. You create a SIMATIC PC and configure a WinCC Runtime on this SIMATIC PC.
   In this way you use the SIMATIC PC with a WinCC Runtime as an HMI device.



2. You create a WinCC Runtime and configure a communication processor to the Runtime.
   In this way you use your configuration PC with a WinCC Runtime as an HMI device.



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 298)

Configuring an HMI connection via PROFINET with a SIMATIC PC (Page 305)

Configuring an HMI connection via PROFINET with a PC (Page 307)

Configuring an HMI connection via PROFINET with a SIMATIC PC

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.

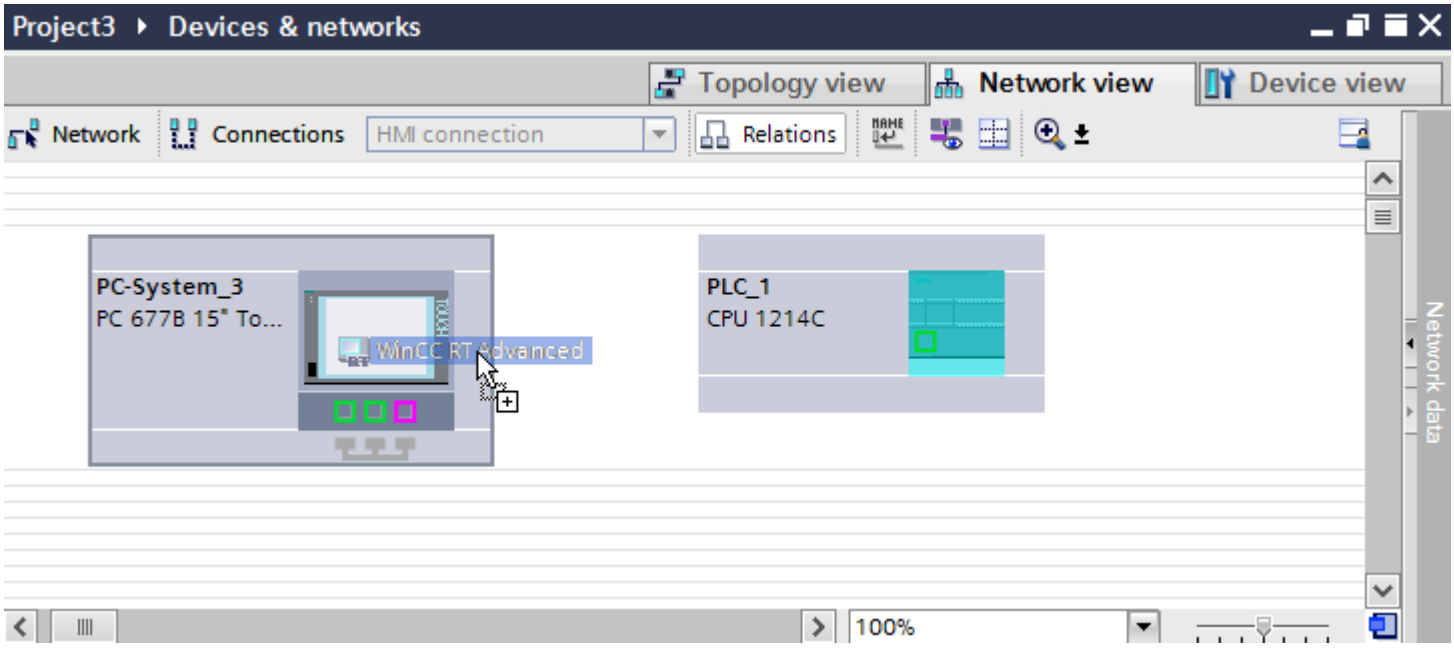
Requirements

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

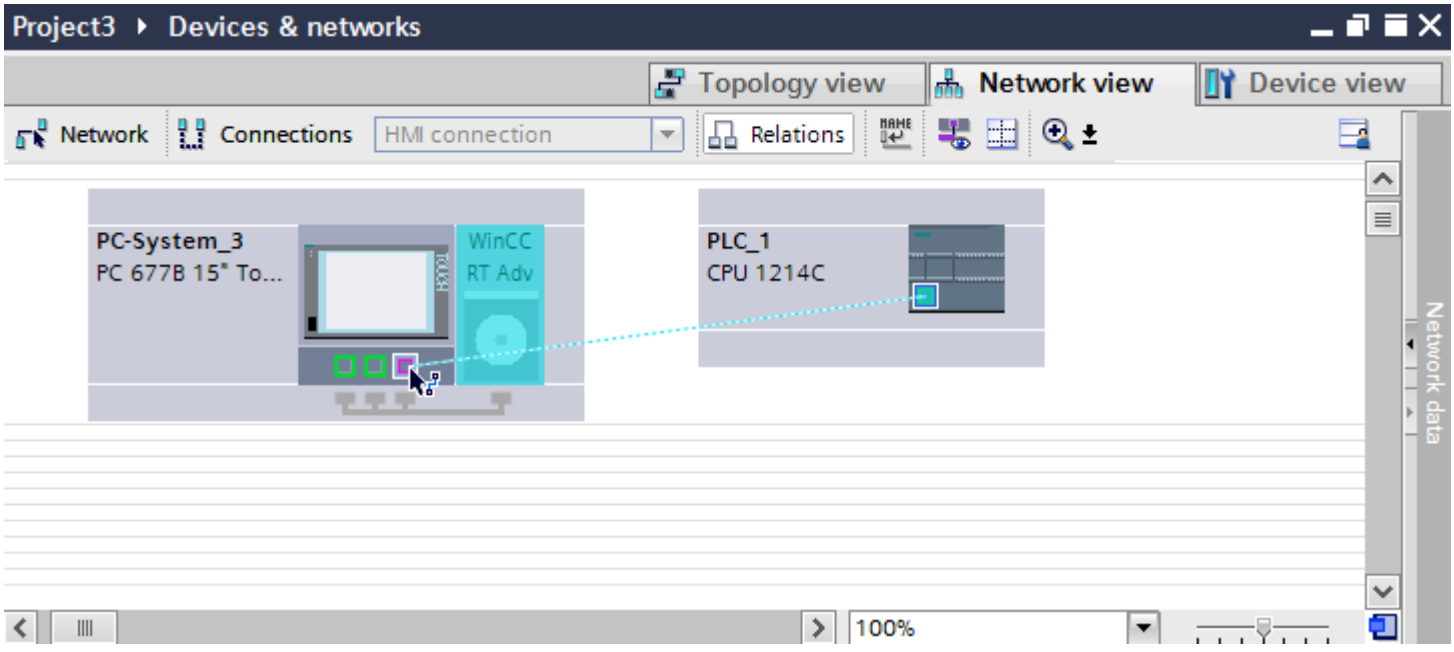
- SIMATIC S7 1200
- SIMATIC PC with PROFINET interface

Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed in the network view.
2. Use a drag-and-drop operation to move a WinCC Runtime from the hardware catalog to the PC.



3. Click the "Connections" button and select "HMI connection" for the connection type.
   The devices available for connection are highlighted in color.
4. Click the PROFINET interface of the PLC and use a drag-and-drop operation to draw a connection to the PROFINET interface of the PC.



5. Click the connecting line.
6. Click "Highlight HMI connection" and select the HMI connection.
   The connection is displayed graphically in the Inspector window.
7. 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 310)".

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 298)

Communication via PROFINET (Page 303)

Configuring an HMI connection via PROFINET with a PC (Page 307)

PROFINET parameters (Page 310)

Configuring an HMI connection via PROFINET with a PC

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.

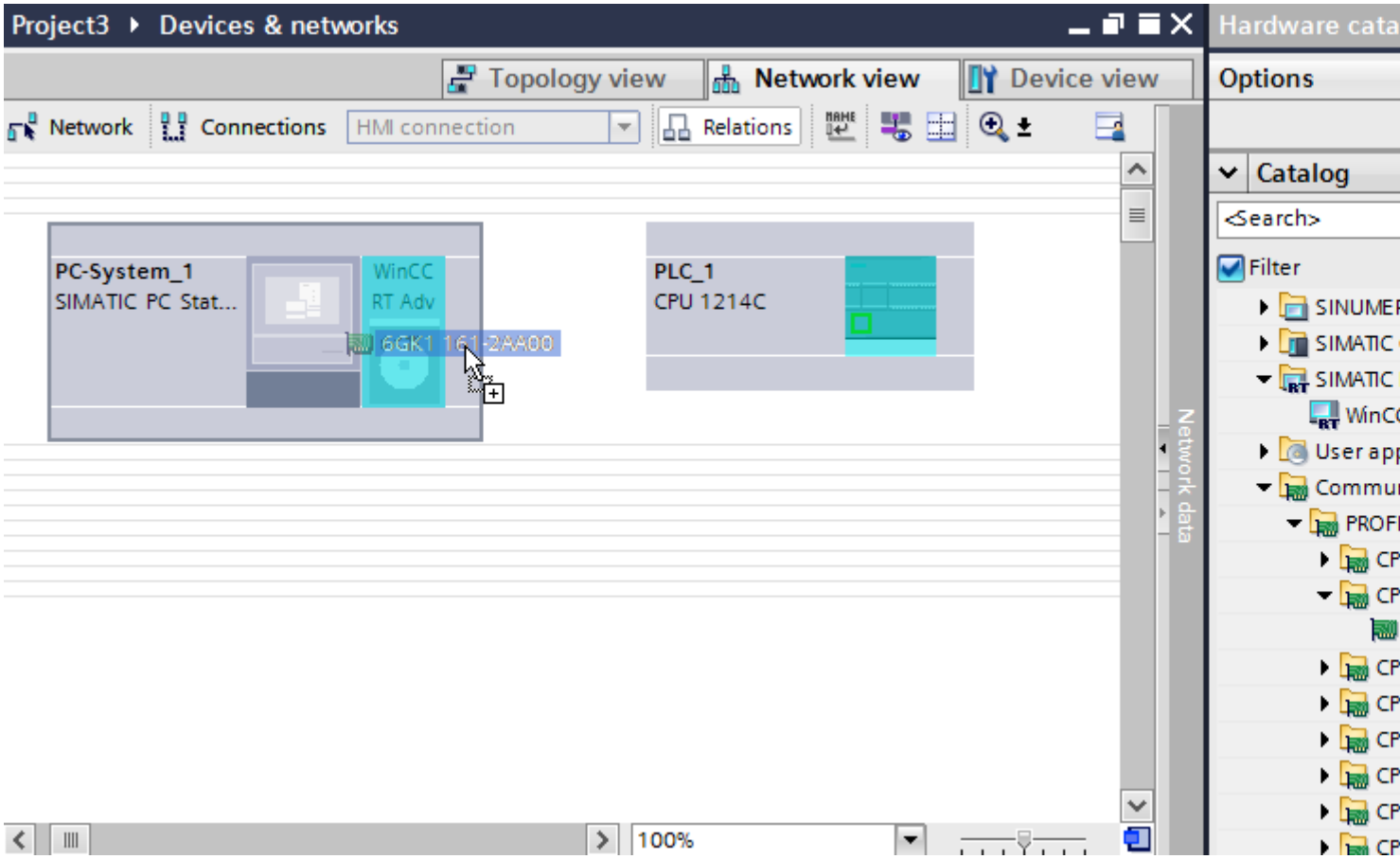
Requirements

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

- SIMATIC S7 1200
- PC station with WinCC RT Advanced or WinCC RT Professional

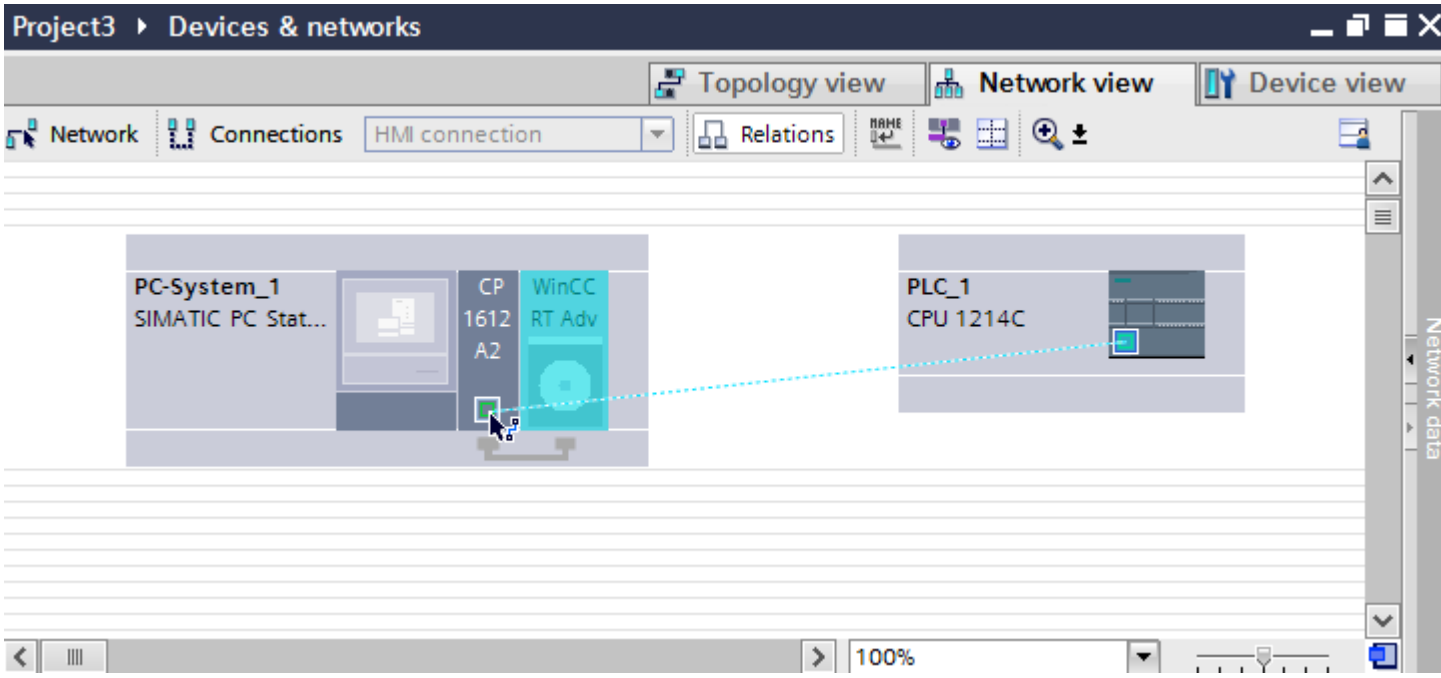
Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed in the network view.
2. Use a drag-and-drop operation to move a PROFINET-capable communication processor from the hardware catalog to the WinCC Runtime.



3. Click the "Connections" button and select "HMI connection" for the connection type.
   The devices available for connection are highlighted in color.

4. Click the PROFINET interface of the PLC and use a drag-and-drop operation to draw a connection to the PROFINET interface of the communication processor.



5. Click the connecting line.
6. Click "Highlight HMI connection" and select the HMI connection.
   The connection is displayed graphically in the Inspector window.
7. 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 310)".

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 298)

Communication via PROFINET (Page 303)

Configuring an HMI connection via PROFINET with a SIMATIC PC (Page 305)

PROFINET parameters (Page 310)

2.7.2.3 PROFINET parameters

PROFINET parameters for the HMI connection

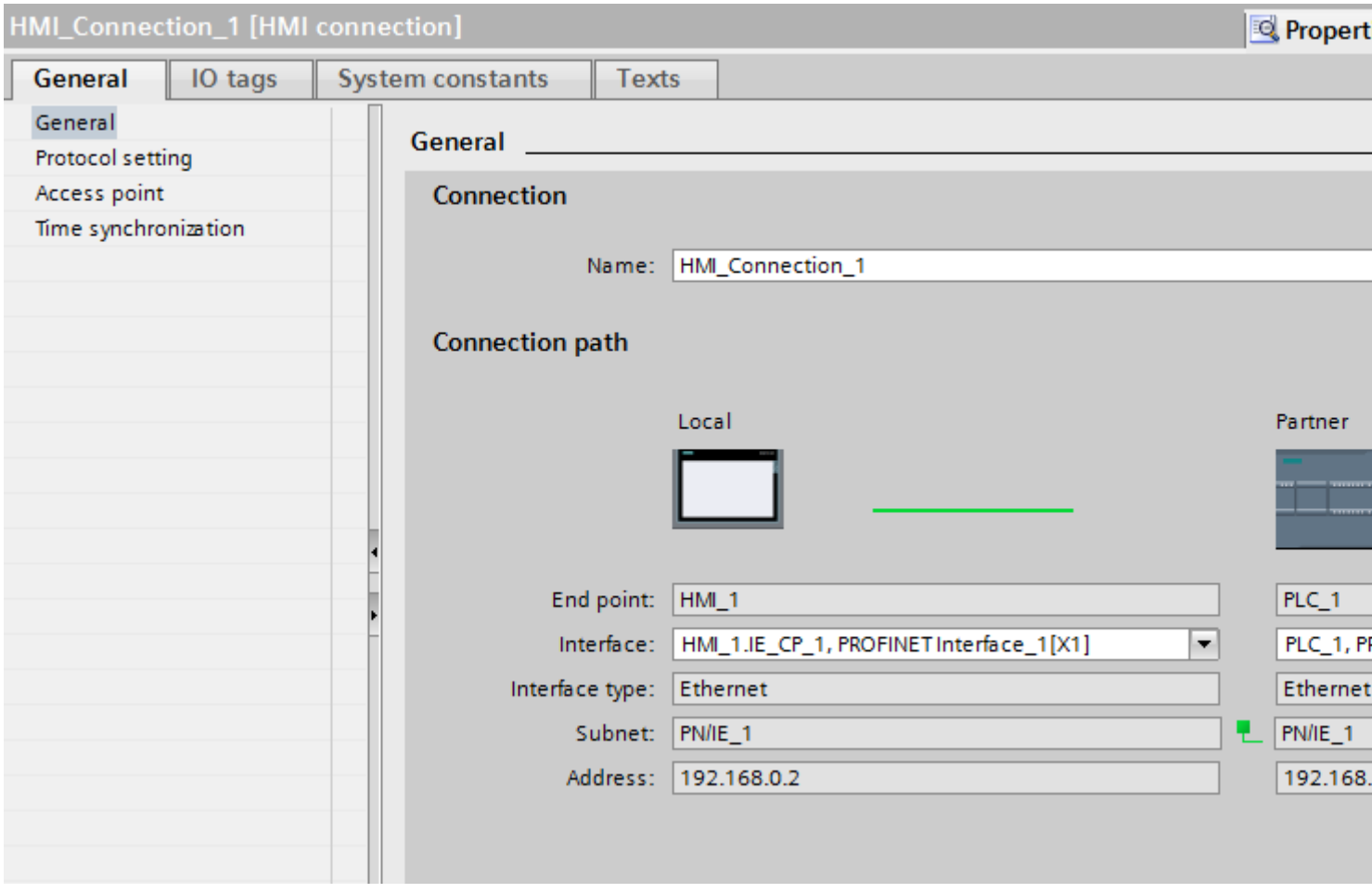
PROFINET parameters for the HMI connection

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 with SIMATIC S7-1200 (Page 298)

PROFINET parameters for the HMI device (Page 311)

PROFINET parameters for the PLC (Page 313)

Configuring Industrial Ethernet (Page 315)

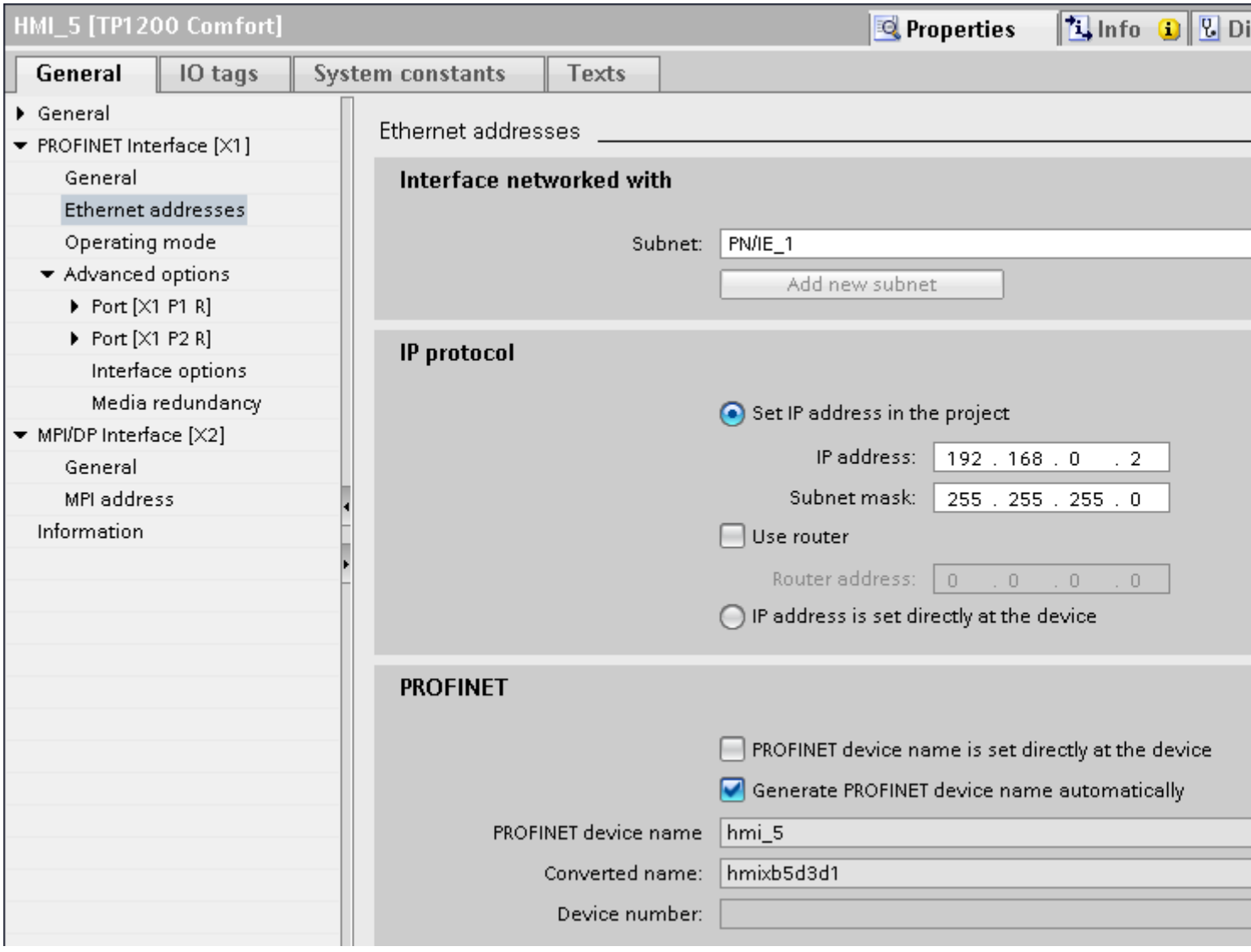
PROFINET parameters for the HMI device

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.
- "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 S7-1200 (Page 298)

PROFINET parameters for the HMI connection (Page 310)

PROFINET parameters for the PLC (Page 313)

Configuring Industrial Ethernet (Page 315)

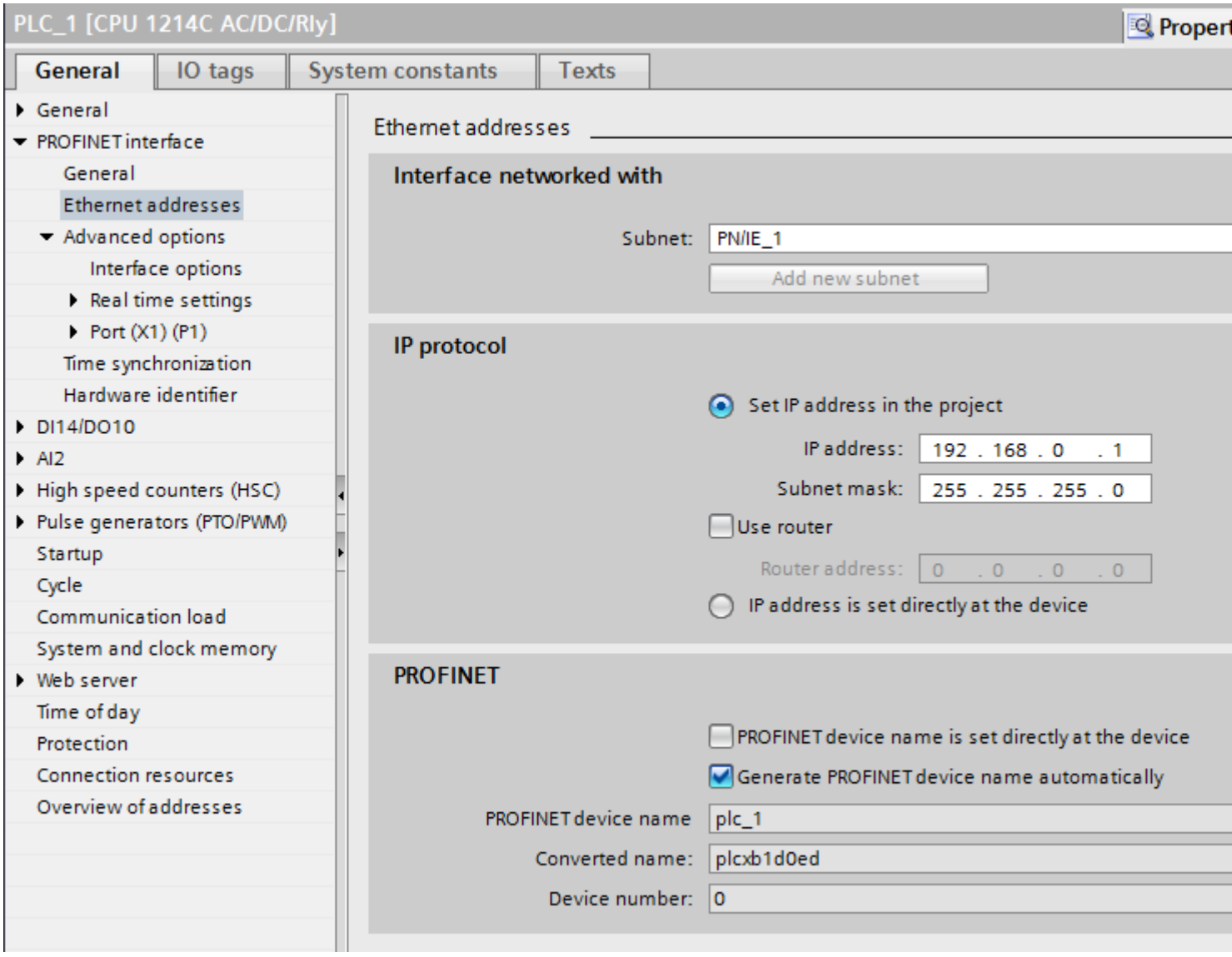
PROFINET parameters for the PLC

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 S7-1200 (Page 298)

PROFINET parameters for the HMI connection (Page 310)

PROFINET parameters for the HMI device (Page 311)

Configuring Industrial Ethernet (Page 315)

Configuring Industrial Ethernet

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 with SIMATIC S7-1200 (Page 298)

PROFINET parameters for the HMI connection (Page 310)

PROFINET parameters for the HMI device (Page 311)

PROFINET parameters for the PLC (Page 313)

Communication security

Security levels

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 318)

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

Communication with SIMATIC S7-1200 (Page 298)

Setting options for the protection (FW as of V4) (Page 318)

PROFINET parameters for the HMI connection (Page 310)

Access password for the HMI connection (Page 321)

Setting options for the protection (FW as of V4)

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

Communication with SIMATIC S7-1200 (Page 298)

PROFINET parameters for the HMI connection (Page 310)

Setting options for the protection level (FW V1 to V3) (Page 320)

Access password for the HMI connection (Page 321)

Security levels (Page 317)

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

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

Communication with SIMATIC S7-1200 (Page 298)

PROFINET parameters for the HMI connection (Page 310)

Setting options for the protection (FW as of V4) (Page 318)

Access password for the HMI connection (Page 321)

Access password for the HMI connection

Introduction

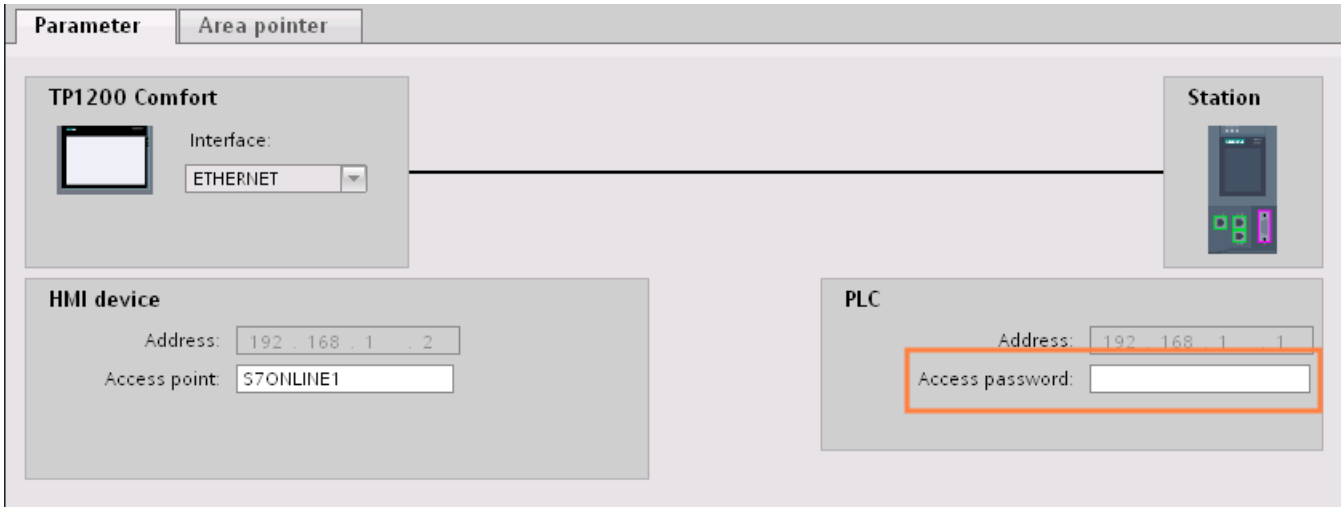
Communication with a PLC with the"Complete protection" protection level is protected by a password. The password is stored in the properties of the PLC.

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

Communication with SIMATIC S7-1200 (Page 298)

PROFINET parameters for the HMI connection (Page 310)

Setting options for the protection (FW as of V4) (Page 318)

Setting options for the protection level (FW V1 to V3) (Page 320)

Security levels (Page 317)

2.7.2.4 Setting port options

Setting the port options

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

Communication with SIMATIC S7-1200 (Page 298)

Wiring rules for disabled autonegotiation (Page 324)

Boundaries at the port (Page 325)

Wiring rules for disabled autonegotiation

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:

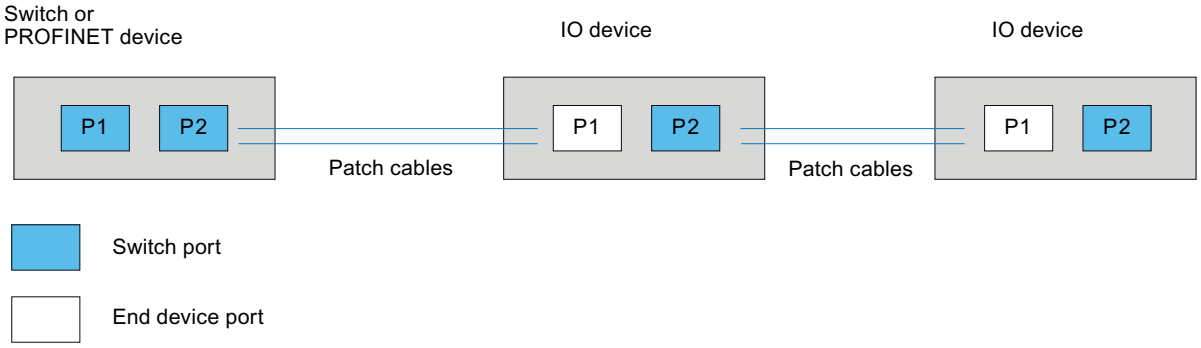
Type of port	PROFINET devices	Note
Switch port with crossed pin assignment	For IO devices: Port 2 For S7 CPUs with 2 ports: Ports 1 and 2	Crossed pin assignment means that the pin assignment for the ports for sending and receiving between the respective PROFINET devices is exchanged internally.
End device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

Communication with SIMATIC S7-1200 (Page 298)

Setting the port options (Page 322)

Boundaries at the port (Page 325)

Boundaries at the port

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

Communication with SIMATIC S7-1200 (Page 298)

Setting the port options (Page 322)

Wiring rules for disabled autonegotiation (Page 324)

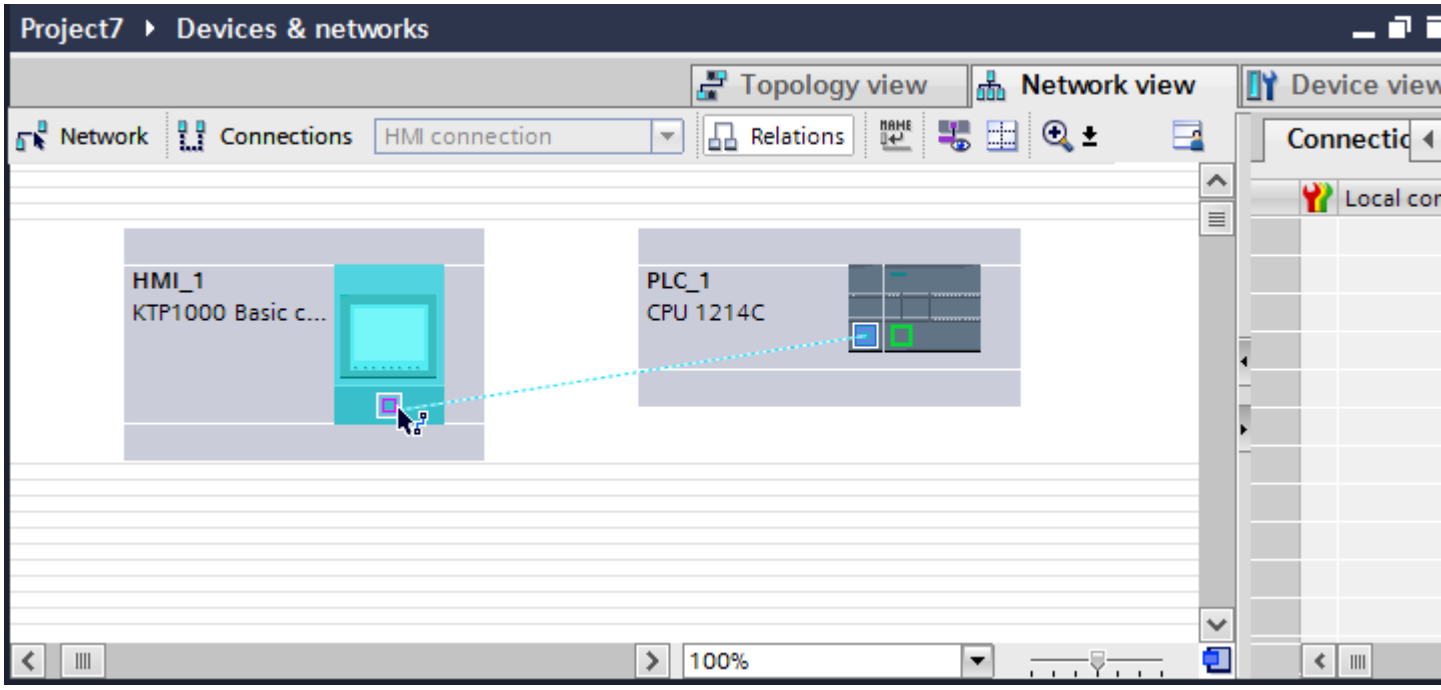
2.7.3 Communication via PROFIBUS

2.7.3.1 Configuring an HMI connection

Communication via PROFIBUS

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 298)

Configuring an HMI connection via PROFIBUS (Page 328)

Configuring an HMI connection via PROFIBUS

Introduction

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

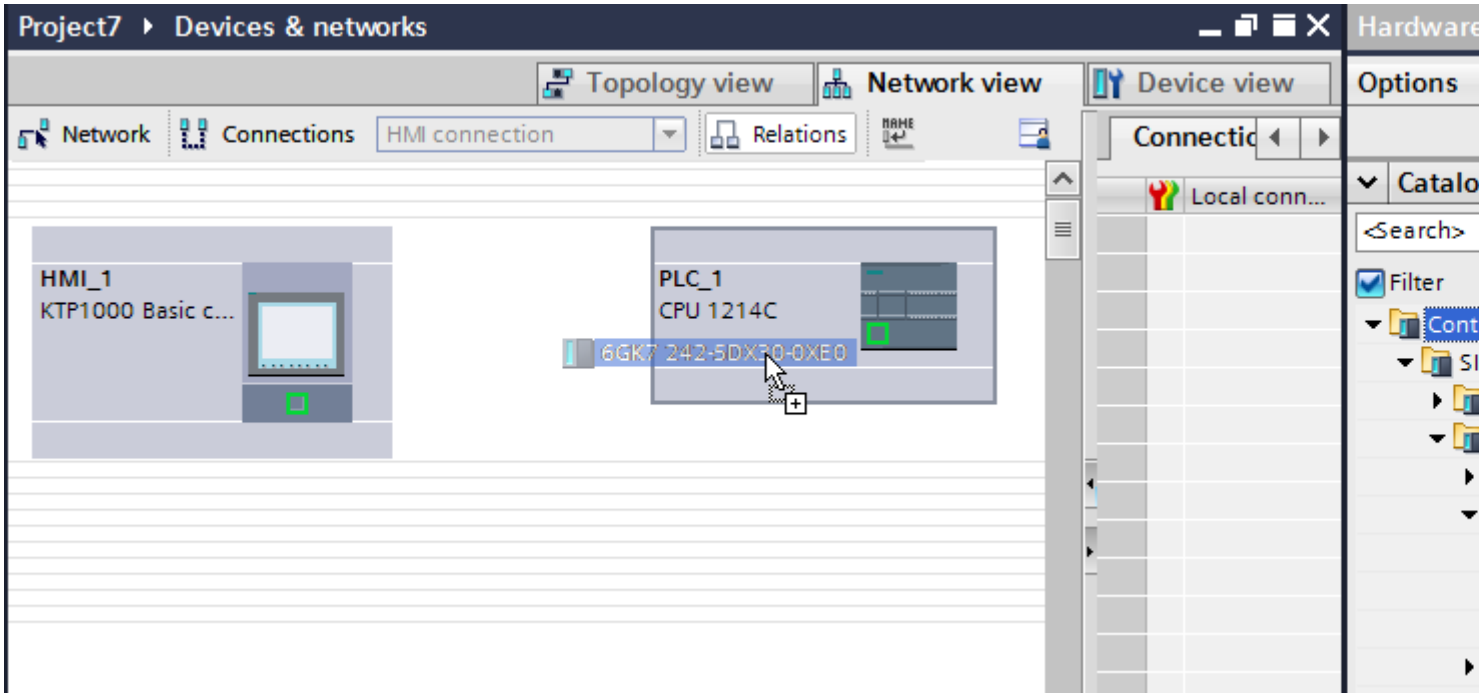
Requirements

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

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

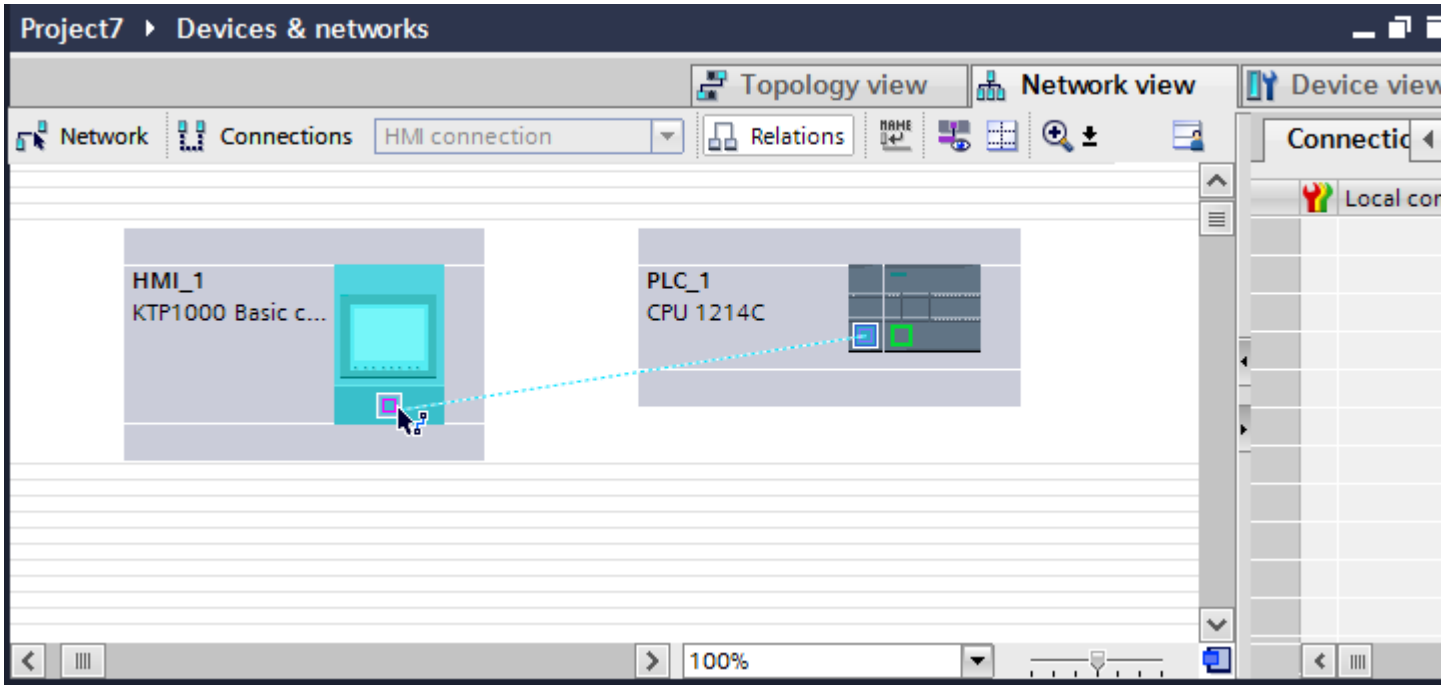
Procedure

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 339)" 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 298)

Communication via PROFIBUS (Page 327)

PROFIBUS parameters (Page 339)

2.7.3.2 Configuring an HMI connection

Communication via PROFIBUS

Communication via PROFIBUS

This section describes the communication between a WinCC Runtime and the SIMATIC S7 1200 PLC via PROFIBUS.

You can use the following WinCC Runtimes as an HMI device:

- WinCC RT Advanced
- WinCC RT Professional

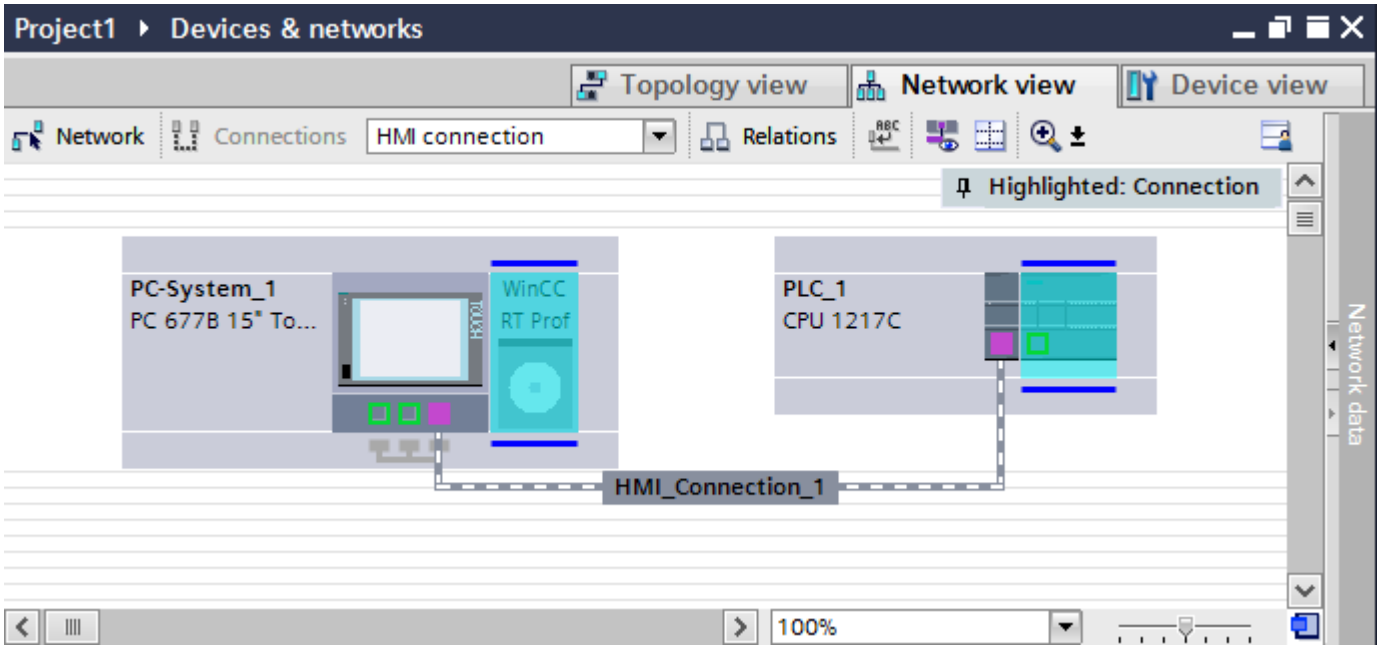
If you want to connect a SIMATIC S7 1200 to an HMI device via PROFIBUS, you must configure a PROFIBUS-capable communication module in a slot of the PLC.

WinCC Runtime as HMI device

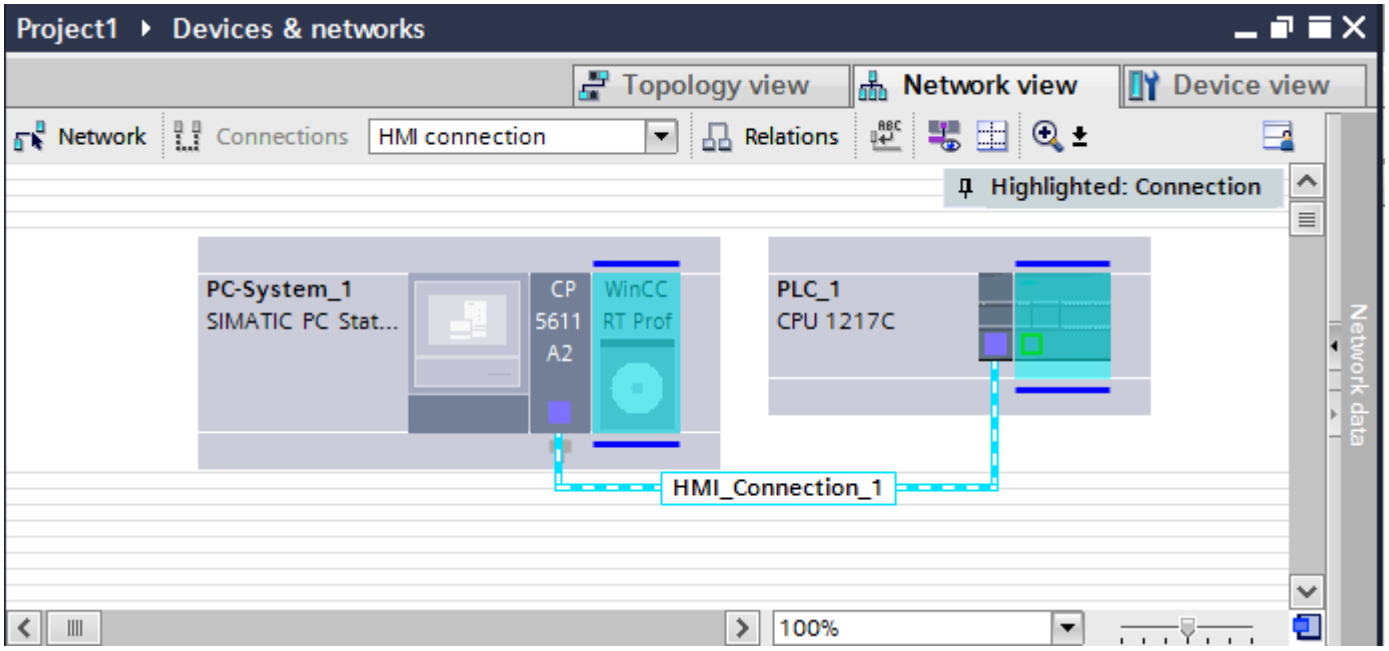
You configure the HMI connections between a WinCC Runtime and SIMATIC S7 1200 in the "Devices & Networks" editor.

If you want to use a WinCC Runtime as an HMI device, you have the following options:

1. You create a SIMATIC PC and configure a WinCC Runtime on this SIMATIC PC.
   In this way you use the SIMATIC PC with a WinCC Runtime as an HMI device.



2. You create a WinCC Runtime and configure a communication processor to the Runtime.
   In this way you use your configuration PC with a WinCC Runtime as an HMI device.



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 PROFIBUS in the "Devices & Networks" editor.

Connection in the "Connections" editor

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

See also

Communication with SIMATIC S7-1200 (Page 298)

Configuring an HMI connection via PROFIBUS with a SIMATIC PC (Page 332)

Configuring an HMI connection via PROFIBUS with a PC (Page 335)

Configuring an HMI connection via PROFIBUS with a SIMATIC PC

Introduction

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

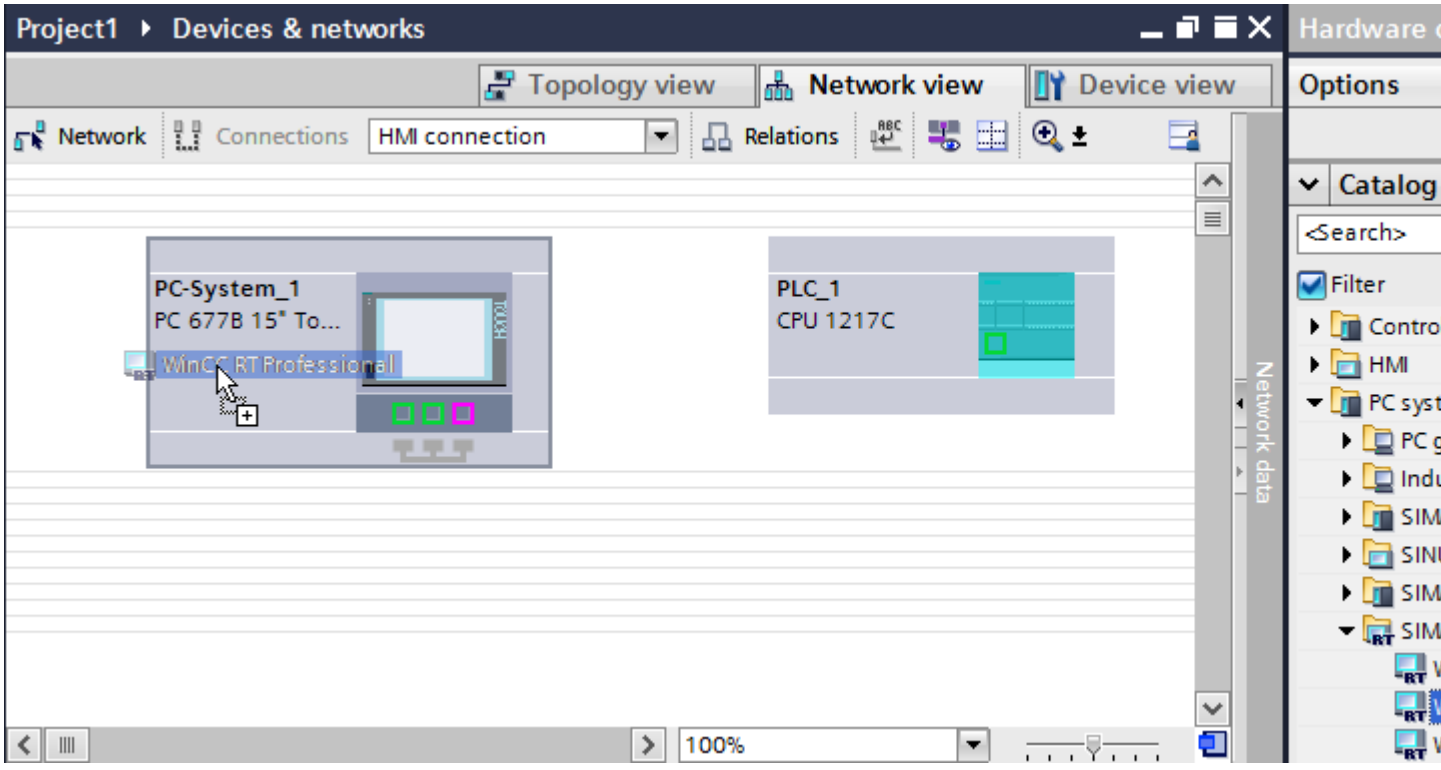
Requirements

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

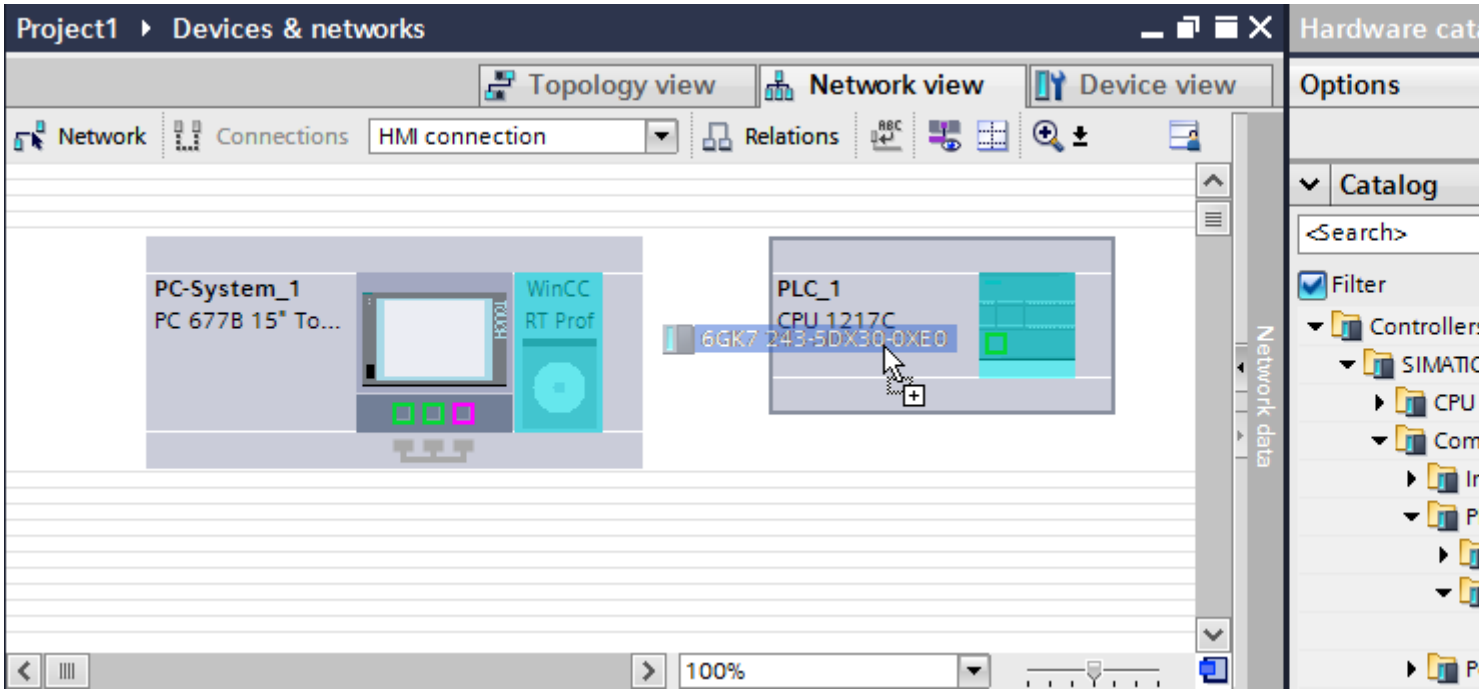
- SIMATIC S7-1200
- SIMATIC PC with PROFIBUS interface

Procedure

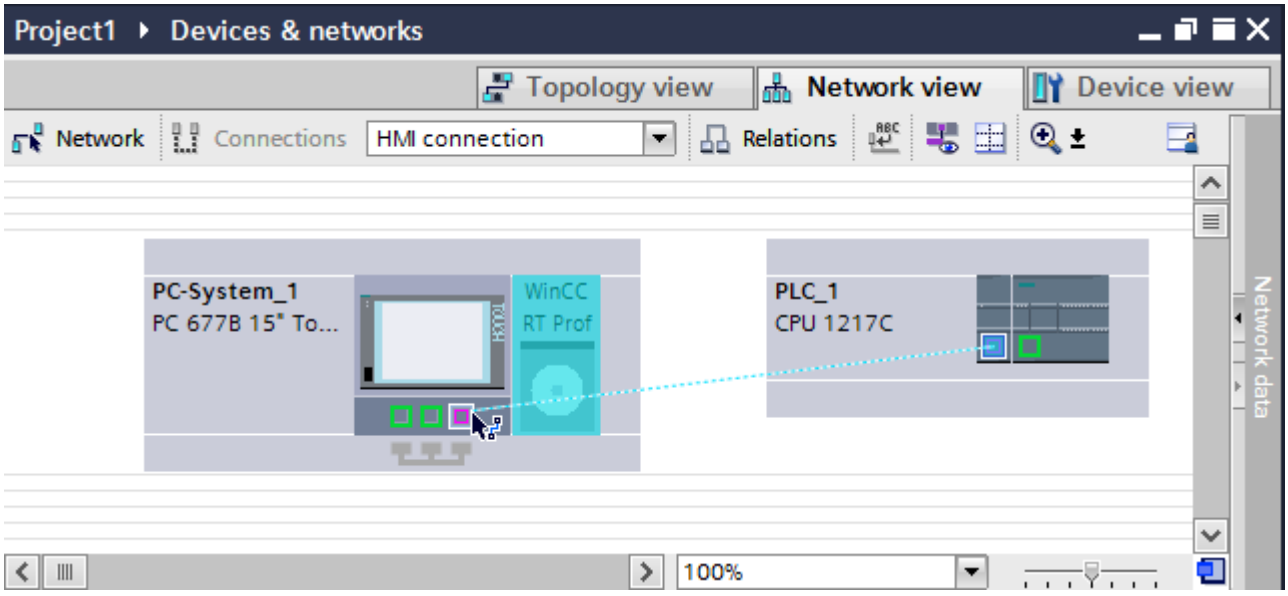
1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed in the network view.
2. Use a drag-and-drop operation to move a WinCC Runtime from the hardware catalog to the PC.



3. Use a drag-and-drop operation to move a PROFIBUS-capable communication module from the hardware catalog to the PLC.



4. Click the "Connections" button and select "HMI connection" for the connection type. The devices available for connection are highlighted in color.
5. Click the PROFIBUS interface of the communication module and use a drag-and-drop operation to draw a connection to the PROFIBUS interface of the PC.



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 339)".

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 298)

Communication via PROFIBUS (Page 330)

Configuring an HMI connection via PROFIBUS with a PC (Page 335)

PROFIBUS parameters (Page 339)

Configuring an HMI connection via PROFIBUS with a PC

Introduction

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

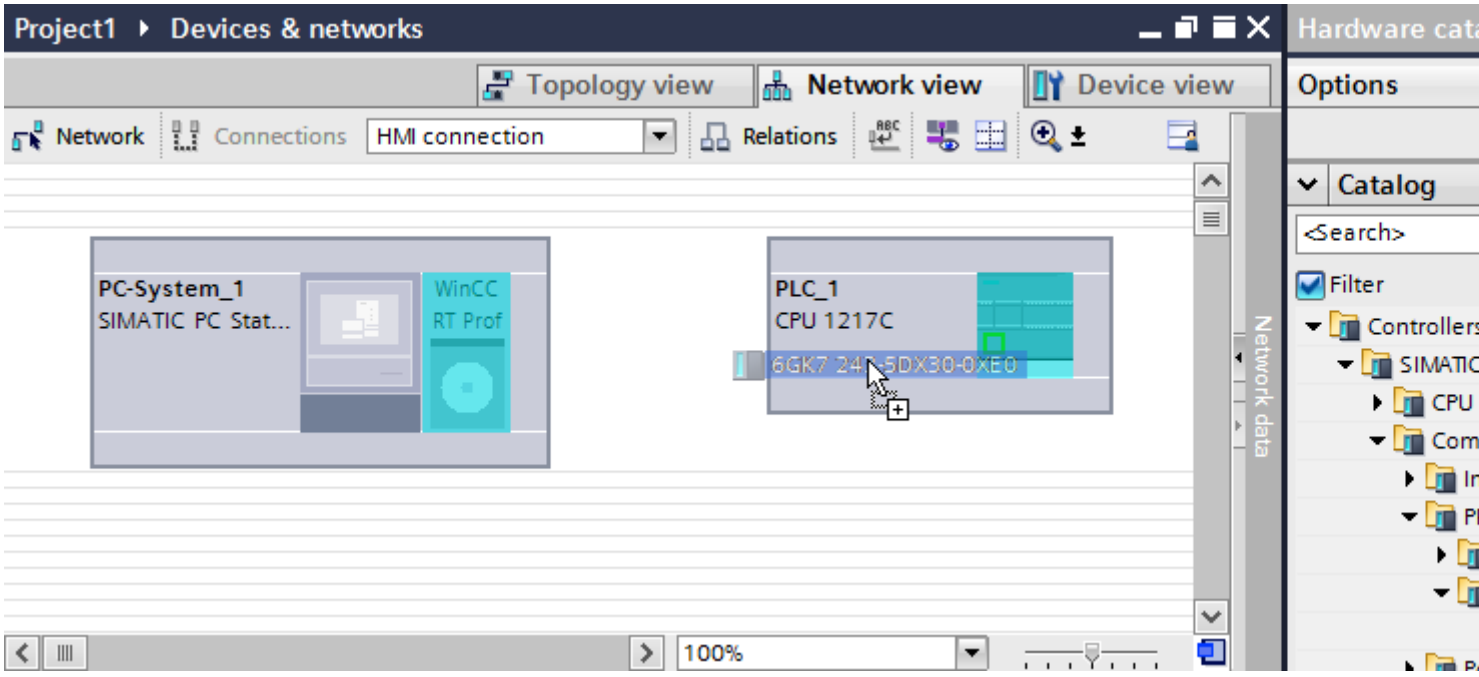
Requirements

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

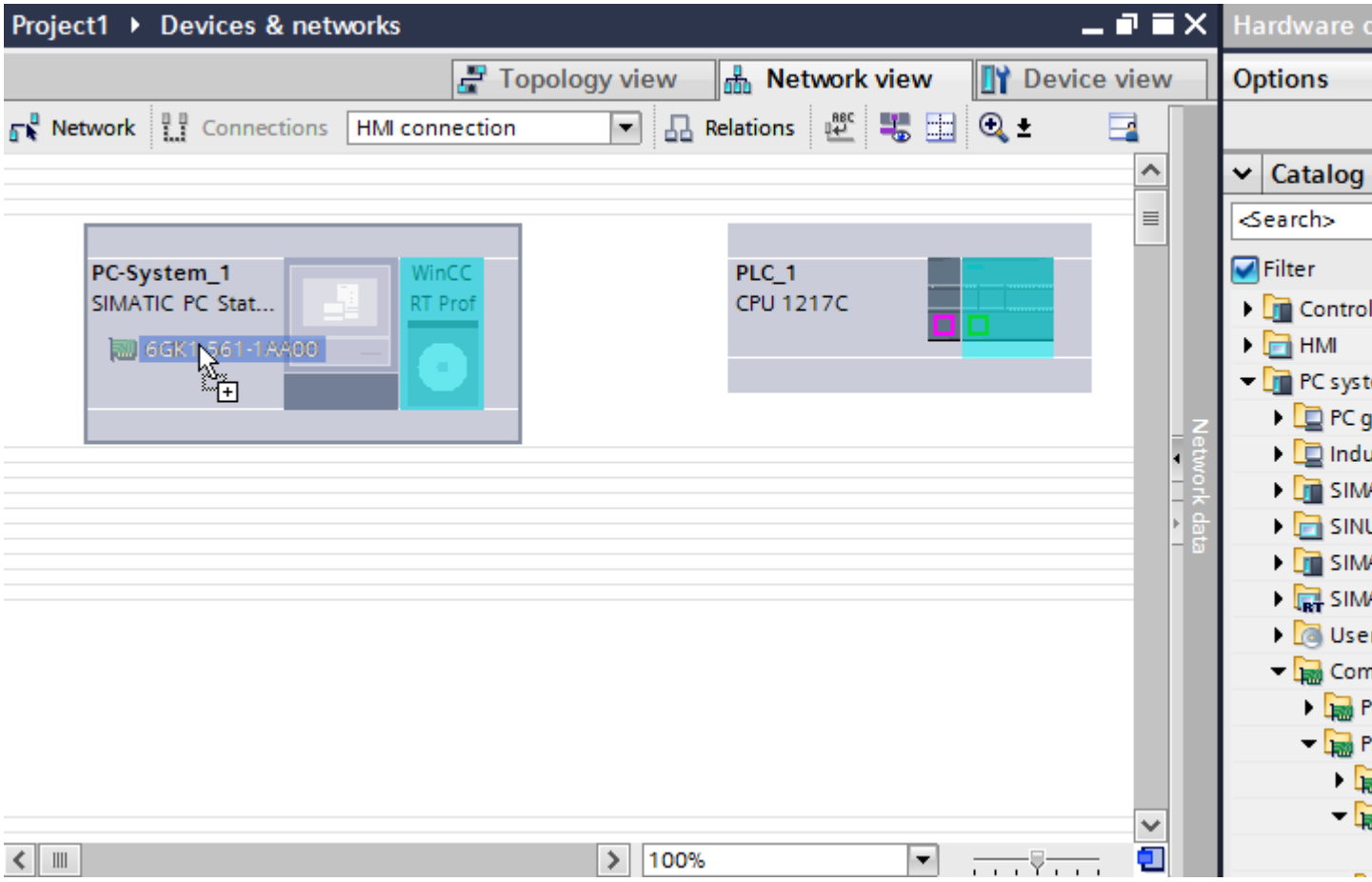
- SIMATIC S7 1200
- WinCC RT Advanced or WinCC RT Professional

Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed in the network view.
2. Use a drag-and-drop operation to move a PROFIBUS-capable communication module from the hardware catalog to the PLC.

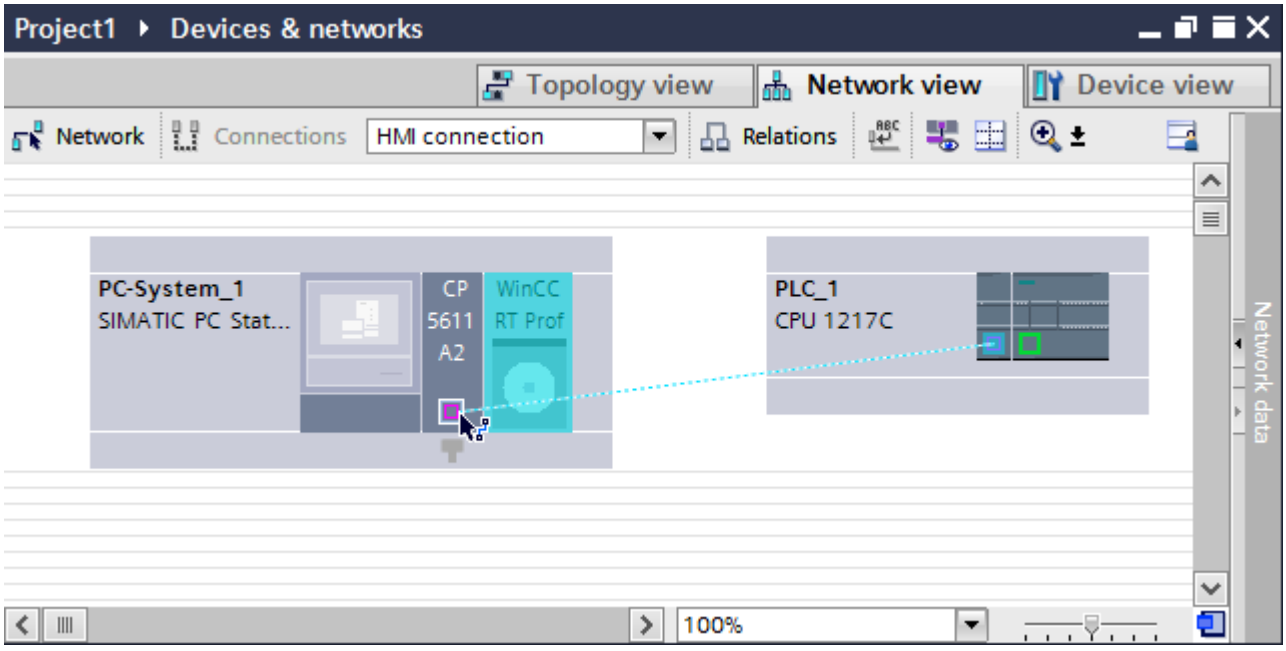


3. Use a drag-and-drop operation to move a PROFIBUS-capable communication processor from the hardware catalog to the WinCC Runtime.



4. Click the "Connections" button and select "HMI connection" for the connection type. The devices available for connection are highlighted in color.

5. Click the PROFIBUS interface of the communication module and use a drag-and-drop operation to draw a connection to the PROFIBUS interface of the communication processor.



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 339)".

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 298)

Communication via PROFIBUS (Page 330)

Configuring an HMI connection via PROFIBUS with a SIMATIC PC (Page 332)

PROFIBUS parameters (Page 339)

2.7.3.3 PROFIBUS parameters

PROFIBUS parameters for the HMI connection

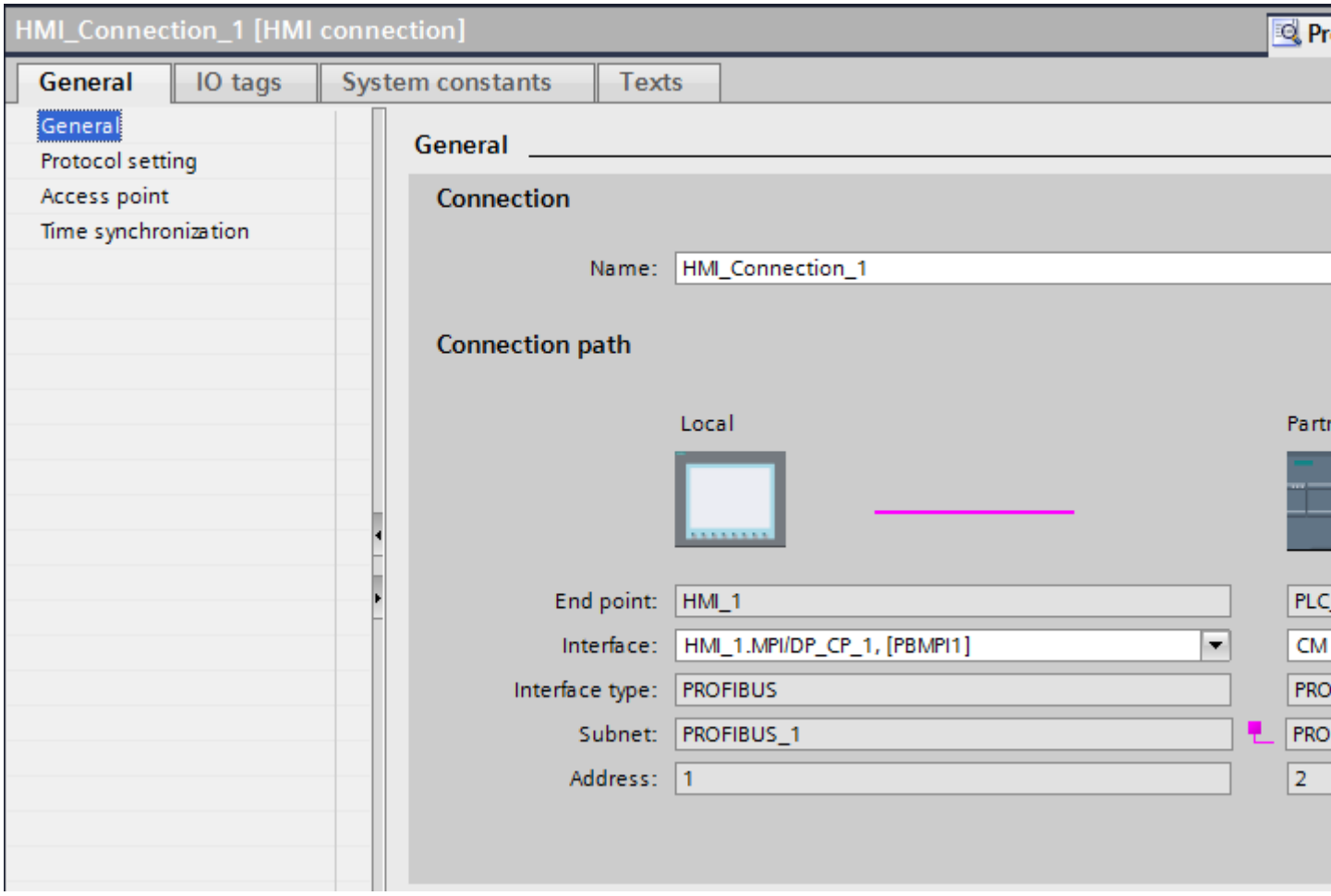
PROFIBUS parameters for the HMI connection

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 with SIMATIC S7-1200 (Page 298)

Access password for the HMI connection (Page 340)

PROFIBUS parameters for the HMI device (Page 341)

PROFIBUS parameters for the PLC (Page 343)

Bus profiles with PROFIBUS (Page 344)

Access password for the HMI connection

Introduction

Secure access to a PLC by assigning a password.

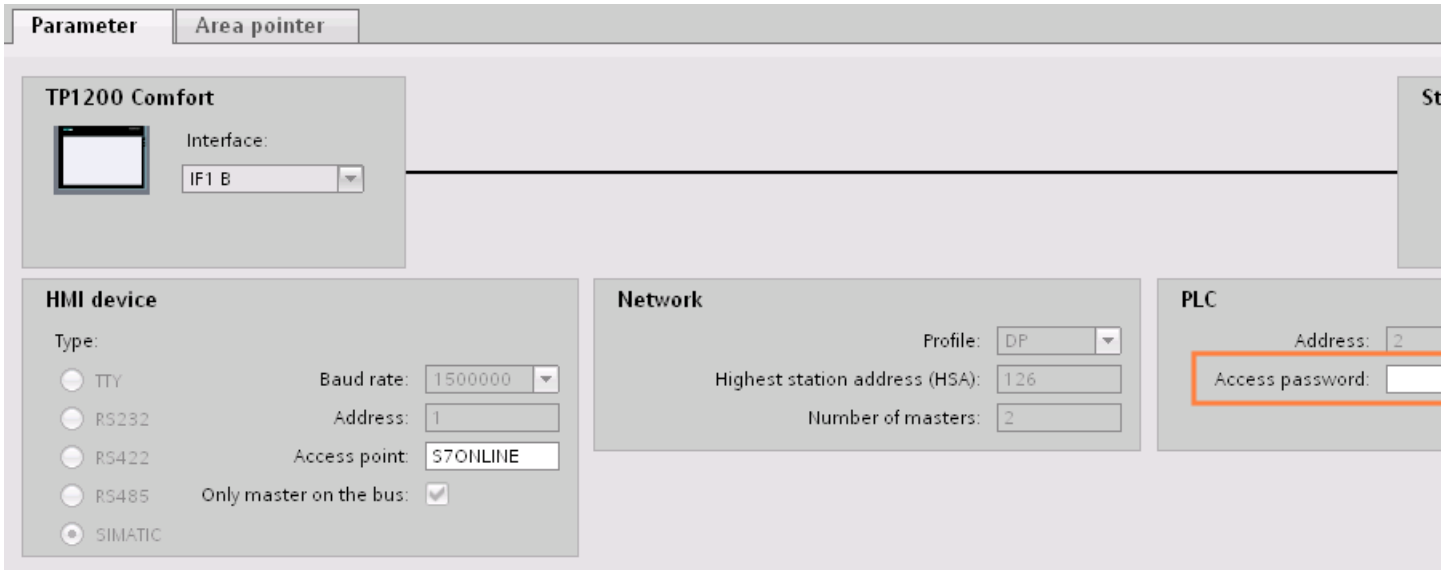
Assign the password when configuring the connection.

Input of the password of the PLC is mandatory as of "Complete protection" security level.

Communication to the PLC is denied if an incorrect password or no password is entered.

Assign password

Enter the "Access password" for the PLC in the "Connections" editor.



See also

Communication with SIMATIC S7-1200 (Page 298)

PROFIBUS parameters for the HMI connection (Page 339)

PROFIBUS parameters for the HMI device (Page 341)

PROFIBUS parameters for the PLC (Page 343)

Bus profiles with PROFIBUS (Page 344)

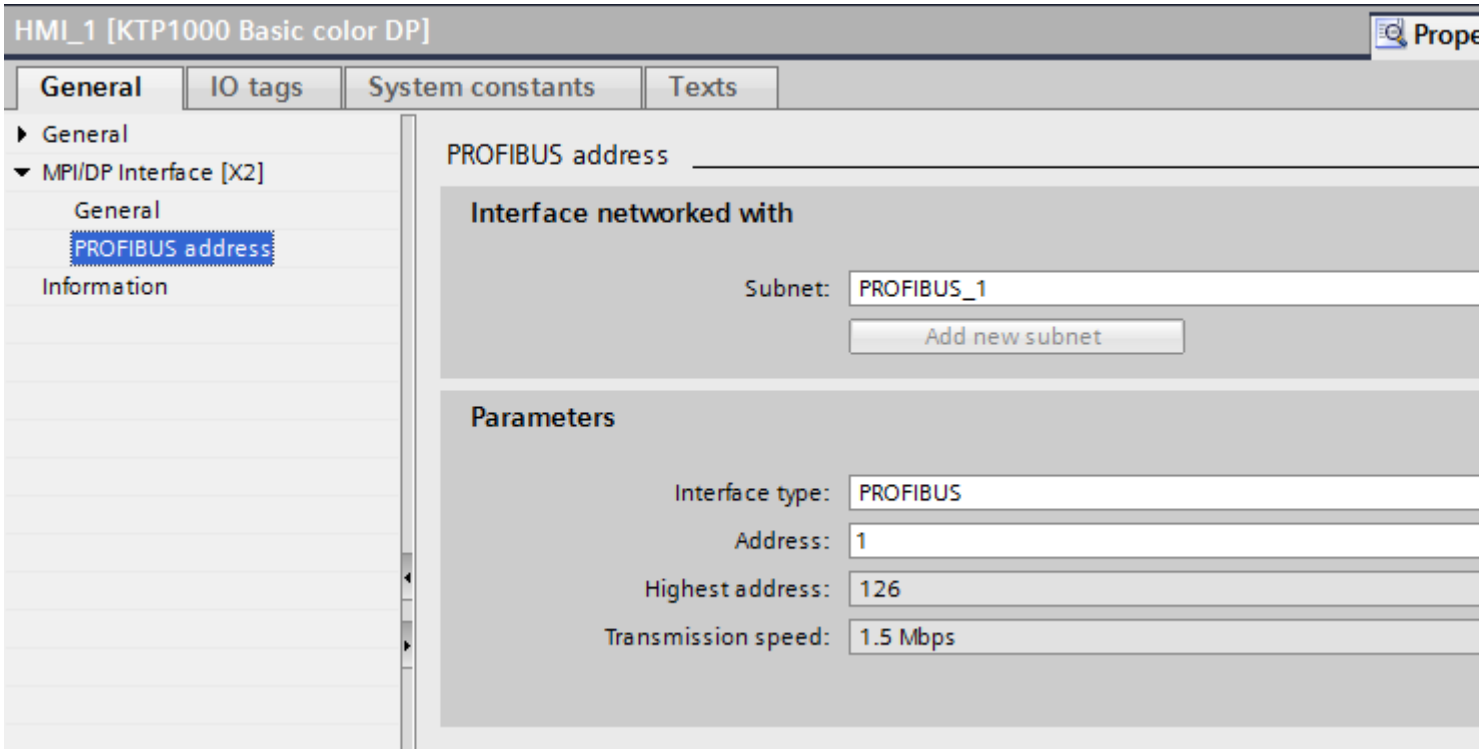
PROFIBUS parameters for the HMI device

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 with SIMATIC S7-1200 (Page 298)

PROFIBUS parameters for the HMI connection (Page 339)

Access password for the HMI connection (Page 340)

PROFIBUS parameters for the PLC (Page 343)

Bus profiles with PROFIBUS (Page 344)

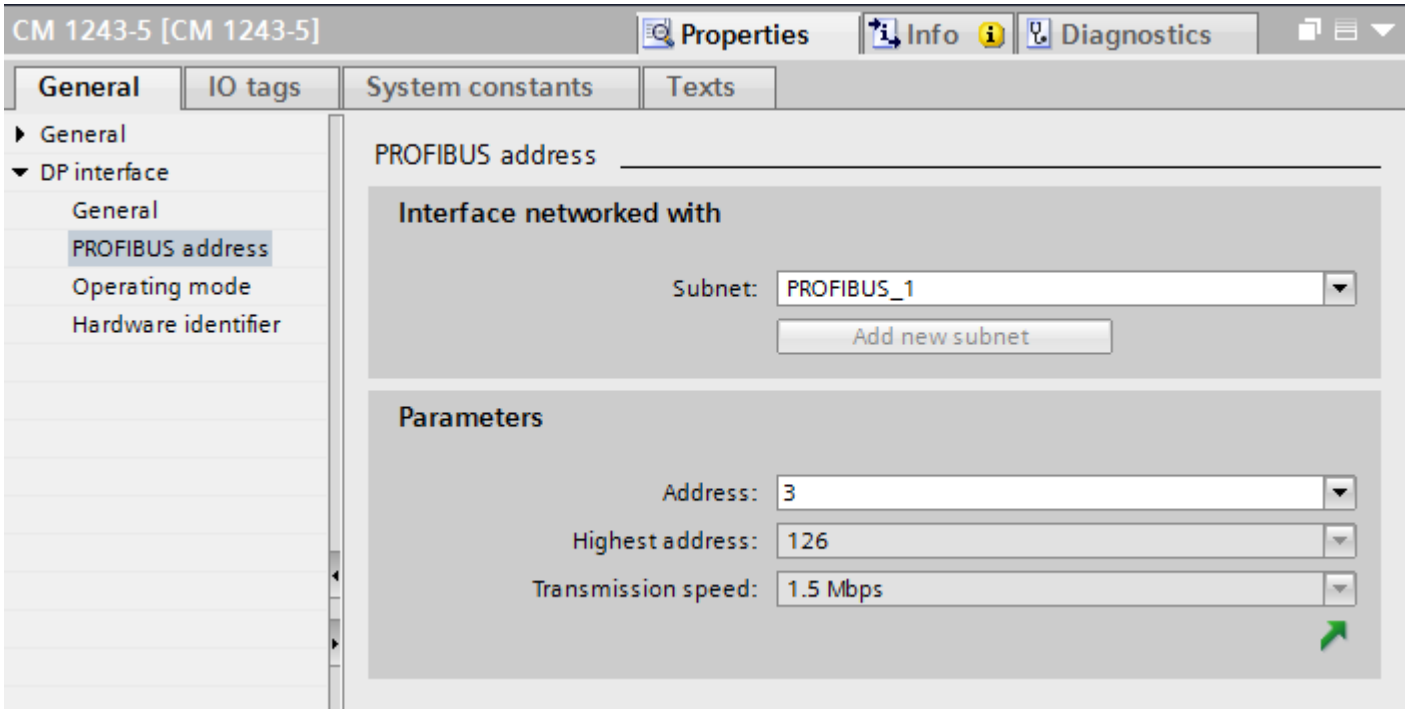
PROFIBUS parameters for the PLC

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 S7-1200 (Page 298)

PROFIBUS parameters for the HMI connection (Page 339)

Access password for the HMI connection (Page 340)

PROFIBUS parameters for the HMI device (Page 341)

Bus profiles with PROFIBUS (Page 344)

Bus profiles with PROFIBUS

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 transfer speeds in Kbps
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 with SIMATIC S7-1200 (Page 298)

PROFIBUS parameters for the HMI connection (Page 339)

Access password for the HMI connection (Page 340)

PROFIBUS parameters for the HMI device (Page 341)

PROFIBUS parameters for the PLC (Page 343)

Communication security

Security levels

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 346)

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

Communication with SIMATIC S7-1200 (Page 298)

PROFIBUS parameters for the HMI connection (Page 339)

Access password for the HMI connection (Page 348)

Setting options for the protection (FW as of V4) (Page 346)

Setting options for the protection (FW as of V4)

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.

Access password for the HMI connection

Introduction

Secure access to a PLC by assigning a password.

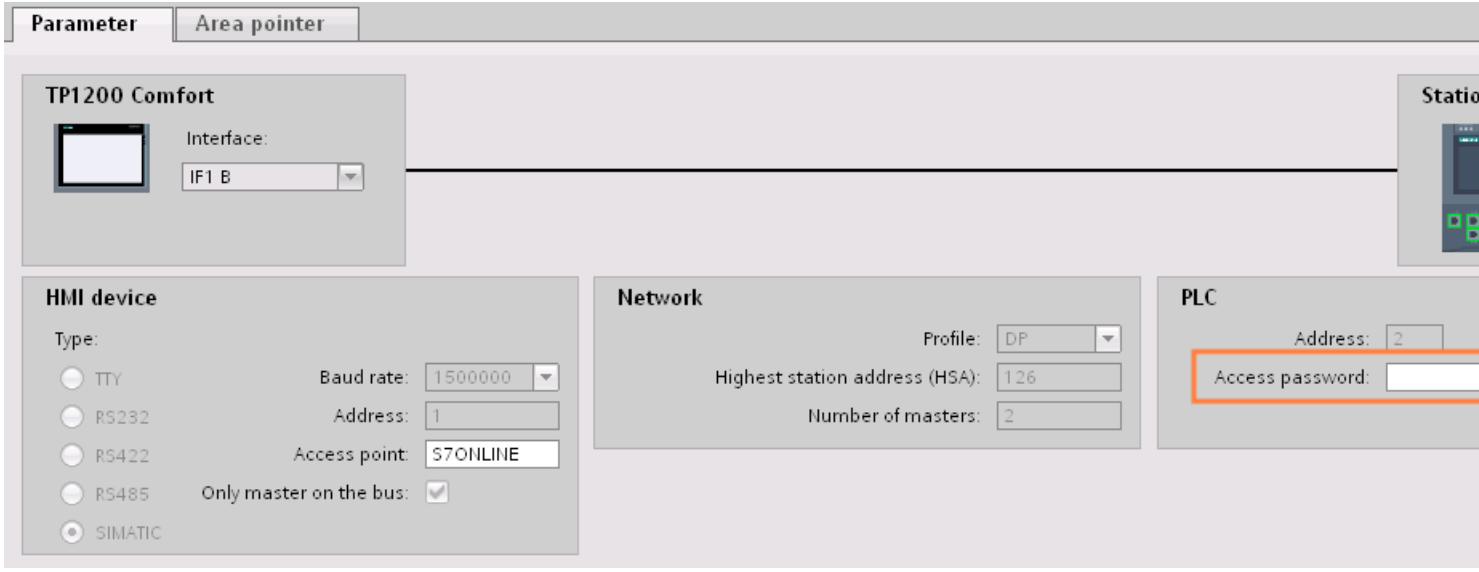
Assign the password when configuring the connection.

Input of the password of the PLC is mandatory as of "Complete protection" security level.

Communication to the PLC is denied if an incorrect password or no password is entered.

Assign password

Enter the "Access password" for the PLC in the "Connections" editor.



See also

Communication with SIMATIC S7-1200 (Page 298)

PROFIBUS parameters for the HMI connection (Page 339)

Security levels (Page 345)

2.7.4 Data exchange

2.7.4.1 Data exchange using area pointers

General information on area pointers

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

Communication with SIMATIC S7-1200 (Page 298)

Data exchange using area pointers (Page 149)

Area pointer "Date/time" (Page 349)

"Date/time PLC" area pointer (Page 351)

Area pointer "Coordination" (Page 352)

Area pointer "Screen number" (Page 354)

Area pointer "Project ID" (Page 355)

Area pointer "Job mailbox" (Page 357)

Area pointer "Date/time"

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

Communication with SIMATIC S7-1200 (Page 298)

General information on area pointers (Page 349)

"Date/time PLC" area pointer (Page 351)

Area pointer "Coordination" (Page 352)

Area pointer "Screen number" (Page 354)

Area pointer "Project ID" (Page 355)

Area pointer "Job mailbox" (Page 357)

"Date/time PLC" area pointer

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

Communication with SIMATIC S7-1200 (Page 298)

General information on area pointers (Page 349)

Area pointer "Date/time" (Page 349)

Area pointer "Coordination" (Page 352)

Area pointer "Screen number" (Page 354)

Area pointer "Project ID" (Page 355)

Area pointer "Job mailbox" (Page 357)

Area pointer "Coordination"

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

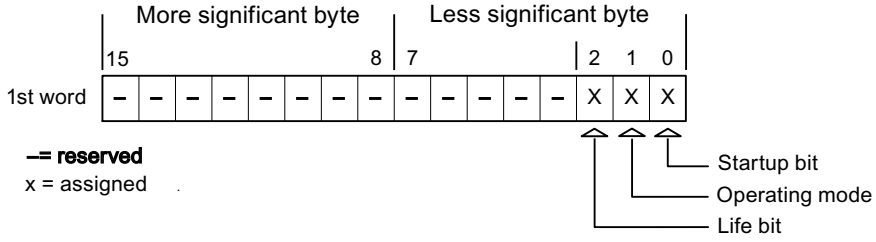
By default, the "Coordination" area pointer has the length of one word and cannot be changed.

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

Communication with SIMATIC S7-1200 (Page 298)

General information on area pointers (Page 349)

Area pointer "Date/time" (Page 349)

"Date/time PLC" area pointer (Page 351)
Area pointer "Screen number" (Page 354)
Area pointer "Project ID" (Page 355)
Area pointer "Job mailbox" (Page 357)

Area pointer "Screen number"

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

Communication with SIMATIC S7-1200 (Page 298)

General information on area pointers (Page 349)

Area pointer "Date/time" (Page 349)

"Date/time PLC" area pointer (Page 351)

Area pointer "Coordination" (Page 352)

Area pointer "Project ID" (Page 355)

Area pointer "Job mailbox" (Page 357)

Area pointer "Project ID"

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

Communication with SIMATIC S7-1200 (Page 298)

General information on area pointers (Page 349)

Area pointer "Date/time" (Page 349)

"Date/time PLC" area pointer (Page 351)

Area pointer "Coordination" (Page 352)

Area pointer "Screen number" (Page 354)

Area pointer "Job mailbox" (Page 357)

Area pointer "Job mailbox"

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 to 23)
	Parameter 2	Left byte: Minutes (0-59) Right byte: Seconds (0 to 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 to 12)
	Parameter 3	Left byte: Year

No.	Function	
14	**Set time (BCD-coded)**	
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.)	
	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

Permitted data types

You can use the following data types when you configure the "Screen number" area pointer:

- Word
- UInt

See also

Communication with SIMATIC S7-1200 (Page 298)

General information on area pointers (Page 349)

Area pointer "Date/time" (Page 349)

"Date/time PLC" area pointer (Page 351)

Area pointer "Coordination" (Page 352)

Area pointer "Screen number" (Page 354)

Area pointer "Project ID" (Page 355)

"Data record" area pointer

"Data mailbox" area pointer

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

Communication with SIMATIC S7-1200 (Page 298)

General information on area pointers (Page 349)

Transfer without synchronization (Page 360)

Transfer with synchronization (Page 361)

Sequence of a transfer started by the operator in the recipe display (Page 363)

Sequence of the transfer triggered by a job mailbox (Page 364)

Sequence of the transfer when triggered by a configured function (Page 366)

Possible causes of error when transferring data records (Page 367)

Transfer without synchronization

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

Communication with SIMATIC S7-1200 (Page 298)

General information on area pointers (Page 349)

"Data mailbox" area pointer (Page 359)

Transfer with synchronization (Page 361)

Sequence of a transfer started by the operator in the recipe display (Page 363)

Sequence of the transfer triggered by a job mailbox (Page 364)

Sequence of the transfer when triggered by a configured function (Page 366)

Possible causes of error when transferring data records (Page 367)

Transfer with synchronization

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

Communication with SIMATIC S7-1200 (Page 298)

General information on area pointers (Page 349)

"Data mailbox" area pointer (Page 359)

Transfer without synchronization (Page 360)

Sequence of a transfer started by the operator in the recipe display (Page 363)

Sequence of the transfer triggered by a job mailbox (Page 364)

Sequence of the transfer when triggered by a configured function (Page 366)

Possible causes of error when transferring data records (Page 367)

Sequence of a transfer started by the operator in the recipe display

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

Communication with SIMATIC S7-1200 (Page 298)
General information on area pointers (Page 349)
"Data mailbox" area pointer (Page 359)
Transfer without synchronization (Page 360)
Transfer with synchronization (Page 361)
Sequence of the transfer triggered by a job mailbox (Page 364)
Sequence of the transfer when triggered by a configured function (Page 366)
Possible causes of error when transferring data records (Page 367)

Sequence of the transfer triggered by a job mailbox

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

Communication with SIMATIC S7-1200 (Page 298)
General information on area pointers (Page 349)
"Data mailbox" area pointer (Page 359)
Transfer without synchronization (Page 360)
Transfer with synchronization (Page 361)
Sequence of a transfer started by the operator in the recipe display (Page 363)
Sequence of the transfer when triggered by a configured function (Page 366)
Possible causes of error when transferring data records (Page 367)

Sequence of the transfer when triggered by a configured function

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.
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

Communication with SIMATIC S7-1200 (Page 298)
General information on area pointers (Page 349)
"Data mailbox" area pointer (Page 359)
Transfer without synchronization (Page 360)
Transfer with synchronization (Page 361)

Sequence of a transfer started by the operator in the recipe display (Page 363)

Sequence of the transfer triggered by a job mailbox (Page 364)

Possible causes of error when transferring data records (Page 367)

Possible causes of error when transferring data records

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

Communication with SIMATIC S7-1200 (Page 298)
General information on area pointers (Page 349)
"Data mailbox" area pointer (Page 359)
Transfer without synchronization (Page 360)
Transfer with synchronization (Page 361)
Sequence of a transfer started by the operator in the recipe display (Page 363)
Sequence of the transfer triggered by a job mailbox (Page 364)
Sequence of the transfer when triggered by a configured function (Page 366)

2.7.4.2 Trends

General information on trends

Trends

A trend is the graphical representation of one or more values from the PLC. The value is read out either time- or bit-triggered, depending on the configuration.

For additional information see:

AUTOHOTSPOT

Note

The value is read out time-triggered for Basic Panels.

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.

Bit-triggered trends

Through a trigger bit set in the trend request tag, the HMI device either reads in a trend value or an entire trend buffer. This setting is defined in the configuration. Bit-triggered trends are normally used to represent fast 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 "HMI tags" editor and connected to trend areas during configuration. The HMI device and PLC then communicate with each other via 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)

See also

Communication with SIMATIC S7-1200 (Page 298)

Trend request and trend transfer (Page 369)

Permitted data types for trends (Page 370)

Trend request and trend transfer

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 closing the screen, the HMI device resets the relevant bits in the trend request area.

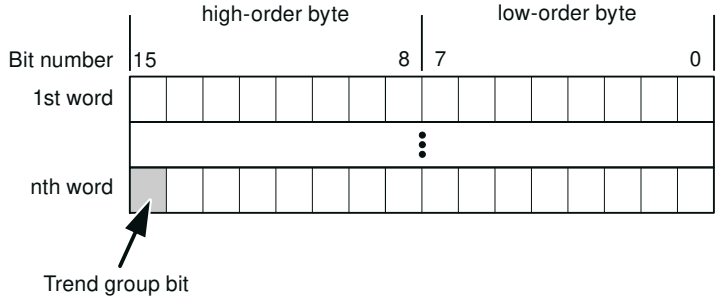
The trend request area can be used for evaluation purposes in the PLC to determine which trend is currently being displayed on the HMI device. Trends can also be triggered without evaluation of the trend request area.

Trend transfer area 1

This area is used to trigger trends. You must set the bit assigned to the trend in the trend transfer area and set the trend group bit in your control program. 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.

The PLC writes to Buffer 2 while the HMI device reads values from Buffer 1, and writes to Buffer 1 when the HMI device is reading Buffer 2. This prevents the PLC from overwriting trend values while the trend is being read by the HMI device.

See also

Communication with SIMATIC S7-1200 (Page 298)

General information on trends (Page 368)

Permitted data types for trends (Page 370)

Permitted data types for trends

For SIMATIC S7

You assign one bit to each trend during configuration. Tags and array tags of the "Word" or "Int" data type are permitted.

See also

Communication with SIMATIC S7-1200 (Page 298)

General information on trends (Page 368)

Trend request and trend transfer (Page 369)

2.7.4.3 Alarms

Configuring alarms

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

Communication with SIMATIC S7-1200 (Page 298)

Acknowledgment of alarms (Page 371)

Acknowledgment of alarms

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

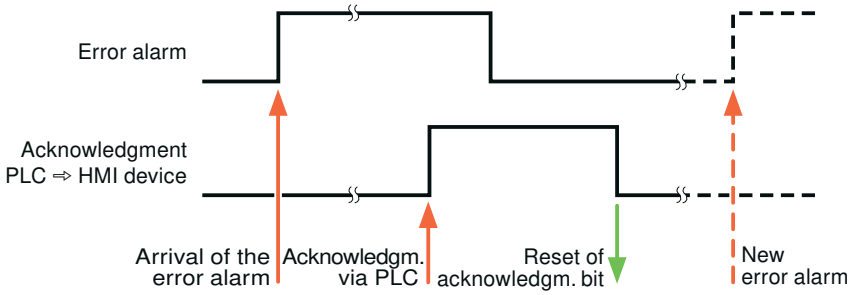
Acknowledgment by the PLC

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.

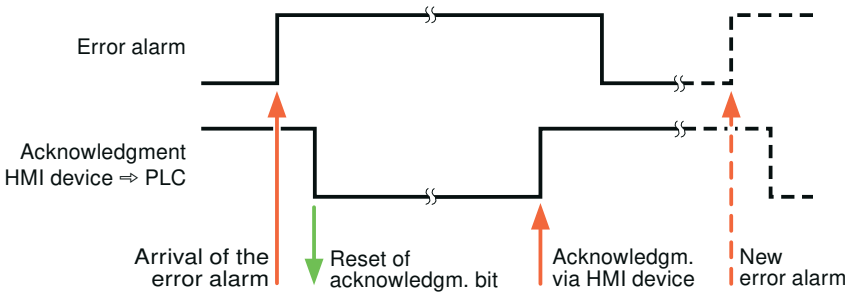
If a discrete alarm subject to acknowledgment is acknowledged by the HMI device, then the corresponding bit in the assigned acknowledgment tag will be set. The entire acknowledgment tag is then written to the PLC by the HMI device. This allows the PLC to recognize that a certain alarm message has been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

Communication with SIMATIC S7-1200 (Page 298)

Configuring alarms (Page 370)

2.7.4.4 LED mapping

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 Comfort Panels	Panel PCs
0	0	Off	Off
0	1	Rapid flashing	Flashing

Bit n+ 1	Bit n	LED function	
		all Mobile Panels, all Comfort Panels	Panel PCs
1	0	Slow flashing	Flashing
1	1	On permanently	On permanently

See also

Communication with SIMATIC S7-1200 (Page 298)

2.7.5 Performance features of communication

2.7.5.1 S7-1200 device dependency

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 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 / DP	No	Yes	Yes	Yes	Yes
KTP900 Basic PN	No	Yes	Yes	Yes	Yes
KTP1200 Basic PN / 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

Basic Panels V14.0.1

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 V15.0

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

Mobile Panels V12.0

HMI devices	SIMAT-IC S7-1200 (V1)	SIMAT-IC S7-1200 (V2)	SIMATIC S7-1200 (V2.2)	SIMATIC S7-1200 (V3)	SIMATIC S7-1200 (V4)
Mobile Panel 177 6" DP	No	Yes	Yes	Yes	Yes
Mobile Panel 177 6" PN	No	Yes	Yes	Yes	Yes
Mobile Panel 277 8"	No	Yes	Yes	Yes	Yes
Mobile Panel 277 8" IW-LAN V2	No	Yes	Yes	Yes	Yes
Mobile Panel 277F 8" IW-LAN V2	No	Yes [1)]	Yes [1)]	Yes [1)]	Yes [1)]
Mobile Panel 277F 8" IW-LAN V2 (RFID tag)	No	Yes [1)]	Yes [1)]	Yes [1)]	Yes [1)]
Mobile Panel 277 10"	No	Yes	Yes	Yes	Yes

Mobile 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)
KTP700 Mobile	No	Yes	Yes	Yes	Yes
KTP700F Mobile	No	Yes	Yes	Yes	Yes
KTP900 Mobile	No	Yes	Yes	Yes	Yes
KTP900F Mobile	No	Yes	Yes	Yes	Yes

Mobile 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)
KTP700 Mobile	No	Yes	Yes	Yes	Yes
KTP700F Mobile	No	Yes	Yes	Yes	Yes
KTP900 Mobile	No	Yes	Yes	Yes	Yes
KTP900F Mobile	No	Yes	Yes	Yes	Yes
KTP400F Mobile	No	Yes	Yes	Yes	Yes

Mobile Panels V14.0.1

HMI devices	SIMATIC S7-1200 (V1)	SIMATIC S7-1200 (V2)	SIMATIC S7-1200 (V2.2)	SIMATIC S7-1200 (V3)	SIMATIC S7-1200 (V4)
KTP700 Mobile	No	Yes	Yes	Yes	Yes
KTP700F Mobile	No	Yes	Yes	Yes	Yes
KTP900 Mobile	No	Yes	Yes	Yes	Yes
KTP900F Mobile	No	Yes	Yes	Yes	Yes
KTP400F Mobile	No	Yes	Yes	Yes	Yes

Mobile Panels V15.0

HMI devices	SIMATIC S7-1200 (V1)	SIMATIC S7-1200 (V2)	SIMATIC S7-1200 (V2.2)	SIMATIC S7-1200 (V3)	SIMATIC S7-1200 (V4)
KTP700 Mobile	No	Yes	Yes	Yes	Yes
KTP700F Mobile	No	Yes	Yes	Yes	Yes
KTP900 Mobile	No	Yes	Yes	Yes	Yes
KTP900F Mobile	No	Yes	Yes	Yes	Yes
KTP400F Mobile	No	Yes	Yes	Yes	Yes

Comfort 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)
KP400 Comfort	No	Yes	Yes	Yes	Yes
KTP400 Comfort	No	Yes	Yes	Yes	Yes
KTP400 Comfort Portrait	No	Yes	Yes	Yes	Yes
KP700 Comfort	No	Yes	Yes	Yes	Yes
TP700 Comfort	No	Yes	Yes	Yes	Yes
TP700 Comfort Portrait	No	Yes	Yes	Yes	Yes
KP900 Comfort	No	Yes	Yes	Yes	Yes
TP900 Comfort	No	Yes	Yes	Yes	Yes
TP900 Comfort Portrait	No	Yes	Yes	Yes	Yes
KP1200 Comfort	No	Yes	Yes	Yes	Yes
TP1200 Comfort	No	Yes	Yes	Yes	Yes
TP1200 Comfort Portrait	No	Yes	Yes	Yes	Yes
KP1500 Comfort	No	Yes	Yes	Yes	Yes
TP1500 Comfort	No	Yes	Yes	Yes	Yes
TP1500 Comfort Portrait	No	Yes	Yes	Yes	Yes
TP1900 Comfort	No	Yes	Yes	Yes	Yes
TP1900 Comfort Portrait	No	Yes	Yes	Yes	Yes
TP2200 Comfort	No	Yes	Yes	Yes	Yes
TP2200 Comfort Portrait	No	Yes	Yes	Yes	Yes

Comfort 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)
KP400 Comfort	No	Yes	Yes	Yes	Yes
KTP400 Comfort	No	Yes	Yes	Yes	Yes

HMI devices	SIMAT-IC S7-1200 (V1)	SIMAT-IC S7-1200 (V2)	SIMATIC S7-1200 (V2.2)	SIMATIC S7-1200 (V3)	SIMATIC S7-1200 (V4)
KTP400 Comfort Portrait	No	Yes	Yes	Yes	Yes
KP700 Comfort	No	Yes	Yes	Yes	Yes
TP700 Comfort	No	Yes	Yes	Yes	Yes
TP700 Comfort Portrait	No	Yes	Yes	Yes	Yes
KP900 Comfort	No	Yes	Yes	Yes	Yes
TP900 Comfort	No	Yes	Yes	Yes	Yes
TP900 Comfort Portrait	No	Yes	Yes	Yes	Yes
KP1200 Comfort	No	Yes	Yes	Yes	Yes
TP1200 Comfort	No	Yes	Yes	Yes	Yes
TP1200 Comfort Portrait	No	Yes	Yes	Yes	Yes
KP1500 Comfort	No	Yes	Yes	Yes	Yes
TP1500 Comfort	No	Yes	Yes	Yes	Yes
TP1500 Comfort Portrait	No	Yes	Yes	Yes	Yes
TP1900 Comfort	No	Yes	Yes	Yes	Yes
TP1900 Comfort Portrait	No	Yes	Yes	Yes	Yes
TP2200 Comfort	No	Yes	Yes	Yes	Yes
TP2200 Comfort Portrait	No	Yes	Yes	Yes	Yes

Comfort Panels V13.0.1

HMI devices	SIMATIC S7-1200 (V1)	SIMAT-IC S7-1200 (V2)	SIMAT-IC S7-1200 (V2.2)	SIMAT-IC S7-1200 (V3)	SIMAT-IC S7-1200 (V4)
KP400 Comfort	No	Yes	Yes	Yes	Yes
KTP400 Comfort	No	Yes	Yes	Yes	Yes
KTP400 Comfort Portrait	No	Yes	Yes	Yes	Yes
KP700 Comfort	No	Yes	Yes	Yes	Yes
TP700 Comfort	No	Yes	Yes	Yes	Yes
TP700 Comfort Portrait	No	Yes	Yes	Yes	Yes
KP900 Comfort	No	Yes	Yes	Yes	Yes
TP900 Comfort	No	Yes	Yes	Yes	Yes
TP900 Comfort Portrait	No	Yes	Yes	Yes	Yes
KP1200 Comfort	No	Yes	Yes	Yes	Yes
TP1200 Comfort	No	Yes	Yes	Yes	Yes
TP1200 Comfort Portrait	No	Yes	Yes	Yes	Yes
KP1500 Comfort	No	Yes	Yes	Yes	Yes
TP1500 Comfort	No	Yes	Yes	Yes	Yes
TP1500 Comfort Portrait	No	Yes	Yes	Yes	Yes

HMI devices	SIMATIC S7-1200 (V1)	SIMAT-IC S7-1200 (V2)	SIMAT-IC S7-1200 (V2.2)	SIMAT-IC S7-1200 (V3)	SIMAT-IC S7-1200 (V4)
TP1900 Comfort	No	Yes	Yes	Yes	Yes
TP1900 Comfort Portrait	No	Yes	Yes	Yes	Yes
TP2200 Comfort	No	Yes	Yes	Yes	Yes
TP2200 Comfort Portrait	No	Yes	Yes	Yes	Yes

Comfort Panels V14.0

HMI devices	SIMATIC S7-1200 (V1)	SIMAT-IC S7-1200 (V2)	SIMAT-IC S7-1200 (V2.2)	SIMAT-IC S7-1200 (V3)	SIMAT-IC S7-1200 (V4)
KP400 Comfort	No	Yes	Yes	Yes	Yes
KTP400 Comfort	No	Yes	Yes	Yes	Yes
KTP400 Comfort Portrait	No	Yes	Yes	Yes	Yes
KP700 Comfort	No	Yes	Yes	Yes	Yes
TP700 Comfort	No	Yes	Yes	Yes	Yes
TP700 Comfort Portrait	No	Yes	Yes	Yes	Yes
KP900 Comfort	No	Yes	Yes	Yes	Yes
TP900 Comfort	No	Yes	Yes	Yes	Yes
TP900 Comfort Portrait	No	Yes	Yes	Yes	Yes
KP1200 Comfort	No	Yes	Yes	Yes	Yes
TP1200 Comfort	No	Yes	Yes	Yes	Yes
TP1200 Comfort Portrait	No	Yes	Yes	Yes	Yes
KP1500 Comfort	No	Yes	Yes	Yes	Yes
TP1500 Comfort	No	Yes	Yes	Yes	Yes
TP1500 Comfort Portrait	No	Yes	Yes	Yes	Yes
TP1900 Comfort	No	Yes	Yes	Yes	Yes
TP1900 Comfort Portrait	No	Yes	Yes	Yes	Yes
TP2200 Comfort	No	Yes	Yes	Yes	Yes
TP2200 Comfort Portrait	No	Yes	Yes	Yes	Yes

Comfort Panels V14.0.1

HMI devices	SIMATIC S7-1200 (V1)	SIMAT-IC S7-1200 (V2)	SIMAT-IC S7-1200 (V2.2)	SIMAT-IC S7-1200 (V3)	SIMAT-IC S7-1200 (V4)
KP400 Comfort	No	Yes	Yes	Yes	Yes
KTP400 Comfort	No	Yes	Yes	Yes	Yes
KTP400 Comfort Portrait	No	Yes	Yes	Yes	Yes
KP700 Comfort	No	Yes	Yes	Yes	Yes
TP700 Comfort	No	Yes	Yes	Yes	Yes

HMI devices	SIMATIC S7-1200 (V1)	SIMAT-IC S7-1200 (V2)	SIMAT-IC S7-1200 (V2.2)	SIMAT-IC S7-1200 (V3)	SIMAT-IC S7-1200 (V4)
TP700 Comfort Portrait	No	Yes	Yes	Yes	Yes
KP900 Comfort	No	Yes	Yes	Yes	Yes
TP900 Comfort	No	Yes	Yes	Yes	Yes
TP900 Comfort Portrait	No	Yes	Yes	Yes	Yes
KP1200 Comfort	No	Yes	Yes	Yes	Yes
TP1200 Comfort	No	Yes	Yes	Yes	Yes
TP1200 Comfort Portrait	No	Yes	Yes	Yes	Yes
KP1500 Comfort	No	Yes	Yes	Yes	Yes
TP1500 Comfort	No	Yes	Yes	Yes	Yes
TP1500 Comfort Portrait	No	Yes	Yes	Yes	Yes
TP1900 Comfort	No	Yes	Yes	Yes	Yes
TP1900 Comfort Portrait	No	Yes	Yes	Yes	Yes
TP2200 Comfort	No	Yes	Yes	Yes	Yes
TP2200 Comfort Portrait	No	Yes	Yes	Yes	Yes

Comfort Panels V15.0

HMI devices	SIMATIC S7-1200 (V1)	SIMAT-IC S7-1200 (V2)	SIMAT-IC S7-1200 (V2.2)	SIMAT-IC S7-1200 (V3)	SIMAT-IC S7-1200 (V4)
KP400 Comfort	No	Yes	Yes	Yes	Yes
KTP400 Comfort	No	Yes	Yes	Yes	Yes
KTP400 Comfort Portrait	No	Yes	Yes	Yes	Yes
KP700 Comfort	No	Yes	Yes	Yes	Yes
TP700 Comfort	No	Yes	Yes	Yes	Yes
TP700 Comfort Portrait	No	Yes	Yes	Yes	Yes
KP900 Comfort	No	Yes	Yes	Yes	Yes
TP900 Comfort	No	Yes	Yes	Yes	Yes
TP900 Comfort Portrait	No	Yes	Yes	Yes	Yes
KP1200 Comfort	No	Yes	Yes	Yes	Yes
TP1200 Comfort	No	Yes	Yes	Yes	Yes
TP1200 Comfort Portrait	No	Yes	Yes	Yes	Yes
KP1500 Comfort	No	Yes	Yes	Yes	Yes
TP1500 Comfort	No	Yes	Yes	Yes	Yes
TP1500 Comfort Portrait	No	Yes	Yes	Yes	Yes
TP1900 Comfort	No	Yes	Yes	Yes	Yes
TP1900 Comfort Portrait	No	Yes	Yes	Yes	Yes
TP2200 Comfort	No	Yes	Yes	Yes	Yes
TP2200 Comfort Portrait	No	Yes	Yes	Yes	Yes

Runtime V12.0

HMI devices	SIMAT-IC S7-1200 (V1)	SIMAT-IC S7-1200 (V2)	SIMATIC S7-1200 (V2.2)	SIMATIC S7-1200 (V3)	SIMATIC S7-1200 (V4)
WinCC RT Advanced	No	Yes	Yes	Yes	Yes

Runtime V13.0

HMI devices	SIMAT-IC S7-1200 (V1)	SIMAT-IC S7-1200 (V2)	SIMATIC S7-1200 (V2.2)	SIMATIC S7-1200 (V3)	SIMATIC S7-1200 (V4)
WinCC RT Advanced	No	Yes	Yes	Yes	Yes

Runtime V13.0.1

HMI devices	SIMAT-IC S7-1200 (V1)	SIMAT-IC S7-1200 (V2)	SIMATIC S7-1200 (V2.2)	SIMATIC S7-1200 (V3)	SIMATIC S7-1200 (V4)
WinCC RT Advanced	No	Yes	Yes	Yes	Yes

Runtime V14.0

HMI devices	SIMAT-IC S7-1200 (V1)	SIMAT-IC S7-1200 (V2)	SIMATIC S7-1200 (V2.2)	SIMATIC S7-1200 (V3)	SIMATIC S7-1200 (V4)
WinCC RT Advanced	No	Yes	Yes	Yes	Yes
WinCC RT Professional	No	Yes	Yes	Yes	Yes

Runtime V14.0.1

HMI devices	SIMAT-IC S7-1200 (V1)	SIMAT-IC S7-1200 (V2)	SIMATIC S7-1200 (V2.2)	SIMATIC S7-1200 (V3)	SIMATIC S7-1200 (V4)
WinCC RT Advanced	No	Yes	Yes	Yes	Yes
WinCC RT Professional	No	Yes	Yes	Yes	Yes

Runtime V15.0

HMI devices	SIMAT-IC S7-1200 (V1)	SIMAT-IC S7-1200 (V2)	SIMATIC S7-1200 (V2.2)	SIMATIC S7-1200 (V3)	SIMATIC S7-1200 (V4)
WinCC RT Advanced	No	Yes	Yes	Yes	Yes
WinCC RT Professional	No	Yes	Yes	Yes	Yes

See also

Communication with SIMATIC S7-1200 (Page 298)

Valid data types for SIMATIC S7-1200 (Page 382)

2.7.5.2 Valid data types for SIMATIC S7-1200

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 298)

S7-1200 device dependency (Page 374)

2.7.6 Configuring connections in the "Connections" editor

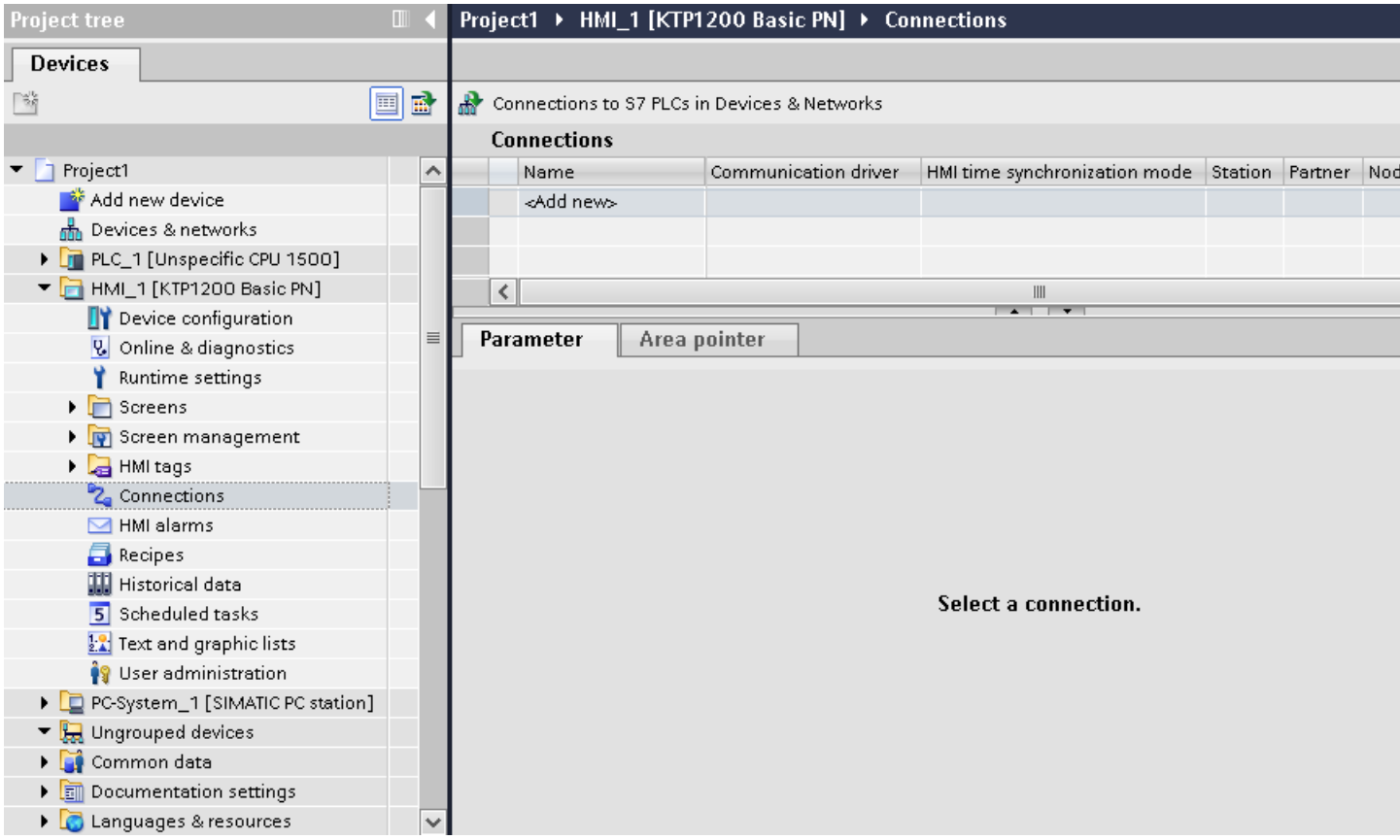
2.7.6.1 Creating a PROFINET connection

Requirements

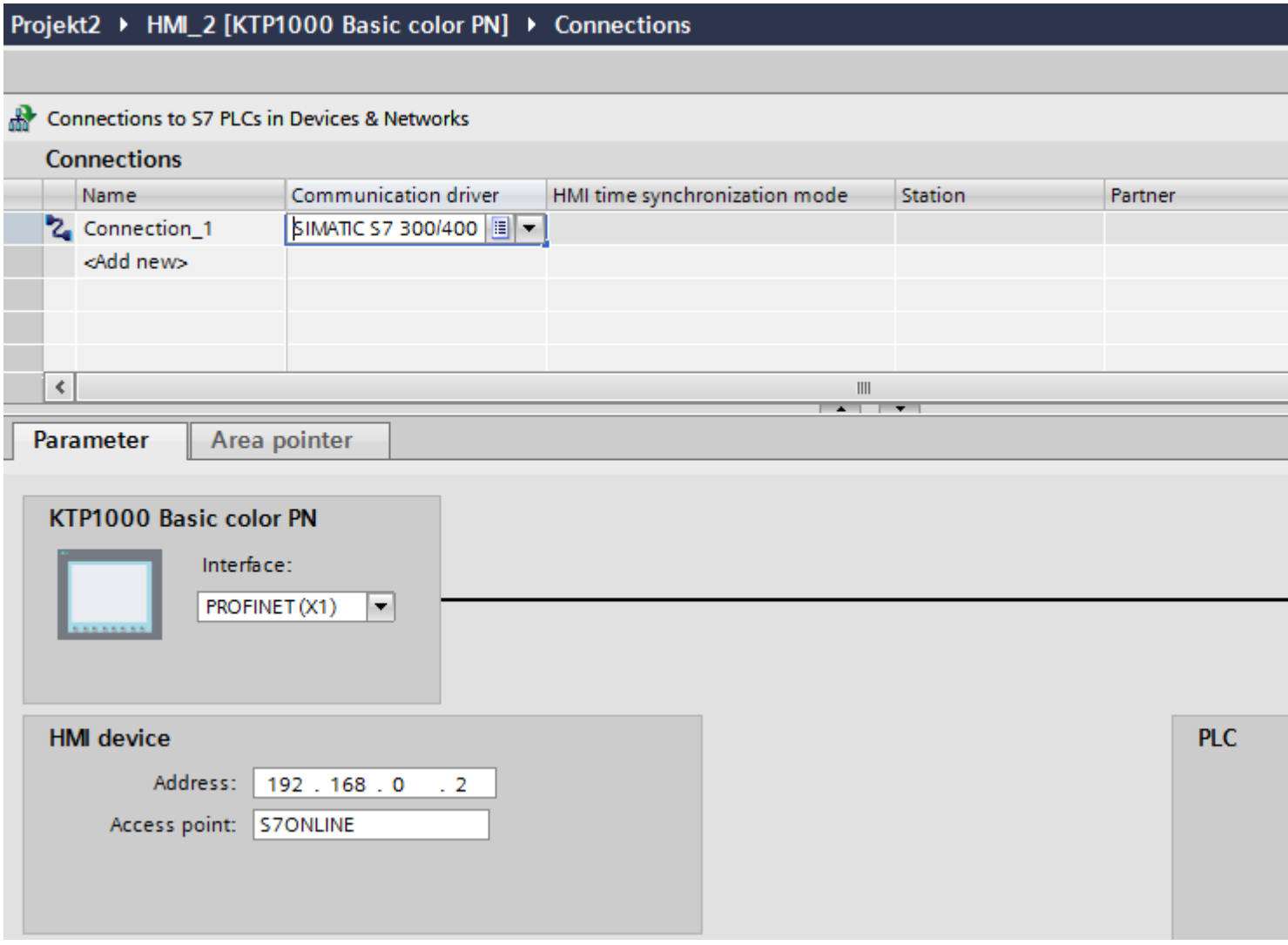
- A project is open.
- An HMI device with a PROFINET interface has been created.

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 with SIMATIC S7-1200 (Page 298)

Creating a PROFIBUS connection (Page 385)

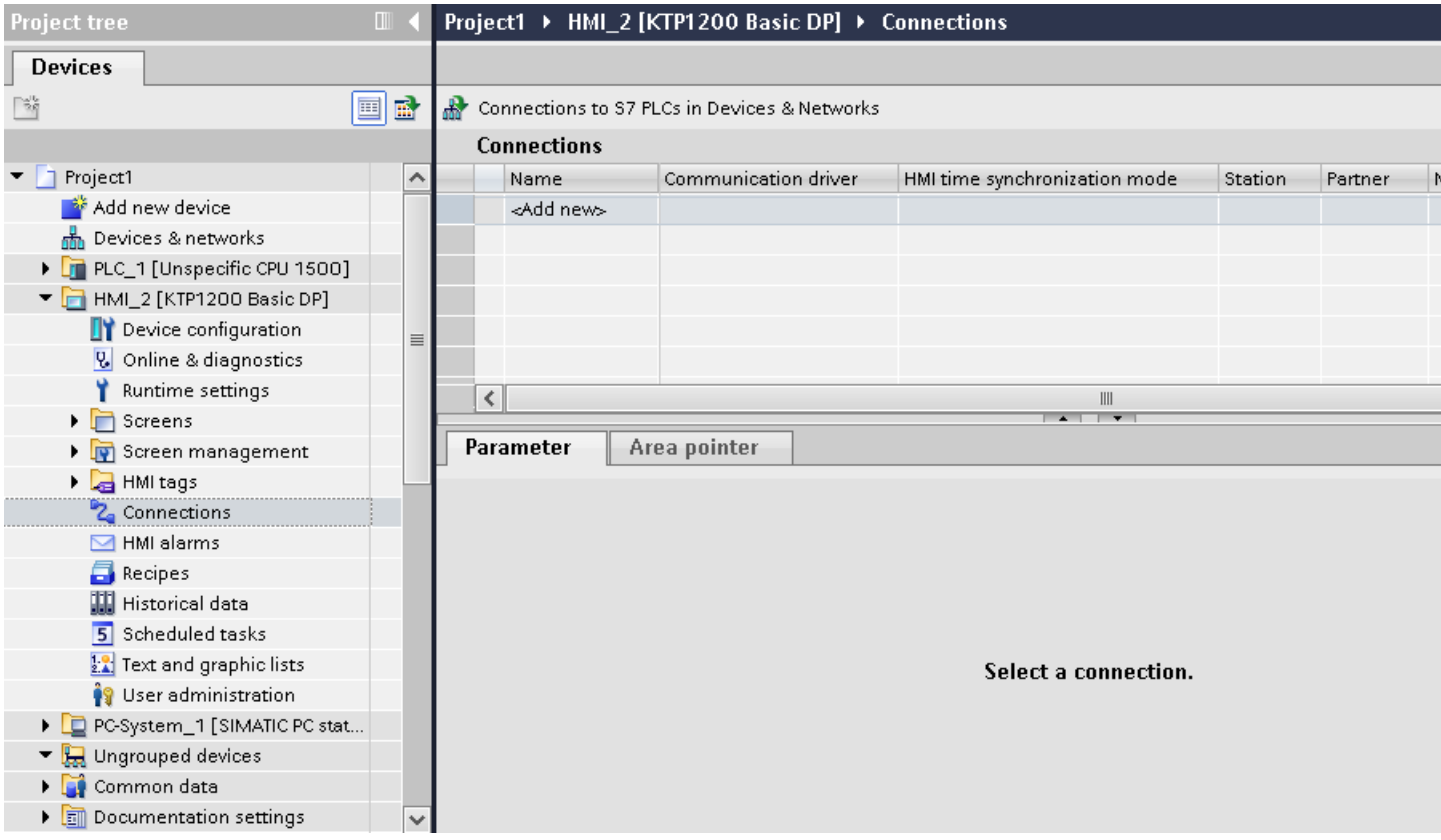
2.7.6.2 Creating a PROFIBUS connection

Requirements

- A project is open.
- An HMI device with a PROFIBUS interface has been created.

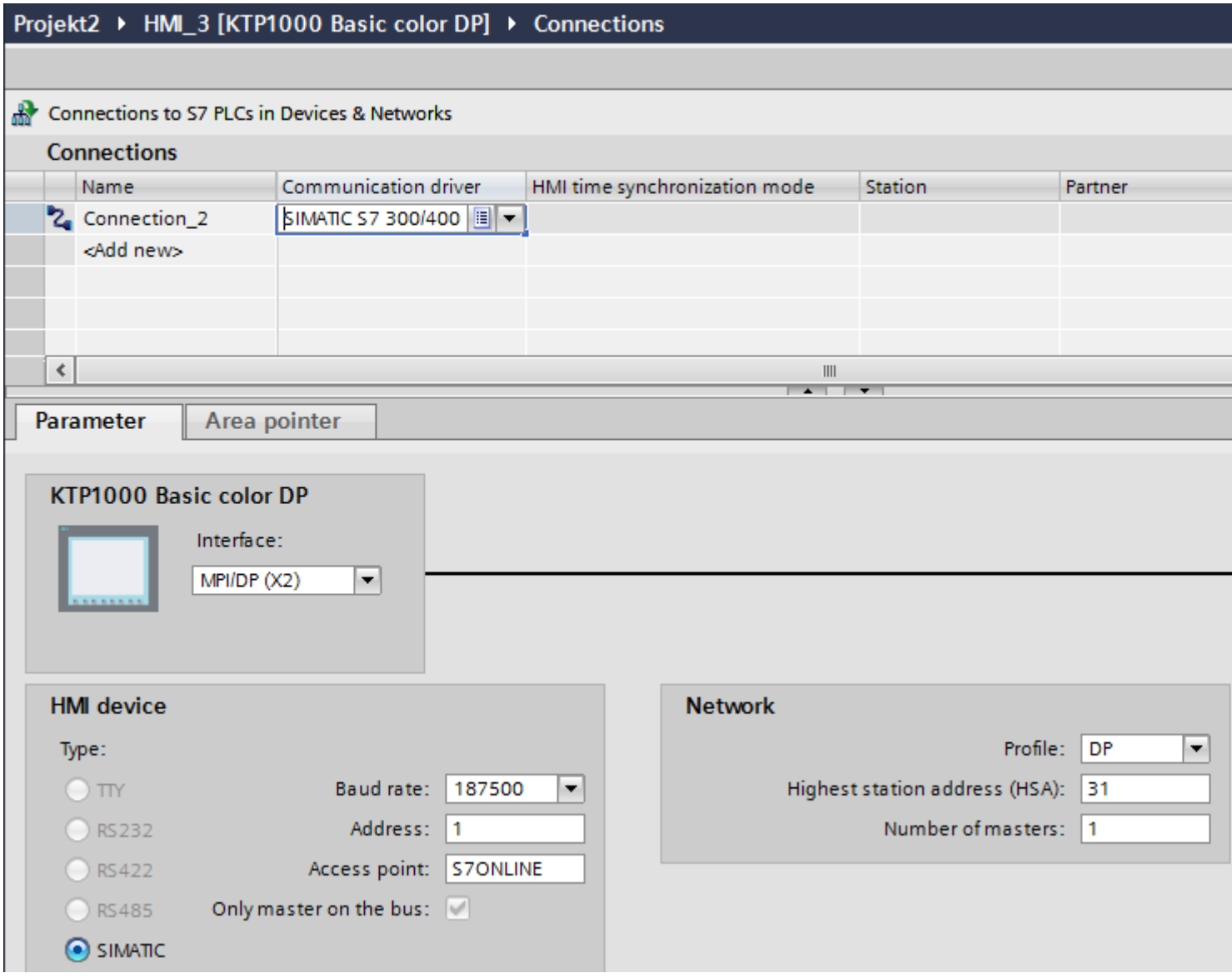
Procedure

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 with SIMATIC S7-1200 (Page 298)
Creating a PROFINET connection (Page 383)

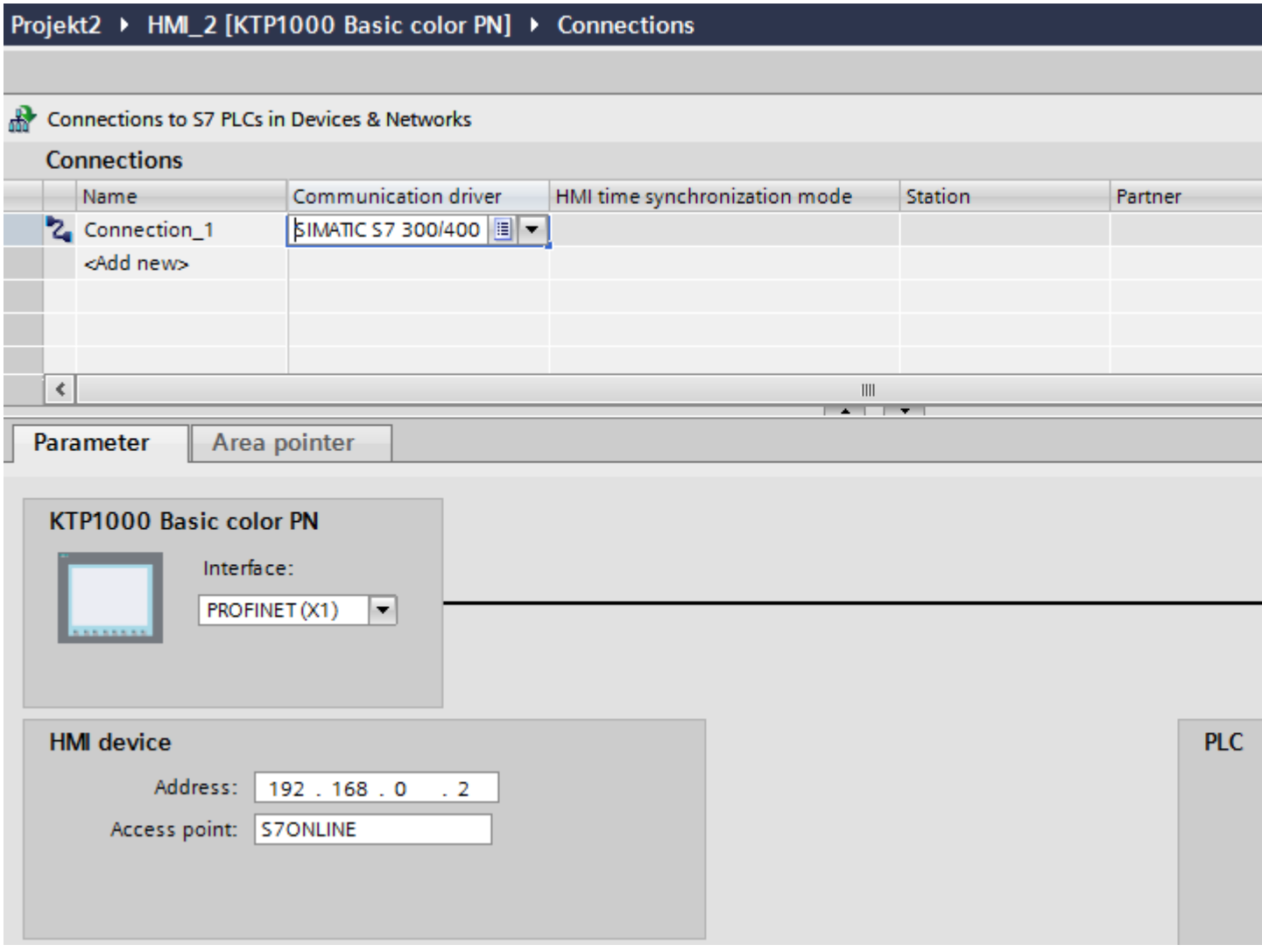
2.7.6.3 Parameters for the connection (SIMATIC S7 1200)

Connection parameters

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 298)

Ethernet parameters (Page 388)

PROFIBUS parameters (Page 389)

Ethernet parameters

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 298)

Connection parameters (Page 387)

PROFIBUS parameters (Page 389)

PROFIBUS parameters

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.
- "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.
- "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 298)

Connection parameters (Page 387)

Ethernet parameters (Page 388)

2.7.7 Configuring connections in the "Connections" editor

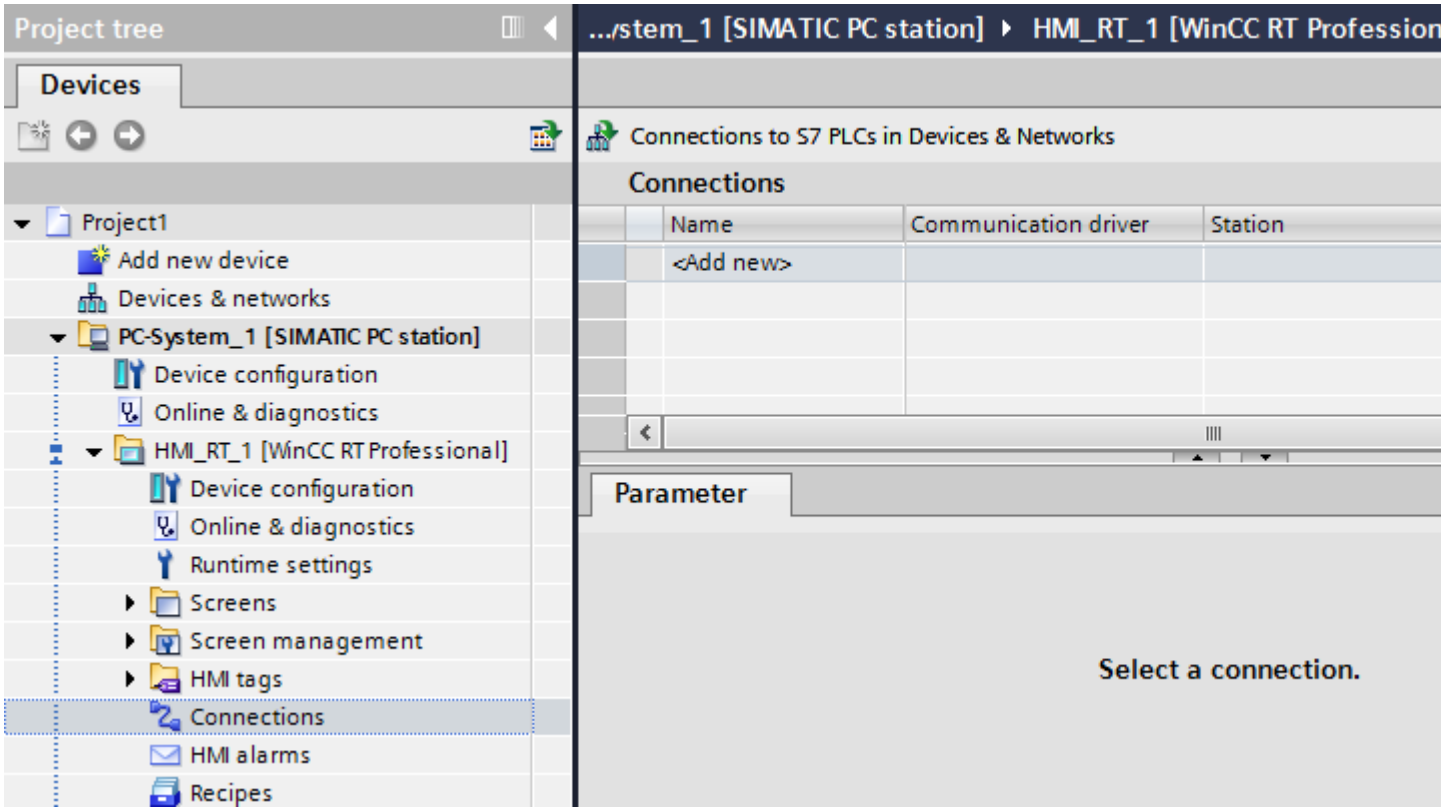
2.7.7.1 Creating a connection

Requirements

- A project is open.
- WinCC RT Professional is 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 an interface of the HMI device in the Inspector window under "Parameters > Interface".
6. Set the parameters for the connection in the Inspector window.

Interfaces

Select one of the following interfaces in the Inspector window under "Parameters > WinCC RT Professional > Interfaces".

- TCP/IP
- PROFIBUS

For additional information on the parameters of the interfaces see:

Parameters for the connection (SIMATIC S7 1200) (Page 393)

See also

Communication with SIMATIC S7-1200 (Page 298)

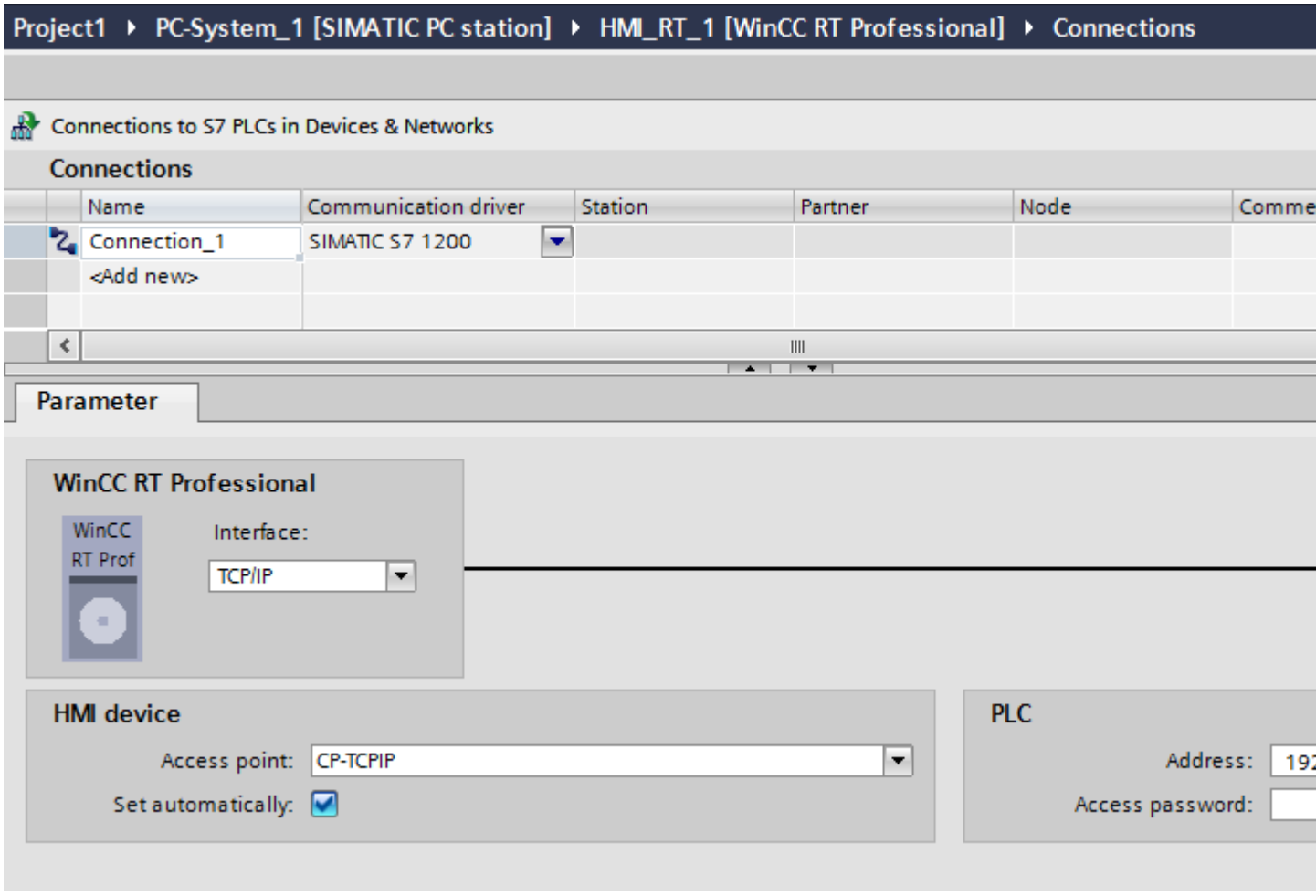
Parameters for the connection (SIMATIC S7 1200) (Page 393)

2.7.7.2 Parameters for the connection (SIMATIC S7 1200)

TCP/IP

Introduction

You assign the parameters for the HMI device and the PLC in the network under "Parameters".



HMI device

- "Access point"
  The access point defines a logical device name through which the communication partner can be reached.
  If only one communications processor is installed for a communication type, select the option "Set automatically". The logical device name is then set automatically when you start Runtime.

PLC

- "Address"
  Enter the station address of the CPU under "Address".

See also

Communication with SIMATIC S7-1200 (Page 298)

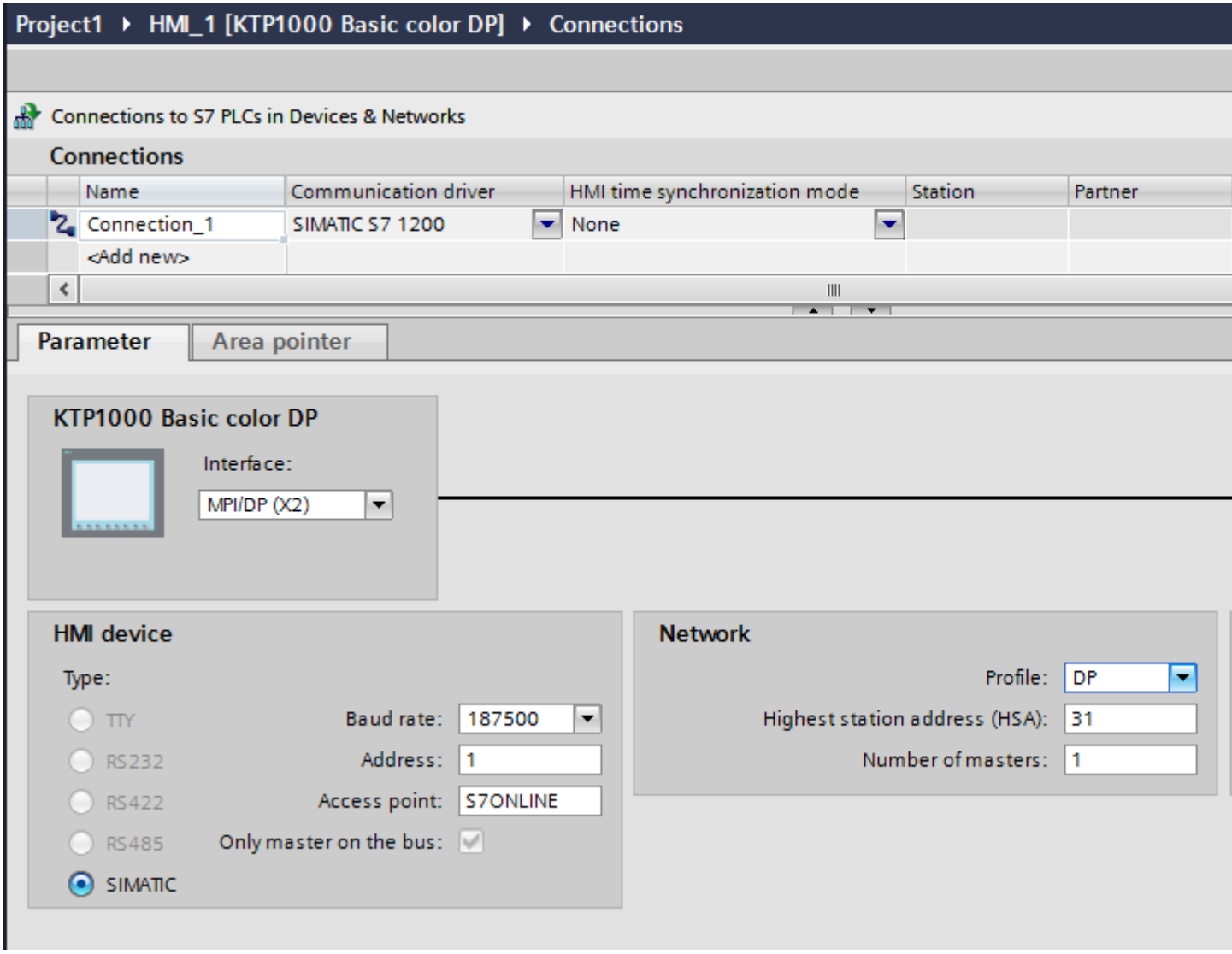
Creating a connection (Page 391)

PROFIBUS (Page 395)

PROFIBUS

Introduction

You assign the parameters for the HMI device and the PLC in the network under "Parameters".



HMI device

- "Access point"
  Select the logical device name of the HMI device under "Access point".

PLC

- "Address"
  Enter the station address of the CPU under "Address".

See also

Communication with SIMATIC S7-1200 (Page 298)
Creating a connection (Page 391)
TCP/IP (Page 393)

2.7.8 Configuring time synchronization

2.7.8.1 Time synchronization

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

Communication with SIMATIC S7-1200 (Page 298)
Time synchronization restrictions (Page 397)
Configuring time synchronization for integrated connections (Page 397)
Configuring time synchronization for non-integrated connections (Page 398)

2.7.8.2 Time synchronization restrictions

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	-
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.

See also

Communication with SIMATIC S7-1200 (Page 298)

Time synchronization (Page 396)

Configuring time synchronization for integrated connections (Page 397)

Configuring time synchronization for non-integrated connections (Page 398)

2.7.8.3 Configuring time synchronization for integrated connections

Introduction

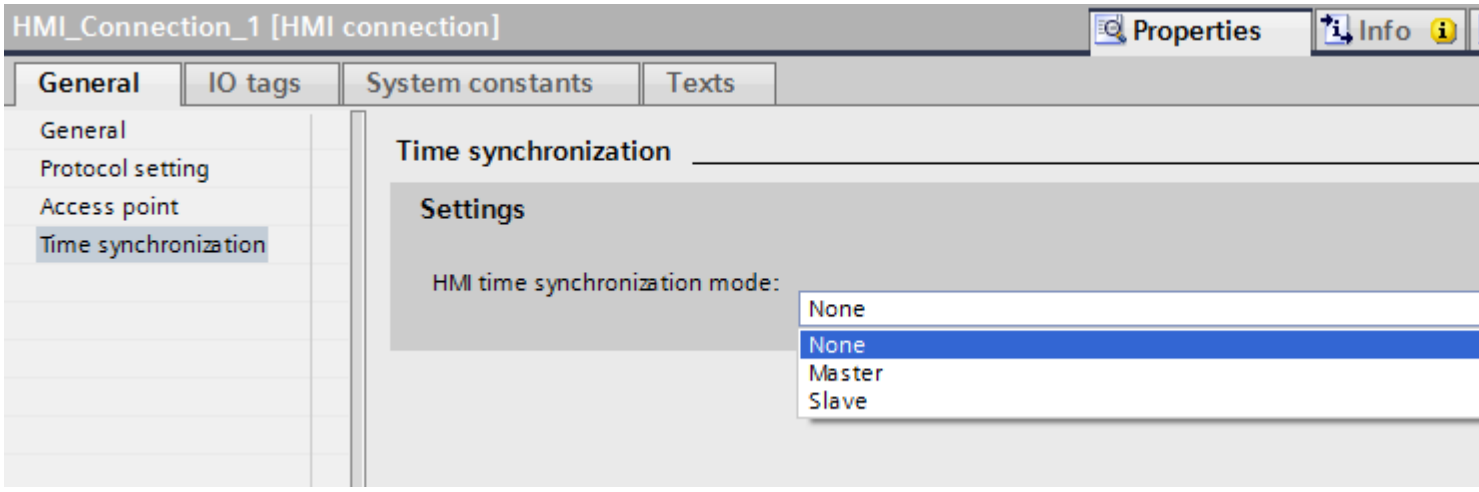
You configure time synchronization for an integrated connection in the "Devices & Networks" editor.

Requirements

- An HMI connection between an HMI device and a SIMATIC S7 1200 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

Communication with SIMATIC S7-1200 (Page 298)

Time synchronization (Page 396)

Time synchronization restrictions (Page 397)

Configuring time synchronization for non-integrated connections (Page 398)

2.7.8.4 Configuring time synchronization for non-integrated connections

Introduction

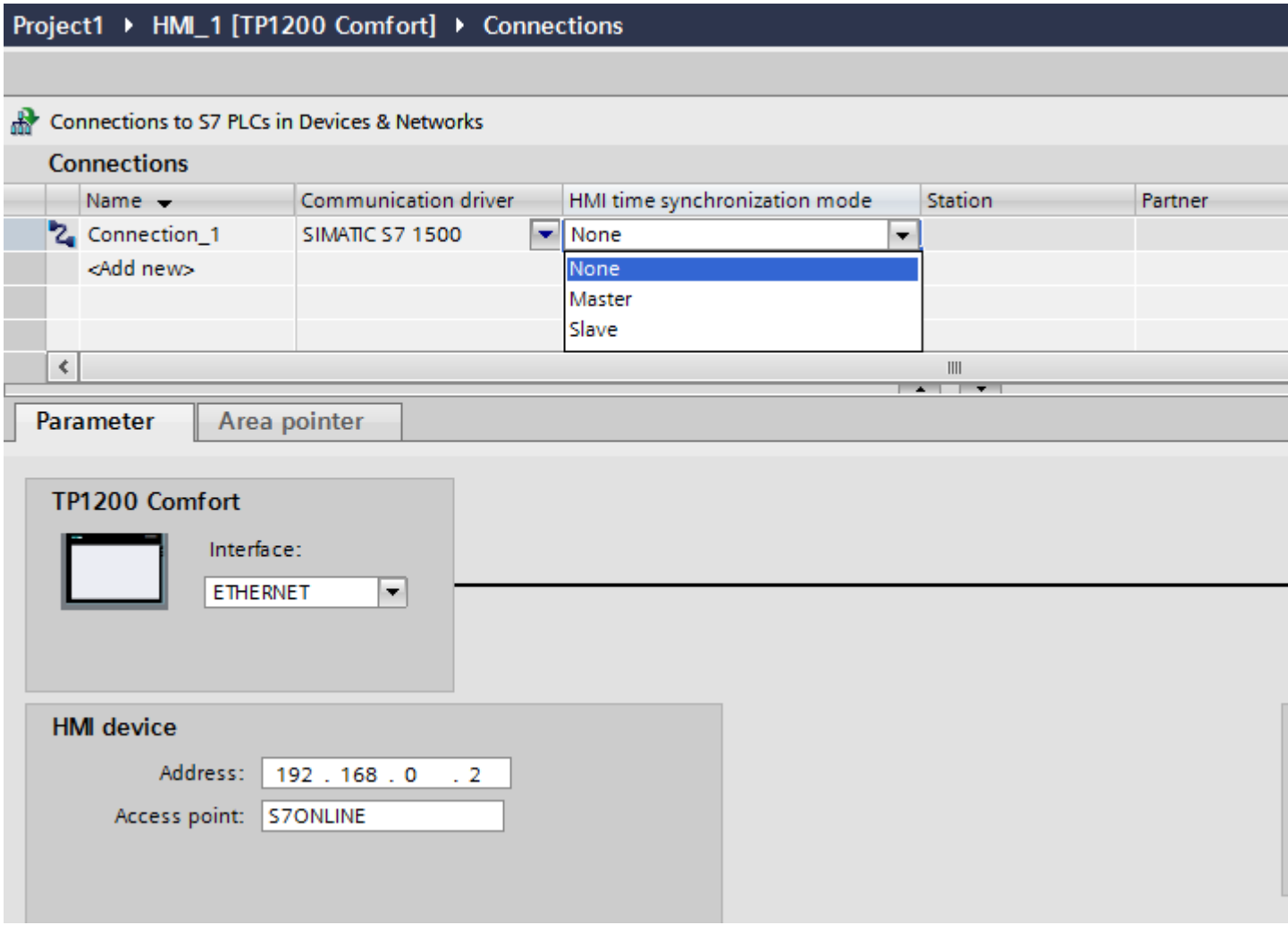
You configure time synchronization for a non-integrated connection in the "Connections" editor.

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 1200" 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

Communication with SIMATIC S7-1200 (Page 298)

Time synchronization (Page 396)

Time synchronization restrictions (Page 397)

Configuring time synchronization for integrated connections (Page 397)

2.7.9 Querying the PLC status

Introduction

In an S7 connection, you can query the status of the PLC using a special tag whose name corresponds to the schema "`@<connection name>@OpStateConfiguredAddress`".

In Runtime this tag is automatically updated every 5 seconds. The values correspond to the state of the PLC.

- 0: Not connected
- 4: Stop
- 8: Run

Querying the PLC status

To query the status of a PLC, proceed as follows:

1. Establish a connection with the communication driver "SIMATIC S7 1500" or "SIMATIC S7 1200". Assign the name "MyConnection" to this connection.
2. Create the tag "@MyConnection@OpStateConfiguredAddress" and assign the data type "UDInt" and the connection "MyConnection" to it.
3. Select "<Absolute access>" for the access mode.

Note

The configured address has no relevance in Runtime. However, it must correspond to a valid value so as not to cause an error when compiling the project.

2.8 Communicating with SIMATIC S7 300/400

2.8.1 Communication with SIMATIC S7 300/400

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.

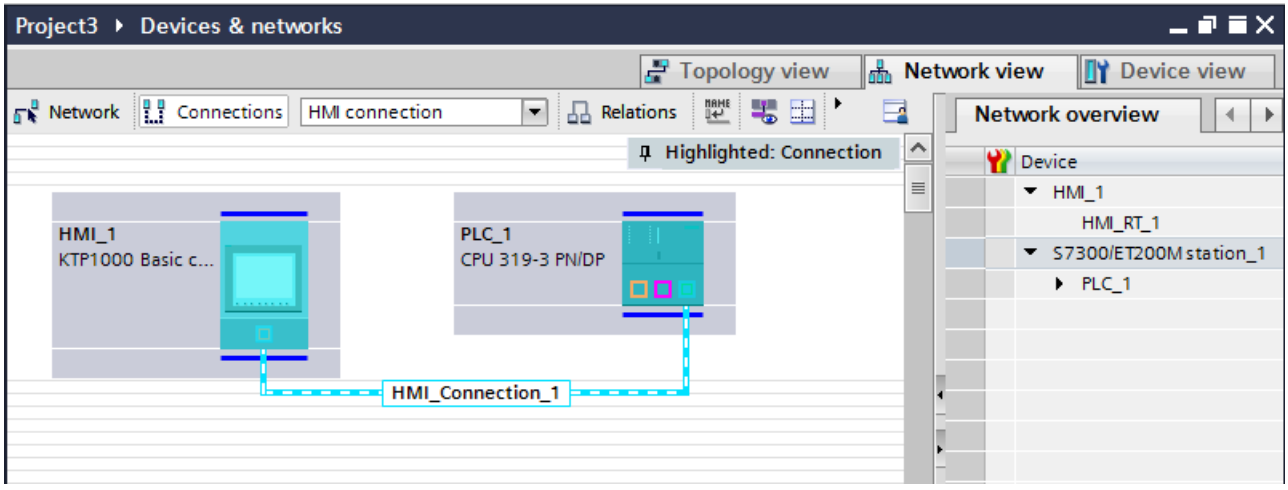
2.8.2 Communication via PROFINET

2.8.2.1 Configuring an HMI connection

Communication via PROFINET

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 401)

Configuring an HMI connection via PROFINET (Page 403)

Configuring an HMI connection via PROFINET

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

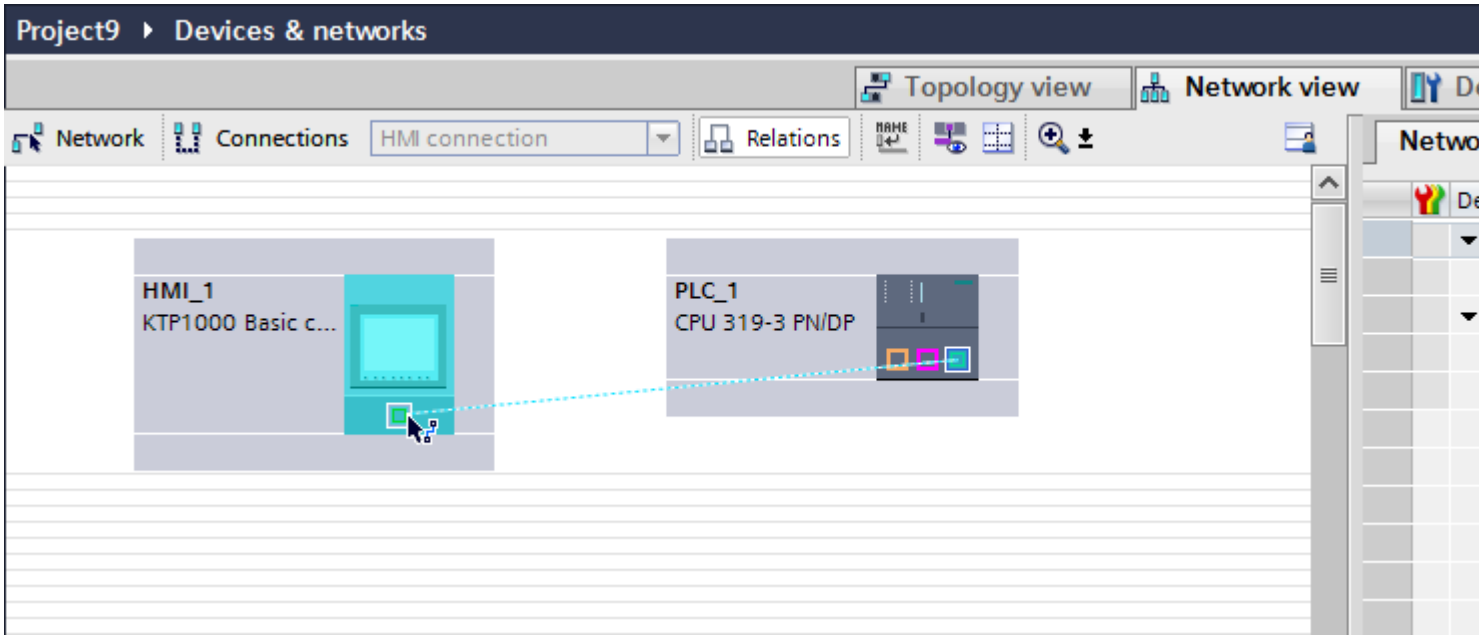
The following communication partners are created in the "Devices & Networks" editor:

- 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 415)".

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 401)

Communication via PROFINET (Page 402)

PROFINET parameters (Page 415)

2.8.2.2 Configuring an HMI connection

Communication via PROFINET

Communication via PROFINET

This section describes the communication between a WinCC Runtime and the SIMATIC S7 300/400 PLC via PROFINET.

You can use the following WinCC Runtimes as an HMI device:

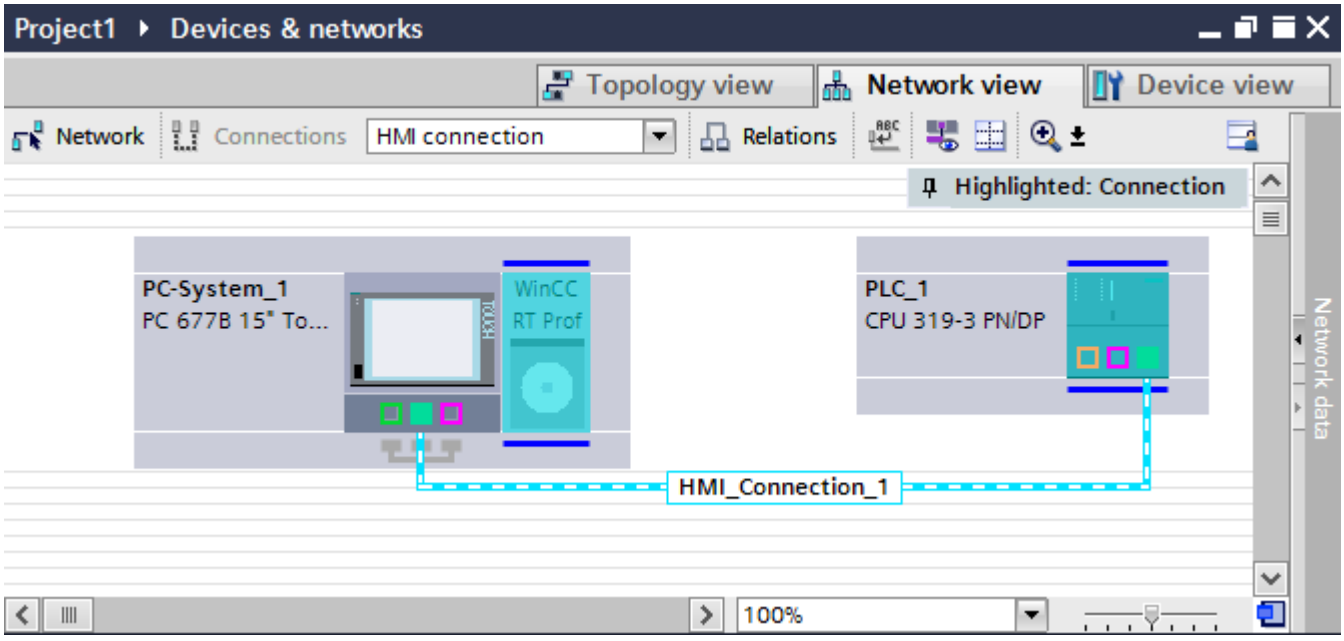
- WinCC RT Advanced
- WinCC RT Professional

WinCC Runtime as HMI device

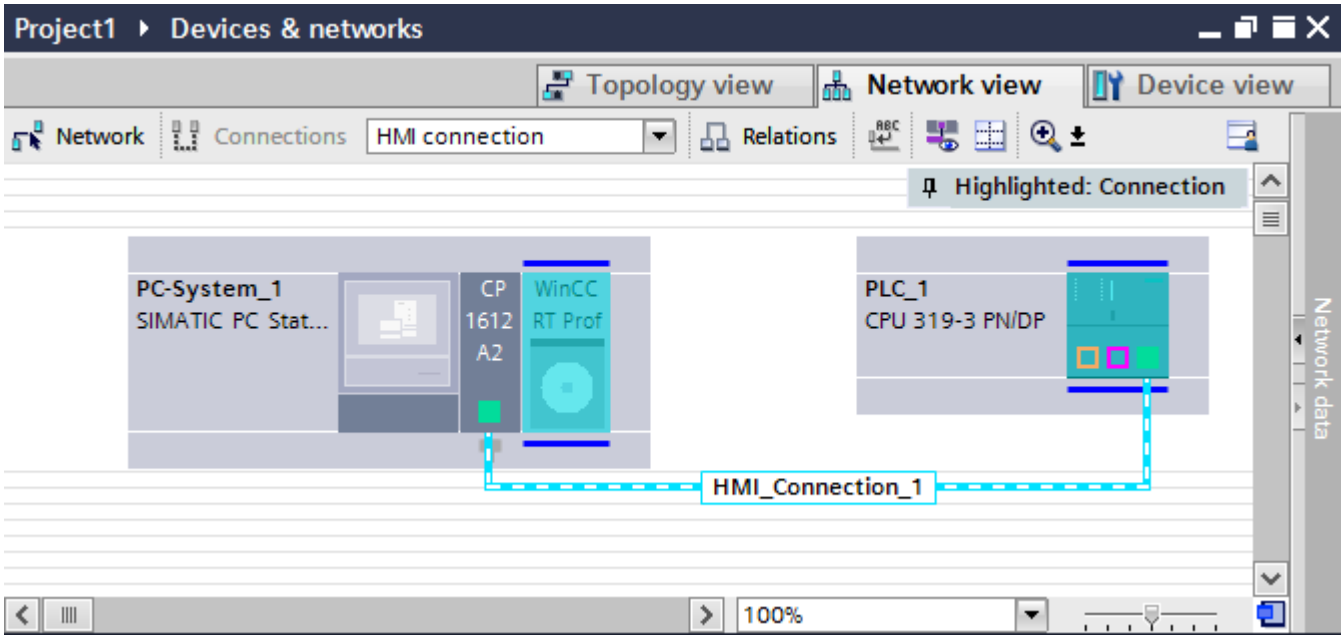
You configure the HMI connections between a WinCC Runtime and SIMATIC S7 300/400 in the "Devices & Networks" editor.

If you want to use a WinCC Runtime as an HMI device, you have the following options:

1. You create a SIMATIC PC and configure a WinCC Runtime on this SIMATIC PC.
   In this way you use the SIMATIC PC with a WinCC Runtime as an HMI device.



2. You create a WinCC Runtime and configure a communication processor to the Runtime.
   In this way you use your configuration PC with a WinCC Runtime as an HMI device.



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 401)

Configuring an HMI connection via PROFINET with a SIMATIC PC (Page 407)

Configuring an HMI connection via PROFINET with a PC (Page 410)

Configuring an HMI connection via PROFINET with a SIMATIC PC

Introduction

You configure an HMI connection between HMI devices and 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.

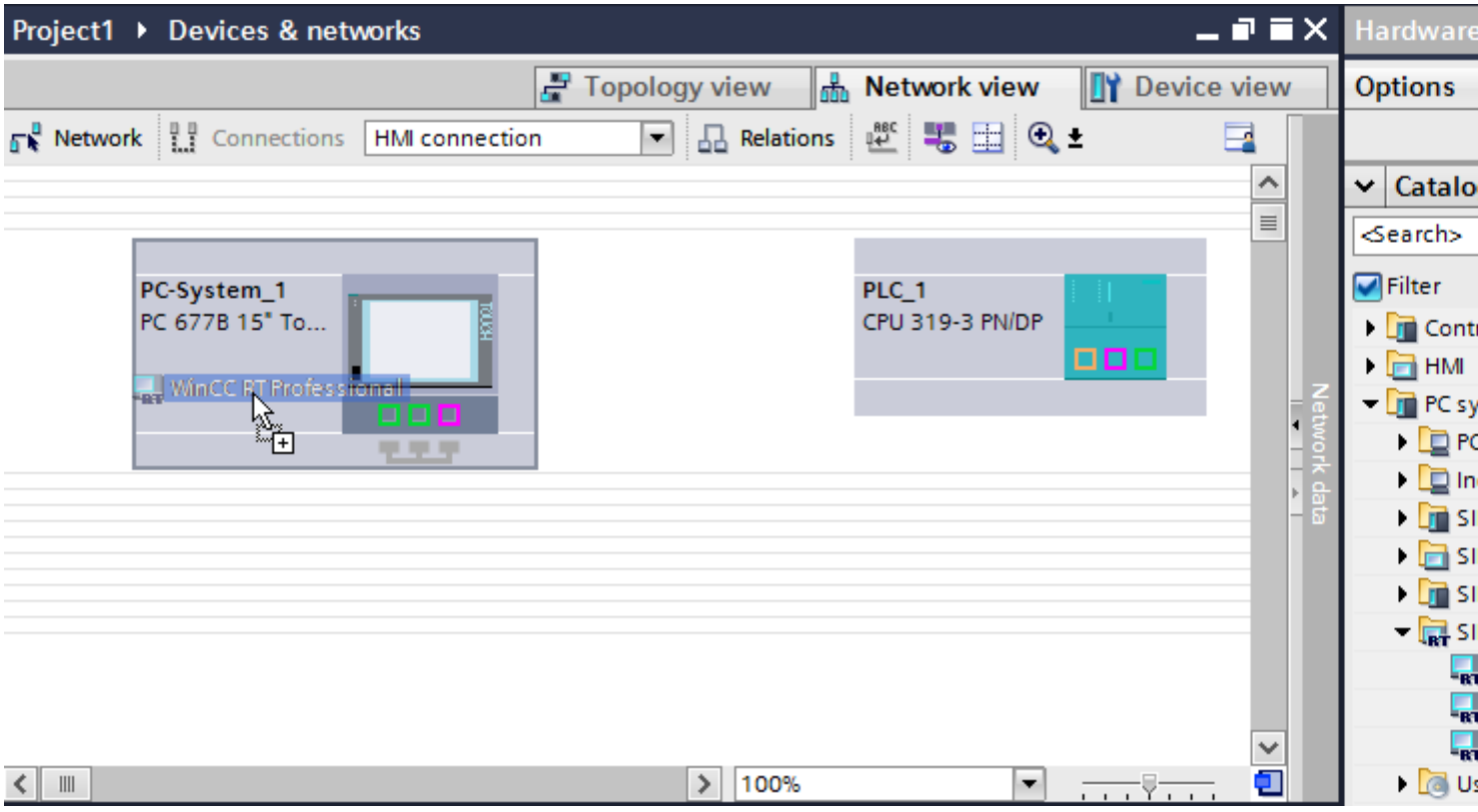
Requirements

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

- SIMATIC S7 300/400 with PROFINET interface
- SIMATIC PC with PROFINET interface

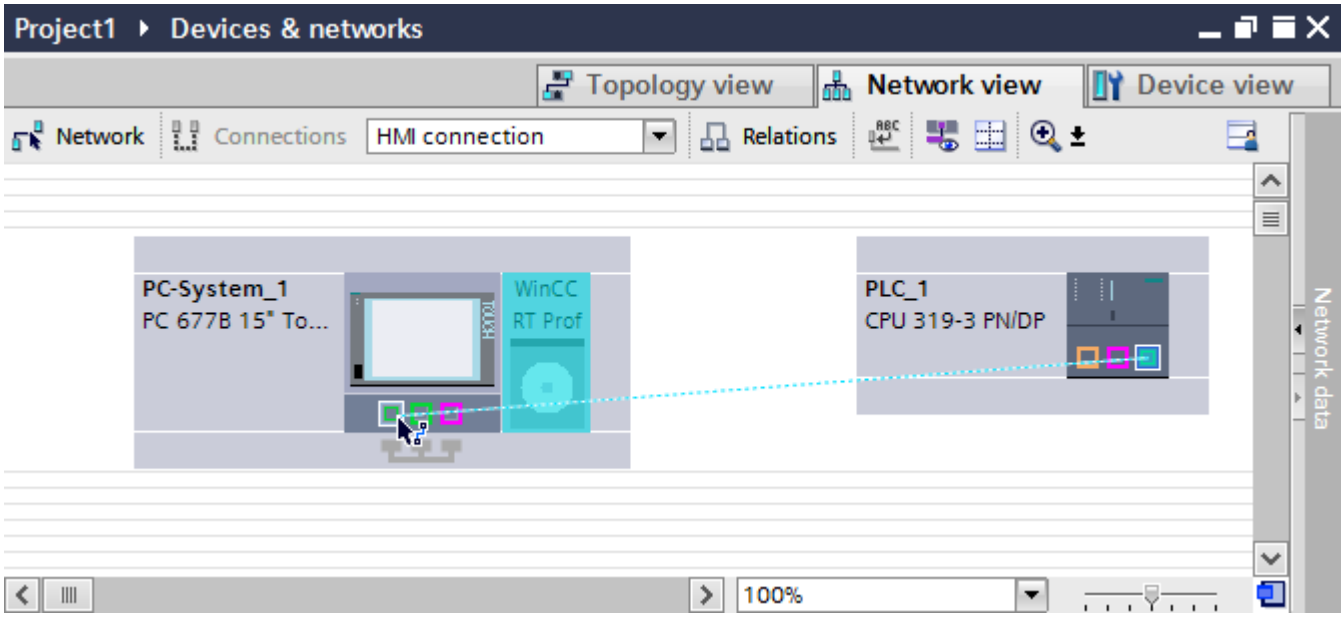
Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed in the network view.
2. Use a drag-and-drop operation to move a WinCC Runtime from the hardware catalog to the PC.



3. Click the "Connections" button and select "HMI connection" for the connection type.
   The devices available for connection are highlighted in color.

4. 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 PC.



5. Click the connecting line.
6. Click "Highlight HMI connection" and select the HMI connection.
   The connection is displayed graphically in the Inspector window.
7. 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 415)".

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 401)

Communication via PROFINET (Page 405)

Configuring an HMI connection via PROFINET with a PC (Page 410)

PROFINET parameters (Page 415)

Configuring an HMI connection via PROFINET with a PC

Introduction

You configure an HMI connection between an HMI device 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.

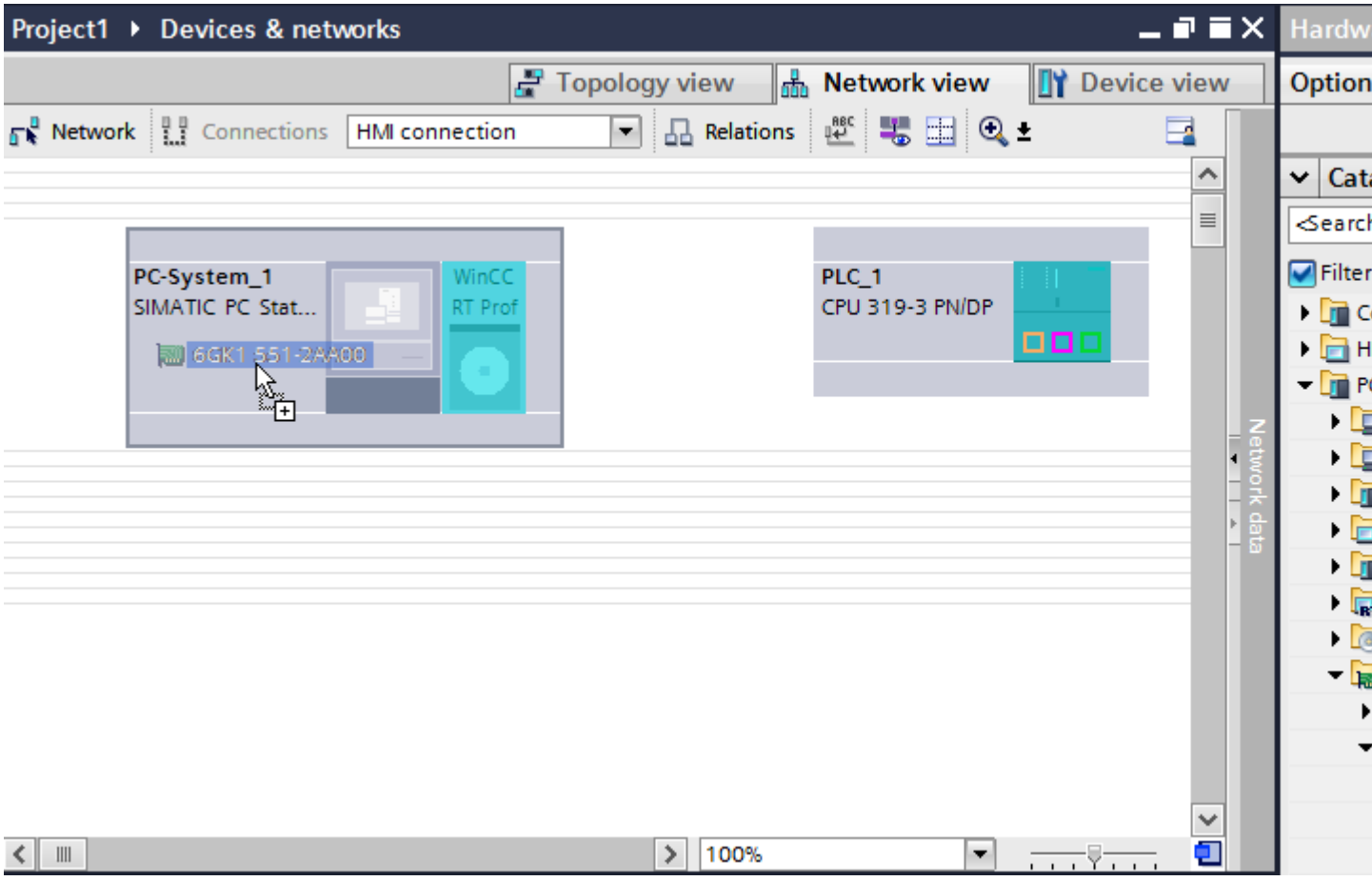
Requirements

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

- SIMATIC S7 300/400 with PROFINET interface
- PC station with WinCC RT Advanced or WinCC RT Professional

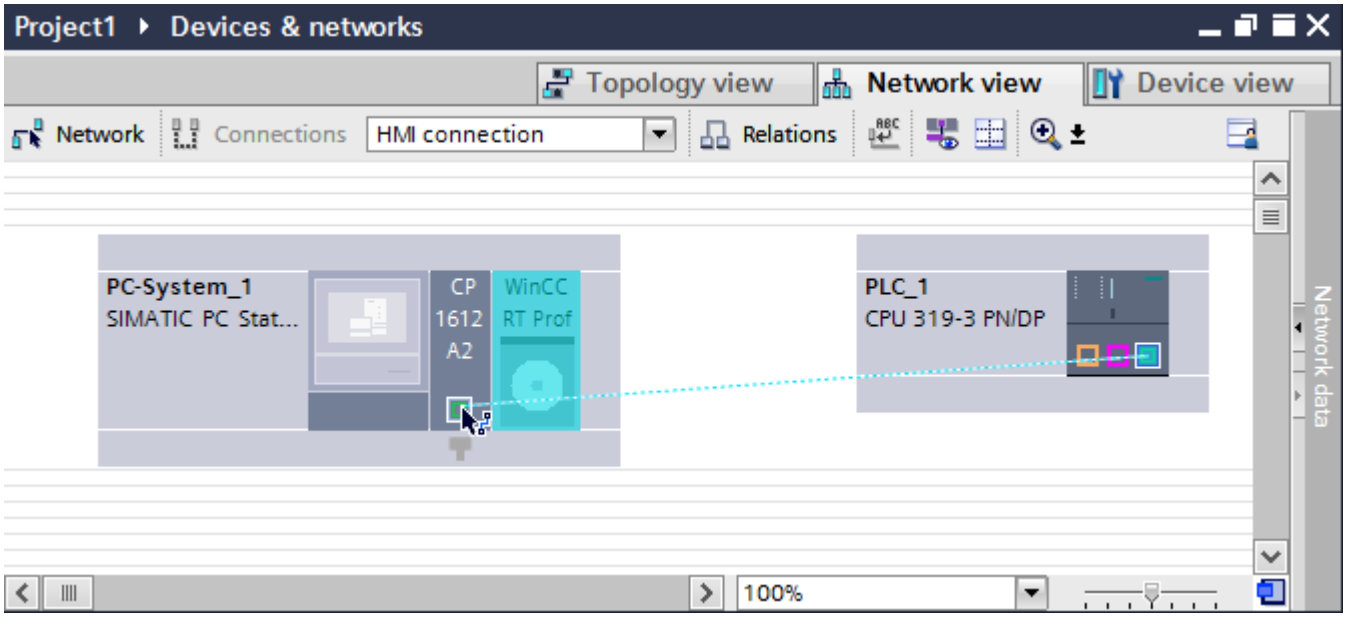
Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed in the network view.
2. Use a drag-and-drop operation to move a PROFINET-capable communication processor from the hardware catalog to the WinCC Runtime.



3. Click the "Connections" button and select "HMI connection" for the connection type.
   The devices available for connection are highlighted in color.

4. Click the PROFINET interface of the PLC and use a drag-and-drop operation to draw a connection to the PROFINET interface of the communication processor.



5. Click the connecting line.
   The connection is displayed graphically in the Inspector window.
6. Click "Highlight HMI connection" and select the HMI connection.
   The connection is displayed graphically in the Inspector window.
7. 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 415)".

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 401)

Communication via PROFINET (Page 405)

Configuring an HMI connection via PROFINET with a SIMATIC PC (Page 407)

PROFINET parameters (Page 415)

2.8.2.3 Named connections

Named connections

Principle of operation

A Named Connection is an integrated, symbolic connection. By this means, WinCC can access non-redundant connections via a symbolic connection name.

The symbolic connections are needed, for example, for communication with the SIMATIC S7-400 in H/F systems.

SDB

The connection information items are saved in system data blocks (SDB). The SDBs are then transferred to the runtime station.

The connection name and the application name are saved in WinCC project for referencing connection information.

The station manager (S7RTM) on the PC station is a requirement for Named Connections.

If no online connection to the PC station exists, then use XDB files. You generate the XDB files with WinCC. The XDB file contains connection information.

The connection names, connection parameters and the application names are stored in a database (*.XDB).

Note

There should only be one XDB file per communication participant in the WinCC system. Hence, a XDB file should not be copied and used on multiple WinCC computers.

Configuring a connection via "Named connections"

Introduction

For setting up a logical connection, one of the symbolic connection names listed in the "Connection name" field is assigned to a selected application name.

The symbolic connection names and application names are configured in STEP 7.

Named connections can only be configured for integrated connections.

Note

In the case of routed HMI connections between WinCC Runtime Professional and the S7-300/400 controllers, "Named Connection" must be used irrespective of the router.

Requirements

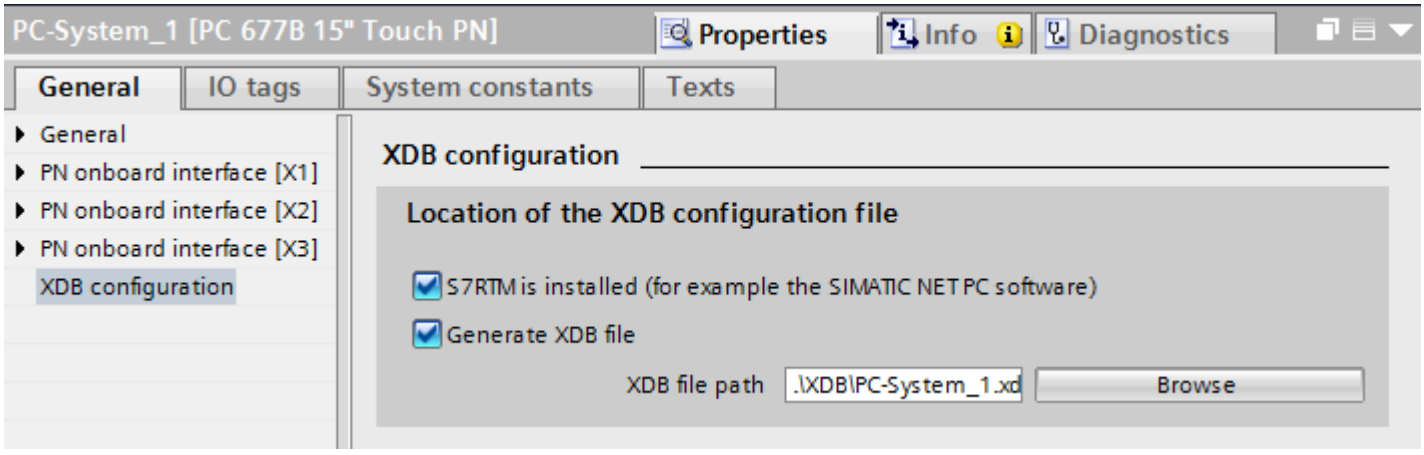
A "Named connection" has been created in STEP 7.

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

- SIMATIC PC with WinCC RT Professional
- SIMATIC S7 300/400

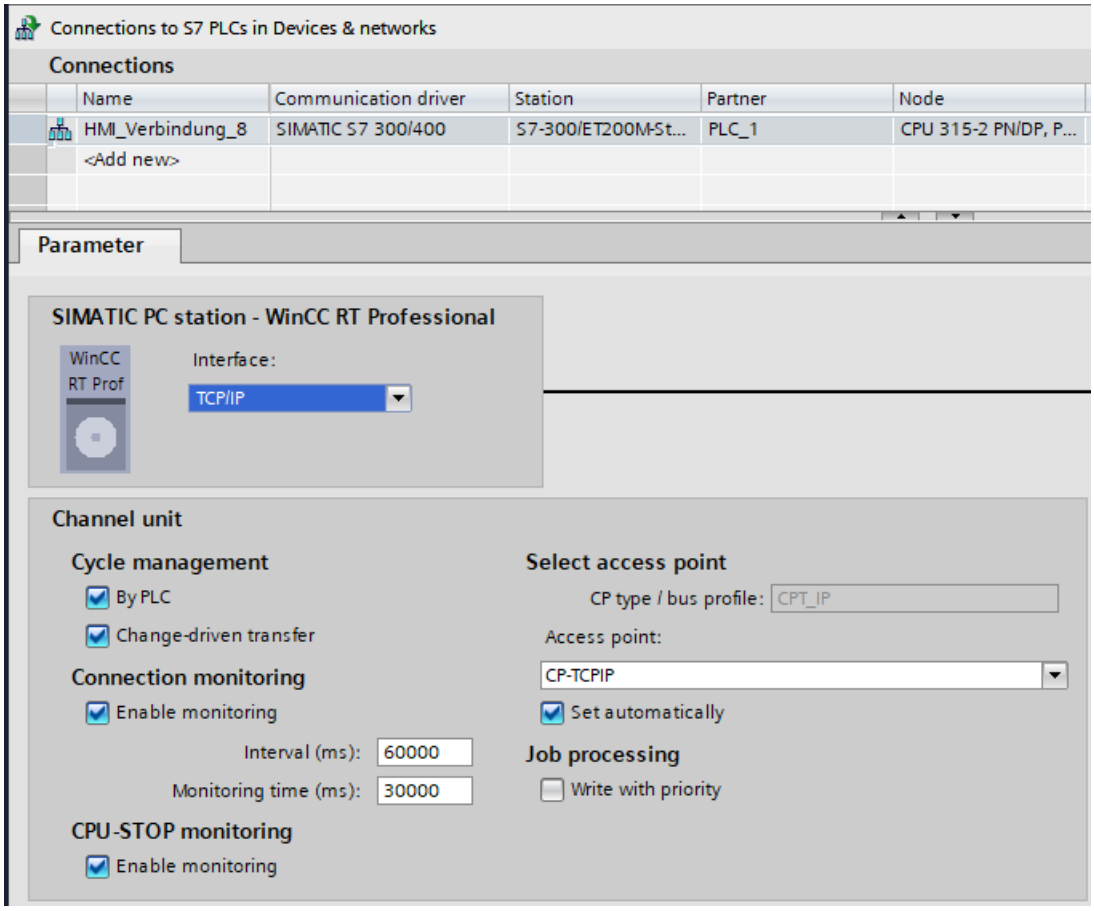
Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed in the network view.
2. Click the HMI device in the "Network view".
3. Select the entry "S7RTM is installed" in the Inspector window under "Parameters > XDB configuration".



   If you have not installed S7RTM, select "Generate XDB file". Then select the path in which the XDB file will be stored.
4. Click the "Connections" button and select "HMI connection" for the connection type.
5. Click the PROFINET interface of the PLC and use a drag-and-drop operation to draw a connection to the PROFINET interface of the PC.
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. Select the interface "Named Conn." in the editor "Connections" > "Parameter" . "Interface" of WinCC RT Professional.



Note

The application and connection names can also be entered manually, e.g. if an XDB file is not available for the symbolic connection name or if the project is to be transferred to another computer. It is necessary to check the correct writing of the name in STEP 7 because there is no name validation in the CS mode.

2.8.2.4 PROFINET parameters

PROFINET parameters for the HMI connection

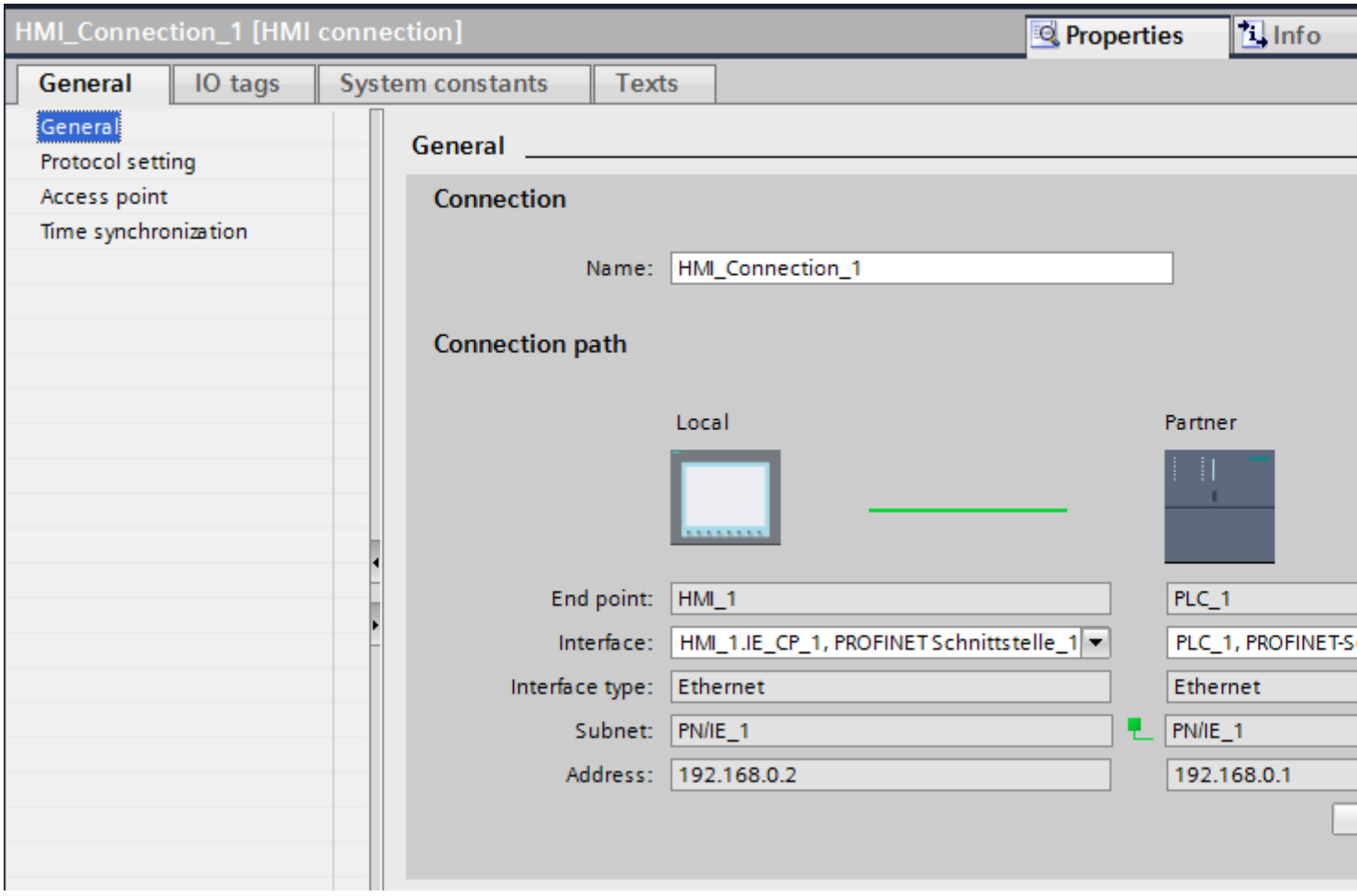
PROFINET parameters for the HMI connection

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 with SIMATIC S7 300/400 (Page 401)

PROFINET parameters for the HMI device (Page 417)

PROFINET parameters for the PLC (Page 419)

Configuring Industrial Ethernet (Page 421)

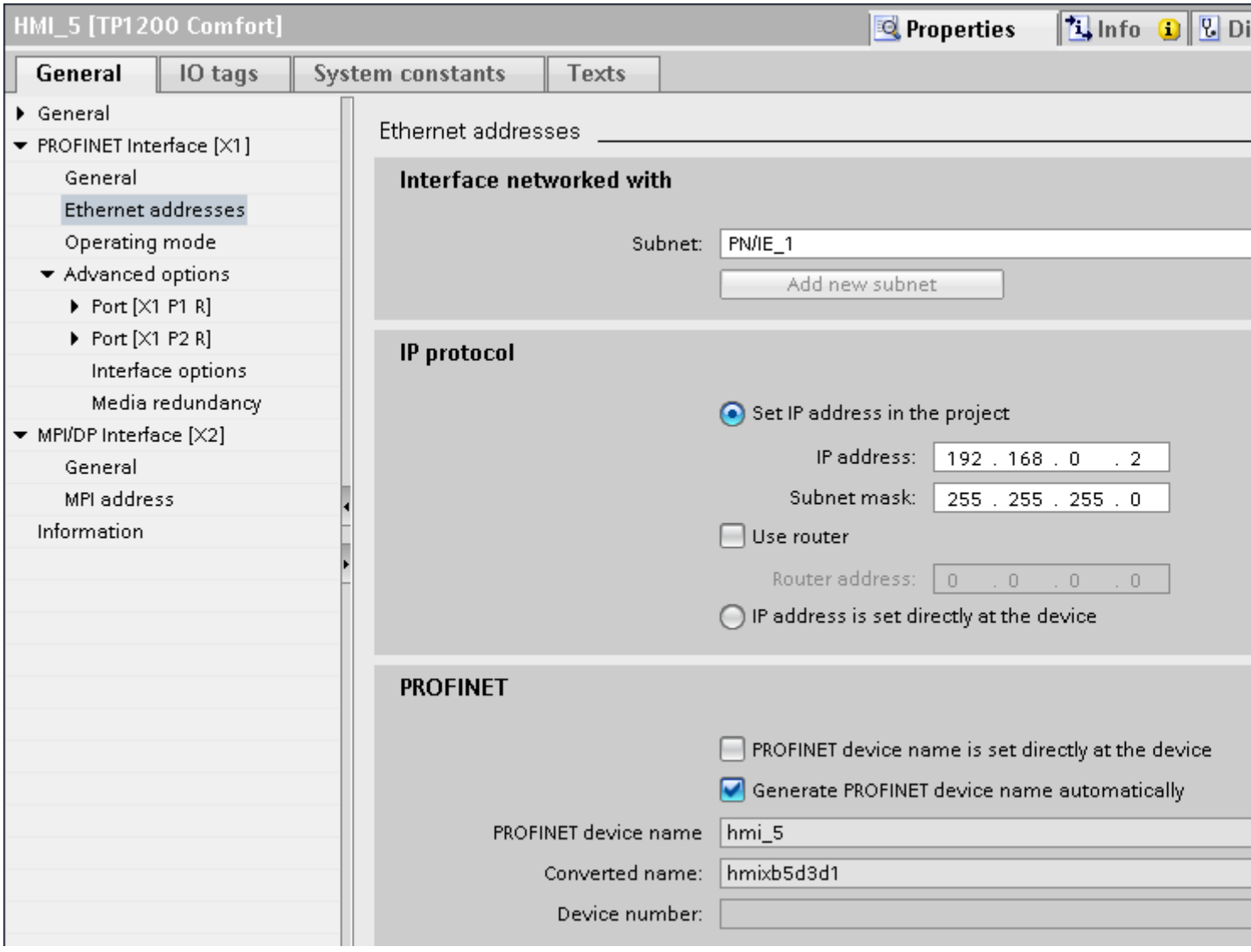
PROFINET parameters for the HMI device

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:
  - Mobile Panel 177 PN
  - Mobile Panel 177 DP

- "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 300/400 (Page 401)

PROFINET parameters for the HMI connection (Page 415)

PROFINET parameters for the PLC (Page 419)

Configuring Industrial Ethernet (Page 421)

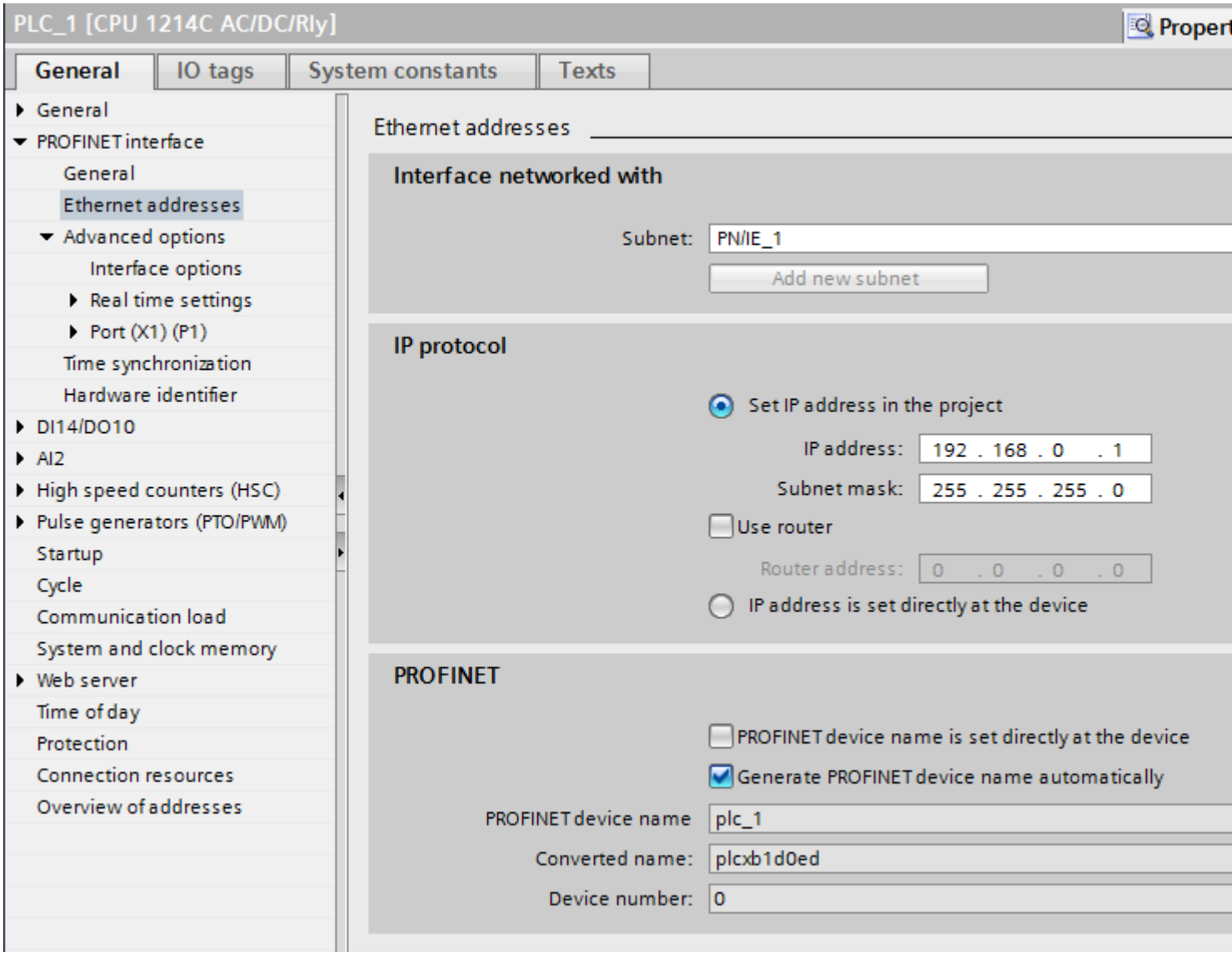
PROFINET parameters for the PLC

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 300/400 (Page 401)

PROFINET parameters for the HMI connection (Page 415)

PROFINET parameters for the HMI device (Page 417)

Configuring Industrial Ethernet (Page 421)

Configuring Industrial Ethernet

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 with SIMATIC S7 300/400 (Page 401)

PROFINET parameters for the HMI connection (Page 415)

PROFINET parameters for the HMI device (Page 417)

PROFINET parameters for the PLC (Page 419)

2.8.2.5 Port options

Setting the port options

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

Communication with SIMATIC S7 300/400 (Page 401)

Wiring rules for disabled autonegotiation (Page 424)

Boundaries at the port (Page 426)

Wiring rules for disabled autonegotiation

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:

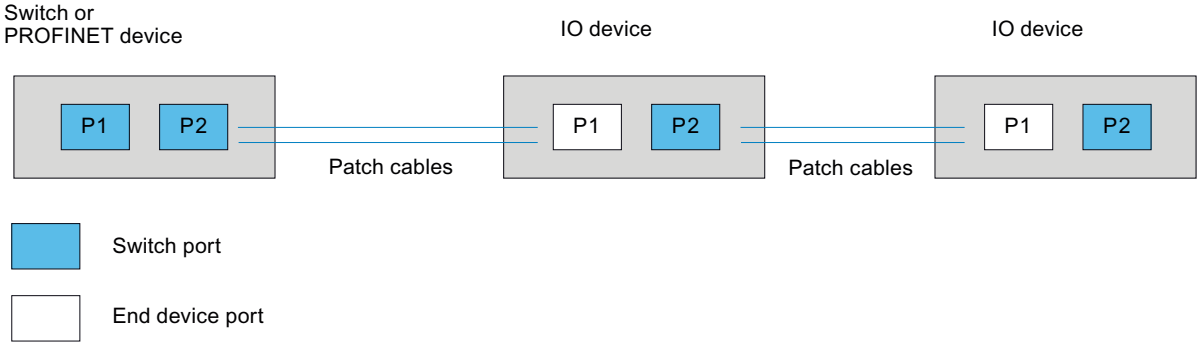
Type of port	PROFINET devices	Note
Switch port with crossed pin assignment	For IO devices: Port 2 For S7 CPUs with 2 ports: Ports 1 and 2	Crossed pin assignment means that the pin assignment for the ports for sending and receiving between the respective PROFINET devices is exchanged internally.
End device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

Communication with SIMATIC S7 300/400 (Page 401)

Setting the port options (Page 423)

Boundaries at the port (Page 426)

Boundaries at the port

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

Communication with SIMATIC S7 300/400 (Page 401)

Setting the port options (Page 423)

Wiring rules for disabled autonegotiation (Page 424)

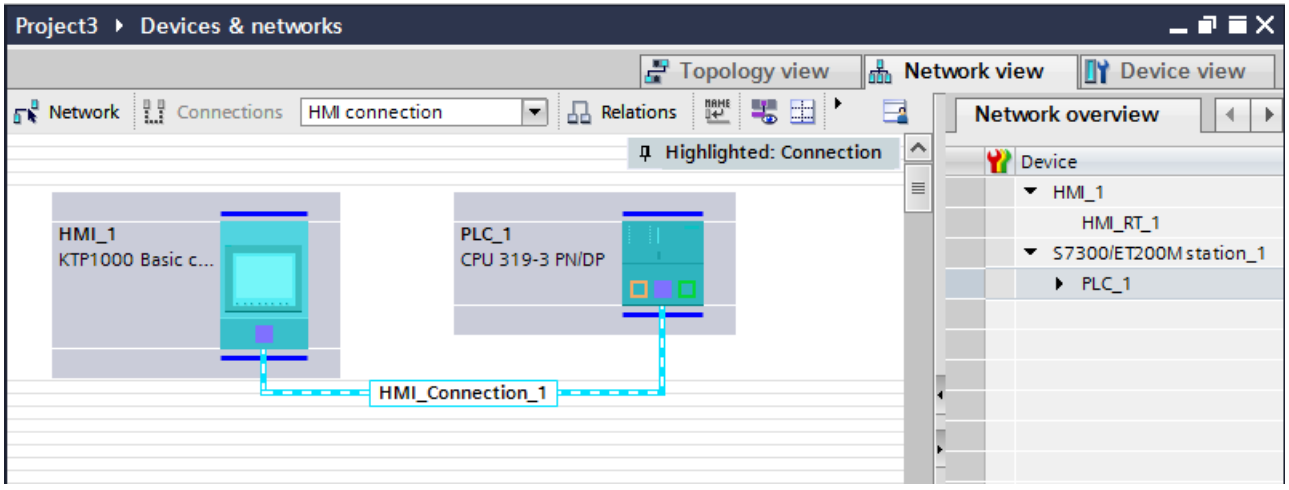
2.8.3 Communication via PROFIBUS

2.8.3.1 Configuring an HMI connection

Communication via PROFIBUS

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 401)

Configuring an HMI connection via PROFIBUS (Page 428)

Configuring an HMI connection via PROFIBUS

Introduction

You configure an HMI connection between HMI devices and a SIMATIC S7 300/400 via PROFIBUS in the "Devices & Networks" editor.

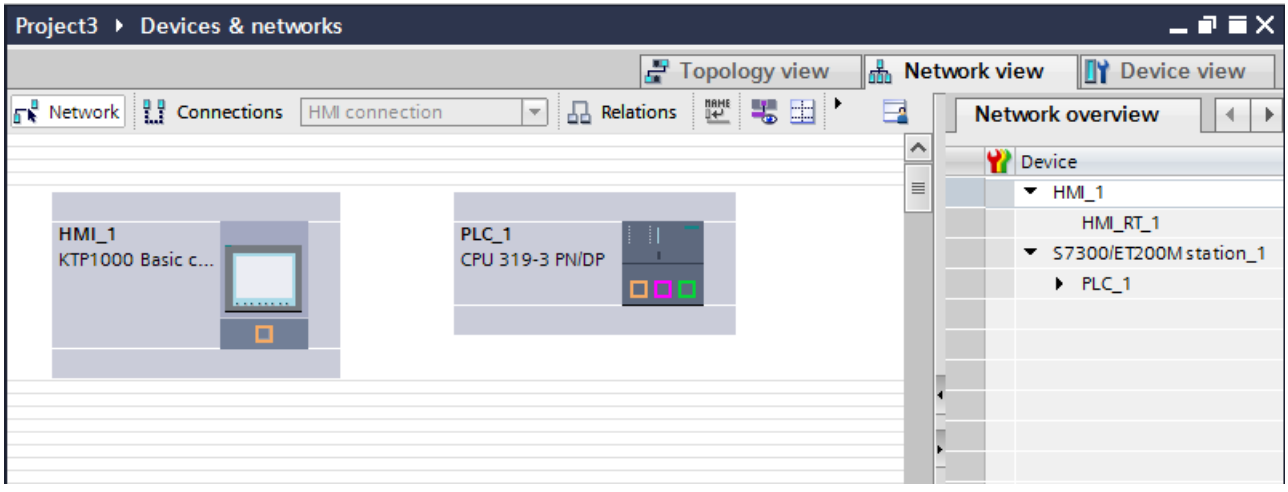
Requirements

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

- HMI device with MPI/DP interface
- SIMATIC S7 300/400 with PROFIBUS interface

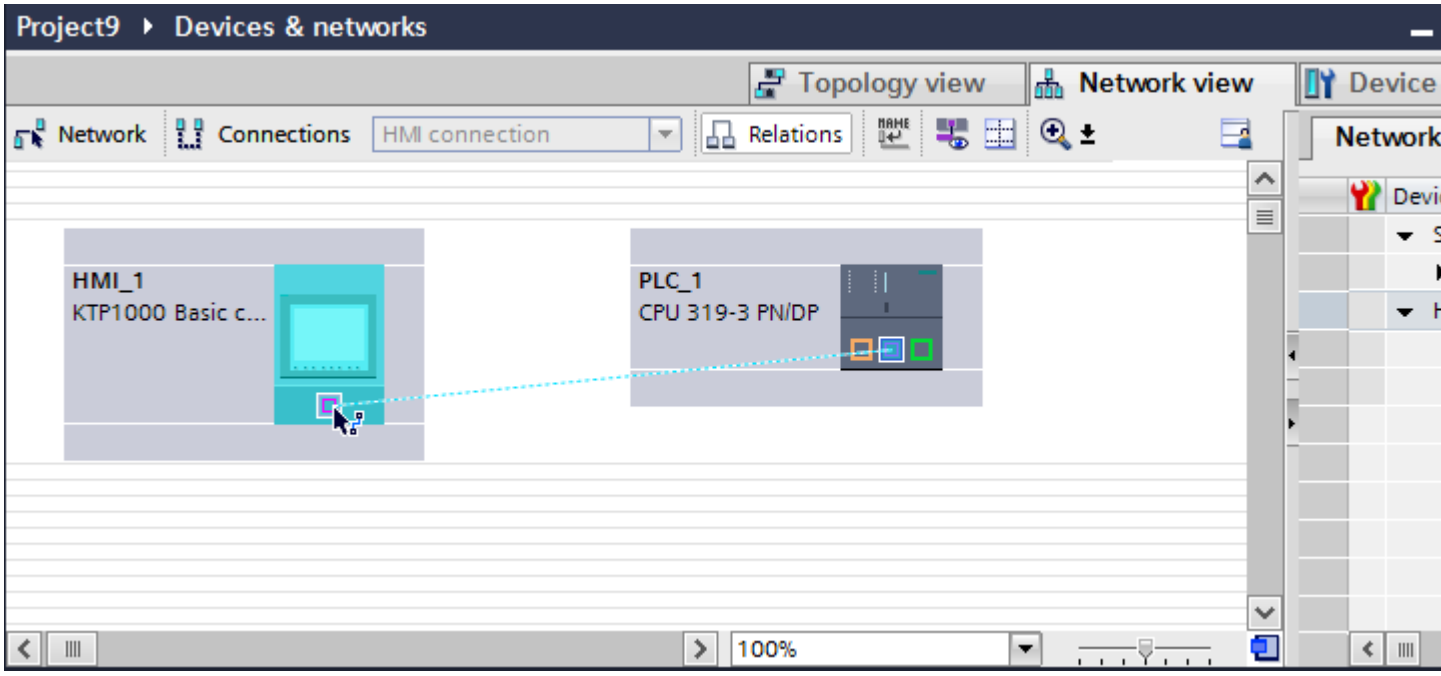
Procedure

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 437)" 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 401)

Communication via PROFIBUS (Page 427)

PROFIBUS parameters (Page 437)

2.8.3.2 Configuring an HMI connection

Communication via PROFIBUS

Communication via PROFIBUS

This section describes the communication between a WinCC Runtime and the SIMATIC S7 300/400 PLC via PROFIBUS.

You can use the following WinCC Runtimes as an HMI device:

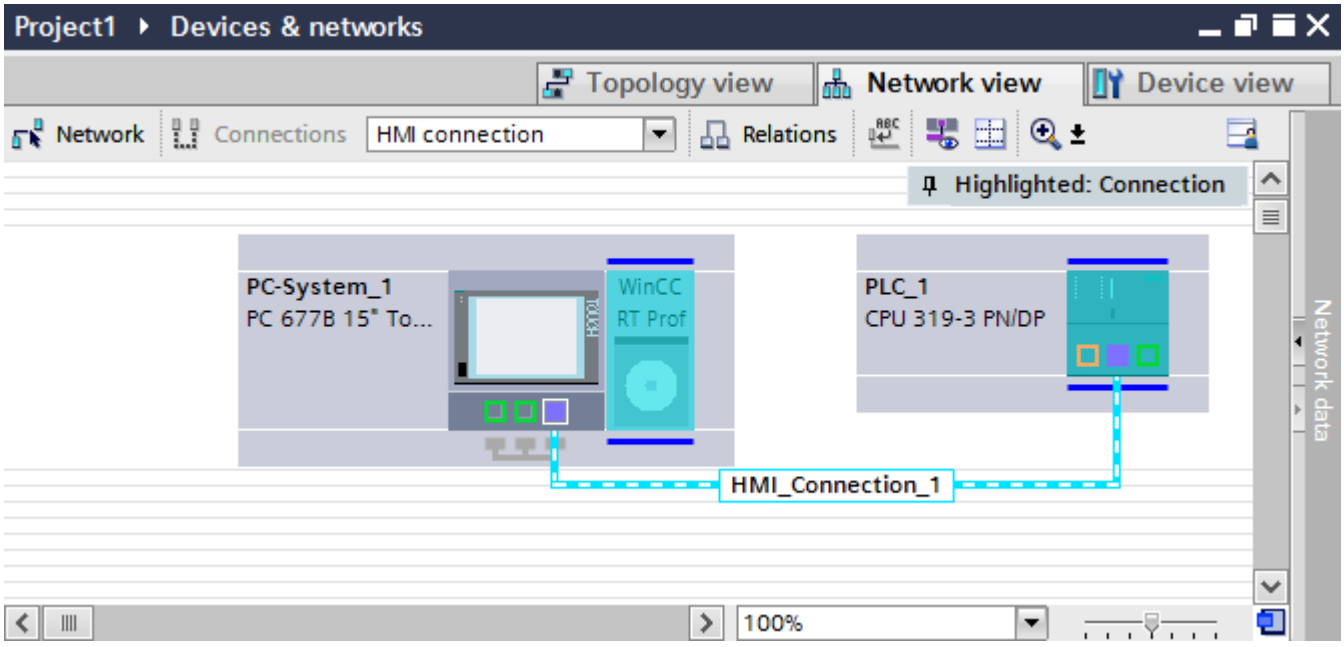
- WinCC RT Advanced
- WinCC RT Professional

WinCC Runtime as HMI device

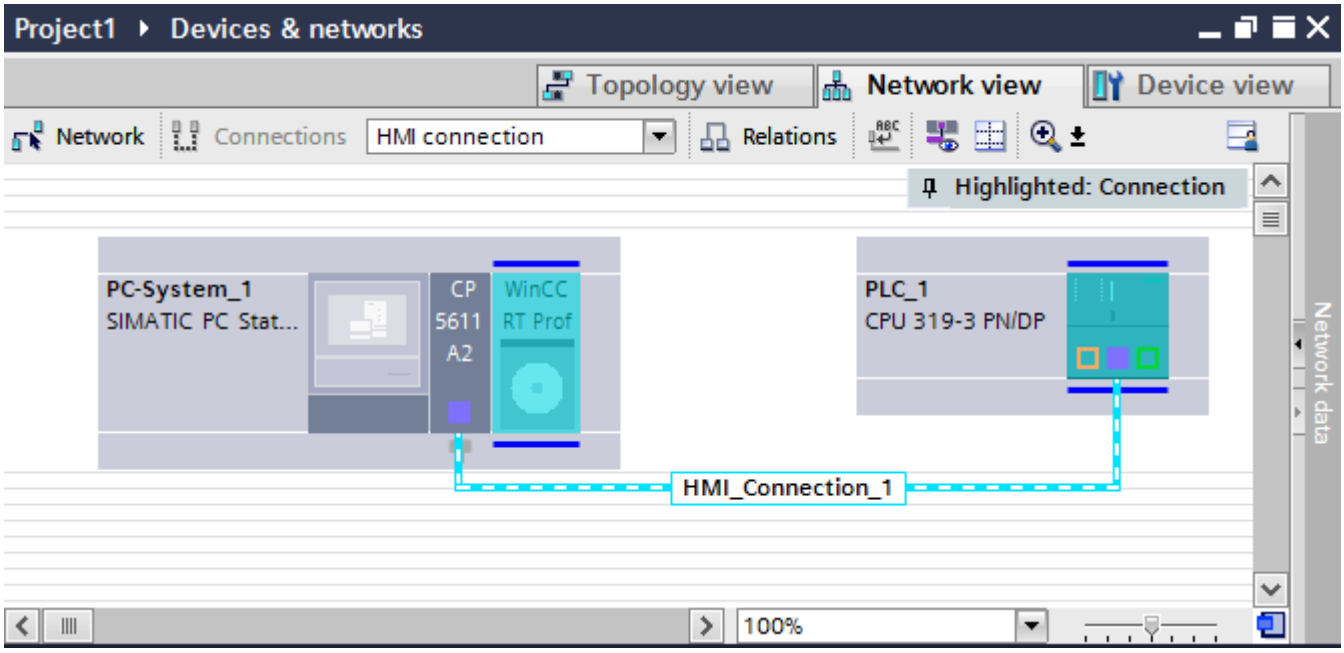
You configure the HMI connections between WinCC Runtime and SIMATIC S7 300/400 in the "Devices & Networks" editor.

If you want to use a WinCC Runtime as an HMI device, you have the following options:

1. You create a SIMATIC PC and configure a WinCC Runtime on this SIMATIC PC.
   In this way you use the SIMATIC PC with a WinCC Runtime as an HMI device.



2. You create a WinCC Runtime and configure a communication processor to the Runtime.
   In this way you use your configuration PC with a WinCC Runtime as an HMI device.



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 PROFIBUS in the "Devices & Networks" editor.

Connection in the "Connections" editor

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

See also

Communication with SIMATIC S7 300/400 (Page 401)

Configuring an HMI connection via PROFIBUS with a SIMATIC PC (Page 432)

Configuring an HMI connection via PROFIBUS with a PC (Page 435)

Configuring an HMI connection via PROFIBUS with a SIMATIC PC

Introduction

You configure an HMI connection between HMI devices and a SIMATIC S7 300/400 via PROFIBUS in the "Devices & Networks" editor.

Requirements

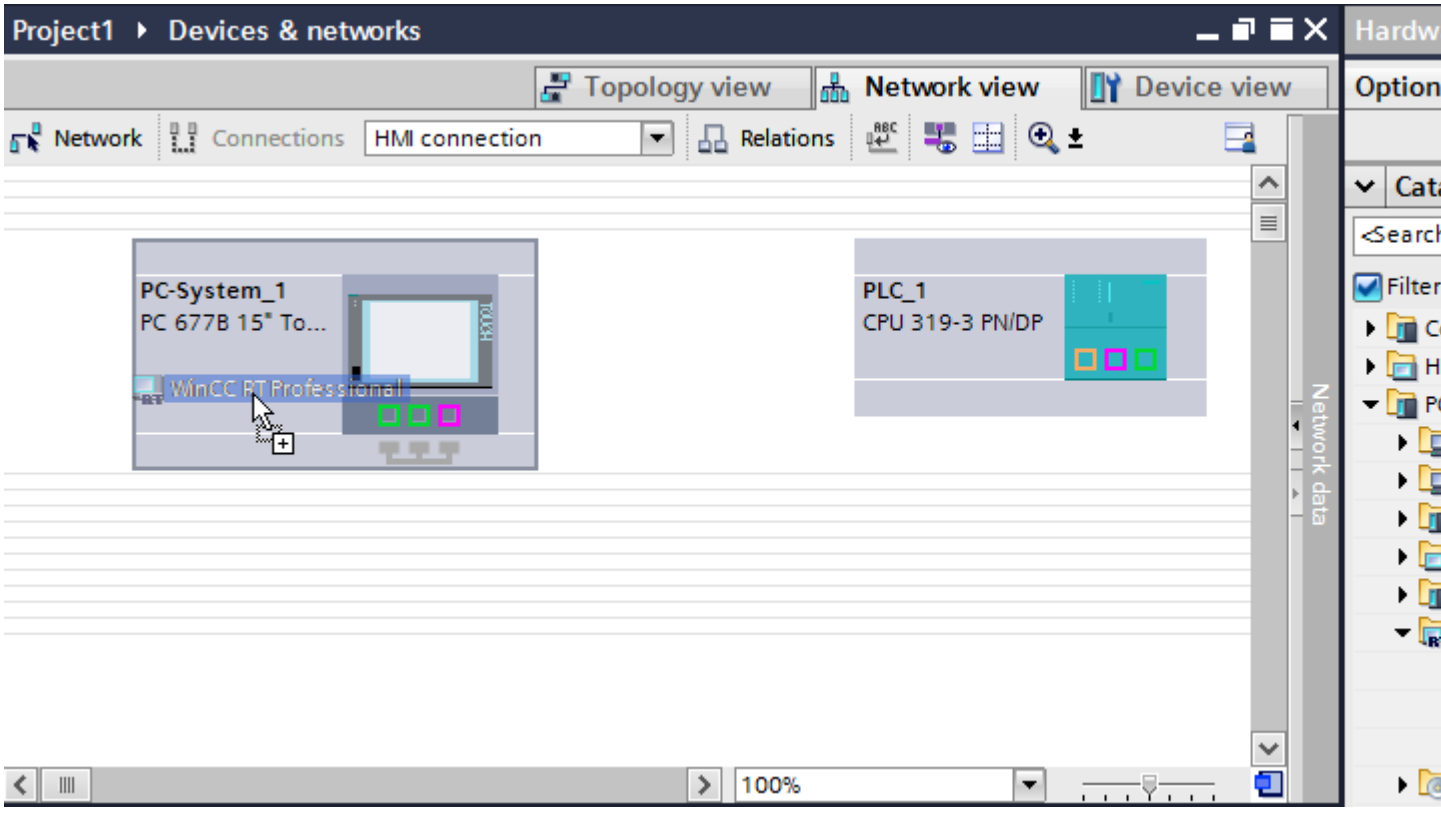
The following communication partners are created in the "Devices & Networks" editor:

- SIMATIC S7 300/400 with PROFIBUS interface
- SIMATIC PC with PROFIBUS interface

Procedure

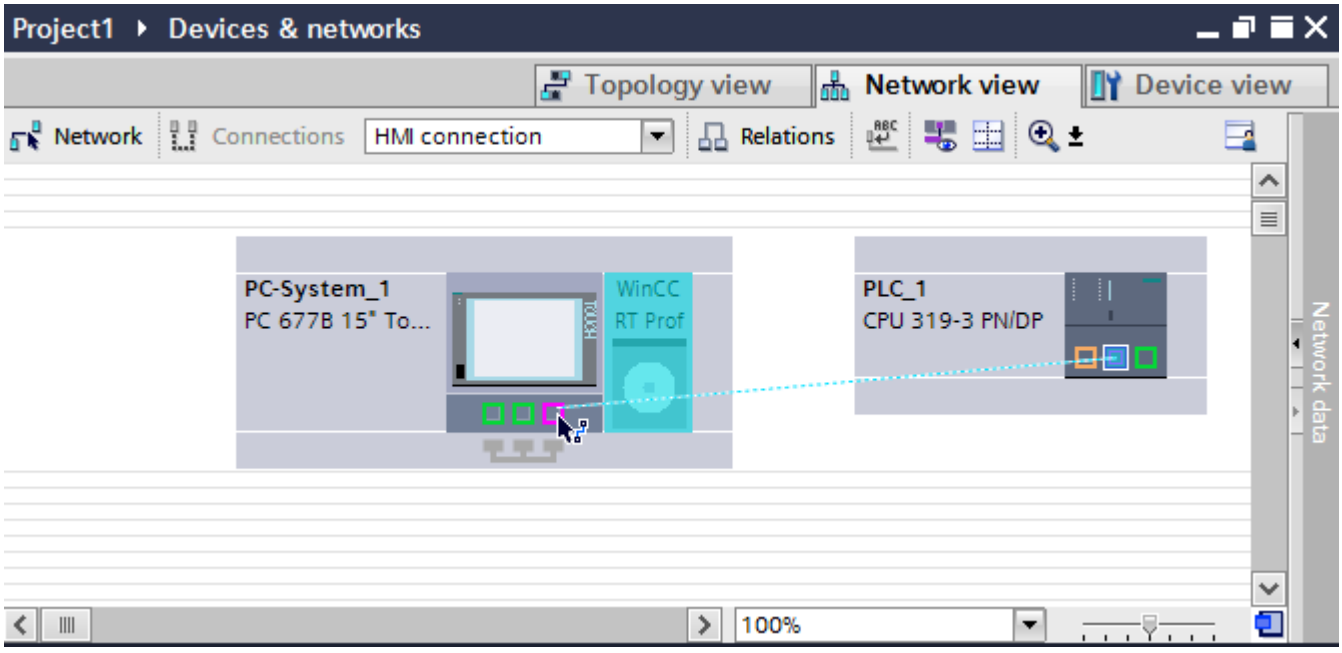
1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed in the network view.
2. Click the MPI interface of the PC and select "PROFIBUS" for the interface type in the Inspector window.

3. Use a drag-and-drop operation to move a WinCC Runtime from the hardware catalog to the PC.



4. Click the "Connections" button and select "HMI connection" for the connection type. The devices available for connection are highlighted in color.

5. Click the PROFIBUS interface of the PLC and use a drag-and-drop operation to draw a connection to the PROFIBUS interface of the PC.



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 437)".

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 401)

Communication via PROFIBUS (Page 430)

Configuring an HMI connection via PROFIBUS with a PC (Page 435)

PROFIBUS parameters (Page 437)

Configuring an HMI connection via PROFIBUS with a PC

Introduction

You configure an HMI connection between an HMI device and a SIMATIC S7-300/400 via PROFIBUS in the "Devices & Networks" editor.

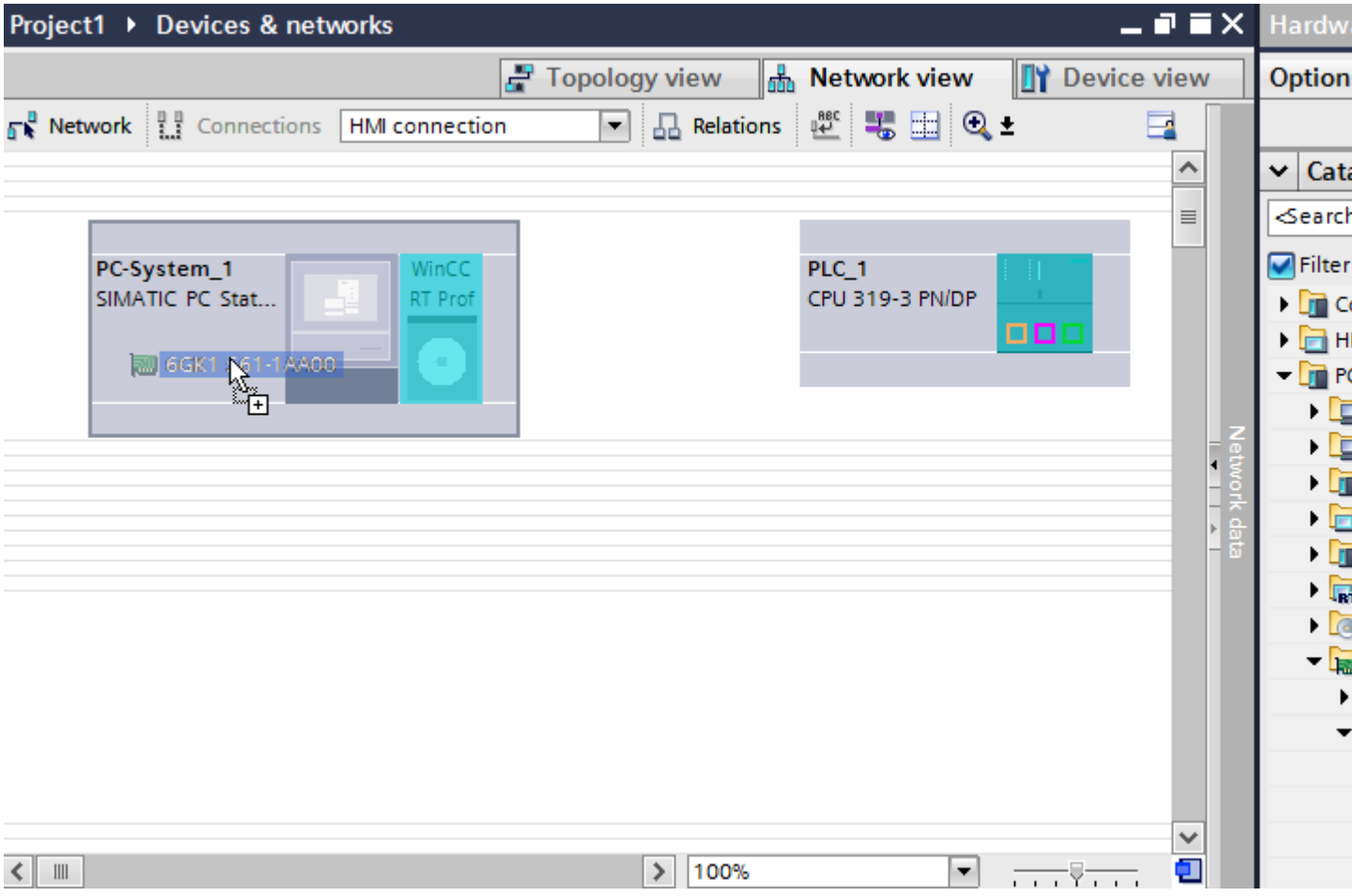
Requirements

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

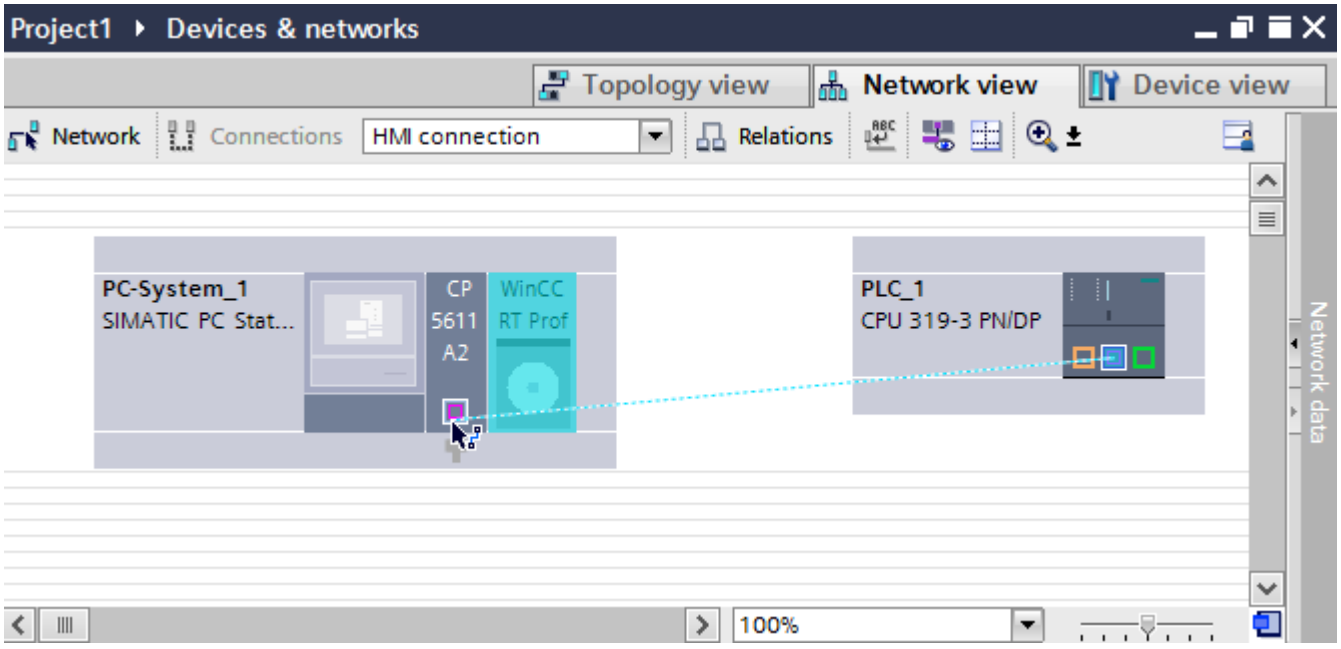
- SIMATIC S7-300/400 with PROFIBUS interface
- PC station with WinCC RT Advanced or WinCC RT Professional

Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed in the network view.
2. Use a drag-and-drop operation to move a PROFIBUS-capable communication processor from the hardware catalog to the WinCC Runtime.



3. Click the "Connections" button and select "HMI connection" for the connection type.
   The devices available for connection are highlighted in color.
4. Click the PROFIBUS interface of the PLC and use a drag-and-drop operation to draw a connection to the PROFIBUS interface of the communication processor.



5. Click the connecting line.
6. Click "Highlight HMI connection" and select the HMI connection.
   The connection is displayed graphically in the Inspector window.
7. 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 437)" 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 300/400 (Page 401)

Communication via PROFIBUS (Page 430)

Configuring an HMI connection via PROFIBUS with a SIMATIC PC (Page 432)

PROFIBUS parameters (Page 437)

2.8.3.3 PROFIBUS parameters

PROFIBUS parameters for the HMI connection

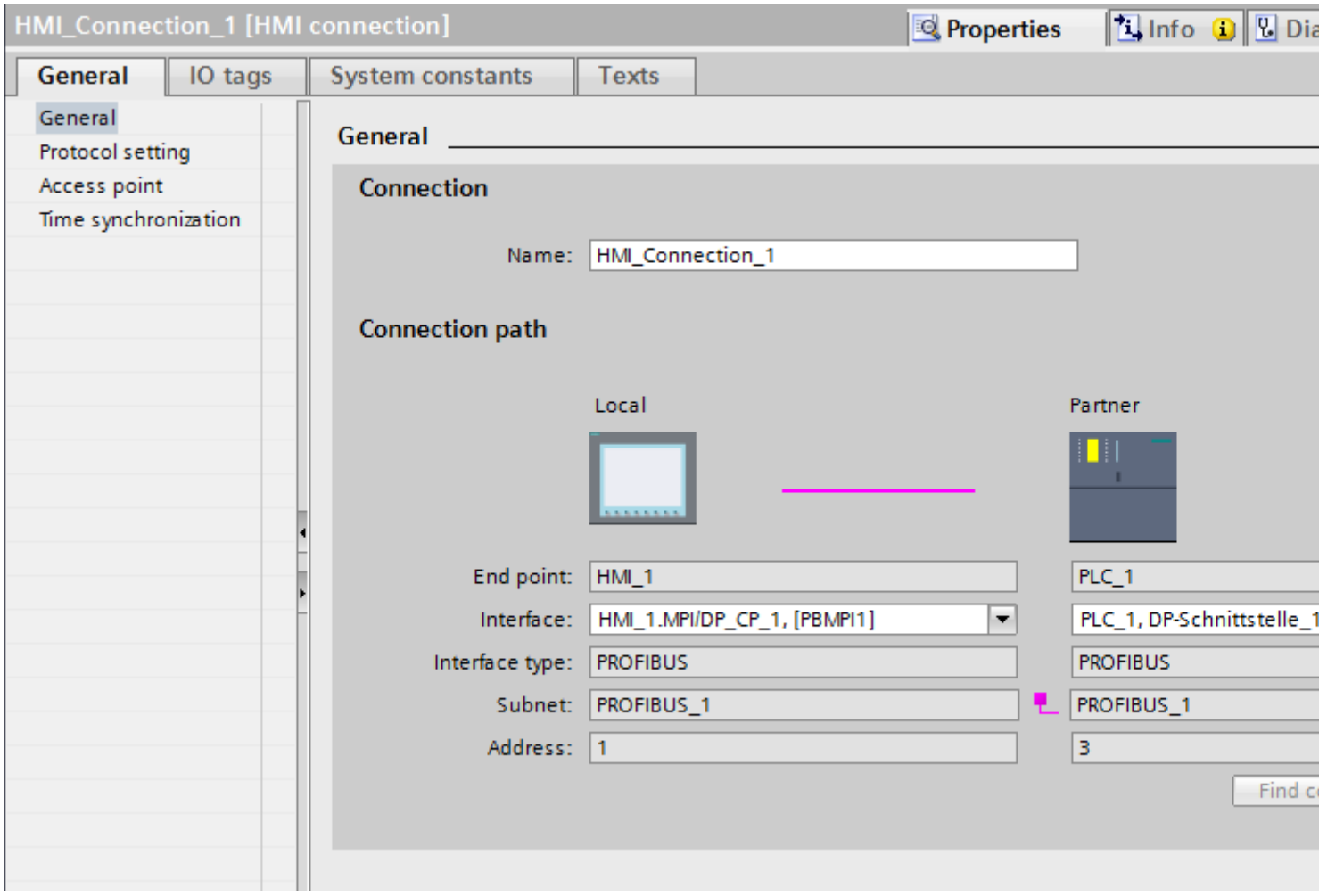
PROFIBUS parameters for the HMI connection

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 with SIMATIC S7 300/400 (Page 401)

PROFIBUS parameters for the HMI device (Page 439)

PROFIBUS parameters for the PLC (Page 441)

Bus profiles with PROFIBUS (Page 442)

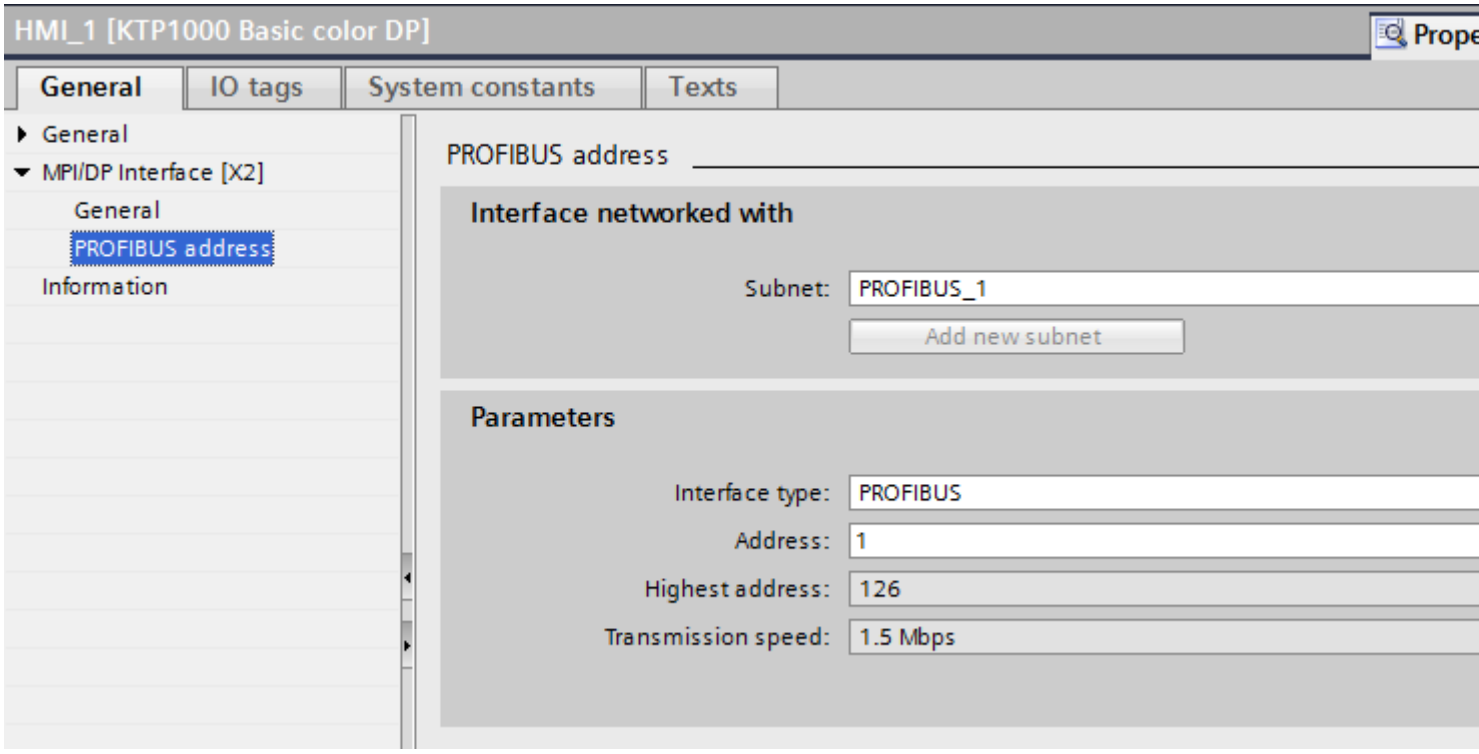
PROFIBUS parameters for the HMI device

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"
  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 300/400 (Page 401)

PROFIBUS parameters for the HMI connection (Page 437)

PROFIBUS parameters for the PLC (Page 441)

Bus profiles with PROFIBUS (Page 442)

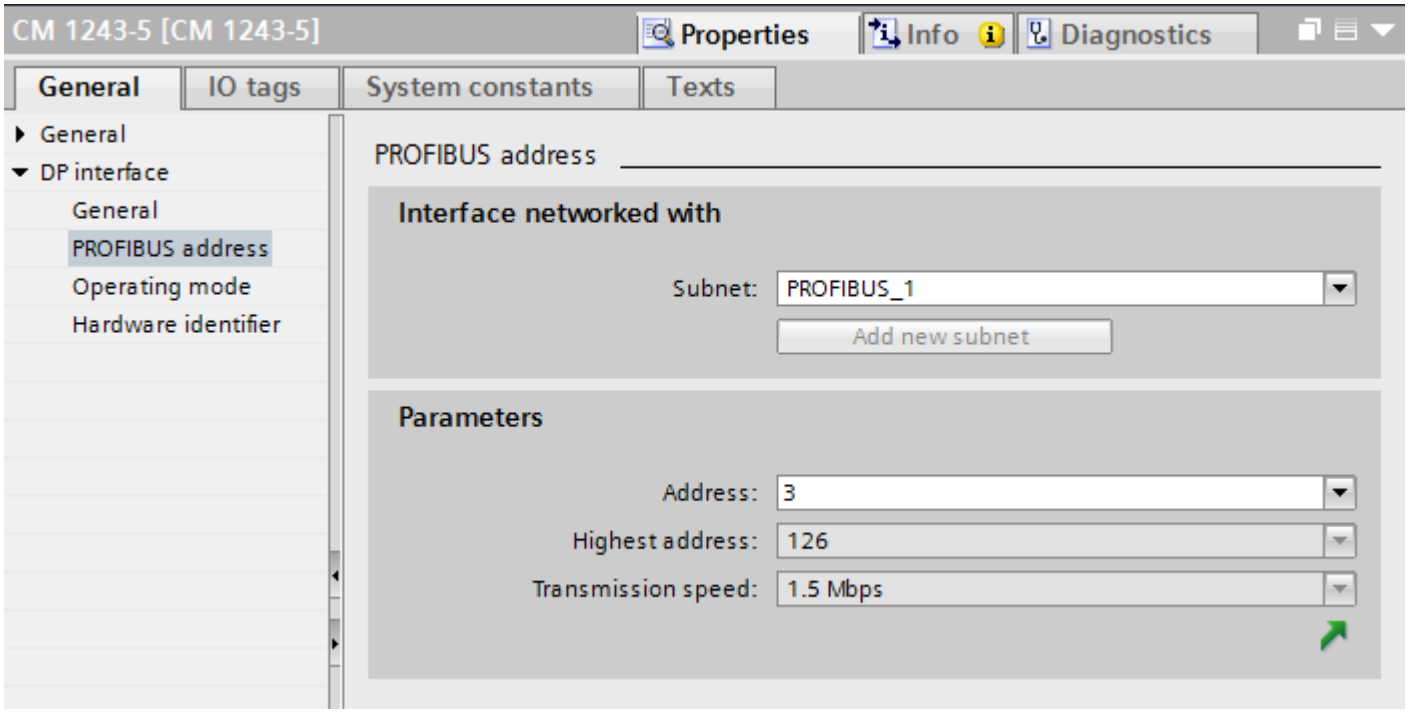
PROFIBUS parameters for the PLC

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 300/400 (Page 401)

PROFIBUS parameters for the HMI connection (Page 437)

PROFIBUS parameters for the HMI device (Page 439)

Bus profiles with PROFIBUS (Page 442)

Bus profiles with PROFIBUS

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 transfer speeds in Kbps
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 with SIMATIC S7 300/400 (Page 401)

PROFIBUS parameters for the HMI connection (Page 437)

PROFIBUS parameters for the HMI device (Page 439)

PROFIBUS parameters for the PLC (Page 441)

2.8.4 Communication via MPI

2.8.4.1 Configuring an HMI connection

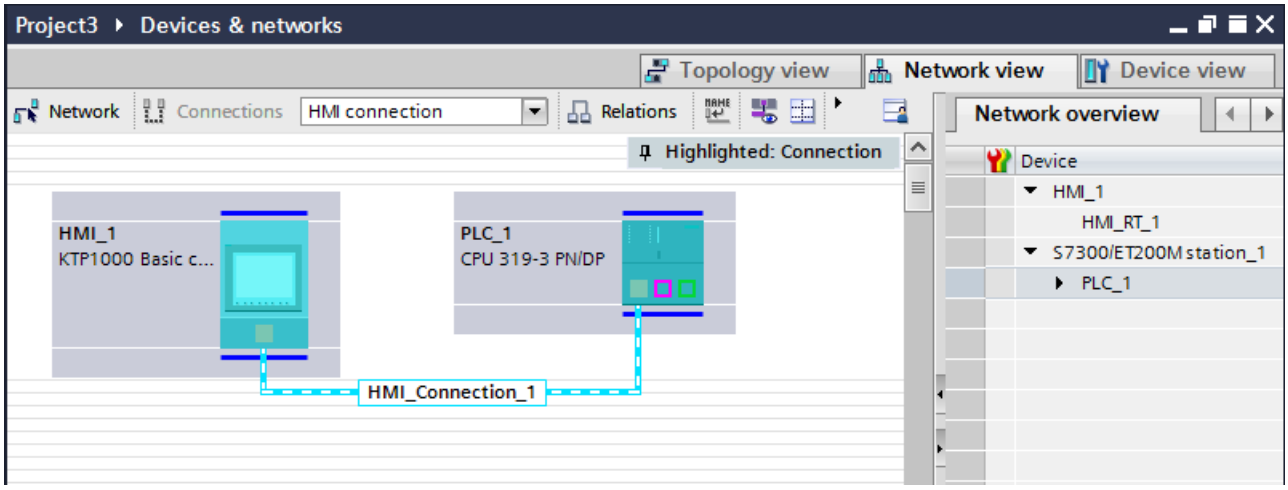
Communication via MPI

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 401)

Configuring an HMI connection via MPI (Page 444)

Configuring an HMI connection via MPI

Introduction

You configure an HMI connection between HMI devices and a SIMATIC S7 300/400 via MPI in the "Devices & Networks" editor.

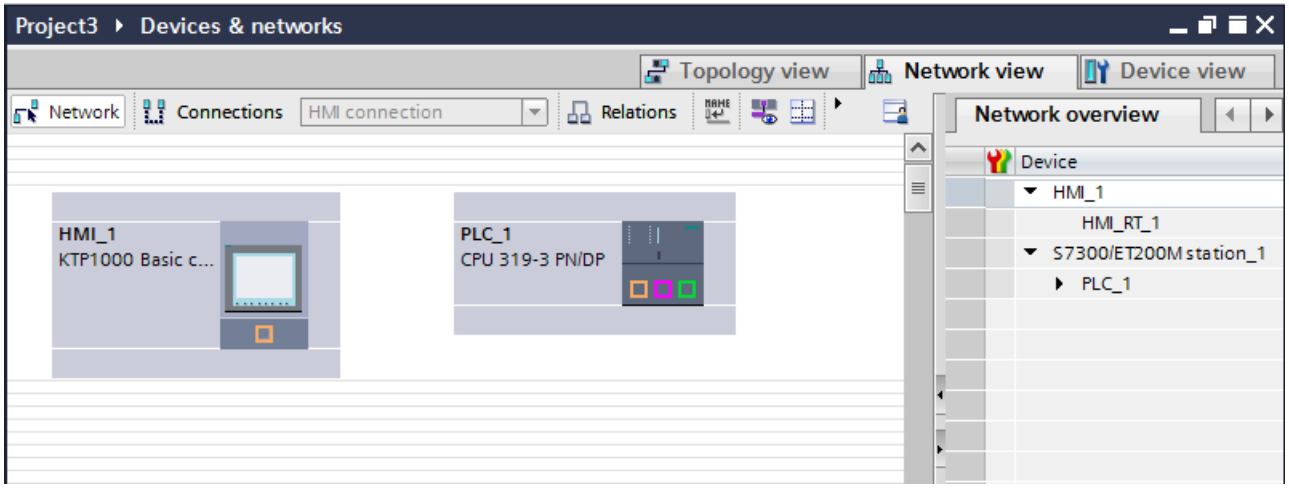
Requirements

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

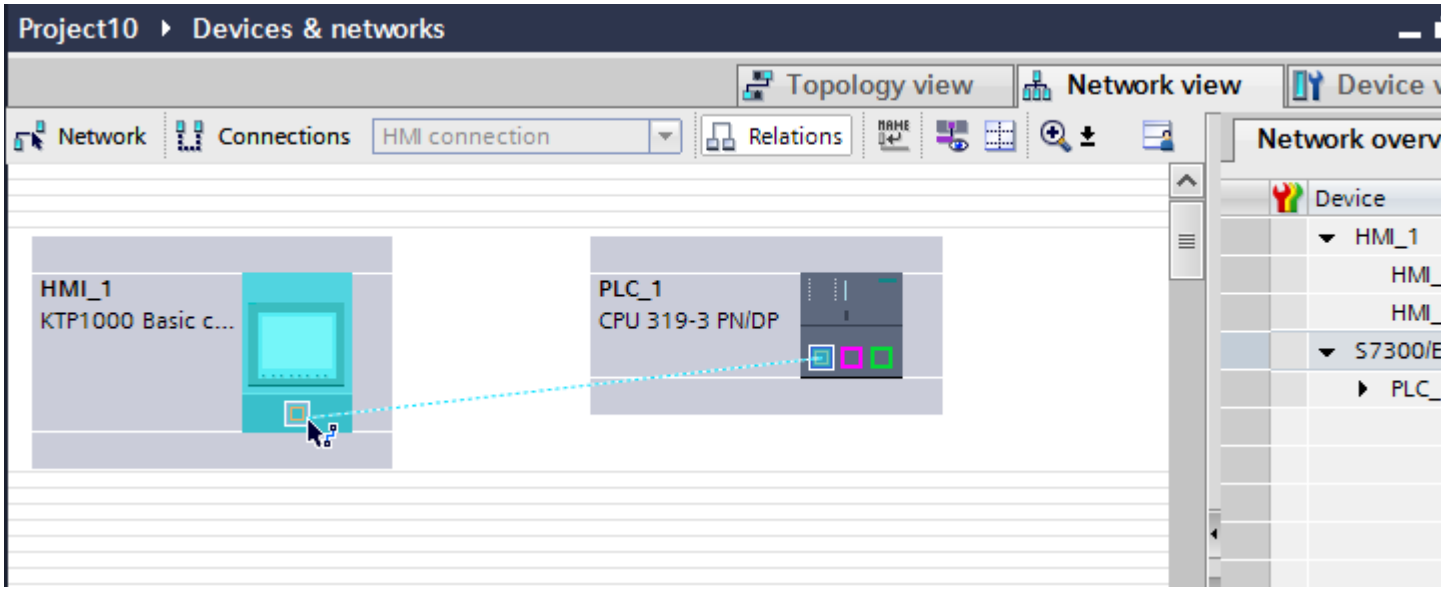
- HMI device with MPI/DP interface
- SIMATIC S7 300/400 with MPI/DP interface

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 453)" 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 401)

Communication via MPI (Page 443)

MPI parameters (Page 453)

2.8.4.2 Configuring an HMI connection

Communication via MPI

Communication via MPI

This section describes the communication between a WinCC Runtime and the SIMATIC S7 300/400 PLC via MPI.

You can use the following WinCC Runtimes as an HMI device:

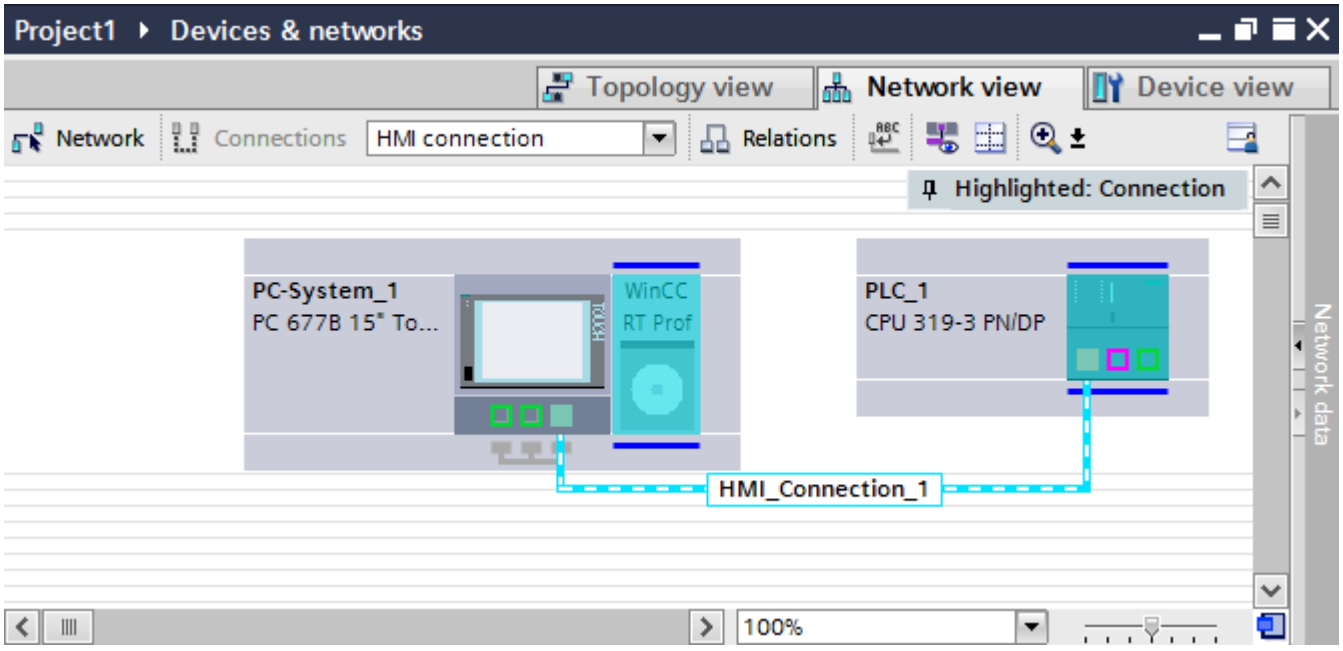
- WinCC RT Advanced
- WinCC RT Professional

WinCC Runtime as HMI device

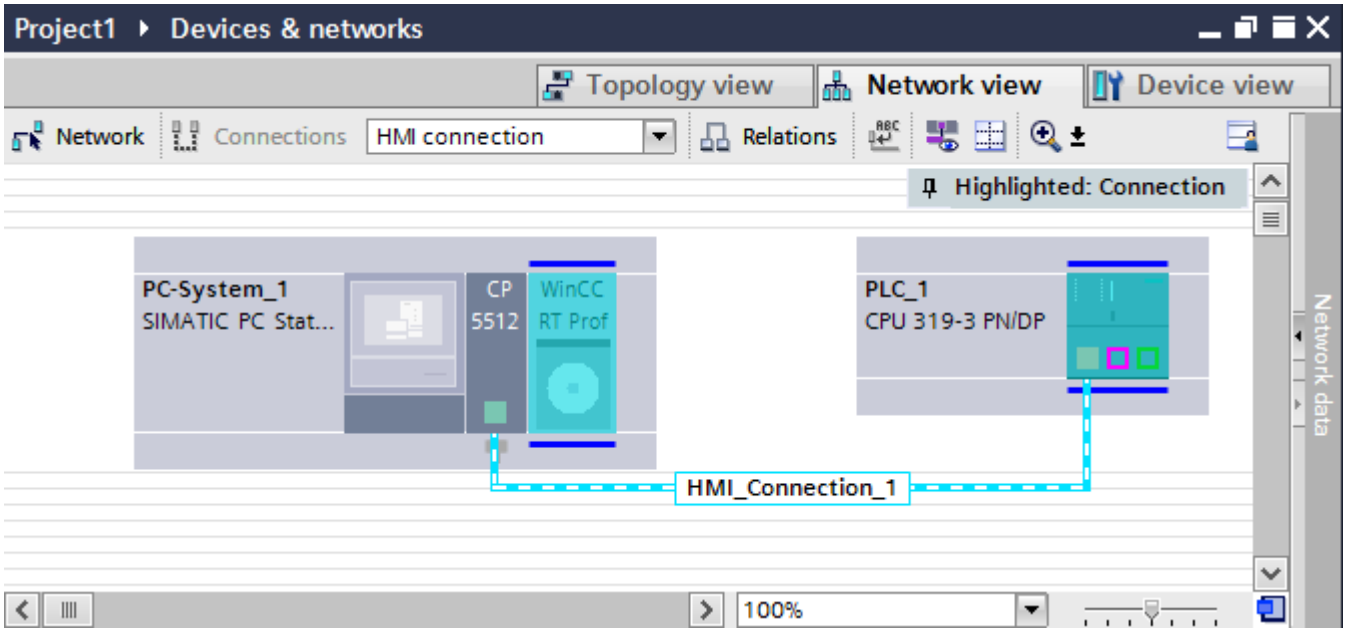
You configure the HMI connections between a WinCC Runtime and SIMATIC S7 300/400 in the "Devices & Networks" editor.

If you want to use a WinCC Runtime as an HMI device, you have the following options:

1. You create a SIMATIC PC and configure a WinCC Runtime on this SIMATIC PC.
   In this way you use the SIMATIC PC with a WinCC Runtime as an HMI device.



2. You create a WinCC Runtime and configure a communication processor to the Runtime.
   In this way you use your configuration PC with a WinCC Runtime as an HMI device.



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 MPI in the "Devices & Networks" editor.

Connection in the "Connections" editor

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

See also

Communication with SIMATIC S7 300/400 (Page 401)

Configuring an HMI connection via MPI with a SIMATIC PC (Page 448)

Configuring an HMI connection via MPI with a PC (Page 451)

Configuring an HMI connection via MPI with a SIMATIC PC

Introduction

You configure an HMI connection between HMI devices and a SIMATIC S7 300/400 via MPI in the "Devices & Networks" editor.

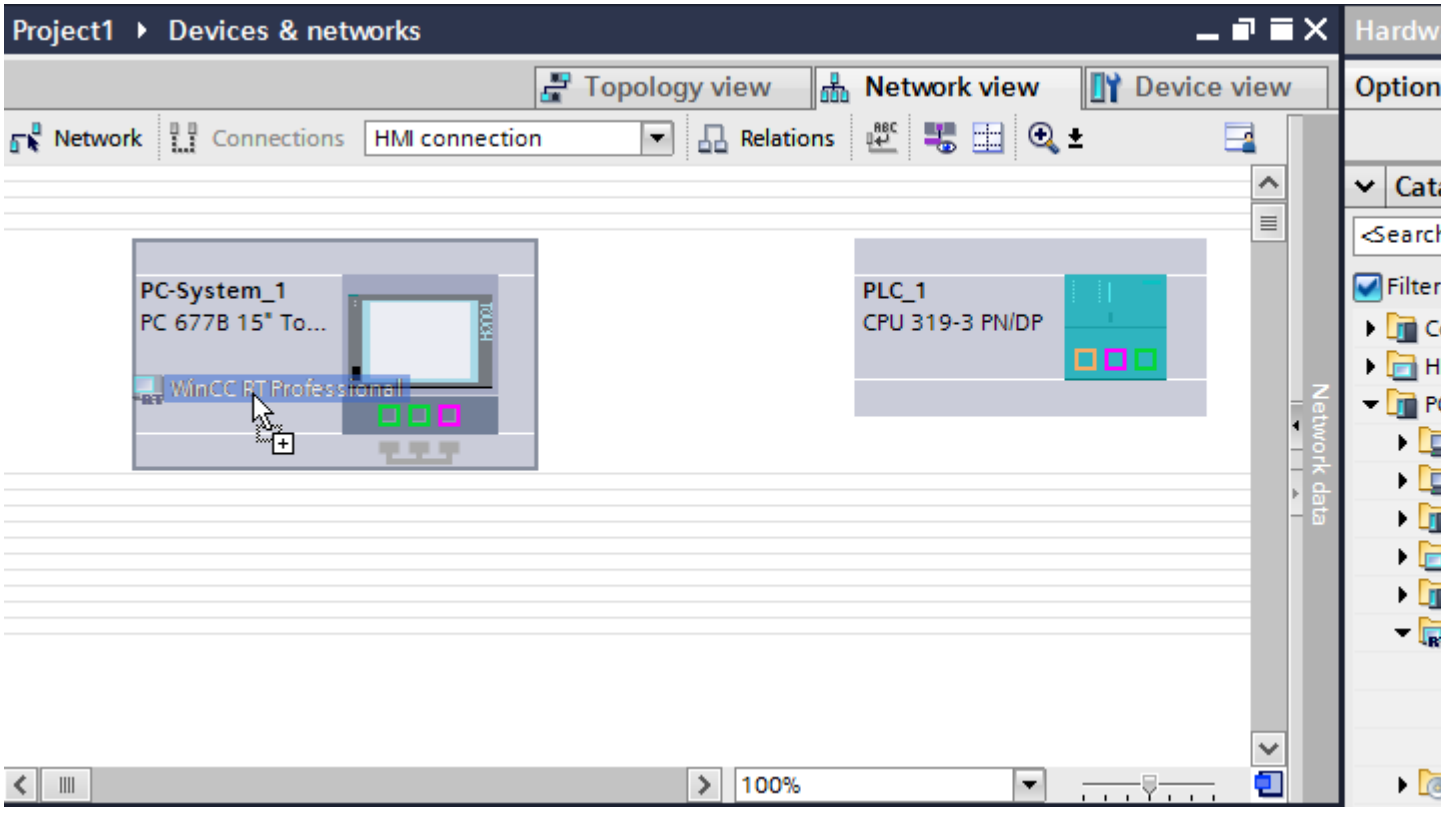
Requirements

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

- SIMATIC S7 300/400 with MPI interface
- SIMATIC PC with MPI interface

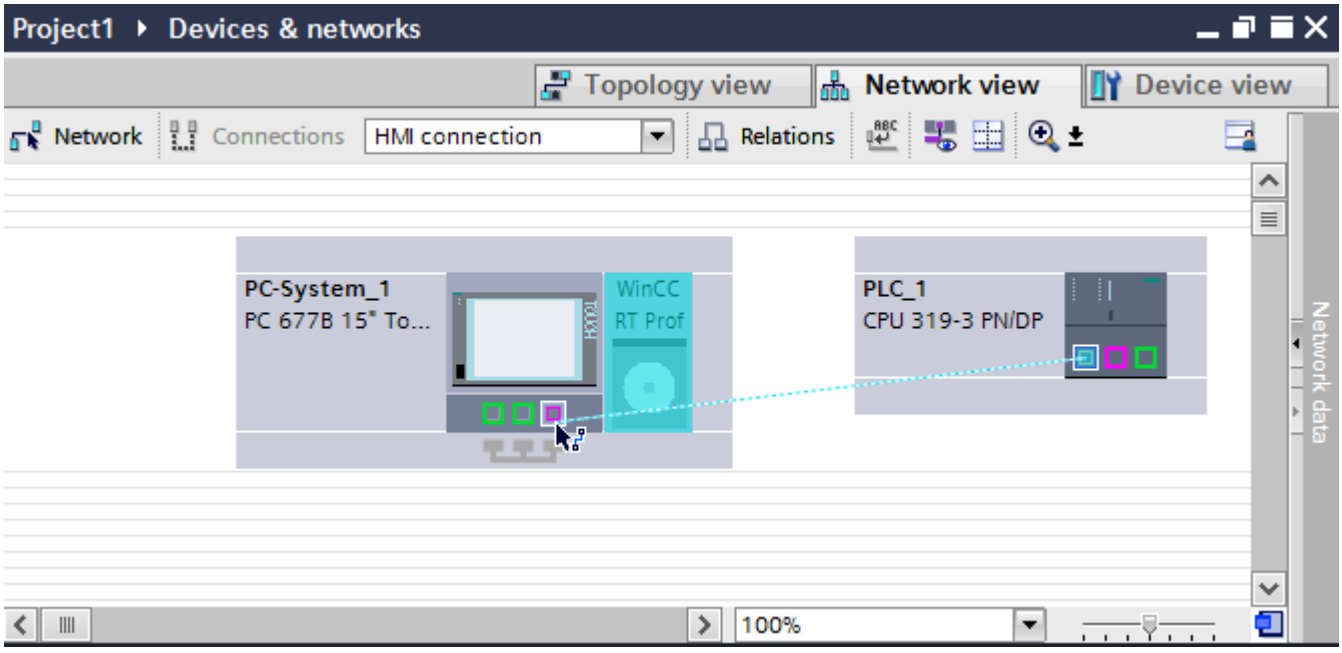
Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed in the network view.
2. Use a drag-and-drop operation to move a WinCC Runtime from the hardware catalog to the PC.



3. Click the "Connections" button and select "HMI connection" for the connection type.
   The devices available for connection are highlighted in color.

4. Click the MPI interface of the PLC and use a drag-and-drop operation to draw a connection to the MPI interface of the PC.



5. Click the connecting line.
6. Click "Highlight HMI connection" and select the HMI connection.
   The connection is displayed graphically in the Inspector window.
7. 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 453)" 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 MPI.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Communication via MPI (Page 446)

Configuring an HMI connection via MPI with a PC (Page 451)

MPI parameters (Page 453)

Configuring an HMI connection via MPI with a PC

Introduction

You configure an HMI connection between HMI devices and a SIMATIC S7 300/400 via MPI in the "Devices & Networks" editor.

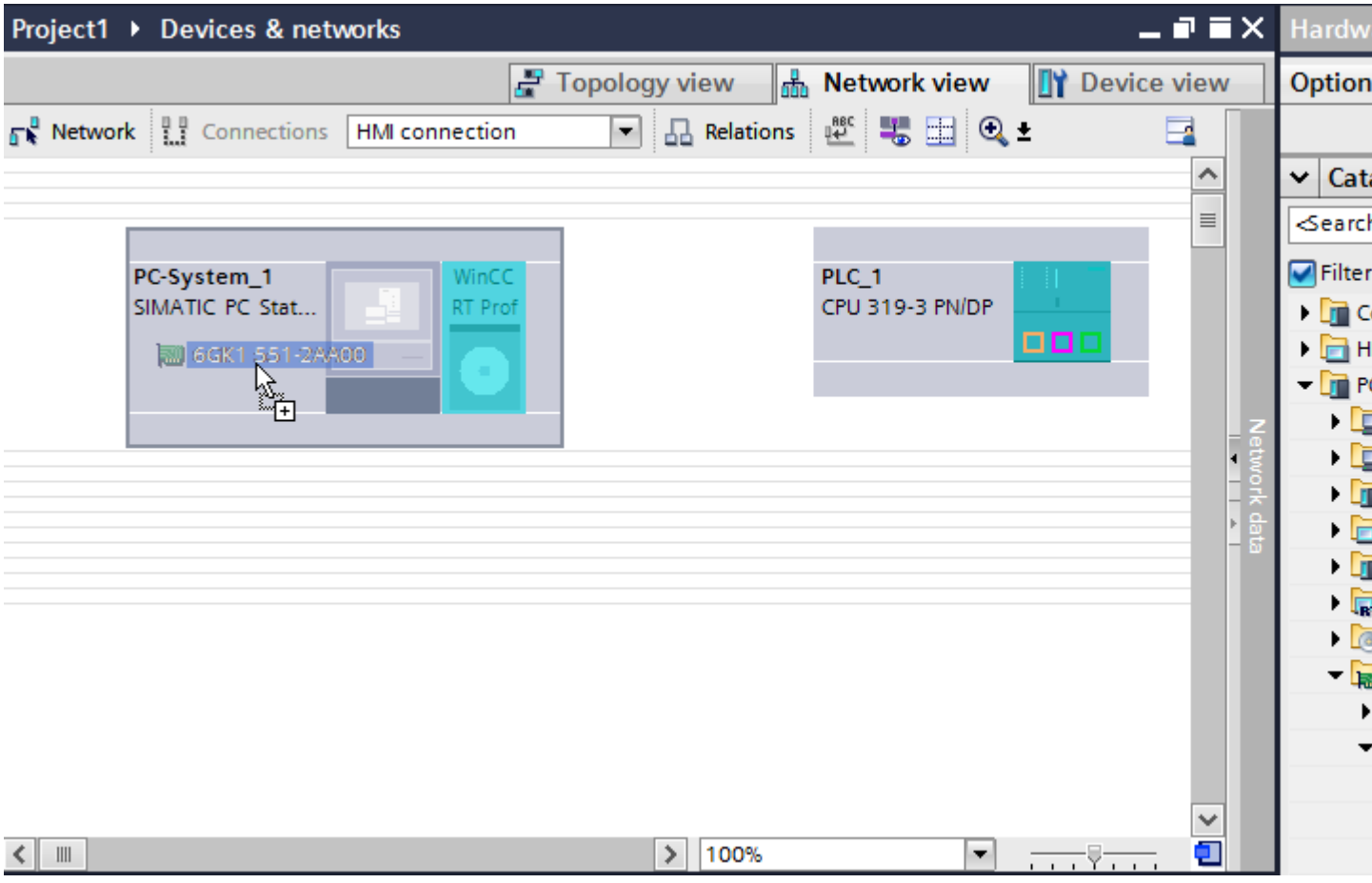
Requirements

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

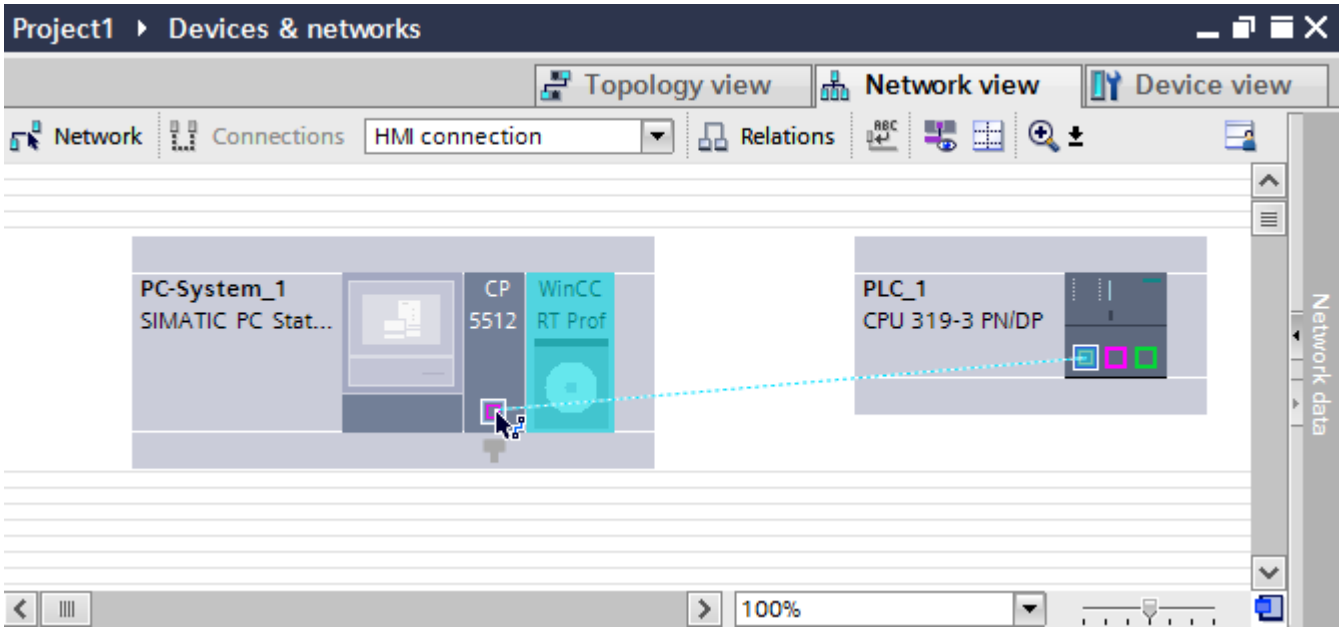
- SIMATIC S7 300/400 with MPI interface
- PC station with WinCC RT Advanced or WinCC RT Professional

Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed in the network view.
2. Use a drag-and-drop operation to move a PROFIBUS-capable communication processor from the hardware catalog to the WinCC Runtime.



3. Click the PROFIBUS interface of the communication processor and change the interface to "MPI".
4. Click the "Connections" button and select "HMI connection" for the connection type.
   The devices available for connection are highlighted in color.
5. Click the MPI interface of the PLC and use a drag-and-drop operation to draw a connection to the MPI interface of the communication processor.



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 MPI parameters in the Inspector window according to the requirements of your project.
   For additional information on this, see section "MPI parameters (Page 453)".

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 MPI.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Communication via MPI (Page 446)

Configuring an HMI connection via MPI with a SIMATIC PC (Page 448)

MPI parameters (Page 453)

2.8.4.3 MPI parameters

MPI parameters for the HMI connection

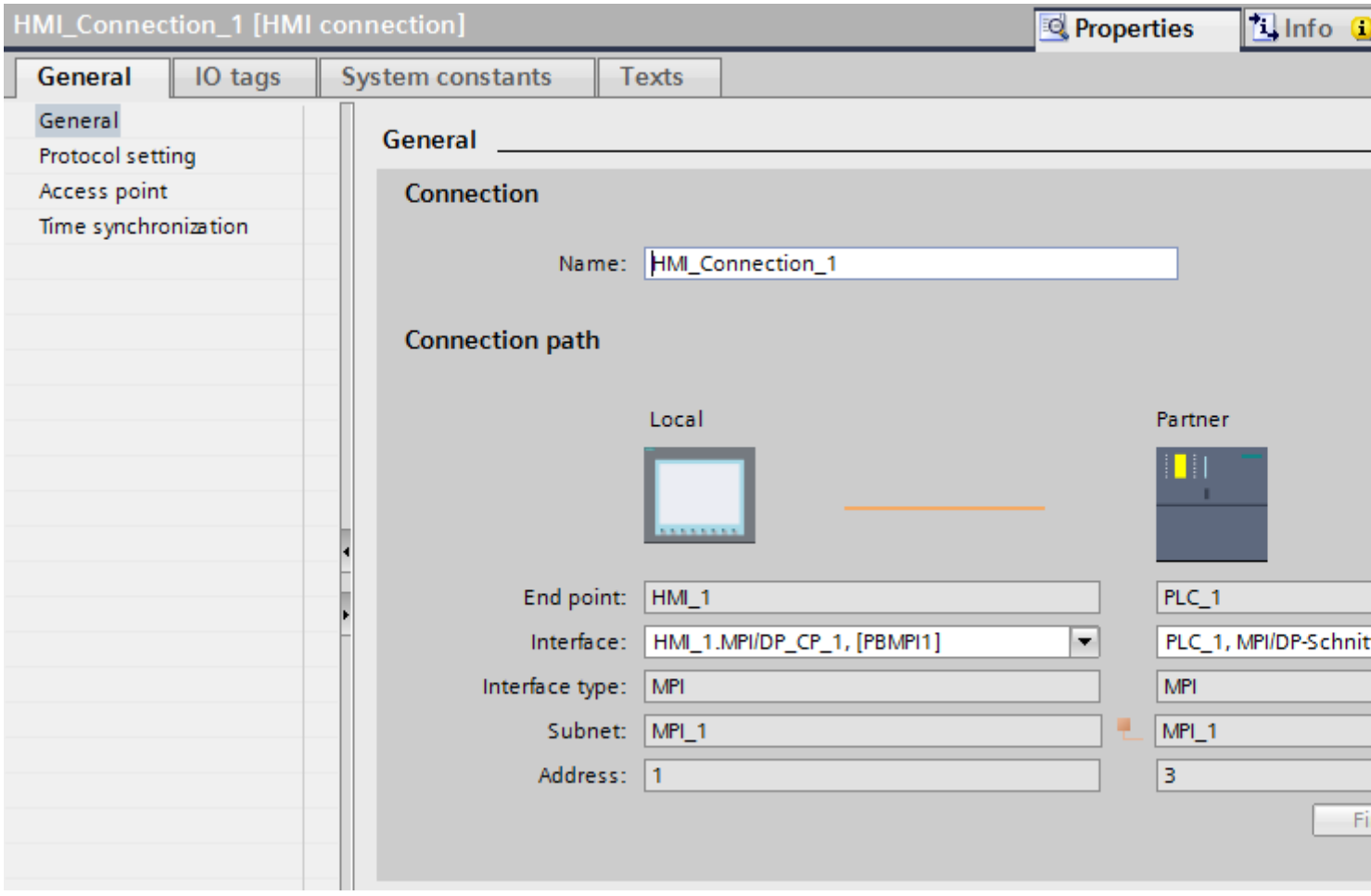
MPI parameters for the HMI connection

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 with SIMATIC S7 300/400 (Page 401)

MPI parameters for the HMI device (Page 455)

MPI parameters for the PLC (Page 457)

Addressing of the PLC via MPI (Page 458)

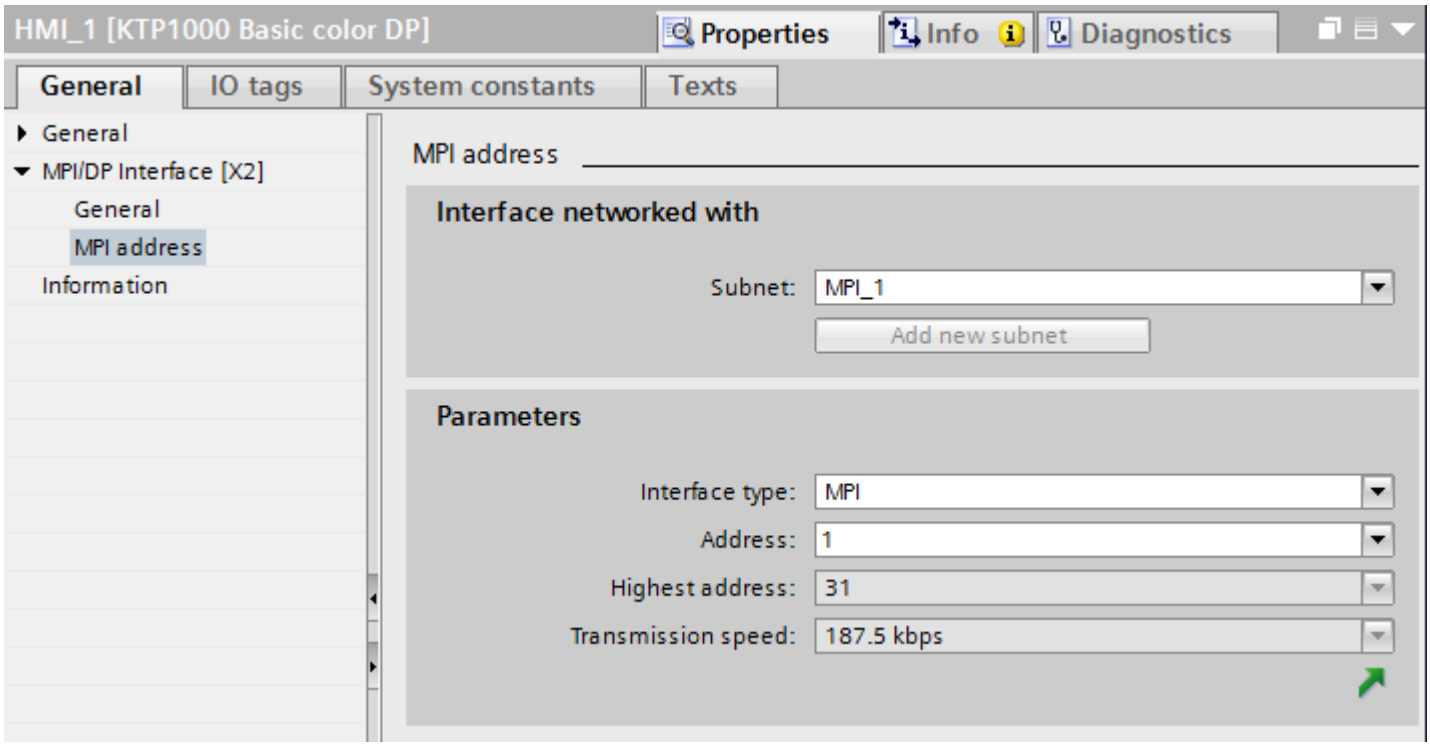
MPI parameters for the HMI device

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 with SIMATIC S7 300/400 (Page 401)

MPI parameters for the HMI connection (Page 453)

MPI parameters for the PLC (Page 457)

Addressing of the PLC via MPI (Page 458)

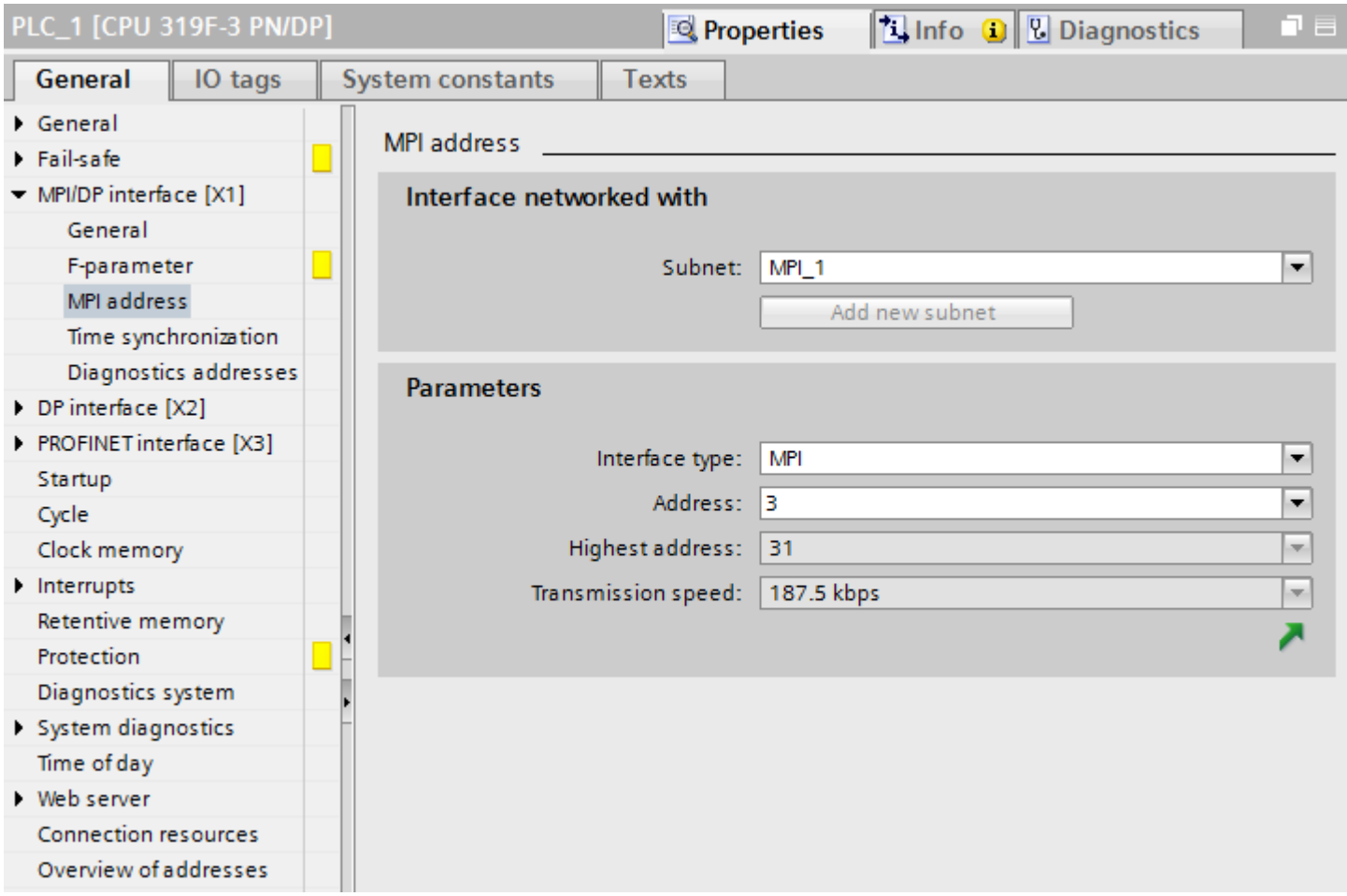
MPI parameters for the PLC

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 with SIMATIC S7 300/400 (Page 401)

MPI parameters for the HMI connection (Page 453)

MPI parameters for the HMI device (Page 455)

Addressing of the PLC via MPI (Page 458)

Addressing of the PLC via MPI

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 with SIMATIC S7 300/400 (Page 401)

MPI parameters for the HMI connection (Page 453)

MPI parameters for the HMI device (Page 455)

MPI parameters for the PLC (Page 457)

2.8.5 Data exchange

2.8.5.1 Data exchange using area pointers

General information on area pointers

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 153)

See also

Communication with SIMATIC S7 300/400 (Page 401)

Configuring area pointers (Page 153)

"Screen number" area pointer (Page 460)

"Date/time" area pointer (Page 461)

"Date/time PLC" area pointer (Page 462)

"Coordination" area pointer (Page 464)

"Project ID" area pointer (Page 465)

"Job mailbox" area pointer (Page 466)

"Screen number" area pointer

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

Communication with SIMATIC S7 300/400 (Page 401)

General information on area pointers (Page 459)

"Date/time" area pointer (Page 461)

"Date/time PLC" area pointer (Page 462)

"Coordination" area pointer (Page 464)

"Project ID" area pointer (Page 465)

"Job mailbox" area pointer (Page 466)

"Date/time" area pointer

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

Communication with SIMATIC S7 300/400 (Page 401)

General information on area pointers (Page 459)

"Screen number" area pointer (Page 460)

"Date/time PLC" area pointer (Page 462)

"Coordination" area pointer (Page 464)

"Project ID" area pointer (Page 465)

"Job mailbox" area pointer (Page 466)

"Date/time PLC" area pointer

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=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

Communication with SIMATIC S7 300/400 (Page 401)
General information on area pointers (Page 459)
"Screen number" area pointer (Page 460)
"Date/time" area pointer (Page 461)
"Coordination" area pointer (Page 464)
"Project ID" area pointer (Page 465)
"Job mailbox" area pointer (Page 466)

"Coordination" area pointer

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

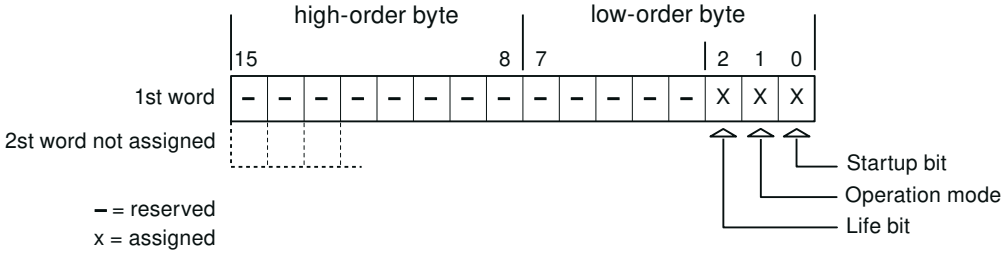
By default, the "Coordination" area pointer has the length of one word and cannot be changed.

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

Communication with SIMATIC S7 300/400 (Page 401)

General information on area pointers (Page 459)

"Screen number" area pointer (Page 460)

"Date/time" area pointer (Page 461)

"Date/time PLC" area pointer (Page 462)

"Project ID" area pointer (Page 465)

"Job mailbox" area pointer (Page 466)

"Project ID" area pointer

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

Communication with SIMATIC S7 300/400 (Page 401)

General information on area pointers (Page 459)

"Screen number" area pointer (Page 460)

"Date/time" area pointer (Page 461)

"Date/time PLC" area pointer (Page 462)

"Coordination" area pointer (Page 464)

"Job mailbox" area pointer (Page 466)

"Job mailbox" area pointer

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 to 23)
	Parameter 2	Left byte: Minutes (0-59) Right byte: Seconds (0 to 59)
	Parameter 3	-
15	**Setting the date (BCD coded)** [2) 3)]	
	Parameter 1	Left byte: - Right byte: Weekday (1-7: Sunday-Saturday)
	Parameter 2	Left byte: Day (1-31) Right byte: Month (1 to 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**	
	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**	

No.	Function	
14	**Set time (BCD-coded)**	
	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**	
	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]	The weekday is ignored on HMI device KTP 600 BASIC PN.
[3]	The weekday is ignored when you configure the "Date/Time PLC" area pointer.

See also

Communication with SIMATIC S7 300/400 (Page 401)

General information on area pointers (Page 459)

"Screen number" area pointer (Page 460)

"Date/time" area pointer (Page 461)

"Date/time PLC" area pointer (Page 462)

"Coordination" area pointer (Page 464)

"Project ID" area pointer (Page 465)

"Data record" area pointer

"Data record" area pointer

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

Communication with SIMATIC S7 300/400 (Page 401)

Transfer without synchronization (Page 470)

Transfer with synchronization (Page 470)

Sequence of a transfer started by the operator in the recipe view (Page 472)

Sequence of the transfer triggered by a job mailbox (Page 473)

Sequence of the transfer when triggered by a configured function (Page 475)

Possible causes of error when transferring data records (Page 476)

Transfer without synchronization

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

Communication with SIMATIC S7 300/400 (Page 401)

"Data record" area pointer (Page 469)

Transfer with synchronization (Page 470)

Sequence of a transfer started by the operator in the recipe view (Page 472)

Sequence of the transfer triggered by a job mailbox (Page 473)

Sequence of the transfer when triggered by a configured function (Page 475)

Possible causes of error when transferring data records (Page 476)

Transfer with synchronization

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

Communication with SIMATIC S7 300/400 (Page 401)

"Data record" area pointer (Page 469)

Transfer without synchronization (Page 470)

Sequence of a transfer started by the operator in the recipe view (Page 472)
Sequence of the transfer triggered by a job mailbox (Page 473)
Sequence of the transfer when triggered by a configured function (Page 475)
Possible causes of error when transferring data records (Page 476)

Sequence of a transfer started by the operator in the recipe view

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

Communication with SIMATIC S7 300/400 (Page 401)

"Data record" area pointer (Page 469)

Transfer without synchronization (Page 470)

Transfer with synchronization (Page 470)

Sequence of the transfer triggered by a job mailbox (Page 473)

Sequence of the transfer when triggered by a configured function (Page 475)

Possible causes of error when transferring data records (Page 476)

Sequence of the transfer triggered by a job mailbox

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)	

	Left byte (LB)	Right byte (RB)
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

Communication with SIMATIC S7 300/400 (Page 401)

"Data record" area pointer (Page 469)

Transfer without synchronization (Page 470)

Transfer with synchronization (Page 470)

Sequence of a transfer started by the operator in the recipe view (Page 472)

Sequence of the transfer when triggered by a configured function (Page 475)
Possible causes of error when transferring data records (Page 476)

Sequence of the transfer when triggered by a configured function

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

Communication with SIMATIC S7 300/400 (Page 401)
"Data record" area pointer (Page 469)

Transfer without synchronization (Page 470)
Transfer with synchronization (Page 470)
Sequence of a transfer started by the operator in the recipe view (Page 472)
Sequence of the transfer triggered by a job mailbox (Page 473)
Possible causes of error when transferring data records (Page 476)

Possible causes of error when transferring data records

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

Communication with SIMATIC S7 300/400 (Page 401)

"Data record" area pointer (Page 469)

Transfer without synchronization (Page 470)

Transfer with synchronization (Page 470)

Sequence of a transfer started by the operator in the recipe view (Page 472)

Sequence of the transfer triggered by a job mailbox (Page 473)

Sequence of the transfer when triggered by a configured function (Page 475)

2.8.5.2 Trends

General information on trends

Trends

A trend is the graphical representation of one or more values from the PLC. The value is read out either time- or bit-triggered, depending on the configuration.

For additional information see:

AUTOHOTSPOT

Note

The value is read out time-triggered for Basic Panels.

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.

Bit-triggered trends

Through a trigger bit set in the trend request tag, the HMI device either reads in a trend value or an entire trend buffer. This setting is defined in the configuration. Bit-triggered trends are normally used to represent fast 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 "HMI tags" editor and connected to trend areas during configuration. The HMI device and PLC then communicate with each other via 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)

See also

Trend request and trend transfer (Page 478)

Permitted data types for trends (Page 479)

Trend request and trend transfer

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 closing the screen, the HMI device resets the relevant bits in the trend request area.

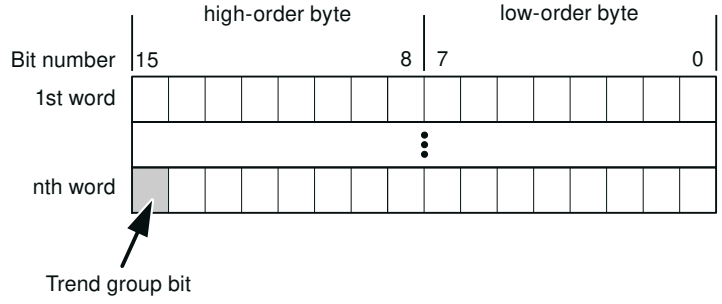
The trend request area can be used for evaluation purposes in the PLC to determine which trend is currently being displayed on the HMI device. Trends can also be triggered without evaluation of the trend request area.

Trend transfer area 1

This area is used to trigger trends. You must set the bit assigned to the trend in the trend transfer area and set the trend group bit in your control program. 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.

The PLC writes to Buffer 2 while the HMI device reads values from Buffer 1, and writes to Buffer 1 when the HMI device is reading Buffer 2. This prevents the PLC from overwriting trend values while the trend is being read by the HMI device.

See also

General information on trends (Page 477)

Permitted data types for trends (Page 479)

Permitted data types for trends

For SIMATIC S7

You assign one bit to each trend during configuration. Tags and array tags of the "Word" or "Int" data type are permitted.

See also

General information on trends (Page 477)

Trend request and trend transfer (Page 478)

2.8.5.3 Alarms

Configuring alarms

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

Communication with SIMATIC S7 300/400 (Page 401)

Acknowledgment of alarms (Page 480)

Acknowledgment of alarms

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

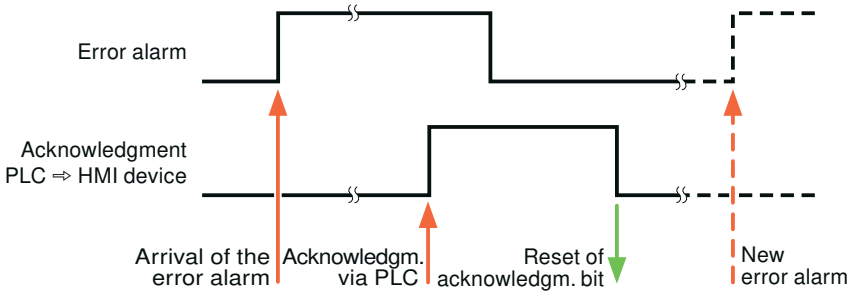
Acknowledgment by the PLC

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.

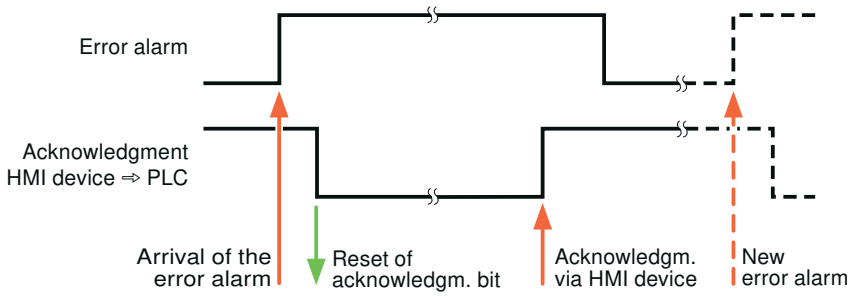
If a discrete alarm subject to acknowledgment is acknowledged by the HMI device, then the corresponding bit in the assigned acknowledgment tag will be set. The entire acknowledgment tag is then written to the PLC by the HMI device. This allows the PLC to recognize that a certain alarm message has been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

Communication with SIMATIC S7 300/400 (Page 401)

Configuring alarms (Page 479)

2.8.5.4 LED mapping

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 Comfort Panels	Panel PCs
0	0	Off	Off
0	1	Rapid flashing	Flashing

Bit n+ 1	Bit n	LED function	
		all Mobile Panels, all Comfort Panels	Panel PCs
1	0	Slow flashing	Flashing
1	1	On permanently	On permanently

See also

Communication with SIMATIC S7 300/400 (Page 401)

2.8.6 Performance features of communication

2.8.6.1 S7 300/400 device dependency

Communication with the SIMATIC S7-300/400 PLC

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 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

Basic Panels V14.0.1

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 V15.0

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

Mobile Panels V12.0

HMI devices	SIMATIC S7-300/400
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 S7-300/400
KTP700 Mobile	Yes
KTP700F Mobile	Yes
KTP900 Mobile	Yes
KTP900F Mobile	Yes

Mobile Panels V14.0

HMI devices	SIMATIC S7-300/400
KTP700 Mobile	Yes
KTP700F Mobile	Yes
KTP900 Mobile	Yes
KTP900F Mobile	Yes
KTP400F Mobile	Yes

Mobile Panels V14.0.1

HMI devices	SIMATIC S7-300/400
KTP700 Mobile	Yes
KTP700F Mobile	Yes
KTP900 Mobile	Yes
KTP900F Mobile	Yes
KTP400F Mobile	Yes

Mobile Panels V15.0

HMI devices	SIMATIC S7-300/400
KTP700 Mobile	Yes
KTP700F Mobile	Yes

HMI devices	SIMATIC S7-300/400
KTP900 Mobile	Yes
KTP900F Mobile	Yes
KTP400F Mobile	Yes

Comfort Panels V12.0

HMI devices	SIMATIC S7-300/400
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

HMI devices	SIMATIC S7-300/400
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 S7-300/400
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 S7-300/400
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 V14.0

HMI devices	SIMATIC S7-300/400
KP400 Comfort	Yes
KTP400 Comfort	Yes
KTP400 Comfort Portrait	Yes
KP700 Comfort	Yes
TP700 Comfort	Yes
TP700 Comfort Portrait	Yes

HMI devices	SIMATIC S7-300/400
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

Comfort Panels V14.0.1

HMI devices	SIMATIC S7-300/400
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

HMI devices	SIMATIC S7-300/400
TP1900 Comfort Portrait	Yes
TP2200 Comfort	Yes
TP2200 Comfort Portrait	Yes

Comfort Panels V15.0

HMI devices	SIMATIC S7-300/400
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 V12.0

HMI devices	SIMATIC S7-300/400
WinCC RT Advanced	Yes

Runtime V13.0

HMI devices	SIMATIC S7-300/400
WinCC RT Advanced	Yes

Runtime V13.0.1

HMI devices	SIMATIC S7-300/400
WinCC RT Advanced	Yes

Runtime V14.0

HMI devices	SIMATIC S7-300/400
WinCC RT Advanced	Yes
WinCC RT Professional	Yes

Runtime V14.0.1

HMI devices	SIMATIC S7-300/400
WinCC RT Advanced	Yes
WinCC RT Professional	Yes

Runtime V15.0

HMI devices	SIMATIC S7-300/400
WinCC RT Advanced	Yes
WinCC RT Professional	Yes

See also

Communication with SIMATIC S7 300/400 (Page 401)
Permitted data types for SIMATIC S7 300/400 (Page 490)
Permitted data types for SIMATIC S7 300/400 (Page 491)

2.8.6.2 Permitted data types for SIMATIC S7 300/400

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)]

Data type	Length
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 401)

S7 300/400 device dependency (Page 483)

Permitted data types for SIMATIC S7 300/400 (Page 491)

2.8.6.3 Permitted data types for SIMATIC S7 300/400

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
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

Data type	Length
TIMER	2 bytes
STRING	(2+n) bytes, n = 0 to 254
RAW*	Raw data type

* Raw data communication to an S7-300 programmable logic controller cannot be set up over a communications processor.

See also

Communication with SIMATIC S7 300/400 (Page 401)

S7 300/400 device dependency (Page 483)

Permitted data types for SIMATIC S7 300/400 (Page 490)

2.8.7 Creating a connection in the "Connections" editor

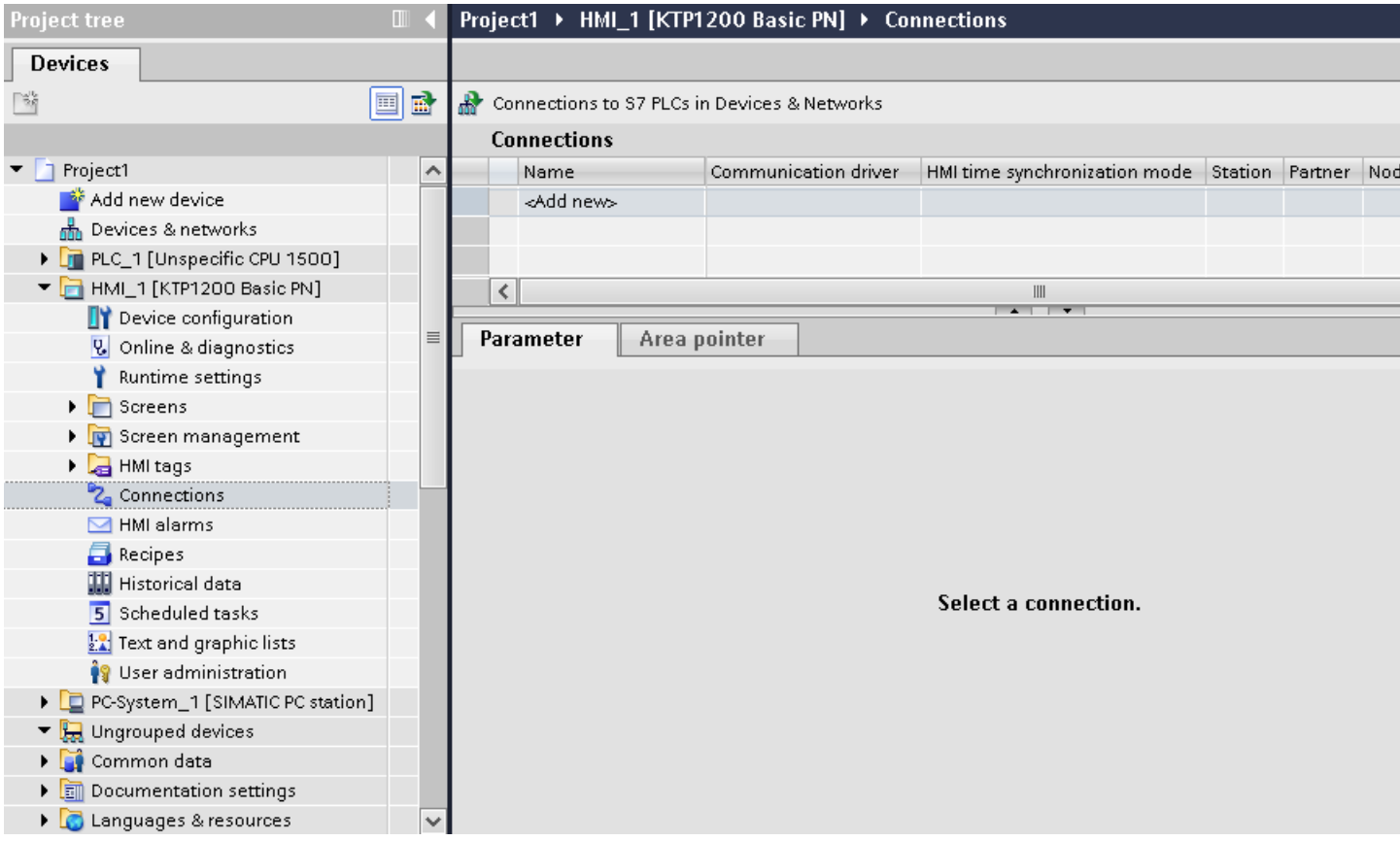
2.8.7.1 Creating a PROFINET connection

Requirements

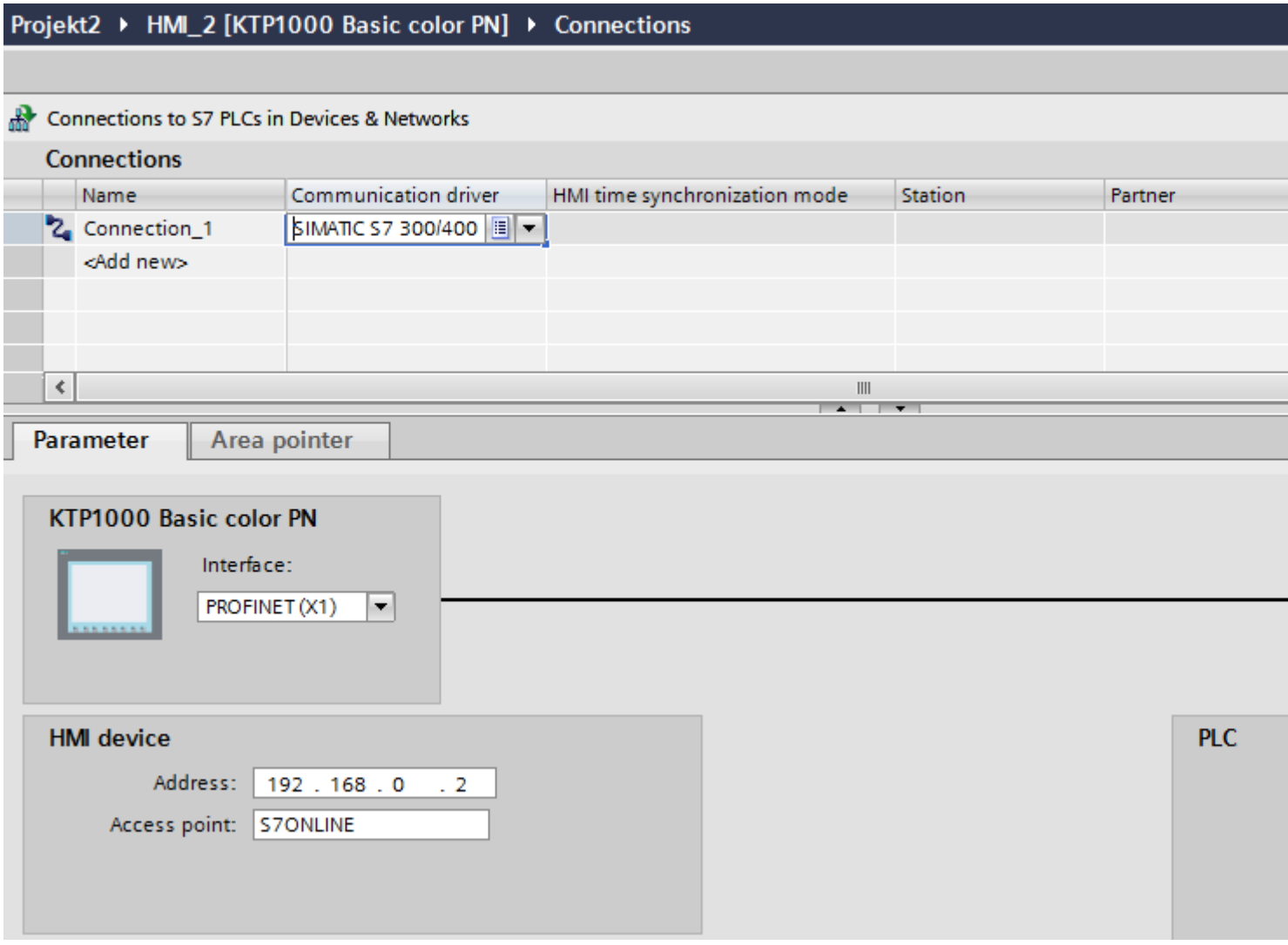
- A project is open.
- An HMI device with a PROFINET interface has been created.

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 with SIMATIC S7 300/400 (Page 401)

Creating a PROFIBUS connection (Page 495)

Creating an MPI connection (Page 497)

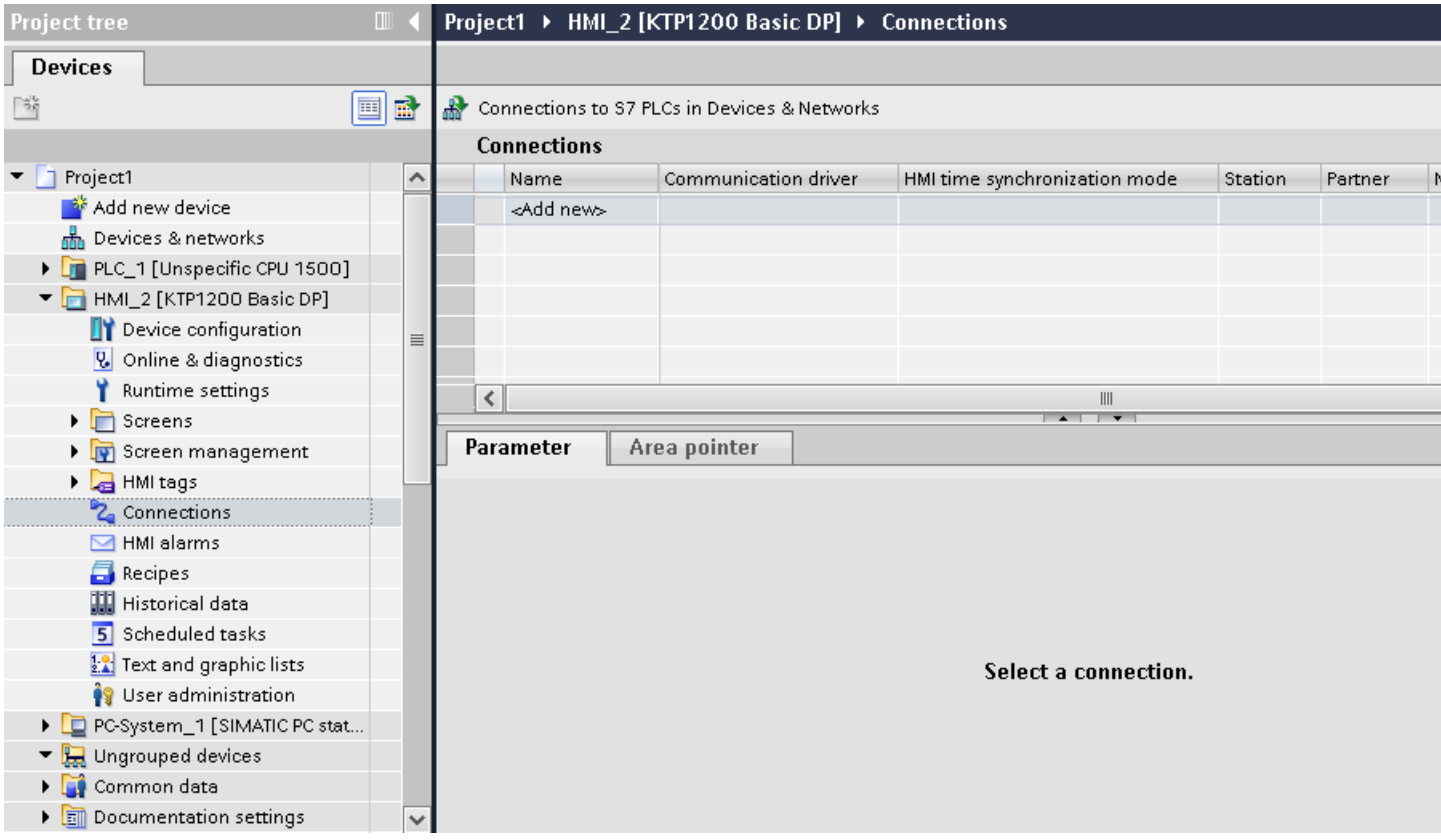
2.8.7.2 Creating a PROFIBUS connection

Requirements

- A project is open.
- An HMI device with a PROFIBUS interface has been created.

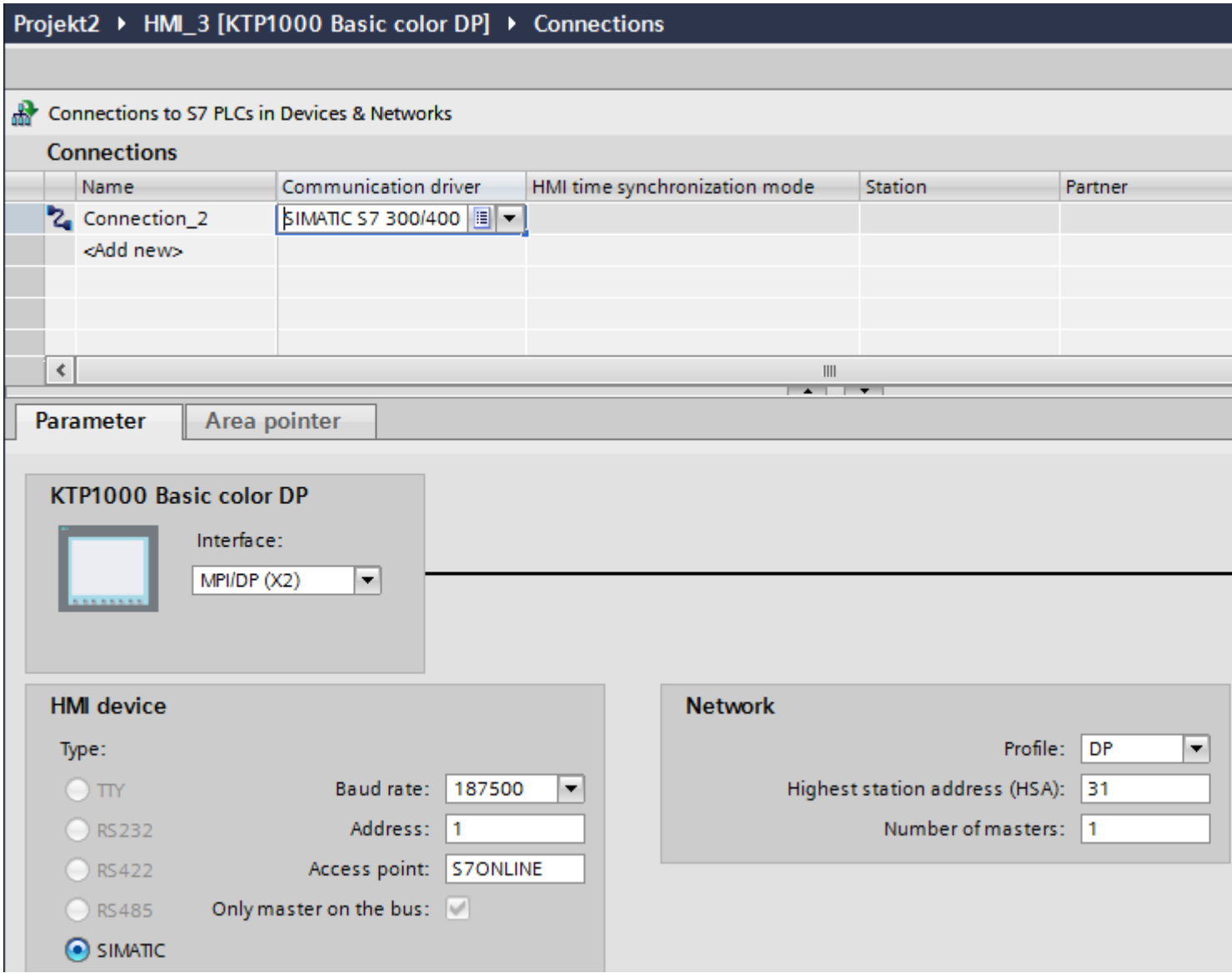
Procedure

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 with SIMATIC S7 300/400 (Page 401)

Creating a PROFINET connection (Page 492)

Creating an MPI connection (Page 497)

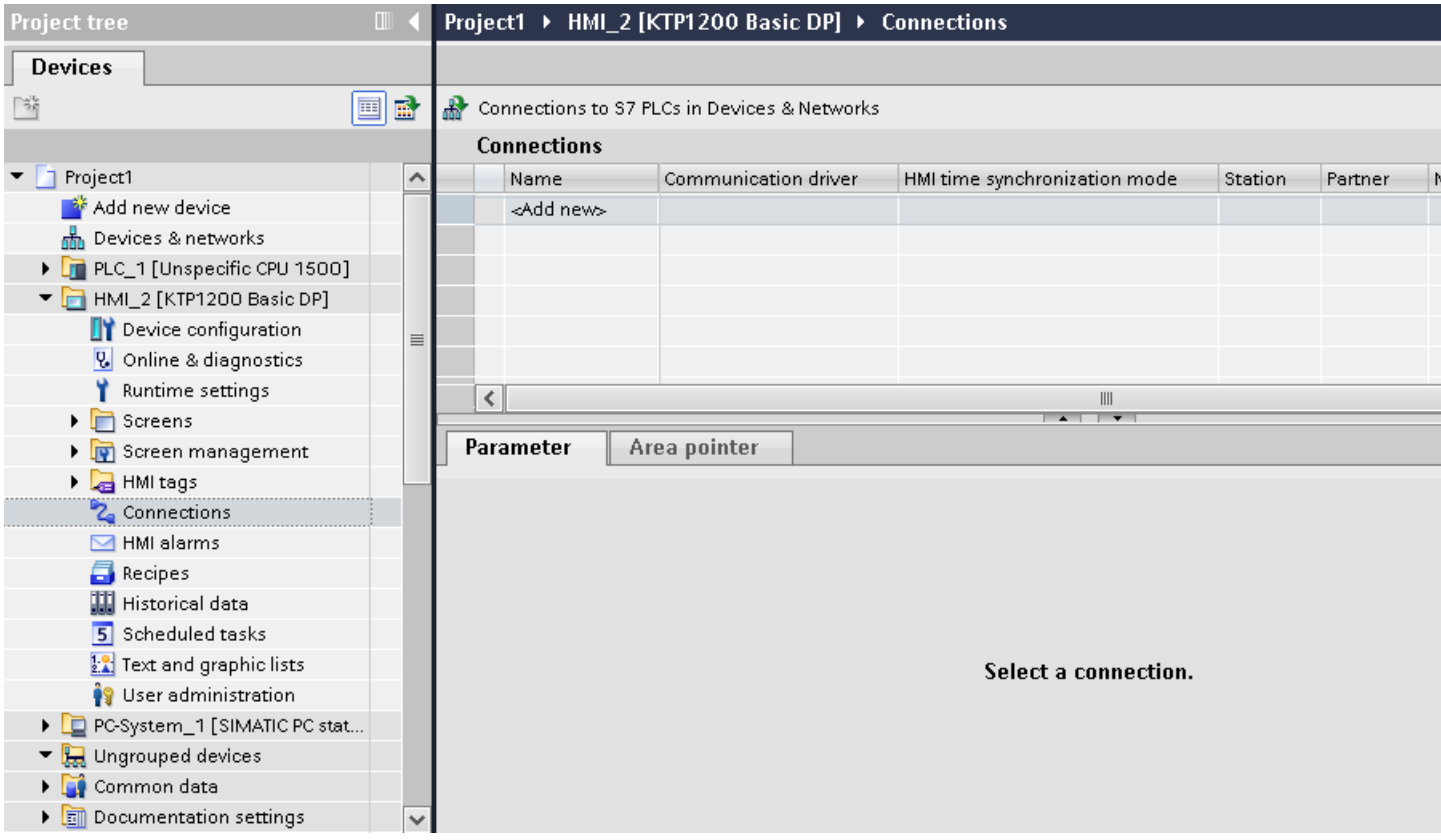
2.8.7.3 Creating an MPI connection

Requirements

- A project is open.
- An HMI device with an MPI interface has been created.

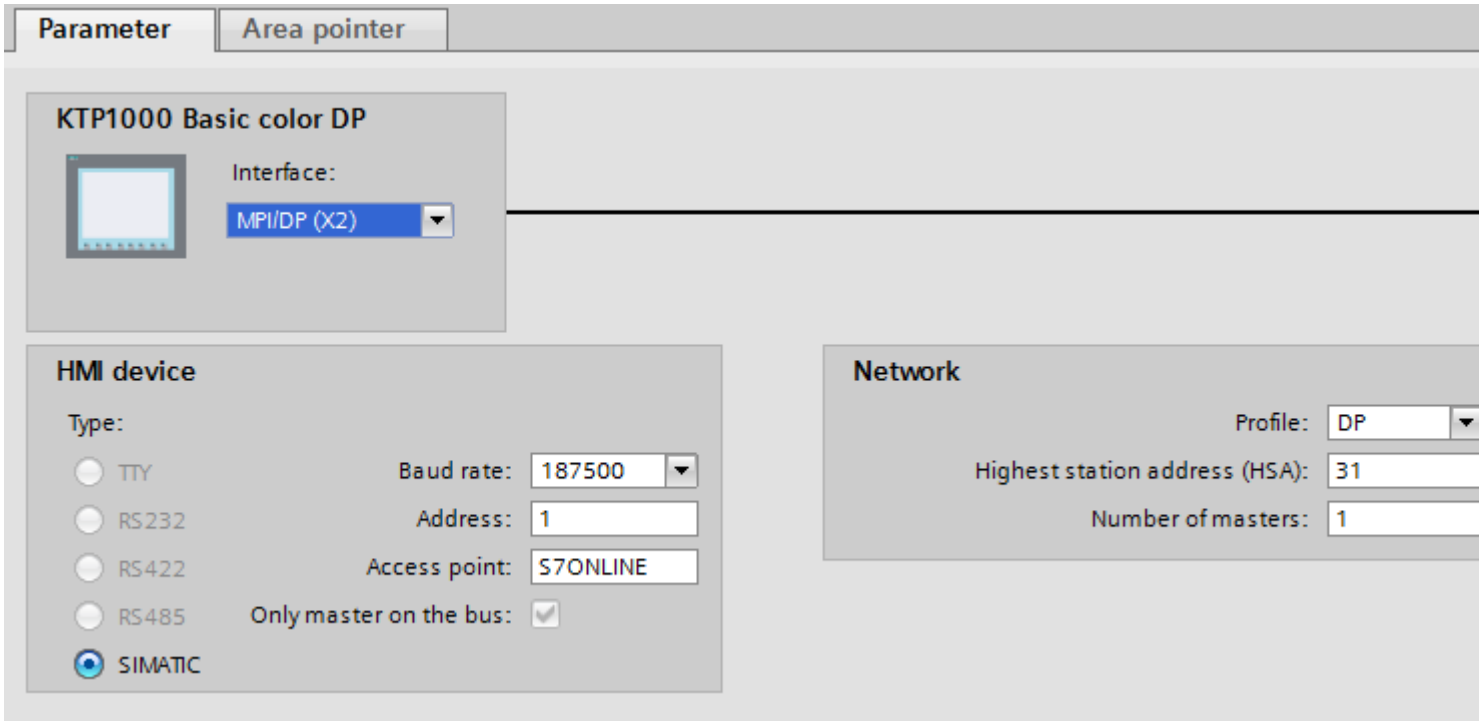
Procedure

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 with SIMATIC S7 300/400 (Page 401)
Creating a PROFINET connection (Page 492)
Creating a PROFIBUS connection (Page 495)

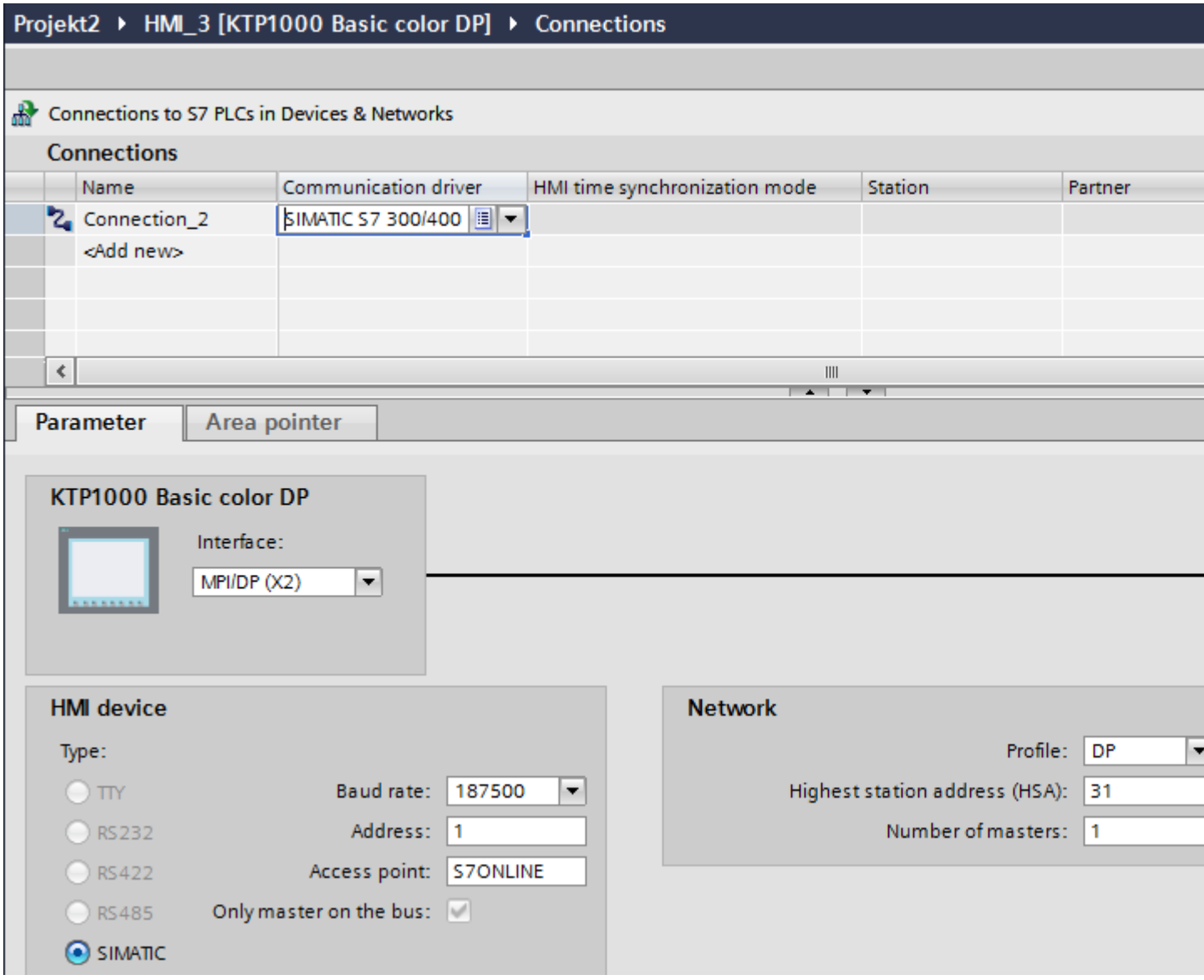
2.8.7.4 Parameters for the connection

Parameters for the connection (SIMATIC S7 300/400)

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 401)

Ethernet parameters (Page 500)

PROFIBUS parameters (Page 501)

MPI parameters (Page 503)

Cyclic operation (Page 504)

Ethernet parameters

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 401)

Parameters for the connection (SIMATIC S7 300/400) (Page 498)

PROFIBUS parameters (Page 501)

MPI parameters (Page 503)

Cyclic operation (Page 504)

PROFIBUS parameters

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.
- "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.
- "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 401)

Parameters for the connection (SIMATIC S7 300/400) (Page 498)

Ethernet parameters (Page 500)

MPI parameters (Page 503)

Cyclic operation (Page 504)

MPI parameters

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.
- "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 401)

Parameters for the connection (SIMATIC S7 300/400) (Page 498)

Ethernet parameters (Page 500)

PROFIBUS parameters (Page 501)

Cyclic operation (Page 504)

Cyclic operation

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 401)
Parameters for the connection (SIMATIC S7 300/400) (Page 498)
Ethernet parameters (Page 500)
PROFIBUS parameters (Page 501)
MPI parameters (Page 503)

2.8.8 Creating a connection in the "Connections" editor

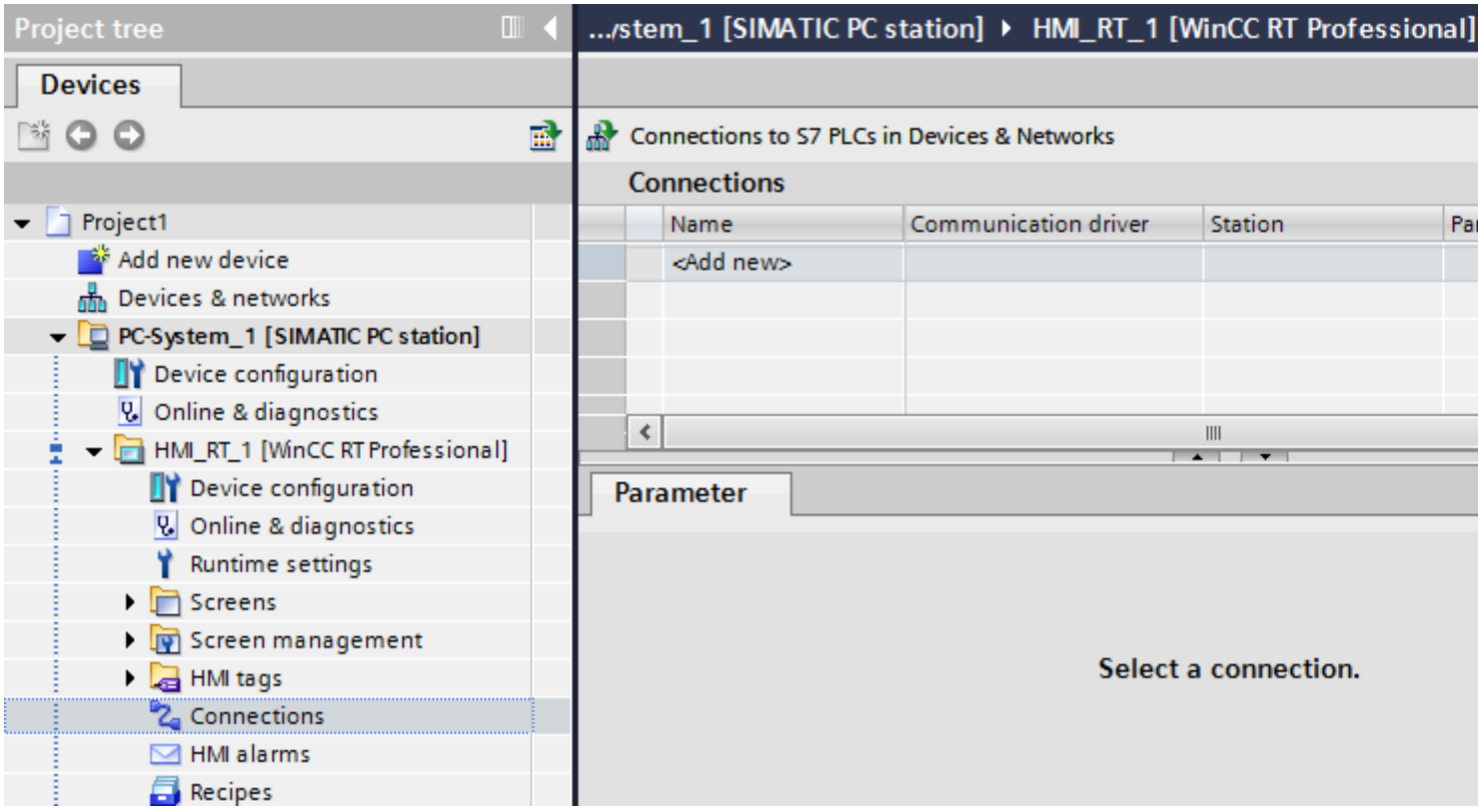
2.8.8.1 Creating a connection

Requirements

- A project is open.
- WinCC RT Professional is 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 an interface of the HMI device in the Inspector window under "Parameters > Interface".
6. Set the parameters for the connection in the Inspector window.

Interfaces

Select one of the following interfaces in the Inspector window under "Parameters > WinCC RT Professional > Interfaces".

- Industrial Ethernet
- MPI
- PROFIBUS
- PLC slot
- Soft PLC
- TCP/IP

For additional information on the parameters of the interfaces see:

Parameters for the connection (SIMATIC S7 300/400) (Page 507)

See also

Communication with SIMATIC S7 300/400 (Page 401)
Parameters for the connection (SIMATIC S7 300/400) (Page 507)

2.8.8.2 Parameters for the connection (SIMATIC S7 300/400)

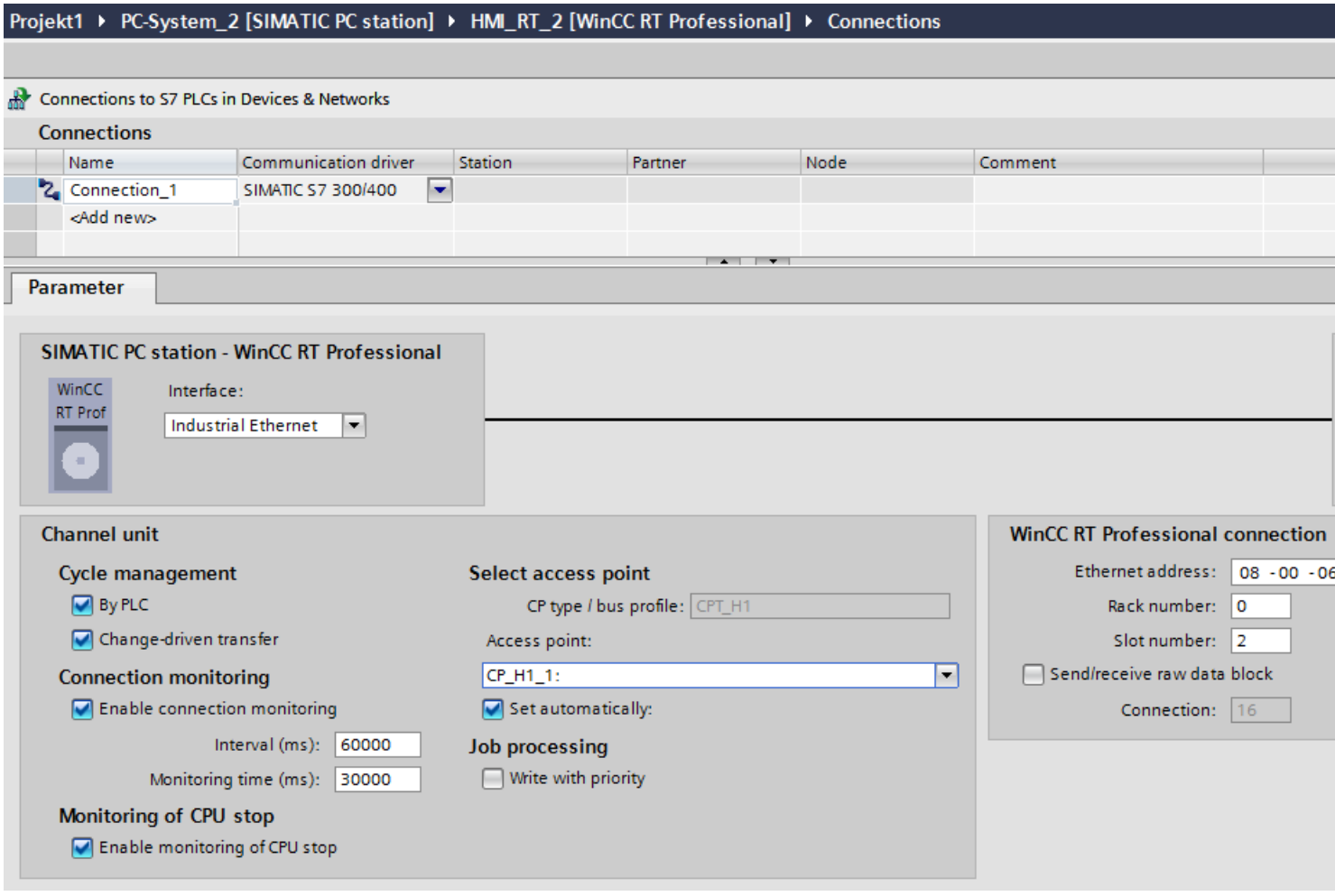
Ethernet

Introduction

You assign the parameters for the HMI device and the PLC in the network under "Parameters".

The parameters described in the following apply to the following interfaces:

- Industrial Ethernet
- TCP/IP



Channel unit

- "Cycle management"
  Select the following under "Cycle management":
  - "By PLC"
    Enabled: Cycle management is enabled by the PLC.
    Disabled: Cycle management is not enabled by the PLC.
  - "Transfer on change"
    Enabled: A data transfer is carried out at data change.
    Disabled: No data transfer is carried out at data change.

  More information: Cyclical reading (Page 510).
- "Select access point"
  Enter the logical device name under "Access point".
  If only one communications processor is installed for a communication type, select the option "Set automatically". The logical device name is then set automatically when you start Runtime.
- "Connection monitoring"
  Connection monitoring is carried out when you select "Enable".
  You can also specify an "interval" and a "Monitoring time" for connection monitoring.
  Interval: Specified in milliseconds
  Monitoring time: Specified in milliseconds
- "Monitoring of CPU stop"
  If you select the "Enable" option under "Monitoring of CPU stop", the CPU state is monitored.
- "Job processing"
  When you activate the "Write with priority" function, writing is transferred to the PLC with a higher priority than reading.

WinCC RT Prof connection

- "Ethernet address" or "TCP/IP address"
  Enter the station address of the CPU under "Ethernet address".
- "Rack number"
  Enter the rack number of the CPU to be addressed under "Rack number".
- "Slot number"
  Enter the number of the slot of the CPU to be addressed of the already selected rack number under "Slot number".
- "Send/receive raw data block"
  When you select the function "Send/receive raw data block", you enable the sending of "raw data blocks" via BSEND/BRCV.

  Note

  Raw data communication to an S7-300 is not possible.

- "Connection"
  This function can only be edited if you have selected "Send/receive raw data block".
  You define the connection resource of the "raw data blocks" with the "Connection" function.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Creating a connection (Page 505)

Serial communication (Page 511)

PLC (Page 513)

Cyclical reading

Description of cycle management

For the interface at the PC system, you can define whether the cyclical read services should be used. In this process, the variables to be read cyclically are compiled in individual requirements, and sent to the AS. The AS sends the requested data once immediately upon receiving them, and thereafter, every time the cycle duration elapses. If the requested data are not required any more, for example, when there is a picture change, WinCC deletes the cyclical read service in the AS.

Normally, the cyclical read services should be read in the AS. Therefore, the corresponding check box is checked by default in the system parameters of the channel unit. Only if the cyclical services are not to be used must this setting be changed.

Only if the cyclical read services are activated can the change transfer also be used. Then, transfer of the data from the AS takes place only if there is a change in the value, and once per AS cycle. The function must be supported by the AS.

The AS and the AS-OS communication is stressed less by the cyclical services and the change transfer, since read jobs do not have to be continuously sent to the AS and processed there.

With acyclic read services, the variables to be read are compiled into a single requirement and sent to the AS. The AS sends the required data once. The cycle management for the requirement is performed by WinCC.

Number of cyclical read services in a CPU

The number of cyclical read services depends on the resources that are available in the AS. Thus, for a S7-300, a maximum of 4 cyclic read services are permitted, for S7-416 or 417, maximum 32 services. This number refers to all the participants in the communication with the AS, i.e. if several WinCC systems communicate with one AS, they have to share the resources of the AS. If the maximum number of services is exceeded, a further cyclical read service is rejected. WinCC must then demand these data via acyclic read jobs and itself perform the cycle management.

Requiring external variables in scripts

After a picture selection, the use of the cyclic read services has no effect on the initial update, if there are no scripts present in the selected picture that demand external variables via the "GetTagWord()" function. If scripts are performed with "GetTagWord()" upon picture selection, a clumsy configuration of these scripts can result in new variable demands being continuously submitted to the channel after a picture change. If external variables are required in a script, then "Variable" should be specified for them as the trigger event.

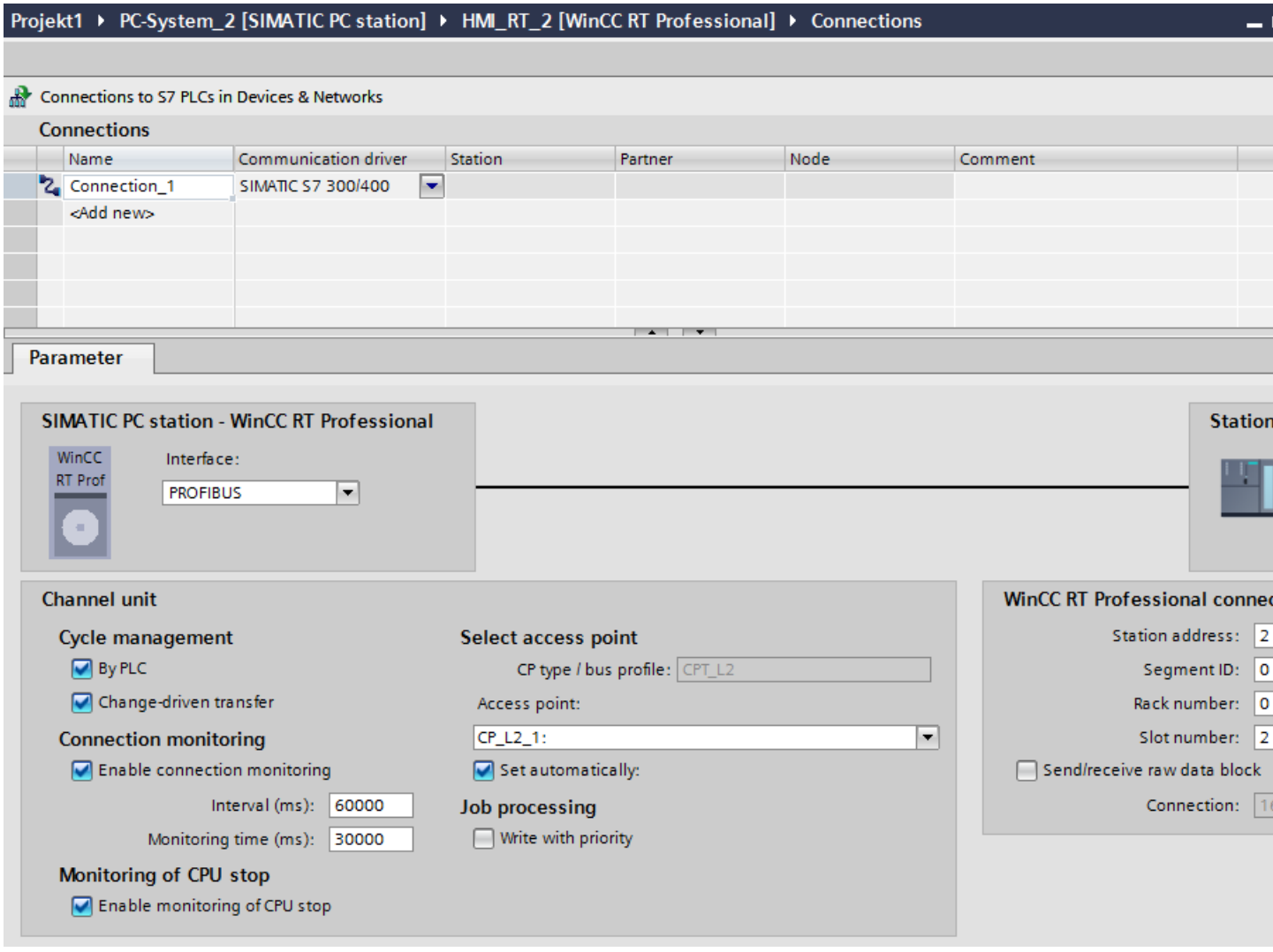
Serial communication

Introduction

You assign the parameters for the HMI device and the PLC in the network under "Parameters".

The parameters described in the following apply to the following interfaces:

- PROFIBUS
- MPI



Channel unit

- "Cycle management"
  Select the following under "Cycle management":
  - "By PLC"
    Enabled: Cycle management is enabled by the PLC.
    Disabled: Cycle management is not enabled by the PLC.
  - "Transfer on change"
    Enabled: A data transfer is carried out at data change.
    Disabled: No data transfer is carried out at data change.
- "Select access point"
  Enter the logical device name under "Access point".
  If only one communications processor is installed for a communication type, select the option "Set automatically". The logical device name is then set automatically when you start Runtime.
- "Connection monitoring"
  Connection monitoring is carried out when you select "Enable".
  You can also specify an "interval" and a "Monitoring time" for connection monitoring.
  Interval: Specified in milliseconds
  Monitoring time: Specified in milliseconds
- "Monitoring of CPU stop"
  If you select the "Enable" option under "Monitoring of CPU stop", the CPU state is monitored.
- "Job processing"
  When you activate the "Write with priority" function, writing is transferred to the PLC with a higher priority than reading.

WinCC RT Prof connection

- "Station address"
  Enter the station address of the CPU under "Station address".
- "Segment ID"
  Enter the segment ID of the CPU under "Segment ID".
- "Rack number"
  Enter the rack number of the CPU to be addressed under "Rack number".
- "Slot number"
  Enter the number of the slot of the CPU to be addressed of the already selected rack number under "Slot number".
- "Send/receive raw data block"
  When you select the function "Send/receive raw data block", you enable the sending of "raw data blocks" via BSEND/BRCV.
- "Connection"
  This function can only be edited if you have selected "Send/receive raw data block".
  You define the connection resource of the "raw data blocks" with the "Connection" function.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Creating a connection (Page 505)

Ethernet (Page 507)

PLC (Page 513)

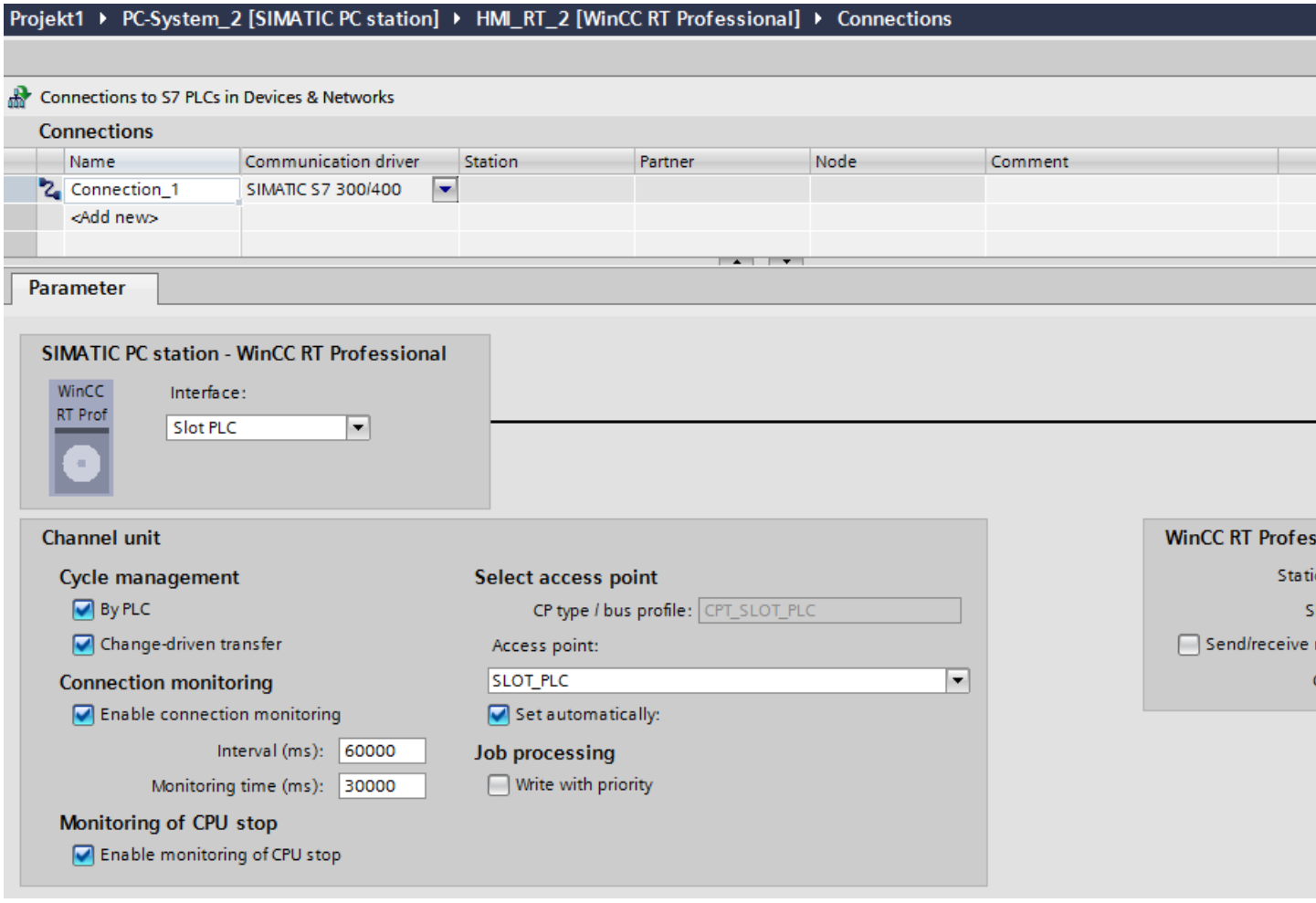
PLC

Introduction

You assign the parameters for the HMI device and the PLC in the network under "Parameters".

The parameters described in the following apply to the following interfaces:

- PLC slot
- Soft PLC



Channel unit

- "Cycle management"
  Select the following under "Cycle management":
  - "By PLC"
    Enabled: Cycle management is enabled by the PLC.
    Disabled: Cycle management is not enabled by the PLC.
  - "Transfer on change"
    Enabled: A data transfer is carried out at data change.
    Disabled: No data transfer is carried out at data change.
- "Select access point"
  Enter the logical device name under "Access point".
  If only one communications processor is installed for a communication type, select the option "Set automatically". The logical device name is then set automatically when you start Runtime.
- "Connection monitoring"
  Connection monitoring is carried out when you select "Enable".
  You can also specify an "interval" and a "Monitoring time" for connection monitoring.
  Interval: Specified in milliseconds
  Monitoring time: Specified in milliseconds
- "Monitoring of CPU stop"
  If you select the "Enable" option under "Monitoring of CPU stop", the CPU state is monitored.
- "Job processing"
  When you activate the "Write with priority" function, writing is transferred to the PLC with a higher priority than reading.

WinCC RT Prof connection

- "Station address"
  Enter the station address of the CPU under "Station address".
- "Slot number"
  Enter the number of the slot of the CPU to be addressed of the already selected rack number under "Slot number".
- "Send/receive raw data block"
  When you select the function "Send/receive raw data block", you enable the sending of "raw data blocks" via BSEND/BRCV.
- "Connection"
  This function can only be edited if you have selected "Send/receive raw data block".
  You define the connection resource of the "raw data blocks" with the "Connection" function.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Creating a connection (Page 505)

Ethernet (Page 507)

Serial communication (Page 511)

2.8.9 Troubleshooting of connection errors (commissioning)

2.8.9.1 Procedure for the localization of errors

S7 channel rectification in case of connection problems

Introduction

If you encounter problems during the commissioning of WinCC S7, the following description provides initial information. The description is for error localization and error correction and does not replace the WinCC description.

Follow the following instructions to locate the error.

- Creating a new project
- Select the correct communication subsystem
- Select the correct channel unit
- Configure the correct AS network address
- Check the system bus
- Check AS with STEP 7
- S7 log function
- Special settings of the S7 channel
- Error codes for connection fault
- API error texts

Error code

The most important error codes are listed in the section "Error codes for connection fault and errors in functions". If an error occurs with an error code that is not in the table, please call the WinCC hotline.

For error codes for a connection fault (SIMATIC S7-300/400) see under numbers 0112 to D406.

You will find errors which occur during editing of channel functions with the display "Error xx occurred in "function name" function!" under the numbers 1 to 100.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Creating a new project (Page 516)

Selection of the correct communication subsystem (Page 516)

Selection of the correct channel unit (Page 517)

Configure the correct AS network address (Page 518)
Checking the bus system (Page 518)
Checking the AS with STEP 7 (Page 519)
S7 log function (Page 519)

Creating a new project

If you communicate in your current project with several automation systems, create a new project with a single connection to a automation system. The errors can be easier localized here.

See also

Communication with SIMATIC S7 300/400 (Page 401)
S7 channel rectification in case of connection problems (Page 515)
Selection of the correct communication subsystem (Page 516)
Selection of the correct channel unit (Page 517)
Configure the correct AS network address (Page 518)
Checking the bus system (Page 518)
Checking the AS with STEP 7 (Page 519)
S7 log function (Page 519)

Selection of the correct communication subsystem

The S7 channel communicates via the communication subsystem SAPI-S7 or S7DOS.

SAPI-S7

SAPI-S7 is a single product and is used for the communication to the S7 automation systems. SAPI-S7 products are adjusted to the corresponding CP. The correct SAPI-S7 product has to be installed to the available CP during communication via SAPI-S7. The S7 channel uses the tag services for reading and writing of tags from SAPI-S7.

For more complex functions, e.g. PMC error message processing or the BSEND/BRCV function, activate the communication subsystem S7DOS.

S7DOS

S7DOS is a part of STEP 7 and is also used for the communication to S7. In addition to the read and write jobs, the more complex PMC services are also activated during communication via S7DOS. For clients who have no STEP 7 installed on the WinCC target machine, S7DOS is also on the WinCC-DVD delivered. For installation of S7DOS see WinCC setup description.

For the single communication subsystems, limitations for supported communication cards are possible. For information, refer to your current product information.

The S7 channel uses the following method to recognize which communication subsystem will be used by the S7 channel for communication: It first attempts to load the S7DOS communication subsystem. If S7DOS cannot be loaded, it attempts to load SAPI-S7. The

error message "Error during loading of S7 communication driver" will be displayed if SAPI-S7 cannot be loaded either.

See also

Communication with SIMATIC S7 300/400 (Page 401)

S7 channel rectification in case of connection problems (Page 515)

Creating a new project (Page 516)

Selection of the correct channel unit (Page 517)

Configure the correct AS network address (Page 518)

Checking the bus system (Page 518)

Checking the AS with STEP 7 (Page 519)

S7 log function (Page 519)

Selection of the correct channel unit

The selection of the channel unit depends on the communication card in your PC. A specific type of communication card is assigned to each channel unit.

Communication with the S7 takes place via so-called 'Logical devices'. The name of a logical device is assigned during installation of the communication card. In the case of some installations you can assign the name also via "Busprofile".

Depending on the installed communication card, the following defaults are standard:

- MPI card as well as the communication cards CP5411 and CP 5511: MPI
- "SIMATIC NET PROFIBUS" communication cards: e.g., CP5611, CP5613, CP5623
- "SIMATIC Industrial Ethernet" communication cards: CP_H1_1:
- Slot PLCs: SLOT_PLC
- TCP/IP communication cards: CP-TCPIP

The S7 communication driver differentiates accordingly the following units:

- MPI for communication via the MPI card;i.e. logical device: MPI
- PROFIBUS communication via the SIMATIC NET PROFIBUS cards (CP5623), which means logical device: CP_L2_1:
- Industrial Ethernet for communication via SIMATIC Industrial Ethernet cards (e.g., CP1623 or CP1628); which means logical device: CP_H1_1:
- Slot PLC for communication via a Slot PLC;i.e. logical device: SLOT_PLC
- TCP/IP for communication via the TCP/IP protocol; i.e. logical device: CP-TCPIP

During installation of the communication drivers the physical device is assigned via which the communication takes place.

You can change the default "logical device names" of the channel unit via the unit-specific setting of the system parameter.

See also

Communication with SIMATIC S7 300/400 (Page 401)
S7 channel rectification in case of connection problems (Page 515)
Creating a new project (Page 516)
Selection of the correct communication subsystem (Page 516)
Configure the correct AS network address (Page 518)
Checking the bus system (Page 518)
Checking the AS with STEP 7 (Page 519)
S7 log function (Page 519)

Configure the correct AS network address

For example, the TSAPs have a fixed setting for the S7 network address. This does not usually need to be configured. The set address as well as the rack and the slot number still have to be configured.

Check the configured network address with the AS configured network address.

The rack number and the slot number have only to be specified, if the communication takes place on AS side via a communication processor and not via the MPI interface installed on the CPU. In this case specify the rack and the slot number of the CPU that is to be addressed. If MPI interface installed in the CPU is used, enter the value 0 for rack and slot number.

See also

Communication with SIMATIC S7 300/400 (Page 401)
S7 channel rectification in case of connection problems (Page 515)
Creating a new project (Page 516)
Selection of the correct communication subsystem (Page 516)
Selection of the correct channel unit (Page 517)
Checking the bus system (Page 518)
Checking the AS with STEP 7 (Page 519)
S7 log function (Page 519)

Checking the bus system

Check the correct setup of the bus system.

SIMATIC NET PROFIBUS

- Check the network for communication stations with the same station address.
- Check if the highest station address (HSA) is correctly set for all communication stations.
- Check if an enabled device , e.g. an AS, is connected at the start and the end of the PROFIBUS.

- Check if the terminating resistors are correctly enabled or disabled. The terminating resistors may only be enabled at the first and the last bus terminal.
- If you have an MPI cable, connect the AS directly to your MPI card or your CP5623.

SIMATIC Industrial Ethernet

- Check the network for communication stations with the same Ethernet addresses.

See also

Communication with SIMATIC S7 300/400 (Page 401)

S7 channel rectification in case of connection problems (Page 515)

Creating a new project (Page 516)

Selection of the correct communication subsystem (Page 516)

Selection of the correct channel unit (Page 517)

Configure the correct AS network address (Page 518)

Checking the AS with STEP 7 (Page 519)

S7 log function (Page 519)

Checking the AS with STEP 7

If you have installed STEP 7, check if you can access the automation system with STEP 7. If problems occur, follow the STEP 7 instructions for error correction.

See also

Communication with SIMATIC S7 300/400 (Page 401)

S7 channel rectification in case of connection problems (Page 515)

Creating a new project (Page 516)

Selection of the correct communication subsystem (Page 516)

Selection of the correct channel unit (Page 517)

Configure the correct AS network address (Page 518)

Checking the bus system (Page 518)

S7 log function (Page 519)

S7 log function

The S7 channel has a logbook function. You can use this file to write the most important status changes and errors to an ASCII file on an external memory. These messages can be used to analyze a communication problem.

The individual messages are described below:

Start and end messages

- S7 channel DLL started
- S7 channel DLL terminated

Unit activated or deactivated

- S7 channel unit "unitname" activated
- S7 channel unit "unitname" deactivated

Version information

- S7DOS version: Version string
- S7CHN version: Version string

Communication error

- Cannot connect to "connectionname": Errorcode 0xhhhh ffff

The message is displayed if communication to the corresponding automation device cannot be established immediately after WinCC is activated. If the connection was established correctly at least once, the following message is displayed if errors occur:

- Connectionerror nnn " connectionname": Errorcode 0xhhhh ffff

	Description
nnn	Number of connection terminations for this connection
Connectionname	Connection name
hhhh	1. Error display in hexa S7DOS / SAPI-S7
ffff	2. Error display in hexa S7DOS / SAPI-S7

Channel API error

- Channel API error: Errorstring

The error string 'errorstring' is passed by the channel to the WinCC Explorer. Depending on the significance of the error, the error string may or may not be displayed in a notice box. For a description of the error strings, please see the API Error Text.

- Max. count of API errors reached - API logbook deactivated

Errors on the API interface can cyclically occur depending on the error and function. To avoid filling the logbook file with these error messages, a maximum of 32 messages are output for an API error.

General channel error messages

- Cannot write storage data
- Cannot read storage data / use default data
- Storage data illegal or destroyed / use default data
- No storage data / use default data

Initialization messages

- Devicename in unit "unitname" changed from "old devicename" to "new devicename"

Message when maximum file length is exceeded

- Max. logbooksize reached - Logbook deactivated

In order to avoid filling of the data medium only with the logbook file, the logbook output has a length monitoring. As soon as the specified length is reached, the logbook is deactivated. The message is only output, when message output causes the max. file length to be exceeded. No message is output, if the file length is changed with an editor or the maximum file length is reduced in the INI file.

The output is not time critical, because the logbook functions are called only during activate/ deactivate and in the case of problems of the communication process.

Example of a logbook protocol:	
17.09.1996/15:16:18.330	S7 channel DLL started
17.09.1996/15:18:05.380	S7DOS version: @(#)TIS-Block Library DLL Version X3.060-DEB-BASIS
17.09.1996/15:18:05.490	S7CHN version: V2.0 / Sep 16 1996 / 18:42:48
17.09.1996/15:18:05.600	S7 channel unit "S7-MPI" activated
17.09.1996/15:36:48.490	Cannot connect to "MPI_CPU_3": Errorcode 0xFFDF 4107
17.09.1996/15:43:29.060	Connectionerror 1 "MPI_CPU_4": Errorcode 0xFFDF 410E
17.09.1996/16:43:44.830	Connectionerror 2 "MPI_CPU_4": Errorcode 0xFFDF 410E
17.09.1996/17:45:29.960	Connectionerror 3 "MPI_CPU_4": Errorcode 0xFFDF 410E
17.09.1996/18:54:02.630	Connectionerror 4 "MPI_CPU_4": Errorcode 0xFFDF 410E
17.09.1996/22:56:47.680	S7 channel unit "S7-MPI" deactivated
17.09.1996/22:56:56.740	S7 channel DLL terminated

The logbook texts are not language-dependent with the exception of "Channel API Errorstring" and are always displayed in English.

The S7 logbook is enabled by default. The logbook is stored in WinCC diagnose folder.

To settings and assigning parameters of the logbook function, see "Setup of an S7 logbook.

See also

Communication with SIMATIC S7 300/400 (Page 401)

S7 channel rectification in case of connection problems (Page 515)

Creating a new project (Page 516)

Selection of the correct communication subsystem (Page 516)

Selection of the correct channel unit (Page 517)

Configure the correct AS network address (Page 518)

Checking the bus system (Page 518)

Checking the AS with STEP 7 (Page 519)

Special settings of the S7 channel

Special settings of the S7 channel

Edit with an ASCII-Editor, e. g. NOTEPAD.EXE, a file with name S7CHN.INI. Via the entries in the INI file you can modify the properties of the S7 channel. The S7CHN.INI file is expected in the directory which also contains the S7 communication driver "SIMATIC S7 300/400.CHN". If no file S7CHN.INI is found, the properties are not modified. The file is set to Windows INI file conventions.

[section]

key=string

In the following the relevant parameters for modification of S7 channel are described. You just have to apply the lines in the INI file which create the setting you want. Comply with the specified examples:

- Change the default of the logical device name of the channel unit
- Change CP type of a channel unit
- Setup S7 logbook

Additional specific settings:

- Modify channel characteristics via the S7chn.ini file

See also

Communication with SIMATIC S7 300/400 (Page 401)

Change the default of the logical device name of a channel unit (Page 522)

Change the CP type of a channel unit (Page 523)

Setup of an S7 logbook (Page 524)

s7chn.ini (Page 525)

Change the default of the logical device name of a channel unit

```
[Unit:unitname]

CpName = name
```

Name

Logical device name of the communication card (for example: CP_L2_1:).

Example: You want to set the channel unit "PROFIBUS" to a new logical device name "mein_cp".

```
 [Unit:PROFIBUS]
CpName = mein_cp
```

Note

The modification via INI file is only for setup of the default. The setting of the logical device name takes place via the unit specific system parameter setting.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Special settings of the S7 channel (Page 522)

Change the CP type of a channel unit (Page 523)

Setup of an S7 logbook (Page 524)

s7chn.ini (Page 525)

Change the CP type of a channel unit

```
[Unit:unitname]
CpName = name
CpType = type
```

Name

Logical device name of the communication card (for example: CP_L2_1:).

Type

Type of the communication card (for example: L2). The following settings are available:

- MPI
- L2
- H1

Example: You want to set the channel unit "PROFIBUS" to a new logical device name "cp_neu". You want to activate the connection configuration mask to configure the AS network address to enter a special AS address according to SCI convention.

```
CpName = cp_neu
CpType = NNN
```

Note

Only change the CP type if you want to activate the connection configuration mask to configure the AS network address to enter a special AS address according to SCI convention.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Special settings of the S7 channel (Page 522)

Change the default of the logical device name of a channel unit (Page 522)

Setup of an S7 logbook (Page 524)

s7chn.ini (Page 525)

Setup of an S7 logbook

```
[Logbook]

FileName = C:\TMP\S7CHN.LOG

MaxKbFileSize = 32

MaxCycleBuffer= 2

Selection = FF.FF.FF.FF
```

FileName

Name of the diagnostics file incl. path specification. If no path is specified, the path of the called module is used. The name of the diagnostic file should be unique in the first 6 characters. In the circular buffer mode the file name is limited to maximum 6 characters and completed with the current cycle number 01 to 99. (Example: FileName = C:\TMP\S7CHN.LOG). The specified folder has to be available when the communication driver is started. The folder is not automatically created by S7 channel DLL.

MaxKbFileSize

Maximal length of the diagnostic file in k byte. If the maximal length is reached, the saving process is completed or switched to another file, depending on the "MaxCycleBuffer" setting. (Example: MaxKbFileSize = 32)

MaxCycleBuffer

Maximum number of circular buffer files. If the parameter is not specified or "MaxCycleBuffer=0" is specified, only one file is created. For input of a number 2..99 a new file is created if the maximum file length is reached. In the circular buffer mode the file name is limited to maximum 6 characters and completed with the current cycle number 01..99. If the maximum buffer number is reached, the saving process of the files continues with the file 01. (Example: MaxCycleBuffer = 2) ==> writing in files: S7CHN01.LOG -> S7CHN02.LOG -> S7CHN01.LOG ...

Selection

Number of the message to be saved. The selection is specified as 32 bit hexastring. Set for each desired kind of message the corresponding bit. (Example: Selection=FFFFFFFF). In order to simplify the input, an ' ' or '.' character can be inserted byte by byte. (Example: Selection = FF.FF.FF.FF)

The individual selection ID has the following values:

Selection ID	Value
00.00.00.01	Start and end error message
00.00.00.02	Error message if unit activated/deactivated
00.00.00.04	Version information
00.00.00.08	Communication errors
00.00.00.10	Channel API error
00.00.00.20	General channel error messages
00.00.00.40	Initialization messages

The selection = FF.FF.FF.FF causes all logbook error messages to be output. The output is not time critical, because the logbook functions are called only during activate/deactivate and in the case of problems of the communication process.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Special settings of the S7 channel (Page 522)

Change the default of the logical device name of a channel unit (Page 522)

Change the CP type of a channel unit (Page 523)

s7chn.ini (Page 525)

s7chn.ini

The characteristics of the S7 channel can be modified via the S7CHN.INI file. The file is only for test or diagnostics and is not required for the normal operation.

Since the runtime characteristics of the S7 channel can be significantly changed using the INI file described below, the options for modifying this file should only be used for internal testing or error analysis!

The interpretation and modification of the parameters is only intended for "insiders". Therefore, we have not gone into more detail on the meaning and application of the parameters and constants.

```
; !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
; !!!!!    A T T E N T I O N    !!!!!      ============================
; !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
; The S7CHN.INI is not required for the runtime!  Only for
; for testing and diagnostics. *) default without S7CHN.INI

;[Debug]
;DebugMode      = 1       ; 0 = off *), 1 = DebugMessage, 2 = S7 Console
;DebugLevel     = 101     ; 100 = Info, 101 = Error *), 102 = CHN API, 103 = Trace, 104 = List
;DebugTime      = 1       ; 0 = off, 1 = on *)
```

```
; !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

;[SAPI]
;TraceTarget    = 1       ; 0 = BUFFER *), 1 = NEW_FILE
;TraceSelect    = 65535   ; 1 = ADMIN_SERVICES *), 65535 = SELECT_ALL
;TraceDepth     = 104     ; 0 = OFF *), 101 = INTERFACE, 103 = LIB0, 104 = LIB0_PDU

;[Timer]
;TimerMode      = 1       ; 0 = external, 1 = internal *)
;TimerCycle     = 1000    ; cycle time in milliseconds
;CycleConnInit = 4000    ; Cycle time for autom. connection establishment
;FunTimeout     = 10000   ; Timeout time for function calls
;CloseTimeout   = 10000   ; Timeout time Close for unit deactivation
;MaxAsCycle     = 5000    ; Max. cycle for cyclic read jobs by AS

;[Channel]
;UnitStorage    = 1       ; Storage Read access: 0 = off, 1 = on
;LocalSlotId    = 0       ; Setting of the local slot number in the TSAP
;ThreadModel    = 1       ; Thread Model (0 without , 1 with, 2 CHNCAP_REENTRANT)
;S7String       = 1       ; String processing (0 with, 1 without control word)

;[Driver]
;ComSystem      = 0       ; 0 = auto *), 1 = SAPI, 2 = S7DOS
;CnfAmqCalling = 0       ; max. rec.-jobs without acknowledgement (0 default)
;CnfAmqCalling = 0       ; max. send.-jobs without acknowledgement (0 default)

;[Unit:Industrial Ethernet]
;CpName         = CP_H1_1:
;CpType         = NNN

;[Diagnosis]
;ChkStopState   = 0       ; Stop check: 0 = off, 1 = on *)
;ChkLifeTime    = 60000   ; cycle time for lifebeat monitoring in milliseconds 0 = off
;LifeTimeAck    = 15000   ; lifebeat acknowledgement timeout in milliseconds

;[Logbook]
;FileName       = C:\TMP\S7CHN.LOG
;MaxKbFileSize = 20
;MaxCycleBuffer= 2
;Selection      = FF.FF.FF.FF

;[ChnTrace]
;FileName       = C:\TMP\S7CHN.TRC
;MaxKbFileSize = 50      ; max. length of the diagnostic file in k byte
```

```
; !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
;MaxCycleBuffer= 10      ; max. number circular buffer files
;DebugMode     = 0       ; 0 = Output in file, 1 = Output via 'OutputDebugString()'
;TimeStampMode = 2       ; 0 = no time stamp, 1 or 2 = with time stamp
;Selection     = 00.00.30.27
                ; 00 00 00 01        // General alarms
                ; 00 00 00 02        // Connection status
                ; 00 00 00 04        // S7 Function calls # Tag services
                ; 00 00 00 08        // S7 Function calls # Tag services

                ; 00 00 00 10        // S7 Function call data
                ; 00 00 00 20        // Message event # Tag services
; 00 00 00 40         // Message event Tag services
                ; 00 00 00 80        // Message event data

                ; 00 00 01 00        // Message on start processing

                ; 00 00 10 00        // S7 raw data function
                ; 00 00 20 00        // S7 raw data

                ; 00 01 00 00        // S7 Channel API # Tag services
                ; 00 02 00 00        // S7 Channel API Tag services
;MaxDump       = 512     ; Max. number bytes on data trace (Hex-Dump) of the channel
diagnosis
```

See also

Communication with SIMATIC S7 300/400 (Page 401)

Special settings of the S7 channel (Page 522)

Change the default of the logical device name of a channel unit (Page 522)

Change the CP type of a channel unit (Page 523)

Setup of an S7 logbook (Page 524)

2.8.9.2 Error codes

Error codes for connection faults (SIMATIC S7-300/400)

The most important error codes are listed in this section. If an error occurs with an error code that is not in the table, please call the WinCC hotline.

- Error 0112
- Error 4022
- Error 4102

- Error 4107
- Error 410E
- Error 4110
- Error 4116
- Error 411A
- Error 411C
- Error 4230
- Error 4231
- Error 4232
- Error 42B0
- Error 42B1
- Error 42B2
- Error 42B3
- Error 42B5
- Error 42B7
- Error 42B8
- Error 42C0
- Error 42C2
- Error 42D2
- Error 7000
- Error 7001
- Error 7002
- Error 7003
- Error 7004
- Error 7005
- Error 7006
- Error 7007
- Error 7008
- Error 7101
- Error 7102
- Error 7900
- Error 8204
- Error 8305
- Error 8404
- Error 8405

- Error 8500
- Error D405
- Error D406

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error 4230 - L4_SCI_E_UNPLUGGED

No further active partner available.

- PC not switched to the bus or power plug not correctly plugged.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 0112 - INVALID_PARAM

An incorrect parameter was transferred to the communication subsystem S7DOS.

- Message processing or CPU Stop Monitoring enabled and an outdated S7DOS version used.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 4022 - S7DOS_SRV_TIMEOUT

No feedback from AS:

- AS did not respond to job within the timeout period.
- Station address available several times on the bus.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 4102 - L4_INVALID_REQ

Request block not permitted.

- An invalid request block was returned by the communication driver.
- A not yet processed request block was returned after a CLOSE_REQ.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 4107 - L4_OK_CLOSED_RESP

Connection was terminated.

- Rack/Slot not right configured.
- Number of maximum permitted connections in AS exceeded.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 410E - L4_REM_ABORT

Connection was aborted.

- AS or CP disabled or reset.
- AS not connected to bus or error in bus system.
- Number of maximum permitted connections in AS exceeded.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 4110 - L4_LOC_TIMEOUT

No connection established. AS denies connection setup.

- Wrong internet address configured.
- AS disabled.
- AS not connected to bus or error in bus system.
- Number of maximum permitted connections in AS exceeded.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 4116 - L4_CONN_REJECT

No connection established. AS denies connection setup.

- Rack/Slot not right configured. For external CP module, the slot of the CPU module has to be indicated.
- Number of maximum permitted connections in AS exceeded.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 411A - L4_ILLEGAL_ADDRESS

Invalid address.

- An invalid network address was specified during connection configuration.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 411C - L4_NETWORK_ERROR

Network error.

- A fatal error was encountered in the network.
- The network address or the segment ID was configured incorrectly.
- Communication module defective.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 4231 - L4_SCI Error 4231 - L4_SCI

BUS disrupted.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 42B0 - L4_DLL_E_NO_HW

No communication hardware found.

- Communication module defective.
- Communication module not installed correctly.
- Wrong port address defined.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 42B1 - L4_DLL_E_HW_DEFECT

Communication module defective.

- Communication module not installed correctly.
- Wrong port address defined.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 42B2 - L4_DLL_E_CNF

- Driver configured incorrectly or invalid parameter in the registry.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 42B3 - L4_DLL_E_BAUDRATE

Incorrect baudrate or incorrect interrupt vector defined.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 42B5 - L4_DLL_E_TS

The defined local participant number (TS_ADR) is already assigned.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 42B7 - L4_DLL_E_INT_NOT_PROV

The defined interrupt vector (IRQ) is not available on the communication module.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 42B8 - L4_DLL_E_INT_BUSY

The defined interrupt vector (IRQ) is already allocated on the communication module.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 42C0 - L4_DLL_E_NO_FILE

The selected communication driver cannot be installed. File not found.

- Communication driver not installed correctly.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 4232 - L4_DLL_E_HSA

Incorrect maximum number of set stations.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

1. S7DOS error indicator

This value indicates the error detecting module. The value is not relevant for the initial error analysis.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

1. SAPI-S7 error indicator

This value indicates the "S7_MINI_DB_RESPONSE_CODE". The value is not relevant for the initial error analysis.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

2. SAPI-S7 error indicator

This value indicates the "s7_last_detailed_err_no". The table "s7_last_detailed_err_nos7_last_detailed_err_no" includes a list of SAPI-S7 error codes without additional details.

In case of problems with SAPI-S7, read the SAPI-S7 documentation or contact the SINEC or WinCC hotline.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

s7_last_detailed_err_no

Table 2-3

Error	Decimal	Hexa
S7_ERR_NO_ERROR	0	00
S7_ERR_UNKNOWN_ERROR	1	01

Error	Decimal	Hexa
S7_ERR_WRONG_CP_DESCR	2	02
S7_ERR_NO_RESOURCE	3	03
S7_ERR_INVALID_PARAMETER	7	07
S7_ERR_TOO_LONG_DATA	8	08
S7_ERR_TOO_MANY_DLL_USERS	9	09
S7_ERR_WRONG_IND_CNF	10	0A
S7_ERR_SERVICE_NOT_SUPPORTED	11	0B
S7_ERR_INVALID_CREF	20	14
S7_ERR_CONN_NAME_NOT_FOUND	23	17
S7_ERR_INVALID_ORDERID	30	1E
S7_ERR_ORDERID_USED	31	1F
S7_ERR_OBJ_UNDEFINED	50	32
S7_ERR_OBJ_ATTR_INCONSISTENT	51	33
S7_ERR_OBJ_ACCESS_DENIED	53	35
S7_ERR_INVALID_DATA_SIZE	80	50
S7_ERR_RECEIVE_BUFFER_FULL	81	51
S7_ERR_FW_ERROR	90	5A
S7_ERR_MINI_DB_TYPE	100	64
S7_ERR_MINI_DB_VALUE	101	65
S7_ERR_SERVICE_VFD_ALREADY_USED	112	70
S7_ERR_SERVICE_CONN_ALREADY_USED	113	71
S7_ERR_CONN_ABORTED	120	78
S7_ERR_INVALID_CONN_STATE	121	79
S7_ERR_MAX_REQ	122	7A
S7_ERR_CONN_CNF	123	7B
S7_ERR_INVALID_CYCL_READ_STATE	130	82
S7_ERR_INSTALL	140	8C
S7_ERR_INTERNAL_ERROR	141	8D
S7_ERR_NO_SIN_SERV	142	8E
S7_ERR_NO_LICENCE	143	8F
S7_ERR_SYMB_ADDRESS	150	96
S7_ERR_SYMB_ADDRESS_INCONSISTENT	151	97

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

No help found for this error.

The help function contains only the most important error codes. Please contact the WinCC hotline.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 42C2 - L4_DLL_E_LOGDEV

The logical device is not defined in the registry.

- Communication driver not installed correctly.
- Entry damaged or deleted in the registry.
- Check the setting of the logical device name with the "Set PG/PC interface" program.
- Check the setting for the logical device name in the "System parameter - Device" mask.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 42D2 - L4_DLL_E_NO_SMC

The connection to the adapter is disturbed.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 7000 - CEC_UDEACT

- Unit will be deactivated.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 7001 - CEC_STPCHK

Connection closed due to Stop check.

- The connection was terminated by the channel because a "STOP", "HOLD" or "DEFECTIVE" VMD status was detected for the CPU.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 7002 - CEC_LACKTO

Lifebeat acknowledgement error.

- The connection was terminated because the acknowledgment of the lifebeat frame was not received within the expected period.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 7003 - CEC_RAWFKT

Connection closed via raw data function

- The connection function was activated via a raw data function.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 7004 - CEC_OIDERR

Order id invalid

- The order ID that has to be processed is invalid.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 7005 - CEC_DTLERR

Data length incorrect.

- In the received PDU the data length is greater than the expected data length or 0. This message is displayed, e.g., when an uneven number byts is read from a AS314 prototype CPU (314-1AE00-0AB0)

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 7006 - CEC_CONERR

Connection closed or deactivated.

- Connection was terminated. That is why the job was rejected.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 7007 - CEC_VMDERR

It is not permitted to determine VMD status via lifebeat frame.

- The connection was terminated by the channel because a "STOP", "HOLD" or "DEFECTIVE" VMD status was detected for the CPU.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 7008 - CEC_CONPRJ

Connection data incorrect or corrupted.

- No connection parameters have been configured.
- The configured connection parameters are not permitted.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 7101 - CEC_NORAM1

No memory available.

- No free memory could be allocated during creation of order list.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 7102 - CEC_NORAM2

No memory available.

- No free memory could be allocated during creation of tag feedback data area.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 7900 - CEC_S7DOSE

Necessary S7DOS version not loaded.

- A function was configured (for example, PMC message processing) that requires a specific version of the communication subsystem S7DOS.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

The connection is ready to use.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

The connection is currently being initialized

Communication is only possible after the completion of the initialization.

- If the state is pending for several seconds, an overloading of the communication on the AS side might be the reason for this.
- Check the PM configuration in your AS.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

The connection to the AS is established

The CPU is in STOP mode.

- To communicate with the CPU via WinCC, switch the CPU to RUN mode.
- You have the possibility to also monitor the PLC in STOP mode. System Diagnostics and Alarming can continue to be run in this way.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 8204 - L7_DEF_OBJ_INCONSISTENT

The object's type format is inconsistent.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 8305 - L7_FUNC_NOT_AVAIL

Functionality not available.

- AS-CPU type does not support the configured function.
- AS CPU does not contain the most recent firmware status.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 8404 - L7_FATAL_ERR

AS protocol error.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 8405 - L7_CPU_IN_PROTECTED_STATE

The remote block is in status DISABLE (PBK). The function cannot be executed.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error 8500 - L7_PDU_SIZE_ERR

Wrong PDU size.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error D405 - L7_DGS_FKT_PAR_SYNTAX_ERR

Service is not supported or syntax error for function parameters.

- AS-CPU type does not support the configured function.
- AS CPU does not contain the most recent firmware status.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

Error D406 - L7_NO_INFO

Desired information not available.

- The requested data are not available.
- In the case of AS300 the message also occur, if two station at the same time access the desired data.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

S7 connection status

Internal function correctly completed.

In this Property Page, the current connection status of the corresponding unit is displayed. You can use the displayed error code to quickly localize a communication error.

Select the corresponding connection and press the <F1> key, or double-click the error code.

Column	Description
Name	Connection name
Status	Connection status: Ready The connection is ready to use. Faulty The connection is faulty. In the Error column is an error code indicated. CPU Stop The AS is in STOP mode. init The connection is currently being initialized or deinitialized.
Fault	Error code in hexadecimal. Note: In each case the error code that occurred first is displayed.
PDU length	The PDU length is displayed here in bytes after the commection has been established.
Number of errors	Number of connection problems that occurred. Note: A connection problem is only counted if an available connection is disturbed , i.e. switches from 'available' to 'disturbed' or 'CPU Stop'.
Asyn. Events	Number of asynchronous S7DOS events.

Note

The table can contain additional columns that are not described here. These data are intended for additional error diagnostics and not relevant for the user.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error codes for connection faults (SIMATIC S7-300/400) (Page 527)

2.8.9.3 Internal error codes and constants

Internal error codes and constants

The following tables contain the most important error codes and constants. The information is intended for "insiders". Therefore, we have not gone into more detail on the meanings of the codes and constants.

- iNA960 messages
- SCI messages
- S7DOS Function types
- S7DOS Error codes
- S7DOS Trace function

See also

Communication with SIMATIC S7 300/400 (Page 401)

iNA960 messages

Table 2-4

General iNA960 messages:			
OK_RESP	1	0x01	Request executed with no errors
OK_EOM_RESP	3	0x03	Data block received with no errors
OK_DECIDE_REQ_RESP	5	0x05	Request executed with no errors
OK_CLOSED_RESP	7	0x07	Connection terminated by local user

iNA960 Error messages:			
INVALID_REQ	2	0x02	Incorrect request block
NO_RESOURCES	4	0x04	No resources free in CP
UNKNOWN_REFERENCE	6	0x06	Incorrect OPEN reference defined
BUFFER_TOO_SHORT	8	0x08	User buffer too short
BUFFER_TOO_LONG	10	0x0A	User buffer too long
ILLEGAL_REQ	12	0x0C	Incorrect "negot_options" defined
REM_ABORT	14	0x0E	Connection terminated by remote station
LOC_TIMEOUT	16	0x10	Timeout
UNKNOWN_CONN_CLASS	18	0x12	Unknown connection class

iNA960 Error messages:			
DUP_REQ	20	0x14	Connection already established
CONN_REJECT	22	0x16	Connection request rejected by remote
NEGOT_FAILED	24	0x18	Connection termination incorrect negot option
ILLEGAL_ADDRESS	26	0x1A	Incorrect transport address
NETWORK_ERROR	28	0x1C	Bus or CP faulted
PROTOCOL_ERR	30	0x1E	Protocol error
ILLEGAL_RB_LENGTH	32	0x20	Incorrect request block length

See also

Communication with SIMATIC S7 300/400 (Page 401)
Internal error codes and constants (Page 543)

SCI messages

See description in the "SINEC Communication Interface SCI" manual (A/5-15).

SCI messages			
SCP_OK	0	0x00	No error
SCP_INCONS	201	0xC9	Minor device number is not 00
SCP_RESOURCE	202	0xCA	DPRAM memory request invalid
SCP_CONFIG	203	0xCB	Configuration error (NUM_PROCS)
SCP_NOCONFIG	204	0xCC	SCP driver not configured
SCP_PARAM	206	0xCE	Incorrect mode
SCP_DEVOPEN	207	0xCF	Open already performed
SCP_BOARD	208	0xD0	Board not inserted/recognized
SCP_SOFTWARE	209	0xD1	IRQ error or software not found
SCP_MEM	210	0xD2	Low memory in DPRAM
SCP_MODE	211	0xD3	Download process not yet completed
SCP_LOADER	212	0xD4	No response from loader
SCP_SIGNAL	213	0xD5	Process started asynchronously
SCP_NOMESS	215	0xD7	No message arrived for the process
SCP_USERMEM	216	0xD8	Buffer length length_of_buffer too small
SCP_WINDOW	217	0xD9	Too many SEND calls
SCP_TIMEOUT	219	0xDB	Timeout on SCP
SCP_ATTACH	220	0xDC	Reset not executed/channel still active
SCP_ILLEGAL_REQUEST	221	0xDD	Incorrect request
SCP_ERECOVERF	223	0xDF	Buffer not retrieved with scp_receive
SCP_ECLOSED	224	0xE0	All buffers assigned for the connection
EUSERMAX	225	0xE1	
SCP_EINTR	226	0xE2	
SCP_BOARD_OPEN	231	0xE7	

SCI messages			
SCP_NO_WIN_SERV	233	0xE9	
EPROTECT	234	0xEA	License not found

SCI messages		
SCP_DB_FILE_DOES_NOT_EXIST	240	0xF0
SCP_DB_FILE_CLOSE_NOT_OK	241	0xF1
SCP_SEND_NOT_SUCCESSFUL	242	0xF2
SCP_RECEIVE_NOT_SUCCESSFUL	243	0xF3
SCP_NO_DEVICE_AVAILABLE	244	0xF4
SCP_ILLEGAL_SUBSYSTEM	245	0xF5
SCP_ILLEGAL_OPCODE	246	0xF6
SCP_BUFFER_TOO_SHORT	247	0xF7
SCP_BUFFER_1_TOO_SHORT	248	0xF8
SCP_ILLEGAL_PROTOCOL_SEQUENCE	249	0xF9
SCP_ILLEGAL_PDU_ARRIVED	250	0xFA
SCP_REQUEST_ERROR	251	0xFB
SCP_NO_LICENSE	252	0xFC

Additional online DLL messages on the SCP interface			
E_TIMER_INIT	768	0x0300	WIN Settimer call unsuccessful
E_INIT_COM	769	0x0301	
E_NO_HW	784	0x0310	MPI module not found
E_HW_DEFECT	785	0x0311	Problem with the hardware
E_CNF	786	0x0312	Incorrect configuration parameter
E_BAUDRATE	787	0x0313	Incorrect baudrate/incorrect IntVector
E_HSA	788	0x0314	Incorrect HSA configured
E_TS	789	0x0315	Configured address already assigned
E_OCC	790	0x0316	HW_Device already assigned
E_INT_NOT_PROV	791	0x0317	Interrupt not available
E_INT_BUSY	792	0x0318	Interrupt allocated
E_SAP	793	0x0319	SAP deactivate: SAP not allocated
E_UNPLUGGED	794	0x031a	No remote station found
E_SYNI	795	0x031b	Syni error occurred
E_AMPRO	796	0x031c	AMPRO 2 reported a system error
E_BUFFSIZE	797	0x031d	No buffer of this size created
E_NO_FILE	800	0x0320	DLL/VxD File not found or entries in registry damaged
E_NO_ENTRY	801	0x0321	Address does not exist in DLL
E_VERSION	816	0x0330	Version conflict between SMC driver and SMC firmware
E_COMCNF	817	0x0331	Problem with the COM port configuration
E_NO_SMC	818	0x0332	SMC no longer responds
E_COMMBADID	819	0x0333	COM port is not configured

Additional online DLL messages on the SCP interface			
E_COMMOPEN	820	0x0334	COM port is not available
E_SMCBUSY	821	0x0335	Serial driver is currently in use with another configuration
E_SMCMODEM	822	0x0336	No connection currently exists to a PC/MPI cable
E_SMCNOLEG	823	0x0337	PC/MPI cable refuses job, required authorization is missing
E_ONLINE	896	0x0380	Internal error at the IOCTL interface
E_LOGDEV	897	0x0381	Logical device not in registry
E_L2DRIVER	898	0x0382	L2DRIVER entry is missing in the registry
E_L4DRIVER	900	0x0384	L4DRIVER entry is missing in the registry
E_SYSERROR	1023	0x03FF	System error

See also

Communication with SIMATIC S7 300/400 (Page 401)

Internal error codes and constants (Page 543)

S7DOS Function types

Table 2-5

S7DOS Function types	
S7O_S7_EVENT	1
S7O_S7AG_BESY_UPDATE	2
S7O_S7AG_BUB_CYCL_READ_CREATE	3
S7O_S7AG_BUB_CYCL_READ_START	4
S7O_S7AG_BUB_CYCL_READ_STOP	5
S7O_S7AG_BUB_CYCL_READ_DELETE	6
S7O_S7AG_BUB_READ_VAR	7
S7O_S7AG_BUB_WRITE_VAR	8
S7O_S7AG_COMPRESS	9
S7O_S7AG_LINK_IN	10
S7O_S7AG_MEM_MODE	11
S7O_S7AG_MSG_MODE	12
S7O_S7AG_PASSWORD	13
S7O_S7AG_READ_SZL	14
S7O_S7AG_READ_TIME	15
S7O_S7AG_RESUME	16
S7O_S7AG_START	17
S7O_S7AG_STOP	18
S7O_S7AG_TEST	19
S7O_S7AG_TEST_DELETE	20
S7O_S7AG_WRITE_TIME	21
S7O_S7BLK_DELETE	22

S7DOS Function types	
S7O_S7BLK_FINDFIRST	23
S7O_S7BLK_FINDNEXT	24
S7O_S7BLK_READ	25
S7O_S7BLK_WRITE	26
S7O_S7DB_CREATE	27
S7O_S7DB_OPEN	28
S7O_S7DB_COPY	29
S7O_S7DB_CLOSE	30
S7O_S7DB_DELETE	31
S7O_S7DOS_RELEASE	32
S7O_S7NET_GET_LIFE_LIST	33
S7O_S7NET_GET_DIRECT_PLC	34
S7O_S7DP_SET_SLAVE_ADDRESS	35
S7O_S7DP_SLAVE_DIAGNOSE	36
S7O_S7AG_PMC_MSG_MODE	37
S7O_S7AG_PMC_ON_OFF	38
S7O_S7AG_BRCV	39
S7O_S7AG_BSND	40
S7O_S7AG_PMC_ACK	41
S7O_S7AG_PMC_MLDG	42
S7O_S7AG_BUB_CYCL_READ_CREATE_CNF	43
S7O_S7AG_MSG_MODE_CNF	44
S7O_S7AG_PMC_UPDATE	45
S7O_S7L7_DOWNLOAD_DOMAIN	128
S7O_S7L7_UPLOAD_DOMAIN	129

See also

Communication with SIMATIC S7 300/400 (Page 401)
Internal error codes and constants (Page 543)

S7DOS Error codes

Table 2-6

Define	Dec	Hex	Error Message (Stand 7.1.99)
INVALID_BLOCK_TYPE_NUM	272	0110	Invalid block type or invalid block number.
INVALID_PARAM	274	0112	Invalid parameter.
INVALID_BLOCK_TYPE	275	0113	Invalid block type.
BLOCK_NOT_FOUND	276	0114	Block not found.
BLOCK_ALREADY_EXIST	277	0115	The block is already available.
BLOCK_IS_PROTECTED	278	0116	The block is read-only.
BLOCK_TOO_LARGE	279	0117	The block is too large.

Define	Dec	Hex	Error Message (Stand 7.1.99)
INVALID_BLOCK_NUM	280	0118	Invalid block number.
INCORRECT_PASSWORD	281	0119	You have entered an incorrect password.
PG_RESOURCE_ERROR	282	011A	PG-Resources error.
PLC_RESOURCE_ERROR	283	011B	AS resource error.
PROTOCOL_ERROR	284	011C	Protocol error.
TOO_MANY_CPU_BLKS	285	011D	Too many blocks (module specific limitation).
DATABASE_DISCONNECTED	286	011E	No connection to the database or S7DOS handle is invalid.
USER_BUFFER_TOO_SHORT	287	011F	The application buffer is too small.
END_OF_BLOCK_LIST	288	0120	End of block list.
CALLOC_ERROR	320	0140	Insufficient memory available.
REQ_INI_ERR	321	0141	The job cannot be processed because of missing resources.
SIMULATION_ERR	368	0170	The simulator could not be found.
DRIVER_NOT_INST	384	0180	The driver is not installed: Incompatible parameter or invalid driver handle.
DRIVER_ALREADY_OPEN	385	0181	The driver is already open or too many channels are open.
VERSION_MISMATCH	448	01C0	The versions are incompatible.
TABLE_STRUCT_MISMATCH	449	01C1	The field setup to the opened database does not correspond to the expected setup.
NO_VALID_SELECTION	450	01C2	s7blk_findnext() was called before s7blk_findfirst().
HEADER_MEMO_INCONSISTENCY	451	01C3	The length format in the block header does not correspond to the actual length of the section in the data storage system.
ID_FILE_ERROR	452	01C4	A problem occurred during processing of Last-ID file.
WRONG_BLOCK_FORMAT	453	01C5	Incorrect block format.
FILE_NOT_FOUND	454	01C6	File not found.
INVALID_BESY_KOMP	455	01C7	Invalid Besy Update Component.
DBCPY_TARGET_EXISTS	456	01C8	The database specified as target is already available.
WR_TABLE_ALREADY_LOCKED	457	01C9	The database is already locked by another application.
SKIP_UNEXPECTED_ERROR	458	01CA	An unexpected error occurred during positioning in the database.
DHCLOSE_ERROR	459	01CB	An invalid DATA4 pointer was transferred to the dhclose() call.
LIB_NOT_FOUND	460	01CC	The TBI-DLL could not be loaded.
FKT_NOT_FOUND	461	01CD	A function in the dynamic downloaded DLL could not be found.
STPCPY_NOT_FOUND	462	01CE	The StopCopy-DLL could not be loaded.
FILE_ACCESS_ERROR	463	01CF	Access protection violation.
PASSWORD_NOT_FOUND	464	01D0	The password was not found.
DIAGNOSE_NOT_STARTED	480	01E0	The diagnosis is disabled.

Define	Dec	Hex	Error Message (Stand 7.1.99)
DIAGNOSE_DATA_NOT_AVAILABLE	481	01E1	No diagnostic data are available.
DIAGNOSE_DATA_INCONSISTENT	482	01E2	The diagnostic data are inconsistent.
NOT_IMPLEMENTED	496	01F0	Function is not implemented.
INTERN_ERR	511	01FF	System error.
L7_UNKNOWN_ROSCTR	2049	0801	Internal error code, is being mapped
L7_UNKNOWN_CN_ID	2050	0802	Internal error code, is being mapped
L7_UNKNOWN_DIENSTKENN	2051	0803	Internal error code, is being mapped
L7_TOO_MUCH_BLOCKS	2052	0804	Internal error code, is being mapped
L7_WRONG_FLAGS	2053	0805	S7 protocol: Invalid flags.
L7_INTERNAL_ERR	2054	0806	Internal error code, is being mapped
L7_UNKNOWN_ID1_ERR	2055	0807	Internal error code, is being mapped
L7_SEND_ERR	2064	0810	S7 protocol: Data could not be correctly sent.
L7_RECEIVE_ERR	2065	0811	S7 protocol: No job for the received data could be found.
L7_DIAG_ERR_UPDATE	3456	0D80	A diagnostic error has occurred.
EPR_VCC_ERROR	8208	2010	Internal error code, is being mapped
EPR_VPP_ERROR	8209	2011	Internal error code, is being mapped
EPR_NO_ADAPT	8210	2012	EPROM: The memory card adapter is missing.
EPR_REM_EXT_ADAPT	8211	2013	EPROM: The external memory card adapter has to be removed.
EPR_NO_MEM_CARD	8212	2014	EPROM: The memory card is missing.
EPR_CHECK_SUM_ERR	8213	2015	Internal error code, is being mapped
EPR_LEN_ERR	8214	2016	Internal error code, is being mapped
EPR_MOD_ADR_ERR	8215	2017	Internal error code, is being mapped
EPR_READ_ONLY	8216	2018	Internal error code, is being mapped
EPR_NOT_IMPLEMENT	8217	2019	Internal error code, is being mapped
EPR_AREA_NOT_PRES	8218	201A	Internal error code, is being mapped
EPR_ALGO_UNKNOWN	8219	201B	EPROM: Unknown configuration algorithm.
EPR_MOD_CHANGE	8220	201C	EPROM: The memory card was changed without authorization.
EPR_NO_OPEN	8221	201D	Internal error code, is being mapped
EPR_ILEG_BREAK	8222	201E	Internal error code, is being mapped
EPR_NO_FLASH	8223	201F	Internal error code, is being mapped
EPR_LEN_SEC_ERR	8224	2020	Internal error code, is being mapped
EPR_ADR_SEC_ERR	8225	2021	Internal error code, is being mapped
EPR_ADR_ERR	8226	2022	Internal error code, is being mapped
EPR_DB_SHRT_ERR	8227	2023	Internal error code, is being mapped
EPR_MODE_ERR	8228	2024	Internal error code, is being mapped
EPR_PROG_ERR	8229	2025	EPROM: Hardware error in the memory card.
EPR_PROG0_ERR	8230	2026	Internal error code, is being mapped
EPR_ERA_ERR	8231	2027	Internal error code, is being mapped
EPR_RD_KENN_ERR	8232	2028	Internal error code, is being mapped
EPR_CMP_ERR	8233	2029	Internal error code, is being mapped

Define	Dec	Hex	Error Message (Stand 7.1.99)
EPR_EMP_ERR	8234	202A	EPROM: The memory card is not empty.
EPR_EE_WR_ERR	8235	202B	Internal error, is being mapped
EPR_EE_TIMOUT_ERR	8236	202C	Internal error, is being mapped
EPR_SYS_ERR	8237	202D	EPROM: Internal error.
EPR_WR_PROT	8238	202E	EPROM: Read-only memory card.
EPR_NO_EXPOWER	8239	202F	EPROM: The external power supply is missing.
EPR_MINUS_12V_ERR	8240	2030	Internal error, is being mapped
EPR_REF_SP_ERR	8241	2031	Internal error, is being mapped
EPR_DEV_NOT_SUPPORTED	8242	2032	EPROM: The EPROM driver cannot support the set configuration interface on this computer.
EPR_PROM_ENTR_ERR	8256	2040	EPROM: External prommer is missing.
EPR_NO_BIOS_CENTADR	8257	2041	Internal error, is being mapped
EPR_QOUT_FACT_ERR	8258	2042	EPROM: Hardware error on the external prommer.
EPR_QUEUE_BUSY	8259	2043	Internal error, is being mapped
EPR_TIMER_ERR	8272	2050	Internal error, is being mapped
EPR_WRONG_WIN_MODE	8288	2060	Internal error, is being mapped
EPR_NO_GLOBAL_MEM	8289	2061	Internal error, is being mapped
EPR_PROGAS_ERR	8324	2084	EPROM: Configuration interface error.
EPR_FATAL_ERR	8328	2088	Internal error, is being mapped
EPR_ERR_KEYNOTFOUND	8452	2104	Internal error, is being mapped
EPR_ERR_RECBUFLEN	8470	2116	Internal error code, is being mapped
EPR_ERR_BLK_LEN	8488	2128	Internal error, is being mapped
EPR_ERR_FILEFULL	8534	2156	Internal error, is being mapped
EPR_DRV_NOT_INST	8560	2170	Internal error, is being mapped
EPR_DRV_ALREADY_OPEN	8561	2171	Internal error, is being mapped
EPR_INVALID_DRV_HANDLE	8562	2172	Internal error, is being mapped
EPR_DRIVER_NOT_OPEN	8563	2173	Internal error, is being mapped
EPR_WRONG_DEVICE_NAME	8564	2174	Internal error, is being mapped
EPR_NO_FLASH_AREA	8566	2176	EPROM: No flash area is available on the memory card.
EPR_WRONG_APPL_KEN	8567	2177	EPROM: The memory card has an incorrect application ID.
EPR_MOD_CONTENS_UNKNOWN	8568	2178	EPROM: The content of the memory card cannot be interpreted.
EPR_MOD_CHKSUM_ERR	8569	2179	A growler error has occurred in the content of the memory card.
EPR_PROGRAMMING_ERR	8592	2190	EPROM: A configuration error has occurred.
SRVERR_NO_CONV_ESTABLISHED	16394	400A	No communication with the S7OTBL server is possible. The server is not started.
SRVERR_SYS_ERROR	16399	400F	System error during the S7OTBL server client communication.
NO_LIST_ENTRY	16415	401F	No corresponding entry is available in the asynchronous list.
LOCAL_ALLOC_ERR	16416	4020	No local memory is available.

Define	Dec	Hex	Error Message (Stand 7.1.99)
LOCAL_LOCK_ERR	16417	4021	The local memory cannot be locked.
S7DOS_SRV_TIMEOUT	16418	4022	Timeout during waiting on WM_ENDE_L7.
WRONG_WIN_MODE	16480	4060	Incorrect operating mode in Windows.
NO_GLOBAL_MEM	16481	4061	No global memory is available.
L4_OK_RESP	16641	4101	Internal error code, is being mapped
L4_INVALID_REQ	16642	4102	Online: Internal error.
L4_OK_EOM_RESP	16643	4103	Online: OK_EOM_RESPONSE.
L4_NO_RESOURCES	16644	4104	Online: No resources in the driver are available.
L4_OK_DECIDE_REQ_RESP	16645	4105	Internal error code, is being mapped
L4_UNKNOWN_REFERENCE	16646	4106	Online: Unknown shortcut.
L4_OK_CLOSED_RESP	16647	4107	Online: The connection was terminated.
L4_BUFFER_TOO_SHORT	16648	4108	Internal error code, is being mapped
L4_NO_SRD_RSP	16649	4109	Online: Sending and receiving of data is not acknowledged.
L4_BUFFER_TOO_LONG	16650	410A	Internal error code, is being mapped
L4_OK_REJECT_CONN_RESP	16651	410B	Internal error code, is being mapped
L4_ILLEGAL_REQ	16652	410C	Internal error code, is being mapped
L4_REM_ABORT	16654	410E	Online: Connection was terminated.
L4_LOC_TIMEOUT	16656	4110	Online: No connection was established. The station does not respond.
L4_UNKNOWN_CONN_CLASS	16658	4112	Internal error code, is being mapped
L4_DUP_REQ	16660	4114	Online: The connection already exists.
L4_CONN_REJECT	16662	4116	Online: No connection was established. The partner refuses to establish the connection.
L4_NEGOT_FAILED	16664	4118	Internal error code, is being mapped
L4_ILLEGAL_ADDRESS	16666	411A	Online: Invalid address.
L4_NETWORK_ERROR	16668	411C	Online: Network error.
L4_PROTOCOL_ERR	16670	411E	Internal error code, is being mapped
L4_ILLEGAL_RB_LENGTH	16672	4120	Internal error code, is being mapped
L4_PPI_T_TI_ERROR	16674	4122	Internal error code, is being mapped
L4_NO_RESOURCE	16897	4201	Online: No resources in the driver are available.
L4_CONNECTION_EXIST	16898	4202	Internal error code, is being mapped
L2_OPEN_DONE	16899	4203	Internal error code, is being mapped
L4_SCI_STATION_NA	16913	4211	Online: No station is available on the subnet.
L4_SCI_STATION_DS	16914	4212	Online: The station is not online.
L4_SCI_STATION_NO	16915	4213	Online: Too many stations.
L4_SCI_NOTIMPLEMENTED	16917	4215	Online: The function is not implemented or is invalid in the current context.
L4_SCI_E_DPX2_ERROR	16918	4216	Online: Incorrect station address of the DP Slave or error message of the DP Slave.
L4_NO_CONFIG_FOUND	16936	4228	The bus parameters could not be retrieved automatically (ONLINE). No stations at the bus that send bus parameter frames. Set your MPI/PROFIBUS interface manually.

Define	Dec	Hex	Error Message (Stand 7.1.99)
L4_ONLY_BAUDRATE_FOUND	16937	4229	Online: Only the baud rates could be retrieved.
L4_ONLY_PROFILE_FOUND	16938	422A	Online: Only the profile could be retrieved.
L4_SCI_E_UNPLUGGED	16944	4230	Online: No other active station is available.
L4_SCI_E_SYNI	16945	4231	Online: The bus is faulted.
L4_DLL_E_HSA	16946	4232	Online: Incorrect maximal number of stations (HIGHEST_ADR).
L4_SCI_E_AMPRO	16947	4233	Online: System error.
L4_DLL_E_TIMER_INIT	17056	42A0	Online: The Windows timer cannot be set.
L4_DLL_E_INIT_COM	17057	42A1	Online: The COM interface cannot be initialized or opened.
L4_DLL_E_NO_HW	17072	42B0	Online: No hardware found.
L4_DLL_E_HW_DEFECT	17073	42B1	Online: The hardware is defective.
L4_DLL_E_CNF	17074	42B2	Online: The driver is configured incorrectly or the registry contains invalid parameters.
L4_DLL_E_BAUDRATE	17075	42B3	Online: An incorrect baud rate or an incorrect interrupt vector is set, or the local MPI address is greater than the maximum station address.
L4_DLL_E_TS	17077	42B5	Online: The set local station address (TS_ADR) is already assigned.
L4_DLL_E_OCC	17078	42B6	Online: The hardware device (DEVICE) cannot be used several times.
L4_DLL_E_INT_NOT_PROV	17079	42B7	Online: The defined interrupt vector (IRQ) is not available on this module.
L4_DLL_E_INT_BUSY	17080	42B8	Online: The set interrupt vector (IRQ) is already allocated.
L4_DLL_E_SAP	17081	42B9	Internal error code, is being mapped
L4_DLL_E_NO_FILE	17088	42C0	Online: The selected communication driver cannot be loaded; the file was not found.
L4_DLL_E_NO_ENTRY	17089	42C1	Online: The function is not realized on the loaded communication driver.
L4_DLL_E_LOGDEV	17090	42C2	Online: The logical device is not defined in the registry.
L4_DLL_E_VERSION	17104	42D0	Online: The output inventory of driver and adapter or PC/MPI cable are incompatible.
L4_DLL_E_COMCNF	17105	42D1	Online: No interrupt of PC/MPI cable was received.
L4_DLL_E_NO_SMC	17106	42D2	Online: The connection to the adapter is disturbed.
L4_DLL_E_COMMBADID	17107	42D3	Online: The COM interface is not configured under Windows.
L4_DLL_E_COMMOPEN	17108	42D4	Online: The COM interface is currently unavailable.
L4_DLL_E_SMCBUSY	17109	42D5	Online: The serial driver is currently used by another application with another configuration.
L4_DLL_E_SMCMODEM	17110	42D6	Online: There is still no remote connection or logical connection to the TS Adapter.

Define	Dec	Hex	Error Message (Stand 7.1.99)
L4_DLL_E_SMCNOLEG	17111	42D7	Online: The TS Adapter refuses the job due to missing required legitimation.
L4_LIB_WIN_SYS_ERR	17120	42E0	Online: Windows system error in communication driver.
L4_LIB_NO_GLOBAL_MEM	17134	42EE	Online: No global memory is available.
L4_LIB_NO_SIN_SERV	17135	42EF	Online: SIN_SERV is not started.
L4_SCI_STATION_NOT_ONLINE	17146	42FA	Online: The station is not online.
L4_SCI_RB_ERR	17147	42FB	Internal error code, is being mapped
L4_SCI_MAX_REQ_NR	17148	42FC	Internal error code, is being mapped
L4_SCI_DRV_ALREADY_OPEN	17149	42FD	Internal error code, is being mapped
L4_SCI_DRV_ERR	17150	42FE	Internal error code, is being mapped
L4_SCI_DRV_NOT_INST	17151	42FF	Internal error code, is being mapped
NO_LOC_SUBNET_ENTRY	17153	4301	Invalid local subnet number in the file S7DPMPI.INI.
NO_TABLE_ENTRY	17154	4302	The station on this subnet is unavailable.
WRONG_RACK_SLOT	17155	4303	Incorrect rack/slot in the module table.
WRONG_NODENAME	17156	4304	Incorrect format of the node name in S7db_open (online).
INVALID_S7_TABUF	17157	4305	Invalid S7-Transport_Address_Buffer.
INVALID_S7_WUSERID	17158	4306	The job cannot be found. Incorrect wUserID or incorrect Window handle.
L7_DPT_ERROR	17456	4430	The data record cannot be read/written.
RES_VON	28672	7000	The area from 0x7000 to 0x7fff is reserved for applications.
RES_BIS	32767	7FFF	The area from 0x7000 to 0x7fff is reserved for applications.
L7_INVALID_CPU_STATE	32769	8001	The function is invalid in this operating mode, or an incorrect minimum scan cycle time is set.
L7_DOMAIN_LOADING_ERR	32771	8003	S7 protocol error: A domain transfer error has occurred. The domain content (e.g. block) is incorrect.
L7_S5_INTERN_ERR	33023	80FF	S7 protocol: Internal error.
L7_COMMON_ERR	33024	8100	Application, general error: Unknown service for the remote module.
L7_MISSING_CONTEXT	33028	8104	The service is not supported. An unknown error in PDU adapter or service has occurred.
L7_DEF_OBJ_INCONSISTENT	33284	8204	The object's type format is inconsistent.
L7_ALREADY_COPIED	33285	8205	A copied object's variant is available.
L7_NO_RAM	33537	8301	The memory space on the module is not enough or the specified memory medium is not available.
L7_NO_RESOURCE	33538	8302	A resource bottleneck is available or the processor resources are unavailable.
L7_ERRCOD_INI_NO_RES	33540	8304	Domain: An additional simultaneous uploading is no longer possible. A resources bottleneck is available.

Define	Dec	Hex	Error Message (Stand 7.1.99)
L7_FUNC_NOT_AVAIL	33541	8305	The functionality is not available. A resources bottleneck is available.
L7_INVALID_SEQUENCE	33793	8401	S7 protocol error: Incorrect sequence of services (e.g. during loading or uploading of a block).
L7_WRONG_PI_STATE	33794	8402	Error during the service sequence. The service cannot be performed due to the mode of the addressed object.
L7_FATAL_ERR	33796	8404	S7 protocol: A significant error has occurred. The function cannot be executed.
L7_CPU_IN_PROTECTED_STATE	33797	8405	The remote block is in status DISABLE (PBK). The function cannot be executed.
L7_PDU_SIZE_ERR	34048	8500	S5 protocol error: Wrong PDU size.
L7_SERVICE_CANCELED	34051	8503	The service has been prematurely canceled.
L7_NO_OBJ_ACCESS	34561	8701	The object access is not supported.
L7_INVALID_ADDRESS	34562	8702	S7 protocol error: Access to a remote object was rejected.
L7_OBJECT_ACCESS_DENIED	34563	8703	Access error: Access to a remote object was rejected.
L7_ACCESS_ERR_OBJ	34564	8704	Access error: The object is damaged.
L7_INVALID_REQ_NB	53249	D001	D001: Protocol error: Job number is invalid.
L7_INVALID_REQ_VER	53250	D002	D002: Parameter error: Job variant is invalid.
L7_INVALID_FKT	53251	D003	D003: Parameter error: The function is invalid.
L7_INVALID_REQ_STAT	53252	D004	D004: Parameter error: Job status is invalid.
L7_INVALID_END_OF_REQ	53253	D005	D005: Parameter error: Job termination is invalid.
L7_INVALID_ABORT_OF_CONN	53254	D006	D006: Parameter error: The ID for the connection termination is invalid.
L7_INVALID_NB_OF_BUF	53255	D007	D007: Parameter error: The number of buffer elements is invalid.
L7_INVALID_GEAR_DOWN	53256	D008	D008: Parameter error: The reduction factor is invalid.
L7_INVALID_NB_OF_EXEC	53257	D009	D009: Parameter error: The execution number is invalid.
L7_INVALID_TRIG_EVENT	53258	D00A	D00A: Parameter error: The trigger event is invalid.
L7_INVALID_TRIG_COND	53259	D00B	D00B: Parameter error: The trigger requirement is invalid.
L7_TRIG_EVENT_ERR_NO_BLK	53265	D011	D011: Parameter error: The block is not available.
L7_TRIG_EVENT_WRONG_OFF	53266	D012	D012: Parameter error: Incorrect address in the block.
L7_TRIG_EVENT_ERR_BLKTYP	53267	D013	D013: Parameter error: The block type is invalid.
L7_TRIG_EVENT_BLK_IS_MOD	53268	D014	D014: Parameter error: The block is being deleted/overloaded.
L7_TRIG_EVENT_ERR_VARADDR	53269	D015	D015: Parameter error: The tag address is invalid.

Define	Dec	Hex	Error Message (Stand 7.1.99)
L7_TRIG_EVENT_ERR_FOUND_BL	53270	D016	D016: Parameter error: No block can be activated.
L7_TRIG_EVENT_ERR_SYS_TRIGGER	53271	D017	D017: Parameter error: The SYS trigger number is invalid.
L7_TRIG_COND_PATH_ERR	53285	D025	D025: Parameter error: The path is invalid.
L7_TRIG_COND_ACC_ERR	53286	D026	D026: Parameter error: The access type is invalid.
L7_TRIG_COND_NB_OF_DB_ERR	53287	D027	D027: Parameter error: The number of DBs is invalid.
L7_SEGMENT_NEXT_ERR	53297	D031	D031: Protocol error: CN-ID, S-ID or ID1 in the following segment do not match the first segment.
L7_SEGMENT_LEN_ERR	53298	D032	D032: Parameter error: Incorrect length of the buffer result.
L7_REQUEST_LEN_ERR	53299	D033	D033: Protocol error: Job length is invalid.
L7_PARAM_CODING_ERR	53311	D03F	D03F: Coding error: Other error in the parameter part (e.g. reserve bytes do not equal NULL).
L7_STALI_ID_ERR	53313	D041	D041: Data error: The stali ID is invalid.
L7_VAR_ADDR_ERR	53314	D042	D042: Data error: The tag address is invalid.
L7_REQ_DOES_NOT_EXIST	53315	D043	D043: Data error: The referenced job is not available.
L7_INVALID_VAR_VALUE	53316	D044	D044: Data error: The tag value is invalid.
L7_BASP_ERR	53317	D045	D045: Data error: Quitting the BASP control (ODIS) during HOLD is invalid.
L7_MEASUREMENT_ERR	53318	D046	D046: Data error: The measuring level during runtime measuring is invalid.
L7_HIERARCHY_ERR	53319	D047	D047: Data error: The hierarchy for 'Read job list' is invalid.
L7_INVALID_DEL_ID	53320	D048	D048: Data error: The delete ID for 'Delete job' is invalid.
L7_INVALID_SUB_ID	53321	D049	D049: The replace ID for 'Replace job' is invalid.
L7_INVALID_ENTRANCE_DATA	53322	D04A	D04A: Error during execution of 'programstatus'.
L7_DATA_CODING_ERR	53343	D05F	D05F: Coding error: Other error in the data part (e.g. reserve bytes do not equal NULL, ...).
L7_REQ_MEM_ERR	53345	D061	D061: Resource error: No memory for the job is available.
L7_FULL_REQ_LIST	53346	D062	D062: Resource error: The job list is full.
L7_TRIG_EVENT_IS_USED	53347	D063	D063: Resource error: The trigger event is allocated.
L7_TOO_SMALL_BUF	53348	D064	D064: Resource error: The memory space for an element of the result buffer is too small.
L7_TOO_MANY_BUF	53349	D065	D065: Resource error: The memory space for several elements of the result buffer is too small.
L7_NO_TIMER_AVAIL	53350	D066	D066: Resource error: Another job allocates the timer for the runtime measurement.

Define	Dec	Hex	Error Message (Stand 7.1.99)
L7_PBUS_ALREADY_USED	53351	D067	D067: Resource error: Another job allocates the P bus during 'Control selection'.
L7_INVALID_FKT_IN_STATE	53377	D081	D081: Operation mode error: The function in the current operation mode is invalid.
L7_OPERATION_STATE_ERR	53378	D082	Operation mode error: Operating mode HOLD cannot be quit.
L7_PROT_LEVEL_ERR	53409	D0A1	D0A1: Protection and coding: The function in the current protection level is not allowed.
L7_MOD_FKT_ERR	53410	D0A2	D0A2: Protection and coding: The memory-modifying OVS function is running.
L7_CTRL_IS_INST	53411	D0A3	D0A3 Protection and coding: "Control selection" is already set up.
L7_FORCE_IS_INST	53412	D0A4	D0A4: Protection and coding: "Forcing" is already set up.
L7_NO_REF_TO_REF	53413	D0A5	D0A5: Protection and coding: The current job references another job in the communication partner which is not available.
L7_TE_AT_ONCE_REQ	53414	D0A6	D0A6: Protection and coding: The job cannot be locked/unlocked.
L7_REQ_NO_DELETE	53415	D0A7	D0A7: Protection and coding: The job cannot be deleted because e.g. it is currently being read.
L7_REQ_NO_CHANGE	53416	D0A8	D0A8: Protection and coding: The job cannot be replaced because e.g. it is currently being read or deleted.
L7_REQ_NO_READ	53417	D0A9	D0A9: Protection and coding: The job cannot be read because e.g. it is currently being deleted.
L7_TIME_LIMIT	53418	D0AA	D0AA: Timeout in process mode is exceeded.
L7_INVALID_INSTRUCT_PARAM	53419	D0AB	D0AB: The job parameter are invalid in process moe.
L7_INVALID_INSTRUCT_DATA	53420	D0AC	D0AC: The job data are invalid in process moe.
L7_OPERATIONS_MODE_REACHED	53421	D0AD	D0AD: Operating mode is already set.
L7_TRIG_EVENT_ON_OTHER_CONN	53422	D0AE	D0AE: The breakpoint is set using another connection. The job cannot be manipulated.
L7_VAR_ACC_ERR	53441	D0C1	D0C1: Processing warning: During accessing tags at least one error was detected.
L7_BZUE_TO_STOP	53442	D0C2	D0C2: Processing warning: Operating mode transition in STOP/HOLD during function call.
L7_BZUE_AFTER_VAR_ACC	53443	D0C3	D0C3: Processing warning: Operating mode transition and access error: Access errors occurred during loading of the tag(s).
L7_FKT_TIMEOUT	53444	D0C4	D0C4: Processing warning: Timer overflow during the runtime measuring.
L7_MOD_BLK_ERR	53445	D0C5	D0C5: Processing warning: The blocks have been deleted/uploaded since transition STOP.

Define	Dec	Hex	Error Message (Stand 7.1.99)
L7_REF_REQ_WAS_DEL	53446	D0C6	D0C6: Processing error: The referenced job was deleted, because all jobs, that referenced it, were deleted.
L7_REQ_WAS_DEL_AFTER_BZUE	53447	D0C7	D0C7: Processing error: The job was deleted due to quitting of the operating mode.
L7_EMPTY_STALI_ID	53448	D0C8	D0C8: Processing error: The status block was interrupted, because an empty stali ID was detected during processing.
L7_VERL_STAT_RES_HGOB	53449	D0C9	D0C9: Processing warning: Quitting the status area via reset of the background OB.
L7_VERL_STAT_RES_HGOB_ZUGERR	53450	D0CA	D0CA: Processing warning: Quitting the status area via reset of the background OB and access error during reading of the tags before quitting.
L7_OUTPUT_LOCK_PA_ON	53451	D0CB	D0CB: The output lock of the I/O outputs is again enabled.
L7_SHORT_RESULT	53452	D0CC	D0CC: The data range for the test functions is limited by the timeout.
L7_INVALID_BL_NAME	53761	D201	D201: Syntax error in block name.
L7_INVALID_ARGUMENT	53762	D202	D202: Syntax error in the function parameters.
L7_INVALID_BL_TYPE	53763	D203	Internal error code, is being mapped
L7_NO_LINKED_BLK	53764	D204	Internal error code, is being mapped
L7_BL_ALREADY_INSERTED	53765	D205	D205: A linked block is already available in the RAM; no conditional copying is possible.
L7_INVALID_BL_NUMBER	53766	D206	D206: A linked block is already available in the EPROM; no conditional copying is possible.
L7_BL_IN_ROM	53767	D207	At least one block is already available in the EPROM.
L7_TOO_MANY_COPIED_BL	53768	D208	The maximum number of copied (unlinked) blocks on the module has been exceeded.
L7_MISSING_BL	53769	D209	D209: At least one unspecified block on the module is unavailable.
L7_TOO_MANY_BL_INS	53770	D20A	D20A: The maximum number of modules that could be linked with one job was exceeded.
L7_TOO_MANY_BL_DEL	53771	D20B	D20B: The maximum number of modules that could be deleted with one job was exceeded.
L7_NO_AE	53772	D20C	D20C: The OB cannot be copied, as the corresponding runtime level is not available.
L7_INVALID_YB	53773	D20D	The SDB cannot be evaluated (e.g. unknown number).
L7_NO_MORE_BL	53774	D20E	D20E: No block available.
L7_INVALID_BL_LEN	53775	D20F	D20F: The module-specific maximum block size was exceeded.
L7_BLK_NB_ERR	53776	D210	D210: The block number is invalid.
L7_RUNTIME_REL_ATTRIB_ERR	53778	D212	D212: Incorrect header attribute (runtime relevant).
L7_TOO_MANY_SDB	53779	D213	D213: Too many SDBs.
L7_INVALID_BL	53780	D214	D214: Incorrect context in the block.

Define	Dec	Hex	Error Message (Stand 7.1.99)
L7_INVALID_USER_PROG	53781	D215	D215: Incorrect user program.
L7_USER_PROG_ERR	53782	D216	D216: Invalid user program- clear/reset module.
L7_SDB0_PROTECTION_ERR	53783	D217	D217: The specified protection level in the SDB0 is invalid.
L7_ACT_PAS_ATTRIB_ERR	53784	D218	D218: Incorrect attribute (active/passive).
L7_BL_LEN_ERR	53785	D219	D219: Incorrect block lengths (e.g. incorrect length of the first section or the whole block).
L7_LOCAL_DATA_LEN_ERR	53786	D21A	D21A: The length of the local data (e.g. odd. for OB<20, local data length/ data stack too large) or read-only ID is incorrect.
L7_COMPRESS_ERR	53787	D21B	D21B: The module cannot compress, or the compress process was prematurely interrupted.
L7_WRONG_CPU_LIMITS	53789	D21D	D21D: The transferred dynamic qualified project specifications are invalid. They do not match the removal of CPU or the current user program. Check your settings and transfer them again.
L7_SDB_LINK_IN_ERR	53790	D21E	D21E: Error during configuration of external modules inside linking of a SDB.
L7_INVALID_LANGUAGE	53792	D220	D220: The language setting is invalid.
L7_MPI_PARAM_ERR	53793	D221	D221: Error in the SDB for the connection management (incorrect MPI parameter in the SDB0 or error in the connection description (SDBs)).
L7_IK_PARAM_ERR	53794	D222	D222: Error in the SDB with GD configuration (incorrect parameter in the GD-SDB).
L7_PBK_PARAM_ERR	53795	D223	D223: Error in instance DB for PBK, or the maximum number of instance DBs was exceeded.
L7_PMC_ERR	53796	D224	There is an error in the setup of the SCAN-SDB.
L7_DP_ERR	53797	D225	There is an error in the setup of the DP-SDB.
L7_BLK_STRUCT_WRONG	53798	D226	D226: A structure error occurred in a module.
L7_BL_TOO_LONG	53808	D230	D230: Incorrect resource specification. The block is longer than specifed, or the information provided at start of loading does not match.
L7_OB_WITHOUT_AE	53809	D231	D231: At least one loaded the OB cannot be copied, because the corresponding runtime level is not available.
L7_LOAD_PRG_BLK_NB_ERR	53810	D232	D232: At least one block number of a loaded module is invalid.
L7_EPR_BLK_ALREADY_EXIST	53811	D233	At least one loaded module is already available in EPROM.
L7_BLK_LOADING_2X	53812	D234	D234: The block is available twice in the specified memory medium or in the job.
L7_BLK_WRONG_CRC	53813	D235	D235: The block contains a checksum error.
L7_BLK_NO_CRC	53814	D236	D236: The block contains no checksum error.
L7_COORD_RULE	53824	D240	D240: The coding regulations were violated.

Define	Dec	Hex	Error Message (Stand 7.1.99)
L7_FEW_PROT_LEVEL	53825	D241	D241: The protection level of the function is too small.
L7_SECURITY_LEVEL	53826	D242	AS protection error.
L7_OSUPDATE_WRONG_VER	53840	D250	D250: The update and module ID or the output release do not match.
L7_OSUPDATE_ORDER_ERR	53841	D251	D251: Incorrect sequence in the operation system components.
L7_OSUPDATE_CHECKSUM_ERR	53842	D252	D252: Checksum error.
L7_OSUPDATE_NO_LOADER	53843	D253	D253: There is no runtime download available, an update is only via the memory card possible.
L7_OSUPDATE_MEM_ERR	53844	D254	D254: Memory error in operating system.
L7_COMPILE_ERR	53888	D280	D280: Compiling error in AS 300.
L7_KOOR1_TRIGGER_ACTIVE	53921	D2A1	D2A1: An additional block function or a trigger on a block is active. Close the other online function.
L7_KOOR2_TRIGGER_ACTIVE	53922	D2A2	D2A2: A trigger is active on a block. Terminate the test function.
L7_KOOR3_TRIGGER_NOT_ACTIVE	53923	D2A3	D2A3: The block is not active (sequenced) or the block has to be deleted. Repeat the function later.
L7_KOOR4_BLOCK_IN_WORK	53924	D2A4	D2A4: The block is currently processed by another block function. Repeat the function later.
L7_KOOR6_PRG_SAVE	53926	D2A6	D2A6: Saving and changing the user program are simultaneously impossible. Repeat the function later.
L7_KOOR7_BLOCK_NOT_RUNNING	53927	D2A7	D2A7: The block has the attribute ´Unlinked´ or is not being processed. A test function on this block is impossible.
L7_KOOR8_TST_RUNNING	53928	D2A8	D2A8: A running test function prevents the CPU configuration. Terminate the test function.
L7_KOOR9_CPU_PARAM	53929	D2A9	D2A9: The CPU is currently being re-configured. "Load user program" is simultaneously impossible. Repeat the function later.
L7_KOOR4_BGR_PARAM	53930	D2AA	D2AA: Modules are being currently re-configured. Repeat the function later.
L7_H_CPU_REORG_MEM	53931	D2AB	D2AB: Modifying the dynamic qualified project specifications is active. The user program is being re-evaluated. Wait until the end of the reevaluation and repeat your job.
L7_INVALID_SZL_ID	54273	D401	D401: Invalid SZL-ID.
L7_INVALID_INDEX	54274	D402	D402: Invalid INDEX.
L7_DGS_CONN_ALREADY_ANNOU	54275	D403	D403: The service is already signed in/signed out (diagnose/PMC).
L7_MAX_USER_NB	54276	D404	D404: The maximum station number was reached.
L7_DGS_FKT_PAR_SYNTAX_ERR	54277	D405	D405: Service is not supported or syntax error for the function parameters.

Define	Dec	Hex	Error Message (Stand 7.1.99)
L7_NO_INFO	54278	D406	D406: The desired information is not available.
L7_DIAGNOSE_ERR	54279	D407	A diagnostics error has occurred.
L7_DIAG_ERR_UPDATE	54280	D408	A diagnostics error has occurred.
L7_DIAG_ERR_DPBUS	54281	D409	A diagnostic error has occurred on the DB bus.
L7_PRT_FKT_PAR_SYNTAX_ERR	54785	D601	D601: Syntax error in function parameter.
L7_PASSWORD_ERR	54786	D602	D602: You have entered an incorrect password.
L7_PRT_CONN_ALREADY_ANNOU	54787	D603	D603: The connection is already allowed.
L7_PRT_CONN_ALREADY_FREE	54788	D604	D604: The connection is already approved.
L7_NO_PASSWORD	54789	D605	D605: Not allowed due to unavailable passord.
L7_INVALID_VAR_ADDR	55297	D801	D801: At least one tag address is invalid..
L7_UNKNOWN_REQ	55298	D802	D802: The specified job is not available.
L7_INVALID_REQ_STATUS	55299	D803	D803: Invalid job status.
L7_INVALID_CYCLIC_TIME	55300	D804	D804: Invalid cycle time (time basis or multiple is invalid).
L7_NO_MORE_CYCLIC_REQ	55301	D805	D805: An additional cyclic reading job cann not be set up.
L7_INVALID_CYCLIC_REQ_STATE	55302	D806	D806: The referenced job is in a status, in which the requested function cannot be executed.
L7_CYCLIC_TIME_TOO_SHORT	55303	D807	D807: Interruption of the function due to overload, i.e., the processing of the reading cycle takes longer than the set cycle time.
L7_WRONG_TSY_FORMAT	56321	DC01	DC01: Error in the specification of date and time.
L7_H_CPU_IS_MASTER	57857	E201	E201: CPU is already master:
L7_H_CPU_AUA_MODULE	57858	E202	E202: Link-up and update impossible due to different user program in flash module.
L7_H_CPU_AUA_FW	57859	E203	E203: Link-up and update impossible due to different firmware.
L7_H_CPU_AUA_MEM	57860	E204	E204: Link-up and update impossible due to different memory expansion.
L7_H_CPU_AUA_SYNC	57861	E205	E205: Abort link-up/update due to synchronization error.
L7_H_CPU_AUA_KOORD	57862	E206	E206: Reject link-up/update due to coordination violation.
L7_INVALID_ID2	61185	EF01	S7 protocol error: Error in ID2: In the job only 00H is valid.
L7_MISSING_CAPABILITY	61186	EF02	S7 protocol error: Error in ID2: The production facility record is not available.
DEMO_RETURN_CODE	65281	FF01	The function in the demo version is invalid.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Internal error codes and constants (Page 543)

S7DOS Trace function

The S7DOS trace can be controlled via the following registry entry:

HKEY_LOCAL_MACHINE/SOFTWARE/SIEMENS/SINEC/TrcParams
"Format"="long"
"Output"="file"
"Level"="0x71000000"

Format

The "Format" parameter can be set to "long" or "short". This affects the line layout of a trace line. A time stamp is input for "long".

Output

The "Output" parameter determines where the output is saved. If output is set to "file", the outputs are written to C:\tmp\S7FILE1.TRC or C:\tmp\S7FILE2.TRC. These files are limited to approx. 2 MB and are written according to the alternating buffer principle.

Level

The parameter level determines what kind of outputs are saved in the tracefile. This parameter is a bit-by-bit "OR" logic operation of the following constants:

Level		
S7TRCAI	0x10000000UL	Application Interface
S7TRCL7	0x01000000UL	Lower Interface (Layer 7 PDUs)
S7TRCL4	0x40000000UL	Lower Interface (Layer 4)
S7TRCL2	0x20000000UL	Lower Interface (Layer 2)

If all possible trace outputs have to be activated, the level has to be set to '0x7FFFFFFF'. The information is evaluated on activation of S7DOS. At this time the folder C:\tmp has to be available. The folder is not automatically created by S7DOS.

See also

Communication with SIMATIC S7 300/400 (Page 401)

Internal error codes and constants (Page 543)

2.8.9.4 API error texts

API error messages

The most important error messages of S7 channel are listed in this section. If an error with an error code that is not in the table occurs, please call the WinCC hotline.

- Error during initialization of the S7 communication driver for 'unitname' unit 'devicename' device.
- Error during loading the S7 communication driver.

- For the configured functions is a S7DOS version xx or higher necessary.
- Error xx occurred in 'functionname' function

See also

Communication with SIMATIC S7 300/400 (Page 401)

Error during initialization of the S7 communication driver for 'unitname' unit 'devicename' device.

An error occurred while the communication drivers were being initialized.

- The configured device is not available.
- An error occurred during the device driver is activated.
- Copy protection error occurred.
- The bus connector to the MPI card is not inserted.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error during loading the S7 communication driver

The communication subsystem SAPI-S7 or S7DOS could not be loaded.

- Communication driver not installed correctly.
- The DLLs required by the communication driver could not be loaded correctly.
- Path setting incorrect or required communication drivers deleted or moved.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

For the configured functions is a S7DOS version xx or higher necessary.

Functions were configured (for example, PMC message processing) that are not supported by the installed communication subsystem.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error xx occurred in 'functionname' function

The error indication xx occurred during the processing of the specified channel function.

The most important error codes are listed in this section. If an error occurs with an error code that is not included in this table, please call the hotline.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 1 - EC_NOIMPL - Function not implemented

The called function is not implemented in the S7 channel.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 2 - EC_STRUFE - structure error

The language DLLs do not belong to the installed S7 channel.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 3 - EC_ILEGAL - Function call not permitted

The call of the function with the transferred parameters in S7 channel is not allowed

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 4 - EC_NO_RAM - No free memory

The S7 channel could not create the memory that is required for the function.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 5 - EC_NOFILE - File not available

The specified file is not available.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 6 - EC_LNGERR - Error during switching languages

An error occurred while switching languages.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 7 - EC_UNITNV - Invalid unit ID

The specified UNIT-ID is invalid.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 8 - EC_NOUNIT - No unit available

The specified UNIT is not available.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 9 - EC_UNITNA - Unit not active

The specified UNIT is not active.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 10 - EC_PTRERR - Incorrect pointer

The pointer that is transferred in the function call is incorrect.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 11 - EC_TIMERR - Error during starting the internal timer

An error occurred during starting of the internal timer.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 12 - EC_S7LERR - Error during loading of the S7 communication driver

The communication subsystem SAPI-S7 or S7DOS could not be loaded.

- Communication driver not installed correctly.
- The DLLs that are required by the communication driver could not be loaded correctly.
- Path setting incorrect or required communication drivers deleted or moved.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 13 - EC_S7IERR - Error during initializing of the S7 communication driver

An error occurred during initializing of the S7 communication driver.

- The configured device is not available.
- An error occurred during the device driver is activated.
- Copy protection error occurred.
- The bus connector to the MPI card is not inserted.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 14 - EC_CONERR - Connection fault

A read/write job was sent to a disrupted connection.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 15 - EC_PARERR - Incorrect parameter supply

An incorrect parameter was specified during a function call.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 16 - EC_DATERR - Data error occurred

The transferred connection data are invalid or damaged.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 17 - EC_CONDAT - Incorrect connection data

The transferred connection data are invalid or damaged.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 18 - EC_WNDERR - SINEC Windows incorrect

The SINEC Windows required for the communication could not be created.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 19 - EC_RAWERR - Error occurred in the raw data structure

The transferred raw data job is invalid or incorrect parameters were specified.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 20 - EC_INTRDY - Internal function completed

Internal function correctly completed.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 21 - EC_EVNERR - EventNumber in MemberName n incorrect

Internal function correctly completed.

Tag configuration for status processing incorrect.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 22 - EC_GETERR - Error during GETVALUECB

Internal function correctly completed.

An error occurred during the function GETVALUECB

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 23 - EC_EVMERR - EventMember variables incomplete

Internal function correctly completed.

Tag configuration for status processing incorrect.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 24 - EC_EIDERR - WinCC EV_ID for status processing incorrect

Internal function correctly completed.

Tag configuration for status processing incorrect.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 25 - EC_S7DOSV - function available after S7DOS version xx

Internal function correctly completed.

Functions were configured (for example, PMC message processing) that are not supported by the installed communication subsystem.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 26 - EC_EVSTDE - data not permitted during writing of .EventState

Internal function correctly completed.

During writing of an .EventState tag incorrect data were transferred.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 27 - EC_CONSTR - Configured connection too long as string

Internal function correctly completed.

The data of the configured connection result in string with invalid length.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 28 - EC_PDULEN - Configured data length greater than PDU length

Internal function correctly completed.

The data length of a configured tag exceeds the maximum PDU length.

The permitted data length for a PDU length of 240 byte is 208 byte. A PDU length of 240 bytes is usual for AS300 and for communication via SAPI-S7.

For a PDU length of 480 bytes, 448 bytes can be transferred. A PDU length of 480 bytes is usual for AS400.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 29 - EC_OBJERR - configured data area is not supported

Internal function correctly completed.

The configured data area (for example area 0x80 I/O) is not supported.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 30 - EC_SYSPAR - Error during the setting of the system parameters

Internal function correctly completed.

An error occurred during the setting of the system parameters.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 31 - EC_SYPWRT - error during writing of the system parameters

Internal function correctly completed.

An error occurred during the writing of the system parameters in the storage file.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 32 - EC_NOVARI - Error during GetTagInfo call

Internal function correctly completed.

An error occurred during call GetTagInfo.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 33 - EC_ACKERR - Error during acknowledge via EventState

Internal function correctly completed.

An error was reported during the sending of jobs to the AS.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 34 - EC_LCKERR - Error during Lock/Release via EventState

Internal function correctly completed.

An error was reported during the sending of jobs to the AS.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 35 - EC_AUAOVL - buffer overflow during link-up and update

Internal function correctly completed.

A buffer overflow due to excess number of jobs occurred during link-up and update.

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

Error 100 - EC_VATERR - Incorrect type of transferred tag

Internal function correctly completed.

Transferred tag not from the right type

See also

Communication with SIMATIC S7 300/400 (Page 401)

API error messages (Page 561)

2.8.10 Raw data tags of the "SIMATIC S7 300/400" communication driver

2.8.10.1 Raw Data Tag

Definition

In WinCC, you can create external and internal tags of the type "Raw data type". The format and the length of a raw data tag are not fixed. Its length can be in the range from 1 and 65535 bytes. It is either defined by the user or results from a specific application.

The contents of a raw data tag are not fixed. Only senders and receivers can interpret the contents of a raw data tag. WinCC does not interpret them.

Note

A raw data tag cannot be displayed in screens.

Potential Applications within WinCC

Raw data tags can be used in the following areas in WinCC:

- For data exchange with the PLC alarm blocks for processing and acknowledgment of alarms
- In scripts for data exchange with the functions "Get/SetTagRaw"
- In tag logs
- To transfer jobs, data, processing acknowledgments between the HMI device and the PLC

Note

When the raw data tag is displayed in an I/O field, you must observe the conventions of the string with a "\0" character at the end.

"Properties address"

The "Properties address" is not the same for all communication drivers with external raw data tags because the parameters of the tag address and the supported raw data tag type depend on the communication driver used.

Format adaptation

There is no format adaptation in WinCC for the "raw data type".

2.8.10.2 Raw data tags of the "SIMATIC S7 300/400" communication driver

Introduction

A tag of the "Raw data type" is a data telegram of the ISO/OSI layer model. The contents of the raw data tag are not fixed and therefore only the sender and the receiver can interpret the transmitted data. There is no format adaptation in WinCC for this data type. The maximum length is 65535 bytes.

WinCC distinguishes between two types of raw data tags:

- Raw data tag for free use
- Raw data tag for handling S7 functions

Raw data tag for free use

Raw data tags for free use are used for transferring user data blocks between the HMI device and the PLC and only contain user data. It distinguishes between:

- Raw data tag as byte array
- Raw data tag for BSEND/BRCV functions

Raw data tag for handling S7 functions

These raw data tags do not have any connection-specific header and are normally used by the alarm system and for process data entry in WinCC.

Supported automation systems

Raw data communication is supported for the following automation systems:

- S7-400
- S7-300
  - CPU319-3 PN/DP V2.5 or higher
  - CPU317-2 PN/DP V2.6 or higher
  - CPU315-2 PN/DP V3.1 or higher

  For S7-300 controllers, a firmware version V3.x or higher is recommended.

- Raw data communication cannot be set up over a communication processor.
- WinAC RTX 2010

2.8.10.3 Raw data tag as byte array

Introduction

Raw data tags as byte array are used for transferring user data blocks between the HMI device and the PLC and only contain user data.

Raw data tags as byte array are handled in the same way as process tags in the communication driver; these are addressed based on the address and length of the data range (e.g. DB 100, DW 20, length 40 bytes).

The raw data length is limited to one transferable data block and must be fully transferable using a PDU (Protocol Data Unit). The PDU length is negotiated during connection establishment. The maximum length of the data block that can be transferred is based on the PDU length minus the header and additional information. The PDU lengths normally used in SIMATIC S7 result in the following maximum lengths:

- S7-300: PDU length 240 Byte, max. data block length 208 Byte
- S7-400: PDU length 480 Byte, max. data block length 448 Byte

Data must be blocked if larger data blocks are to be transferred. This blocking is implemented by the S7 software in the PLC and by scripts in WinCC.

Exchanging large data volumes

The following application example shows you how to use raw data to transfer large amounts of data from the PLC to WinCC:

- Example - Data transfer (https://support.industry.siemens.com/cs/ww/en/view/37873547)

Configuring a raw data tag as byte array

The raw data tags for transferring data blocks are configured as raw data of the "Send/receive raw data block" type with one address and one length detail.

The following figures show an example of a configuration for a data range that is 40 bytes long in the data block 100 as of data word 20:

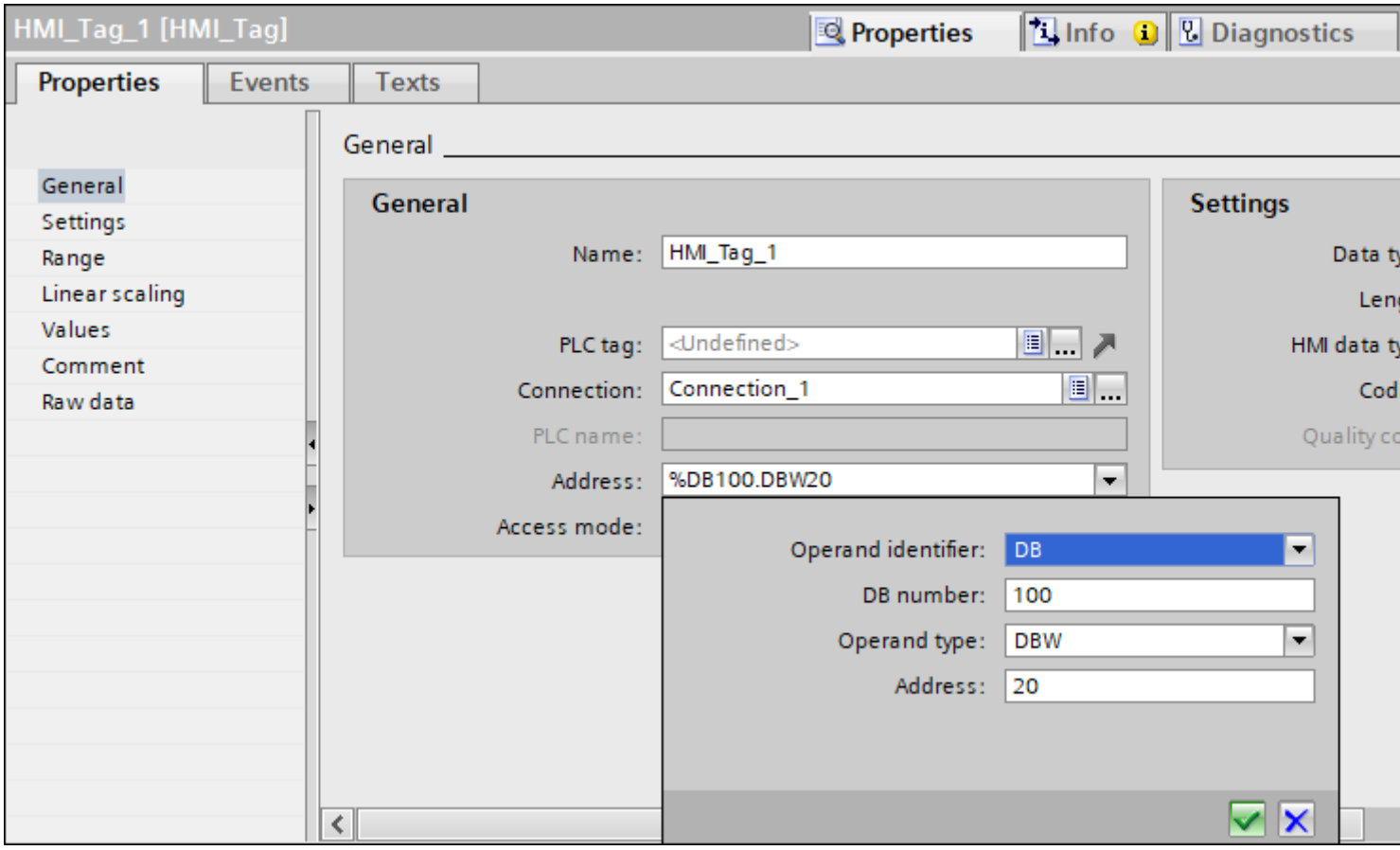


Figure 2-6 Set raw data address

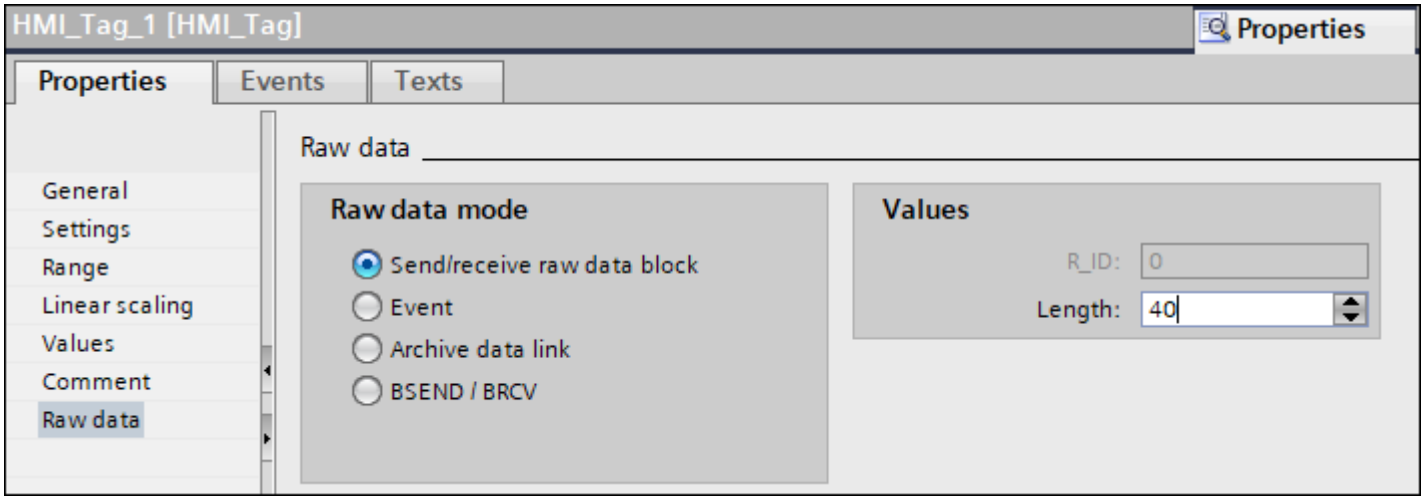


Figure 2-7 Raw data mode "Send/receive raw data block" and the set data length

Read a Raw data tag as Byte Array

This raw data tag is read in the same way as a process tag. The corresponding data block is requested from the PLC and transferred to the HMI device after the data has been received.

Data is transferred always at the initiative of WinCC. Sporadic or event-driven data receipt initiated by the PLC is not possible using this raw data tag.

Write a Raw data tag as Byte Array

This raw data tag is written in the same way as a process tag. After sending the data block and receiving a positive acknowledgement from the PLC, the data block is transferred to the tag image in the HMI device.

2.8.10.4 How to configure a raw data tag as byte array

Introduction

In this section you will learn how a raw data tag of the "SIMATIC S7 300/400" communication driver is configured as a byte array.

The configuration is the same for all interfaces of the communication driver. The example uses the "MPI" interface.

Procedure

1. Double-click "Connections" in the project navigation and create a connection.
2. Select the "SIMATIC S7 300/400" communication driver and the "MPI" interface.

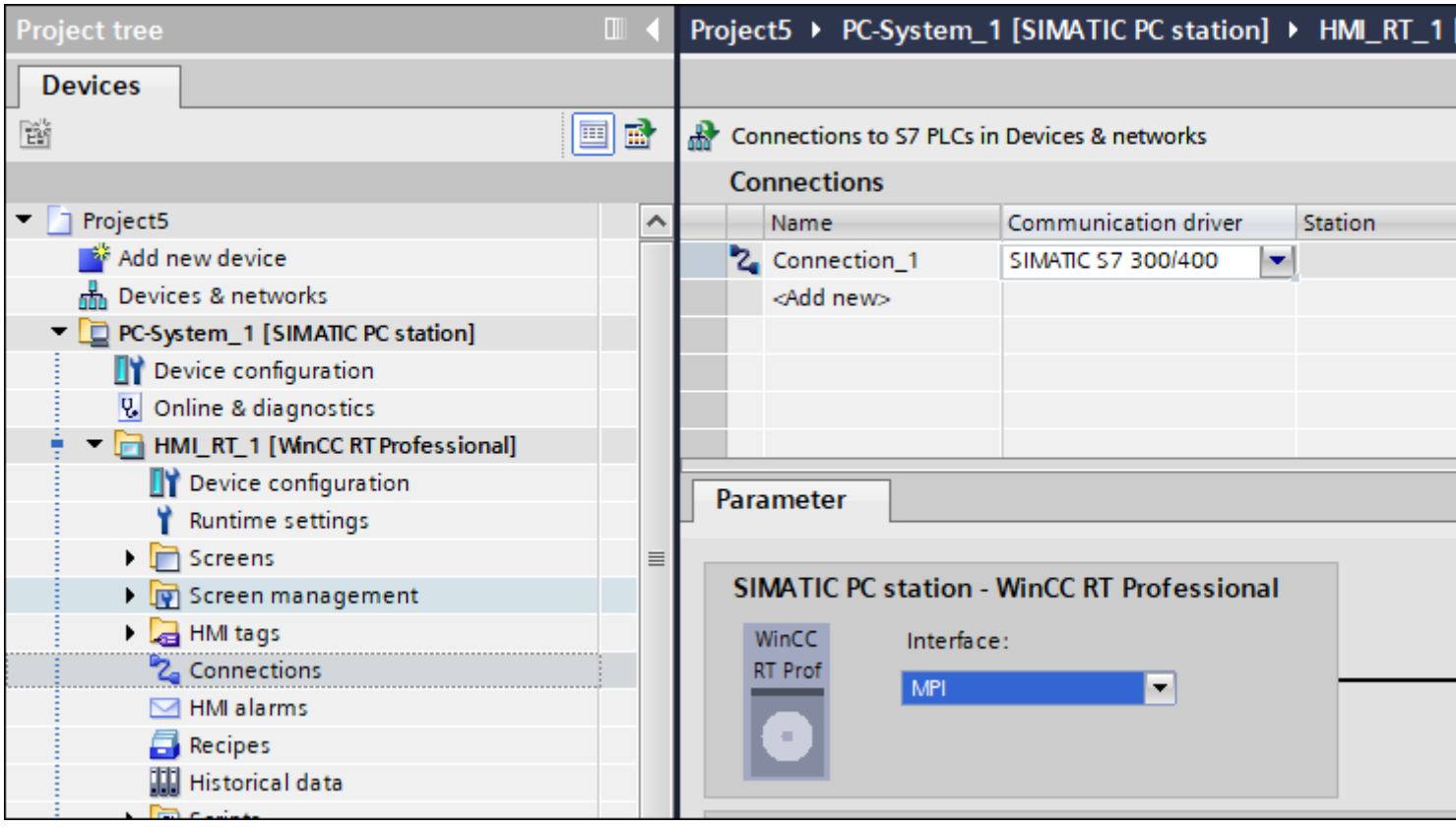
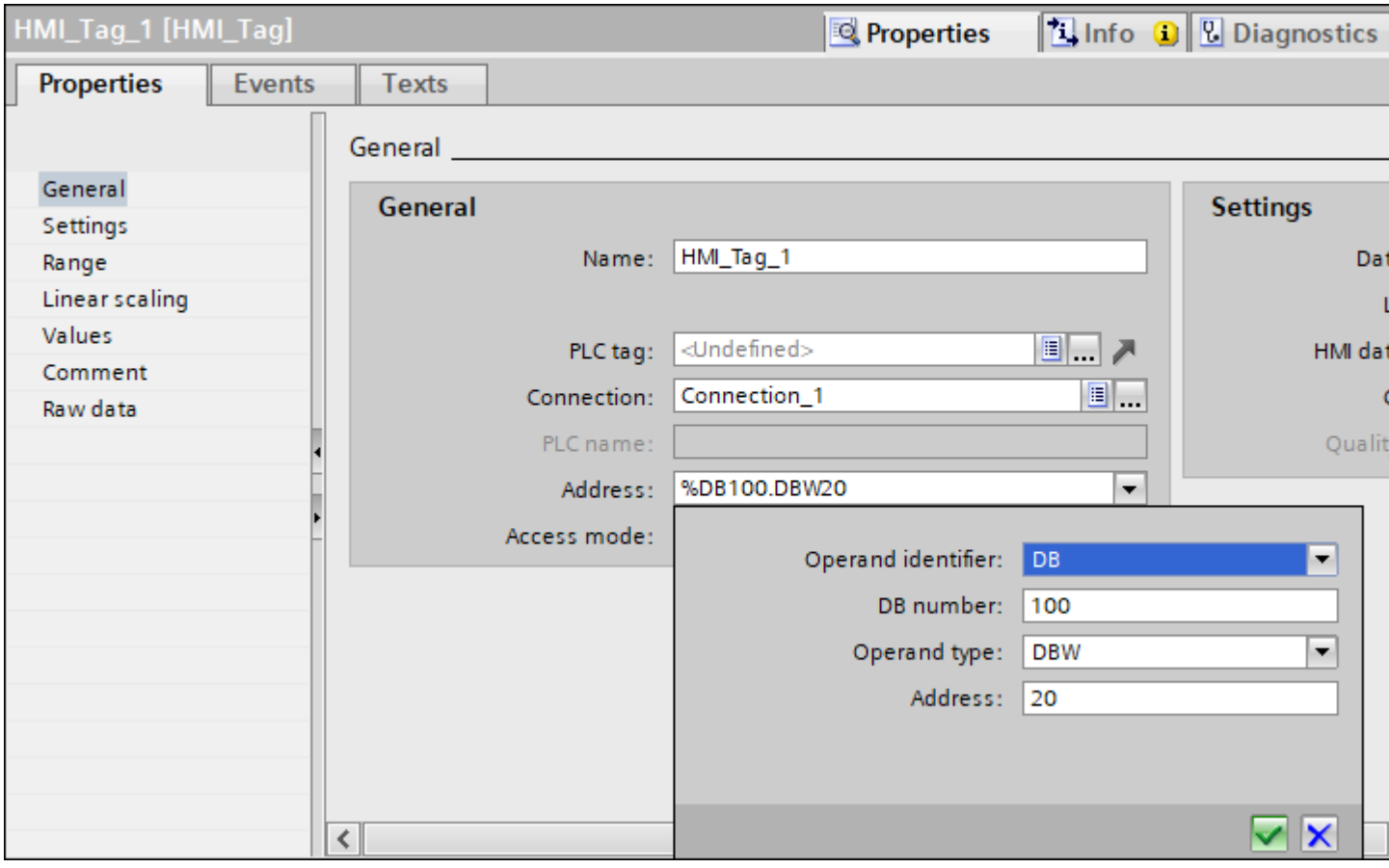


Figure 2-8 Select communication driver and interface

3. Double-click "HMI tags > Show all tags" in the project navigation and create an HMI tag under "HMI tags".
4. Select the data type "Raw" in the "Data type" field.
5. Select "Connection_1" in the "Connection" field and then confirm the selection.

6. Select "Properties > Properties > General" in the Inspector window.



7. Open the "Address" field and enter the desired address details.
8. Select the "DB" operand identifier.
9. Enter the number of the data block.
10. Set the "DBW" operand type.
11. Enter the value of the start address.
12. Enter the desired length of the data range in the "Settings" field.
13. Select "Properties > Properties > Raw data" in the Inspector window.
14. The type "Send/receive raw data block" in the "Raw data mode" area is automatically set.

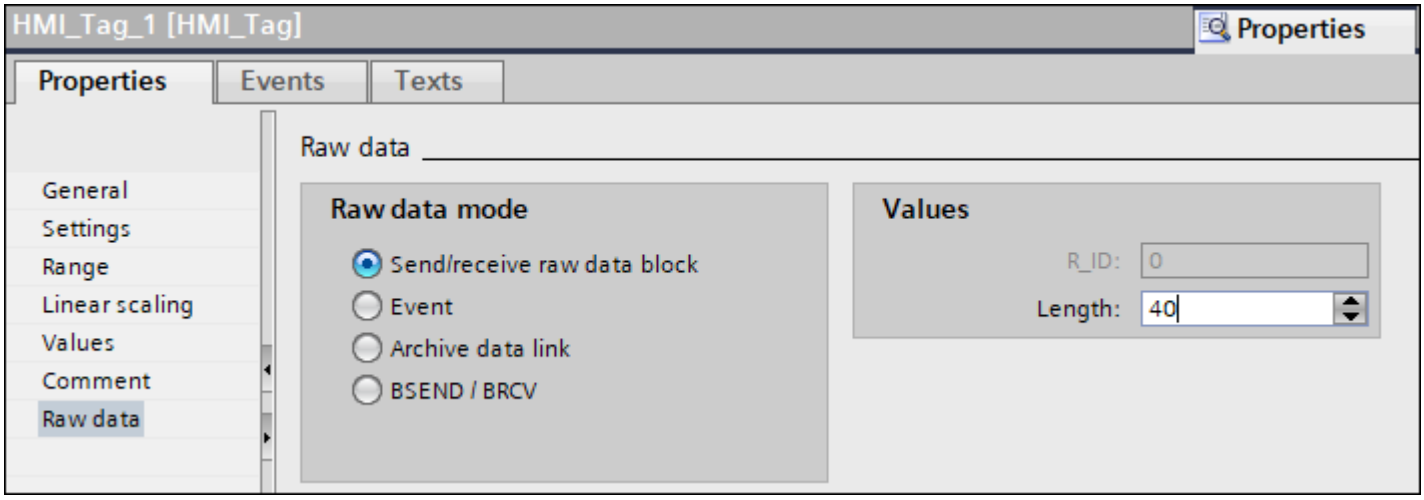


Figure 2-9 Set raw data mode

Note

The raw data length is limited to one transferable data block and must be fully transferable using a PDU (Protocol Data Unit). The PDU length is negotiated during connection establishment. The maximum length of the data block that can be transferred is based on the PDU length minus the header and additional information. The PDU lengths normally used in SIMATIC S7 result in the following maximum lengths:

- S7-300: PDU length 240 Byte, max. data block length 208 Byte
- S7-400: PDU length 480 Byte, max. data block length 448 Byte

An incorrectly specified length leads to rejection of the read/write job with a display.

2.8.10.5 Raw data tag for BSEND/BRCV functions of S7 communication

Introduction

Raw data tags for "BSEND/BRCV" functions are used for transferring user data blocks between the HMI device and the PLC and only contain user data.

This raw data type can be used to access the "BSEND/BRCV" functions of S7 communication.

The "BSEND/BRCV" raw data communication via Named Connections is supported for the following automation systems:

- S7-400
- S7-300
  - CPU319-3 PN/DP V2.5 or higher
  - CPU317-2 PN/DP V2.6 or higher
  - CPU315-2 PN/DP V3.1 or higher

  For S7-300 controllers, a firmware version V3.x or higher is recommended.
- Raw data communication cannot be set up over a communication processor.

- WinAC RTX 2010

For resource reasons, it is advisable to keep the number of BSEND/BRCV raw data tags low.

Resource limitation for the use of S7 functions "AR_SEND" and "BSEND/BRCV"

The maximum data volume that can be sent simultaneously using AR_SEND and/or BSEND/BRCV functions from the PLC to the HMI device is limited:

- 16 Kbytes for the S7-400
- 8 Kbytes for the S7-300

Examples with the S7-400:

- 1x BSEND with a max. of 16 Kbytes
- 1x AR_SEND with 8 kBytes + 1x BSEND with 8 kBytes
- 1x AR_SEND with 10 Kbytes + 1x AR_SEND with 2 Kbytes + 1x BSEND with 4 Kbytes

Note

Coordination of the write jobs

If the data block of a write job is transferred to the PLC and has not yet been deleted or fully deleted from the receiving buffer, then the next write job will be rejected with an error message.

During such an error display, write jobs with R_ID > 0x8000 0000 are written to a connection-specific queue and the system tries to repeat the write job for 6 seconds.

The responsibility for time co-ordination for transfer rests with the user and needs to be noted as shorter time intervals for write jobs.

Configuring a PBK Connection for Using "BSEND/BRCV" functions

"BSEND/BRCV" functions can only be used via a "hard-configured connection", a so-called PBK connection (programmed component communication).

To configure a hard-configured connection, you must specify a connection resource (hex: 10 ... DF) in the connection parameters.

This connection resource will be assigned by STEP 7 when the connection is configured in the PLC.

The connection must be configured as passive connection endpoint in the PLC.

Read/write jobs

A hard-configured connection can also be used to handle other read and write jobs.

When very large data areas are to be transferred via the connection, then the data blocks are transferred in multiple PDUs.

For performance reasons, it would therefore be better to create a separate connection for "BSEND/BRCV" functions.

Configuring Raw data tag for BSEND/BRCV functions

Raw data tags for transferring "BSEND/BRCV" data blocks are configured as raw data of type "BSEND/BRCV" with an "R_ID".

The data length is derived implicitly from the sent or received data volume.

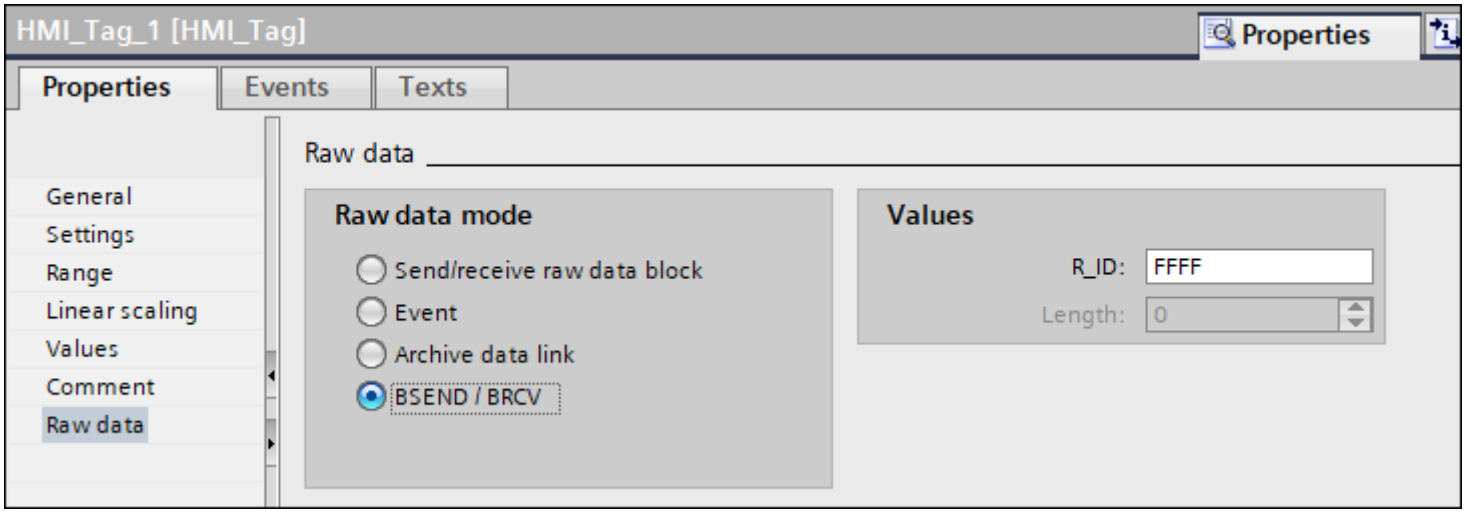


Figure 2-10 Raw data tag for "BSEND/BRCV" functions

"R_ID" Parameter

For the "BSEND/BRCV" functionality, you must specify a 32-bit long R_ID as hexadecimal number.

The R_ID is assigned at the time of configuration in the PLC and is used for distinguishing multiple data block transfers over one connection.

The send and receive calls are always notified with reference to this R_ID in the underlying communication sub-system (SIMATIC Device Drivers).

A raw data tag is thus always assigned to one unique R_ID.

Sending a "BSEND/BRCV" raw data tag

Sending of a "BSEND/BRCV" raw data tag takes place in the same way as writing a process tag.

After sending the data block and receiving a positive acknowledgement from the PLC, the data block is transferred to the tag image in the HMI device.

Receiving a "BSEND/BRCV" raw data tag

Initiated by the PLC, "BSEND/BRCV" raw data is sporadically sent to the communication driver.

It is therefore not possible to explicitly read S7 raw data tags.

Synchronization

The BSEND/BRCV mechanisms do not include any synchronization functions.

If no partner has logged in to receive the data during the start-up phase, the data blocks sent by the PLC will be discarded on the receiver side.

Hence, the user has to take care of the synchronization and, for example, release the sending direction on the PLC by setting a flag with a data word.

2.8.10.6 How to configure a raw data tag for "BSEND/BRCV" functions

Introduction

In this section, you will learn how a raw data tag of the "SIMATIC S7 300/400" communication driver is configured for "BSEND/BRCV" functions.

The configuration is the same for all interfaces of the communication driver. The example uses the "MPI" unit.

Procedure

1. Double-click "Connections" in the project navigation and create a connection.
2. Select the SIMATIC S7 300/400 communication driver and the "MPI" interface.

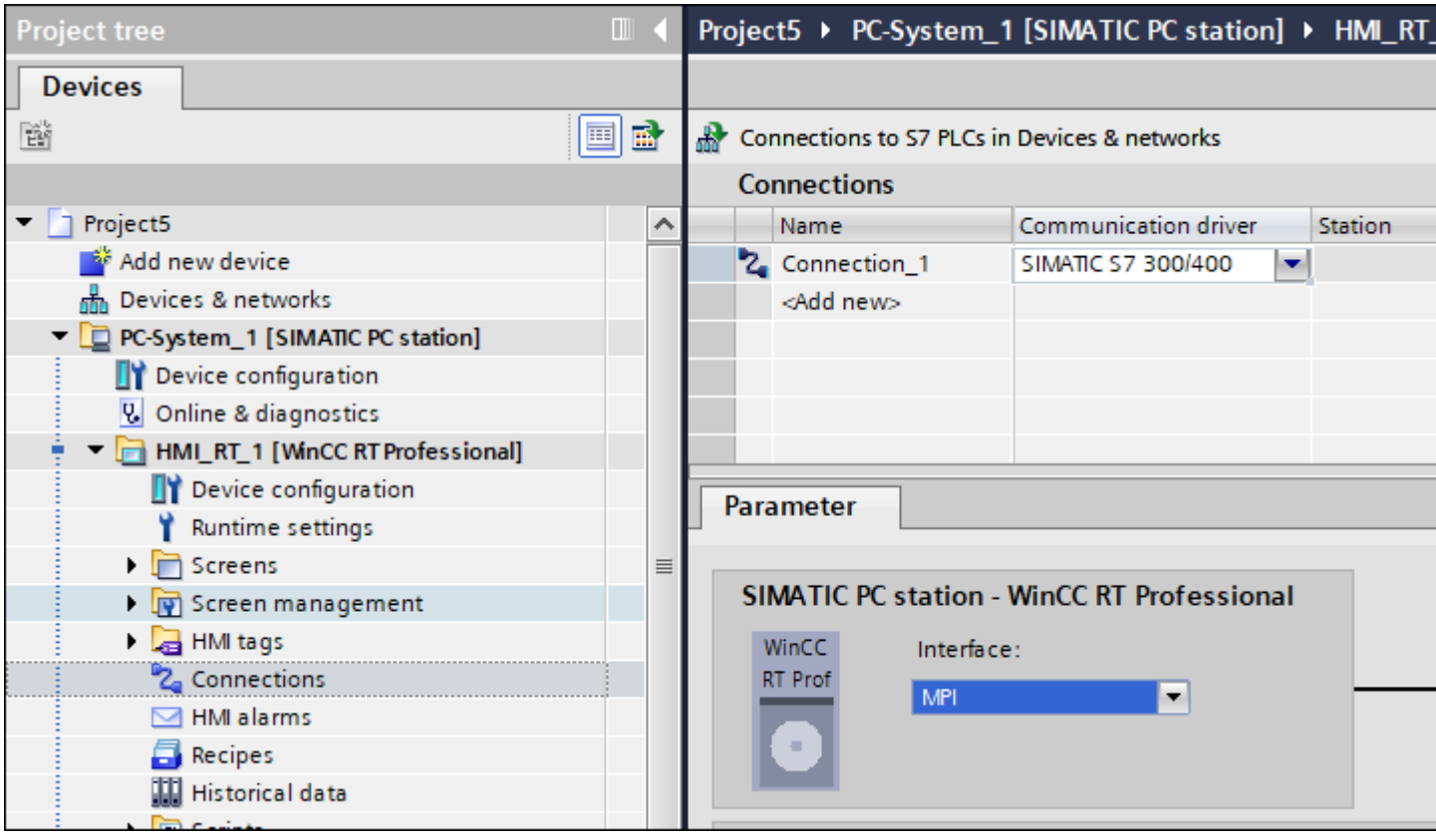


Figure 2-11 Set communication driver and interface

3. Double-click "HMI tags > Show all tags > HMI tags" in the project navigation and create an HMI tag.

4. Select the data type "Raw" in the "Data type" field.
5. Select "Connection_1" in the "Connection" field and then confirm the selection.
6. Select "Properties > Properties > Raw data" in the Inspector window.

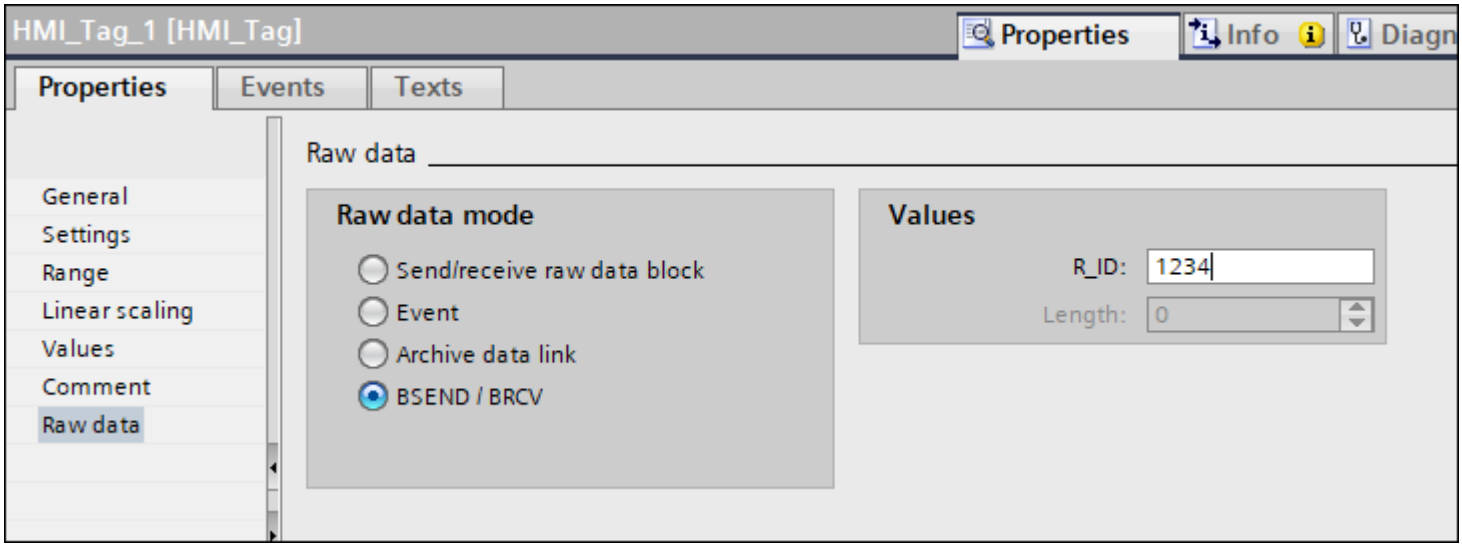


Figure 2-12 Select raw data mode and specify R_ID

7. Select "BSEND/BRCV" as the type in the "Raw data mode" area.
8. Enter the hexadecimal value of the ID in the "R_ID" field. The R_ID is assigned at the time of configuration in the PLC.

2.9 Communicating with the SIMATIC S7-1500 Software Controller

2.9.1 Communication with SIMATIC S7-1500 Software Controller

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 215).

HMI connection for communication

You configure connections between the HMI device and SIMATIC S7-1500 Software Controller in the "Devices & Networks" editor.

See also

Communication via PROFIBUS (Page 215)

2.9.2 Communication via PROFINET

2.9.2.1 Basics of HMI connections to an SIMATIC S7-1500 Software Controller

Introduction

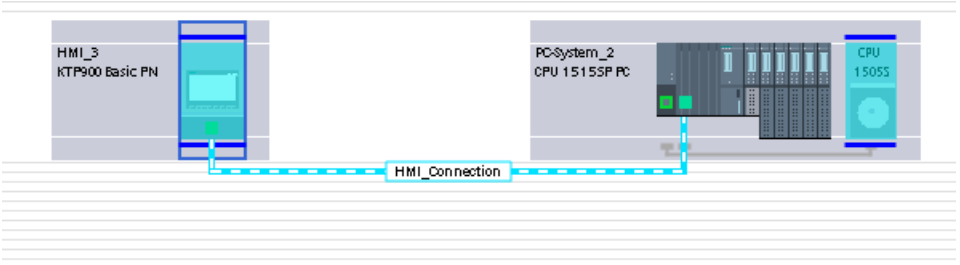
In a PC station, the SIMATIC S7-1500 Software Controller can support the following HMI connections depending on the network:

- Direct HMI connection
- Routed HMI connection to a PLC in another subnet

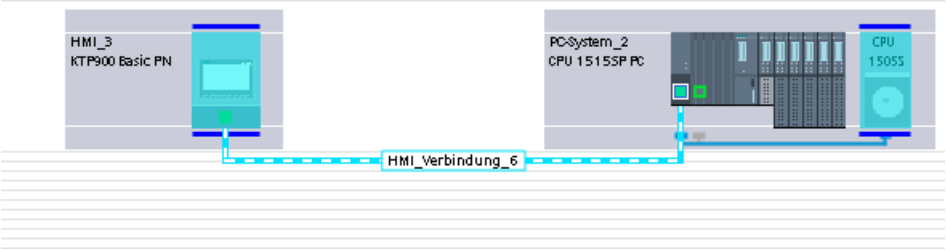
You can find additional information on networking a PC station with an SIMATIC S7-1500 Software Controller in the documentation for the SIMATIC S7-1500 Sofware Controller.

Direct HMI connection to an SIMATIC S7-1500 Software Controller

If a local interface is available on the PC station with the SIMATIC S7-1500 Software Controller, this interface is automatically assigned to the SIMATIC S7-1500 Software Controller when the HMI connection is established. An HMI connection created like this is a direct connection.

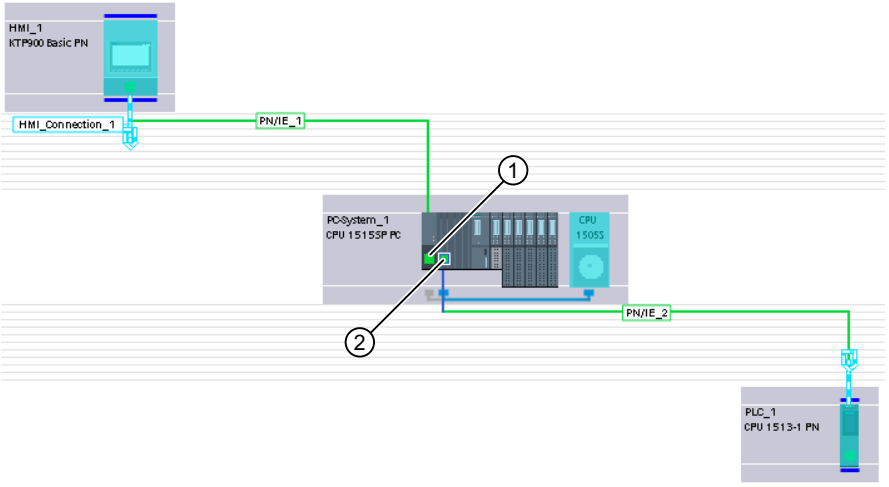


If an interface other than the local interface has been configured for communication on the PC station, the HMI connection is classified as a "routed connection". The HMI connection is, however, shown as a direct connection:



Routed HMI connection over an SIMATIC S7-1500 Software Controller

The SIMATIC S7-1500 Software Controller supports the routing functionality. The PC station must, however, have at least two PN/IE interfaces. The figure below shows the configuration of an HMI connection to an S7 CPU routed through the SIMATIC S7-1500 Software Controller.



① Local interface or interface retrofitted over a communication processor
Required settings under "Interface assignment":
- "PC station"

② "X1" local interface for process communication
Required settings under "Interface assignment":
- "Software_PLC_1" (set automatically)

See also

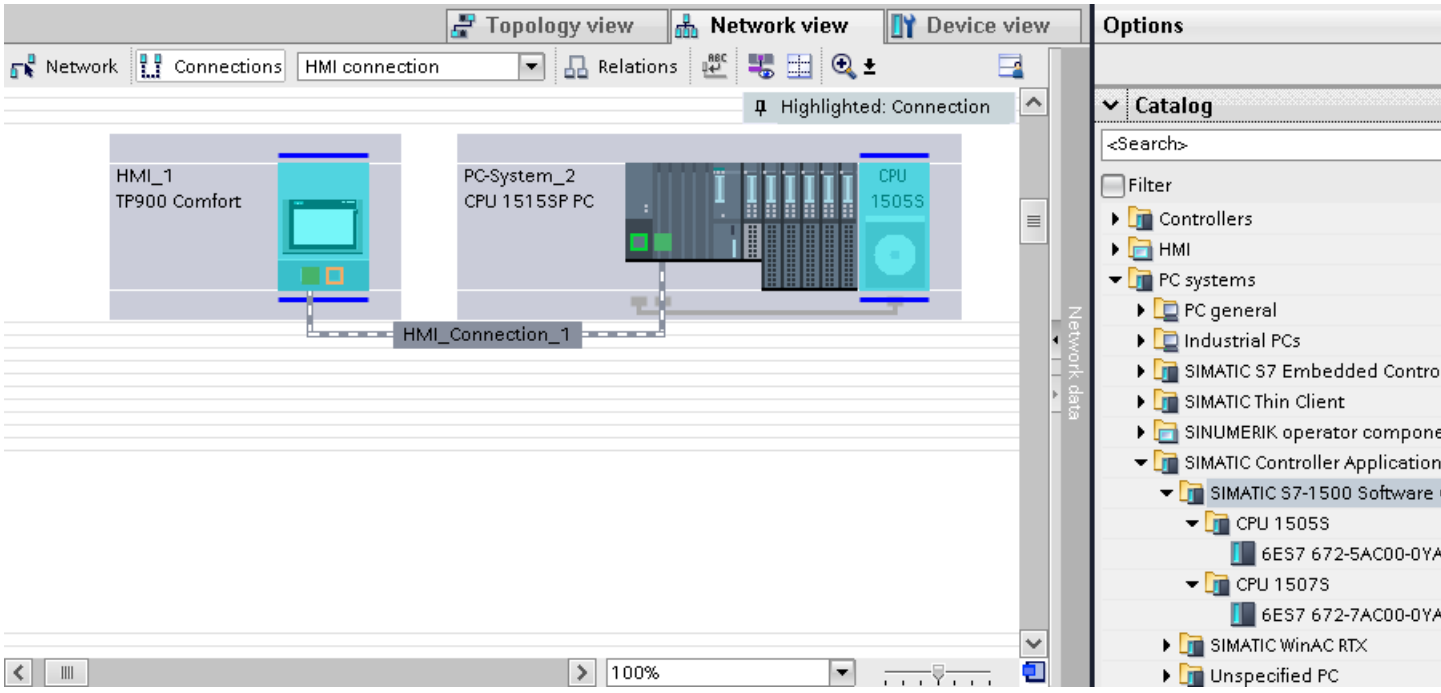
Communication with SIMATIC S7-1500 Software Controller (Page 582)

2.9.2.2 Configuring an HMI connection

Communication via PROFINET

HMI connections via PROFINET

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



You can also connect multiple HMI devices to a single SIMATIC S7-1500 Software Controller and multiple SIMATIC S7-1500 Software Controller 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 582)

Basics of HMI connections to an SIMATIC S7-1500 Software Controller (Page 583)

Configuring an HMI connection via PROFINET (Page 586)

Configuring an HMI connection via PROFINET

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

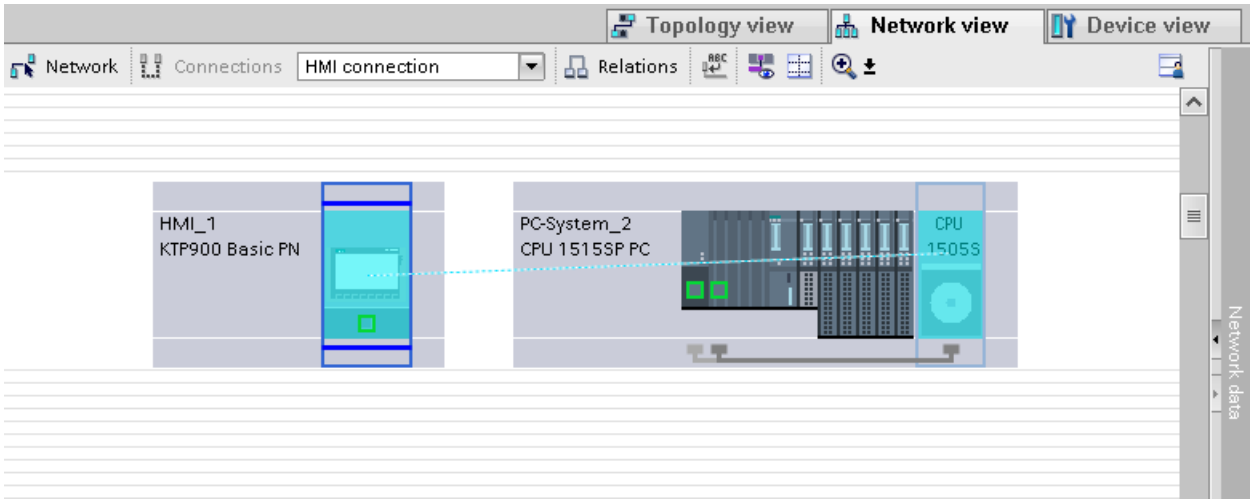
The following communication partners are created in the "Devices & Networks" editor:

- 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 595)".

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 582)

Basics of HMI connections to an SIMATIC S7-1500 Software Controller (Page 583)

Communication via PROFINET (Page 585)

PROFINET parameters (Page 595)

2.9.2.3 Configuring an HMI connection

Communication via PROFINET

Communication via PROFINET

This section describes communication over PROFINET between a WinCC Runtime and an SIMATIC S7-1500 Software Controller.

You can use the following WinCC Runtimes as an HMI device:

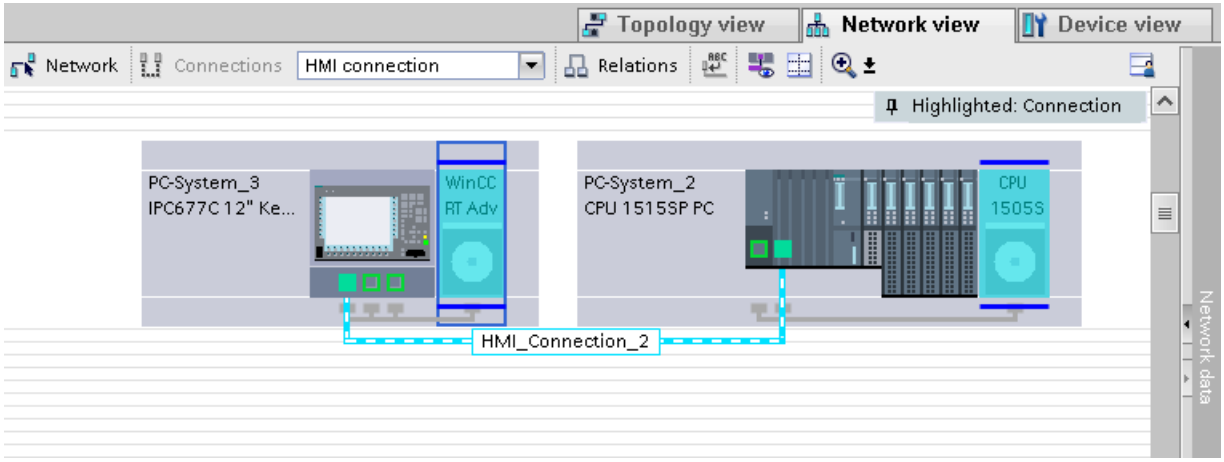
- WinCC RT Advanced
- WinCC RT Professional

WinCC Runtime as HMI device

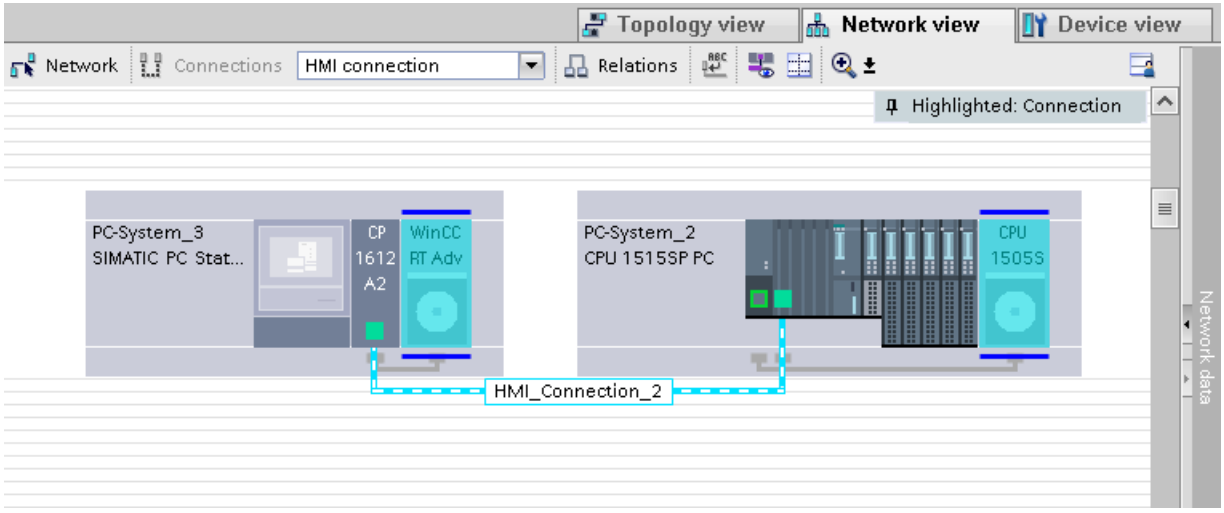
Configure the HMI connections between a WinCC Runtime and an SIMATIC S7-1500 Software Controller in the "Devices & Networks" editor.

If you want to use a WinCC Runtime as an HMI device, you have the following options:

1. You create a SIMATIC PC and configure a WinCC Runtime on this SIMATIC PC.
   In this way you use the SIMATIC PC with a WinCC Runtime as an HMI device.



2. You create a WinCC Runtime and configure a communication processor to the Runtime.
   In this way you use your configuration PC with a WinCC Runtime as an HMI device.



You can also connect multiple HMI devices to a single SIMATIC S7-1500 Software Controller and multiple SIMATIC S7-1500 Software Controller 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 582)

Basics of HMI connections to an SIMATIC S7-1500 Software Controller (Page 583)

Configuring an HMI connection via PROFINET with a PC (Page 590)

Configuring an HMI connection via PROFINET with a SIMATIC PC (Page 593)

Configuring an HMI connection via PROFINET with a PC

Introduction

You configure an HMI connection between an HMI device 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.

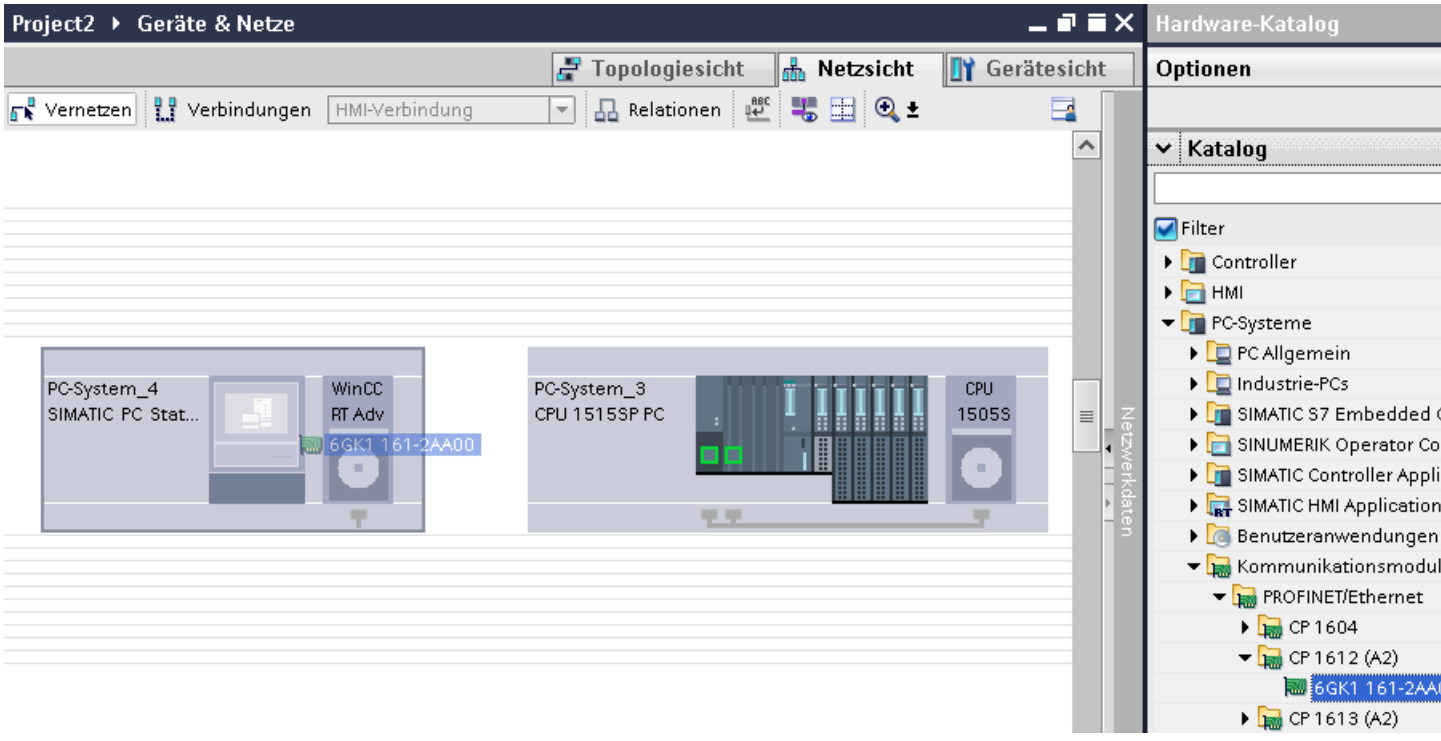
Requirements

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

- PC station with an SIMATIC S7-1500 Software Controller with PROFINET interface
- PC station with a "WinCC Runtime" application

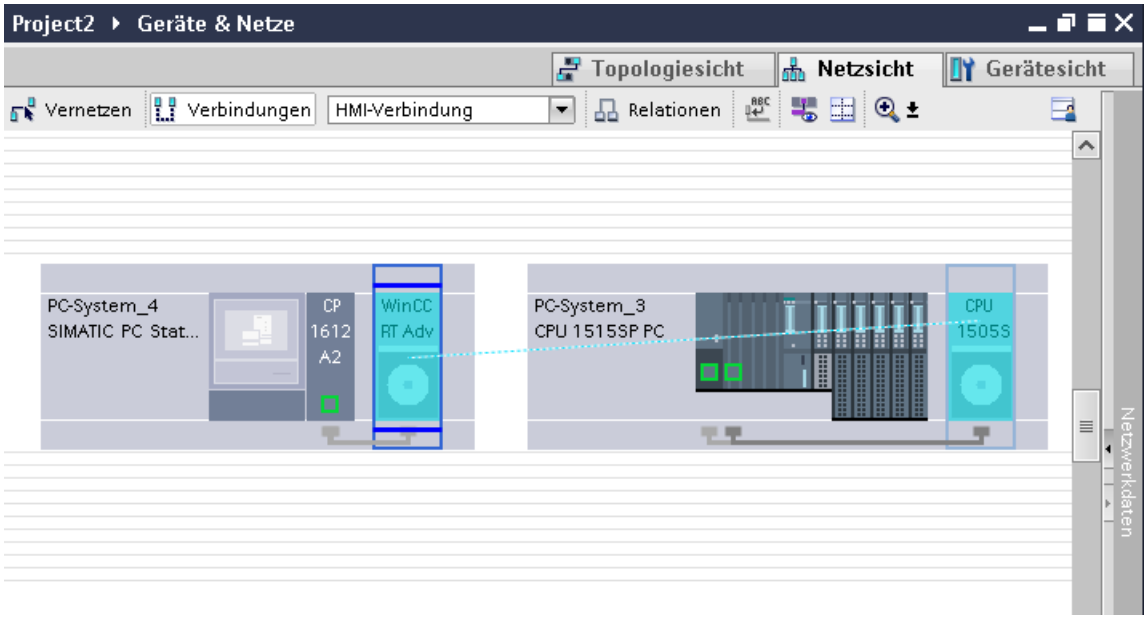
Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed in the network view.
2. Use a drag-and-drop operation to move a PROFINET-capable communication processor from the hardware catalog to the WinCC Runtime.



3. Click the "Connections" button and select "HMI connection" for the connection type.
   The devices available for connection are highlighted in color.

4. Click on WinCC Runtime in the PC station and create a connection to the SIMATIC S7-1500 Software Controller with drag-and-drop.



5. Click on the connected interface in the PC station and select the "PC station" entry under "Interface assignment" in the Inspector window.
6. Click the connecting line.
   The connection is displayed graphically in the Inspector window.
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.
   For additional information on this, see section "PROFINET parameters (Page 595)".

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 582)

Basics of HMI connections to an SIMATIC S7-1500 Software Controller (Page 583)

Communication via PROFINET (Page 588)

Configuring an HMI connection via PROFINET with a SIMATIC PC (Page 593)
PROFINET parameters (Page 595)

Configuring an HMI connection via PROFINET with a SIMATIC PC

Introduction

You configure an HMI connection between an HMI device 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.

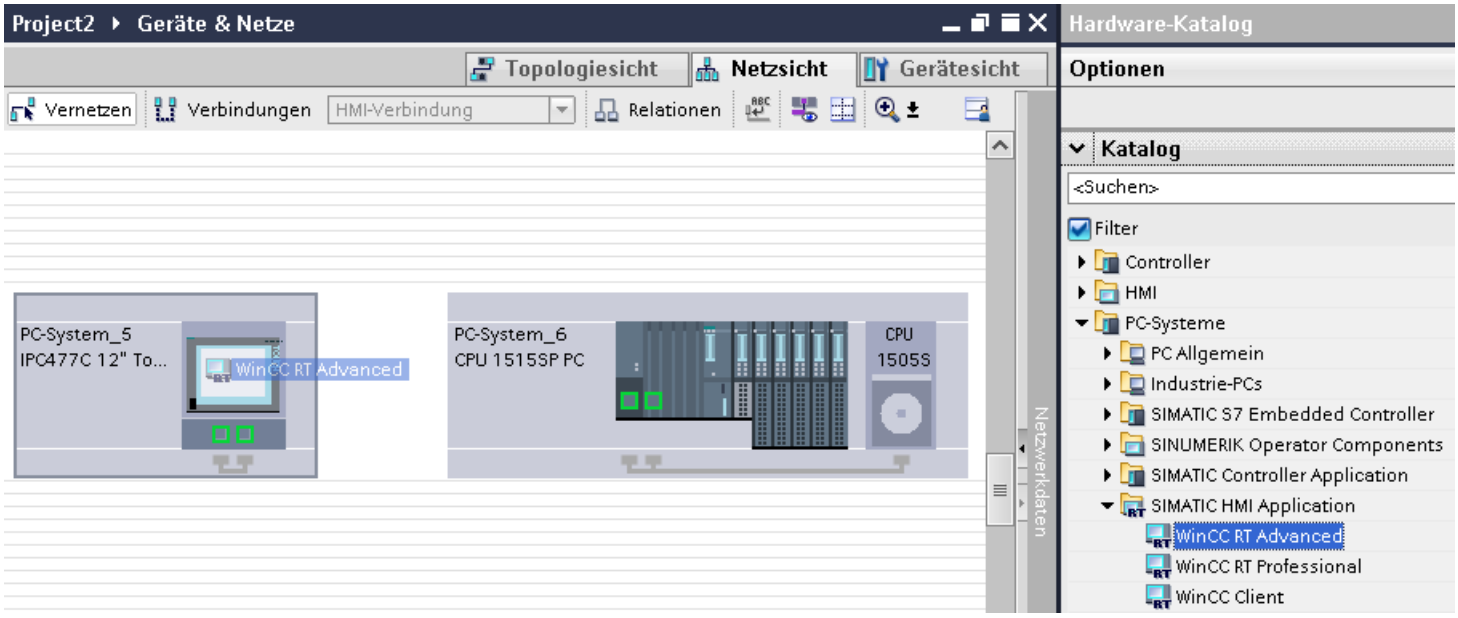
Requirements

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

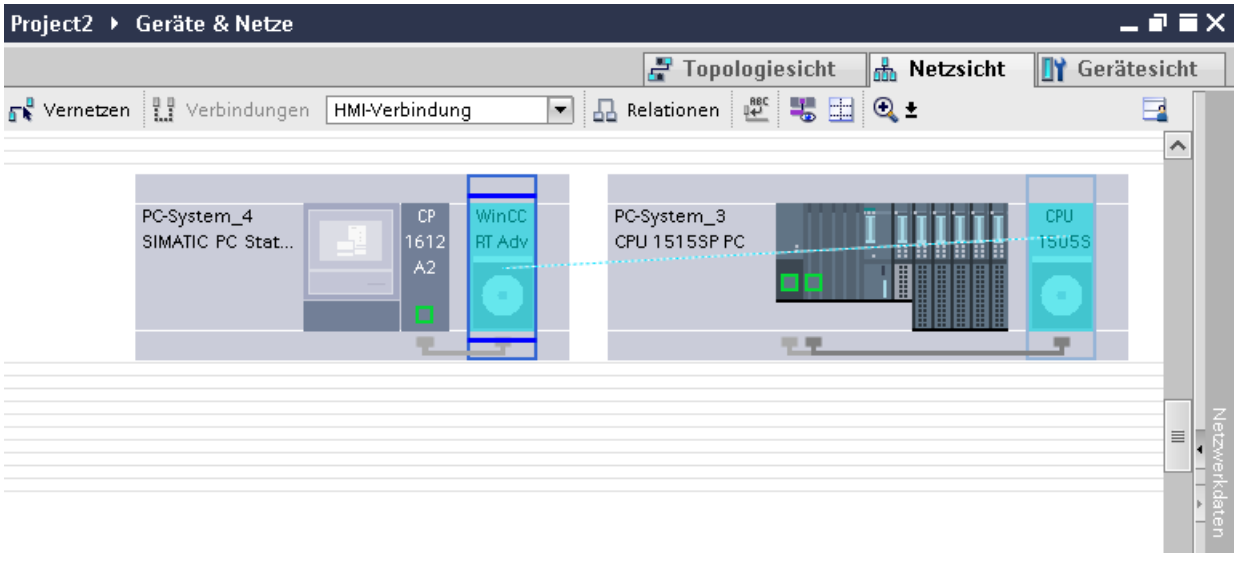
- PC station with an SIMATIC S7-1500 Software Controller with PROFINET interface
- SIMATIC PC with PROFINET interface

Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed in the network view.
2. Use a drag-and-drop operation to move a WinCC Runtime from the hardware catalog to the PC.



3. Click the "Connections" button and select "HMI connection" for the connection type.
   The devices available for connection are highlighted in color.
4. Click on WinCC Runtime in the PC station and create a connection to the SIMATIC S7-1500 Software Controller with drag-and-drop.



5. Click on the connected interface in the PC station and select the "PC station" entry under "Interface assignment" in the Inspector window.
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.
   For additional information on this, see section "PROFINET parameters (Page 595)".

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 582)

Basics of HMI connections to an SIMATIC S7-1500 Software Controller (Page 583)

Communication via PROFINET (Page 588)

Configuring an HMI connection via PROFINET with a PC (Page 590)

PROFINET parameters (Page 595)

2.9.2.4 PROFINET parameters

PROFINET parameters for the HMI connection

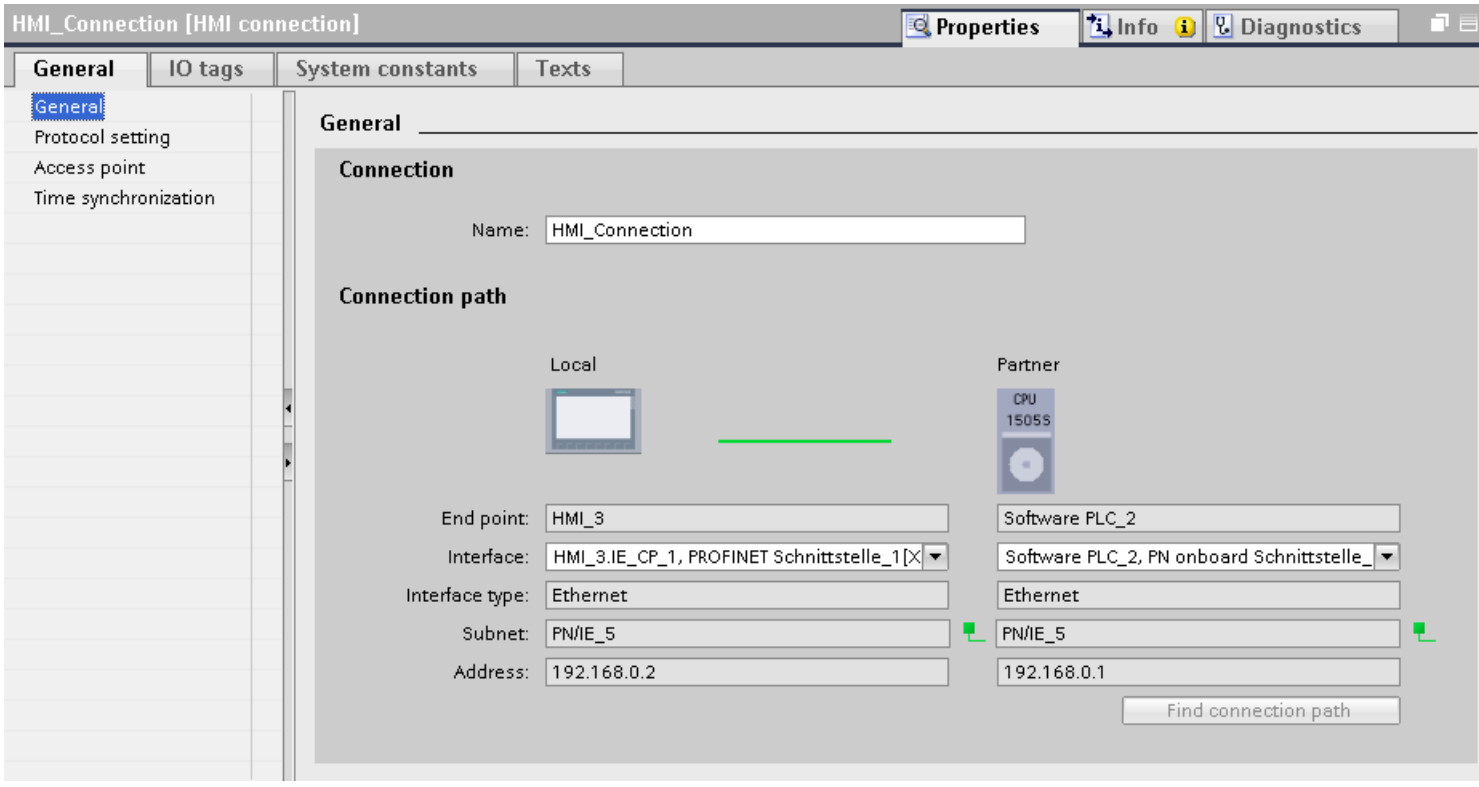
PROFINET parameters for the HMI connection

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 Software Controller (Page 582)

Basics of HMI connections to an SIMATIC S7-1500 Software Controller (Page 583)

PROFINET parameters for the HMI device (Page 597)

PROFINET parameters for the PLC (Page 599)

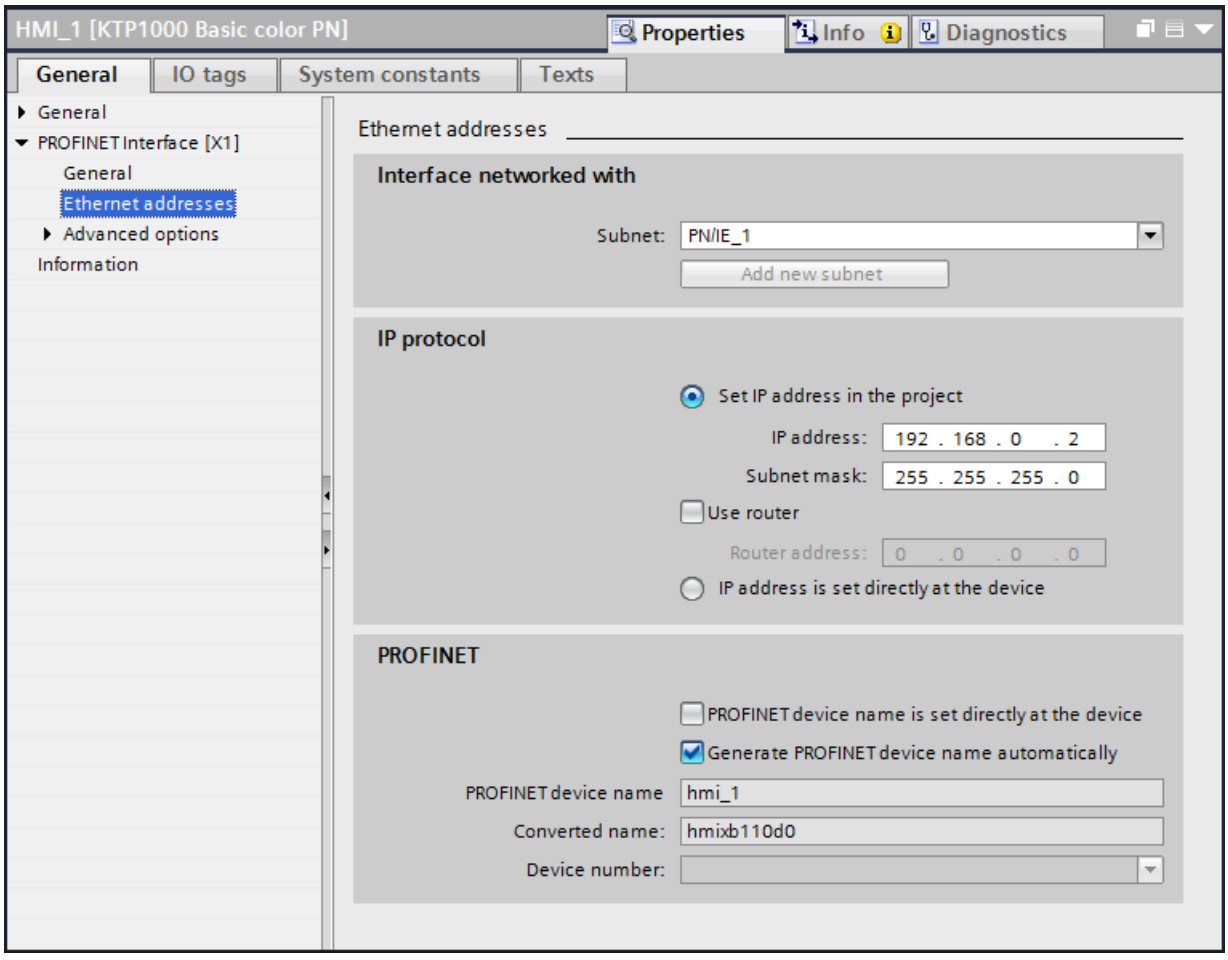
PROFINET parameters for the HMI device

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:
  - Mobile Panel 177 PN
  - Mobile Panel 177 DP

- "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 with SIMATIC S7-1500 Software Controller (Page 582)

Basics of HMI connections to an SIMATIC S7-1500 Software Controller (Page 583)

PROFINET parameters for the HMI connection (Page 595)

PROFINET parameters for the PLC (Page 599)

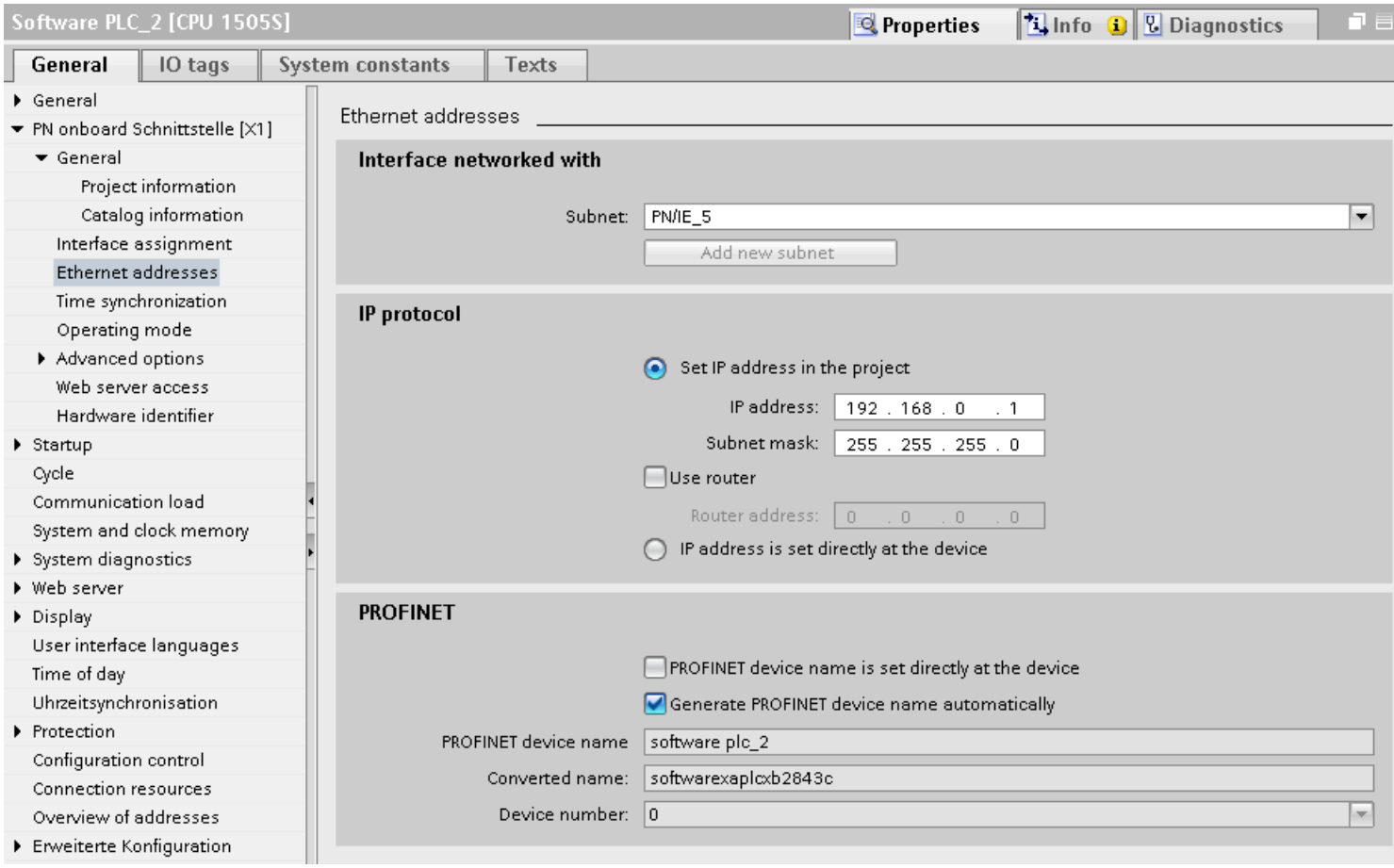
PROFINET parameters for the PLC

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 Software Controller (Page 582)

Basics of HMI connections to an SIMATIC S7-1500 Software Controller (Page 583)

PROFINET parameters for the HMI connection (Page 595)

PROFINET parameters for the HMI device (Page 597)

Protection of communication

Security levels

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

Communication with SIMATIC S7-1500 Software Controller (Page 582)

Basics of HMI connections to an SIMATIC S7-1500 Software Controller (Page 583)

Access password for the HMI connection (Page 602)

Access password for the HMI connection

Introduction

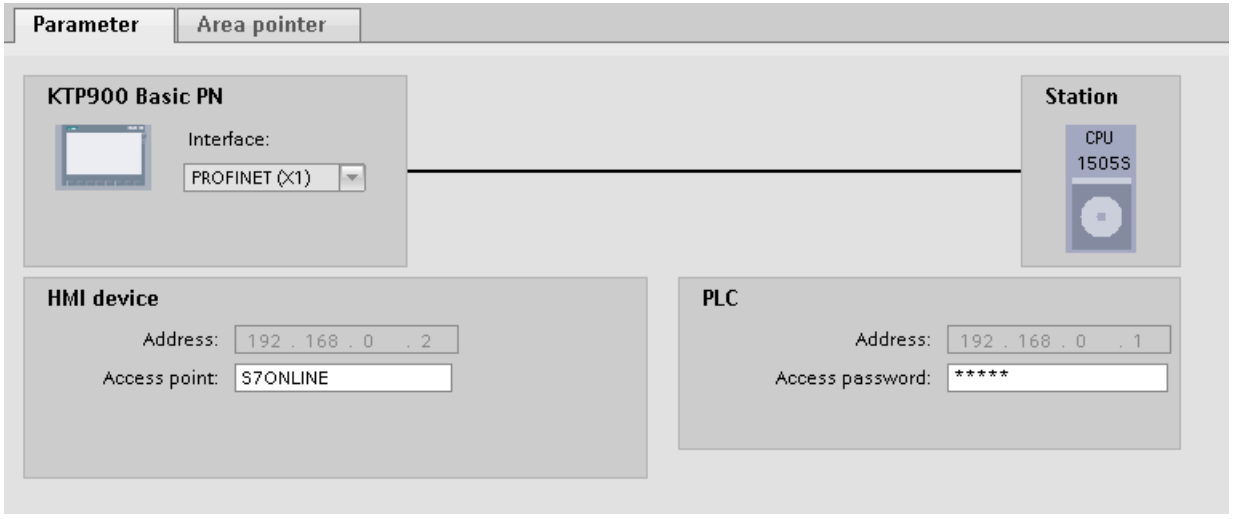
Communication with a PLC with the"Complete protection" protection level is protected by a password. The password is stored in the properties of the PLC.

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

Communication with SIMATIC S7-1500 Software Controller (Page 582)

Basics of HMI connections to an SIMATIC S7-1500 Software Controller (Page 583)

Security levels (Page 601)

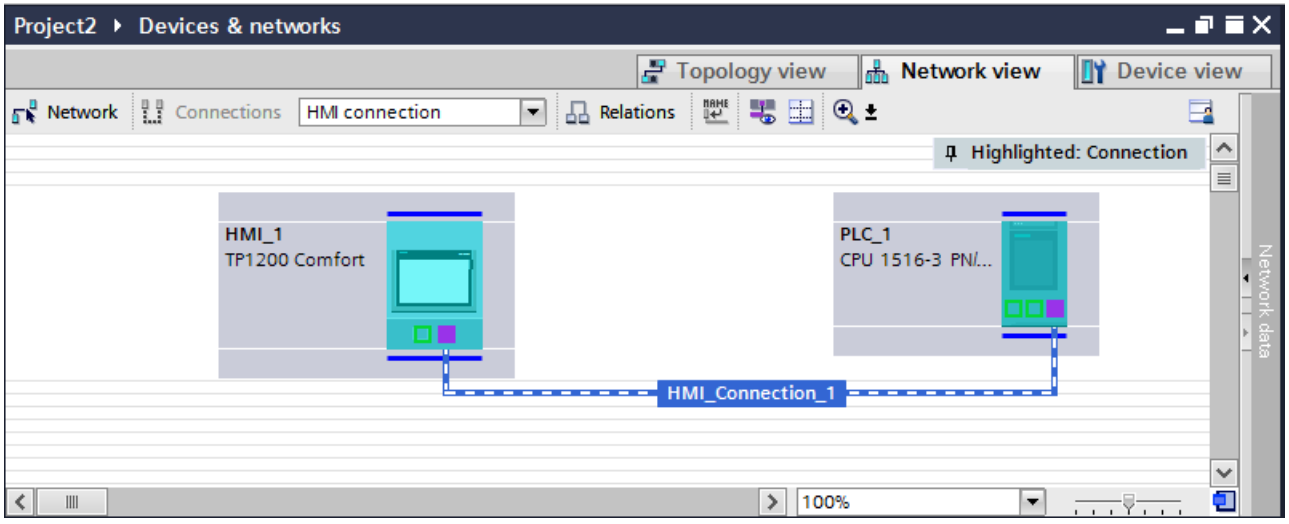
2.9.3 Communication via PROFIBUS

2.9.3.1 Configuring an HMI connection

Communication via PROFIBUS

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.

Configuring an HMI connection via PROFIBUS

Introduction

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

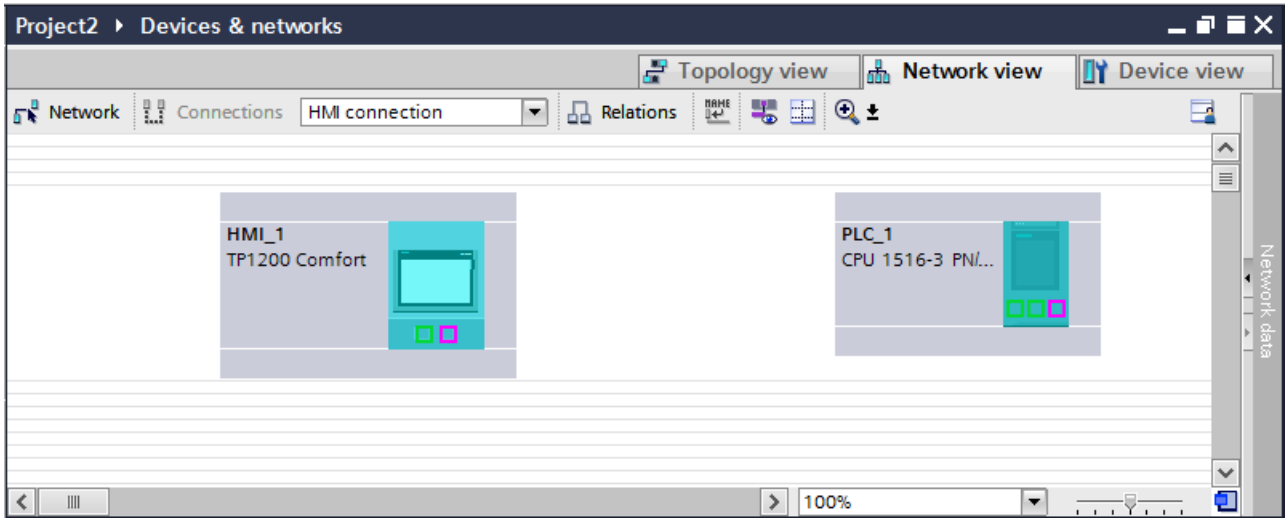
Requirements

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

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

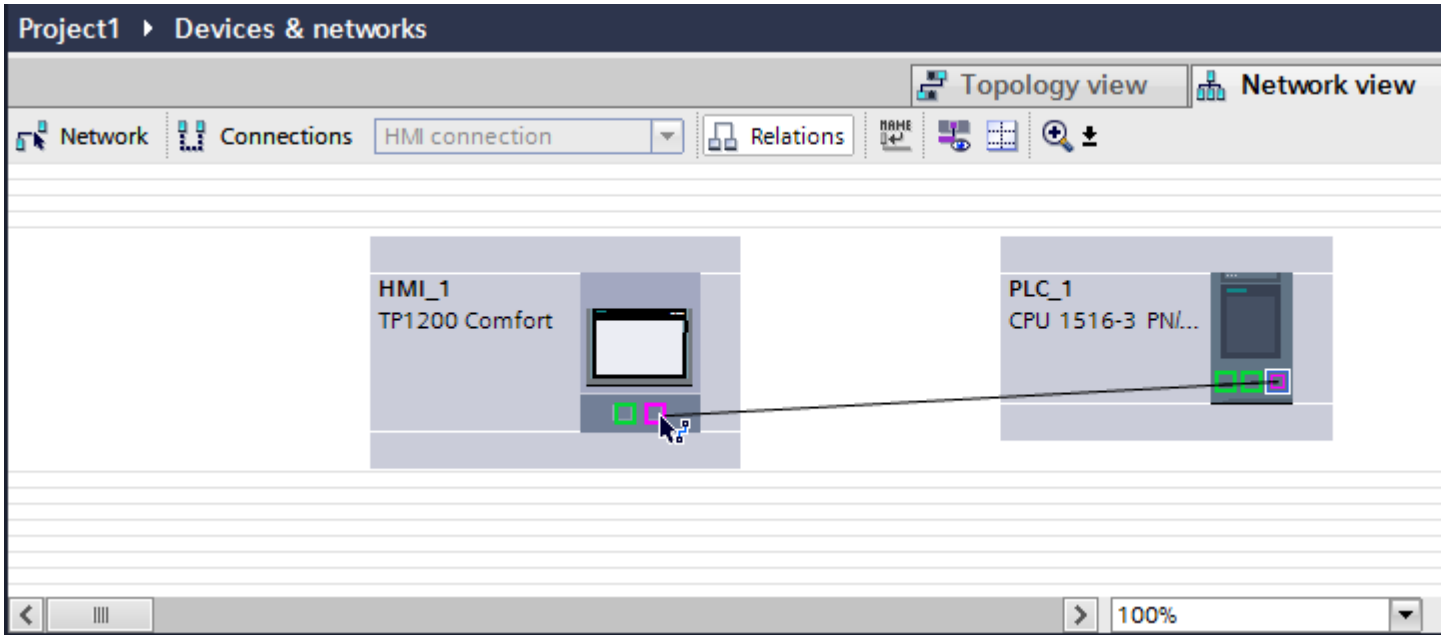
Procedure

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 PROFINET parameters in the Inspector window according to the requirements of your project.
   See the "PROFIBUS parameters (Page 613)" 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 over PROFIBUS between an HMI device and a SIMATIC S7 1500.

See also

PROFIBUS parameters (Page 613)

2.9.3.2 Configuring an HMI connection

Communication via PROFIBUS

Communication via PROFIBUS

This section describes the communication between a WinCC Runtime and the SIMATIC S7 1500 controller via PROFIBUS.

You can use the following WinCC Runtimes as an HMI device:

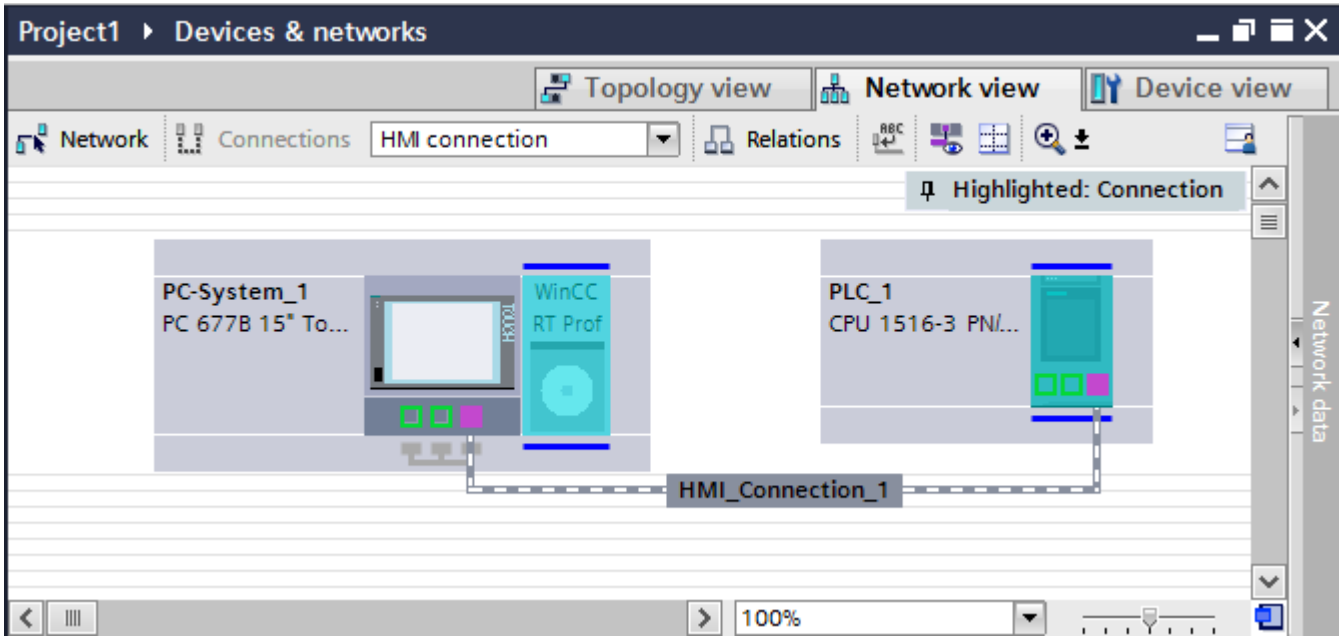
- WinCC RT Advanced
- WinCC RT Professional

WinCC Runtime as HMI device

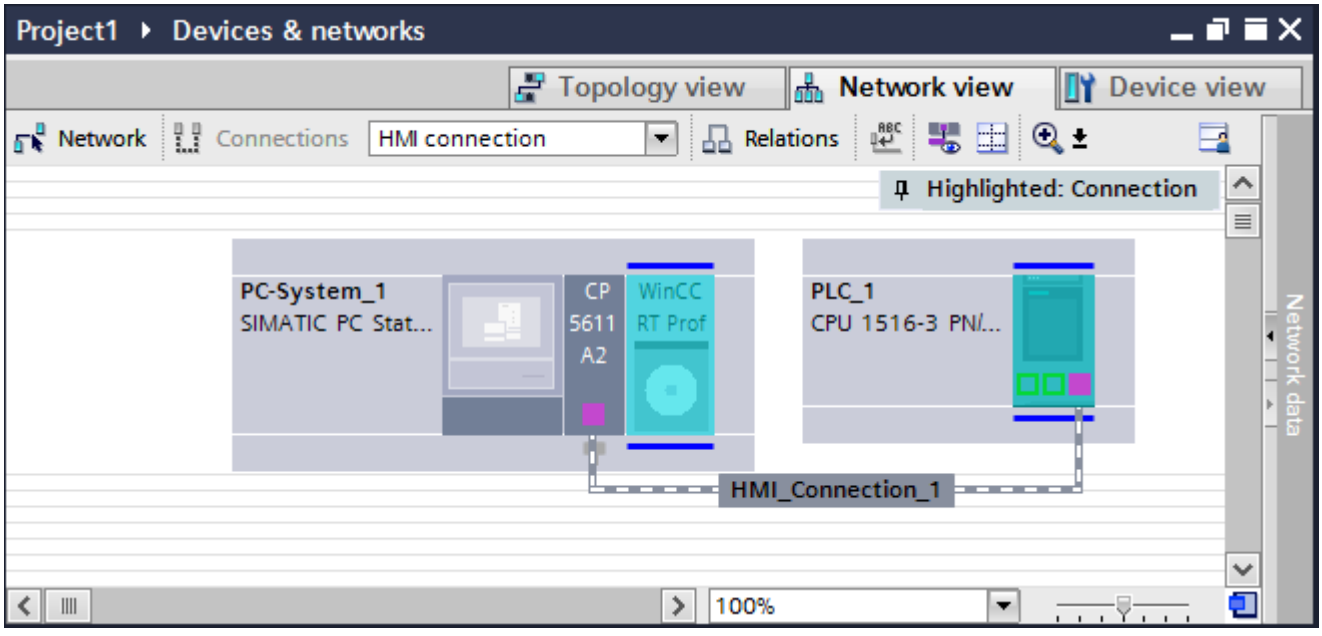
You configure the HMI connections between WinCC Runtime and SIMATIC S7 1500 in the "Devices & Networks" editor.

If you want to use a WinCC Runtime as an HMI device, you have the following options:

1. You create a SIMATIC PC and configure a WinCC Runtime on this SIMATIC PC.
   In this way you use the SIMATIC PC with a WinCC Runtime as an HMI device.



2. You create a WinCC Runtime and configure a communication processor to the Runtime.
   In this way you use your configuration PC with a WinCC Runtime as an HMI device.



You can also connect multiple HMI devices to one SIMATIC S7 1500 and multiple SIMATIC S7 1500 controllers 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 PROFIBUS in the "Devices & Networks" editor.

Connection in the "Connections" editor

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

Configuring an HMI connection via PROFIBUS with a PC

Introduction

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

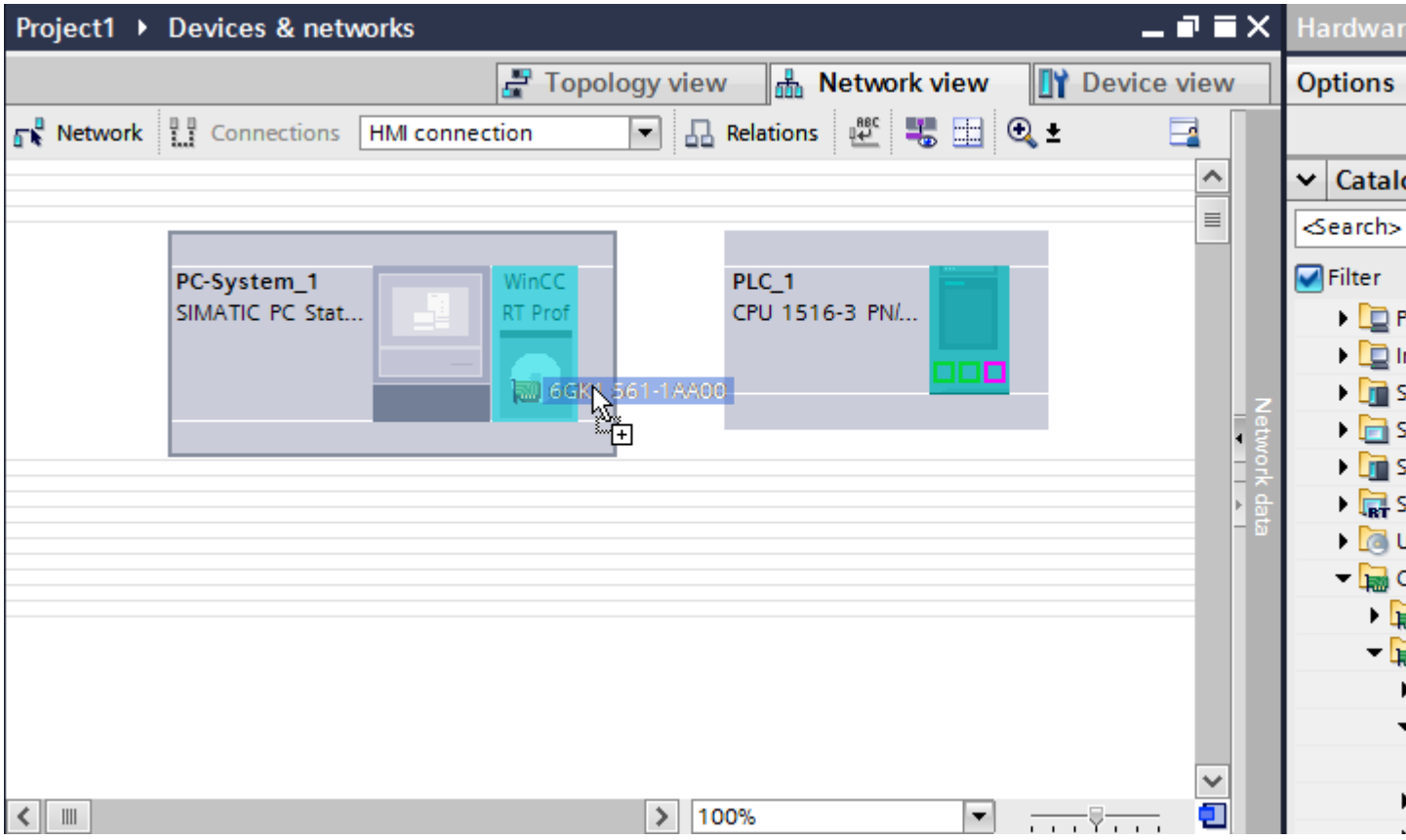
Requirements

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

- SIMATIC S7 1500 with PROFIBUS interface
- PC station with WinCC RT Advanced or WinCC RT Professional

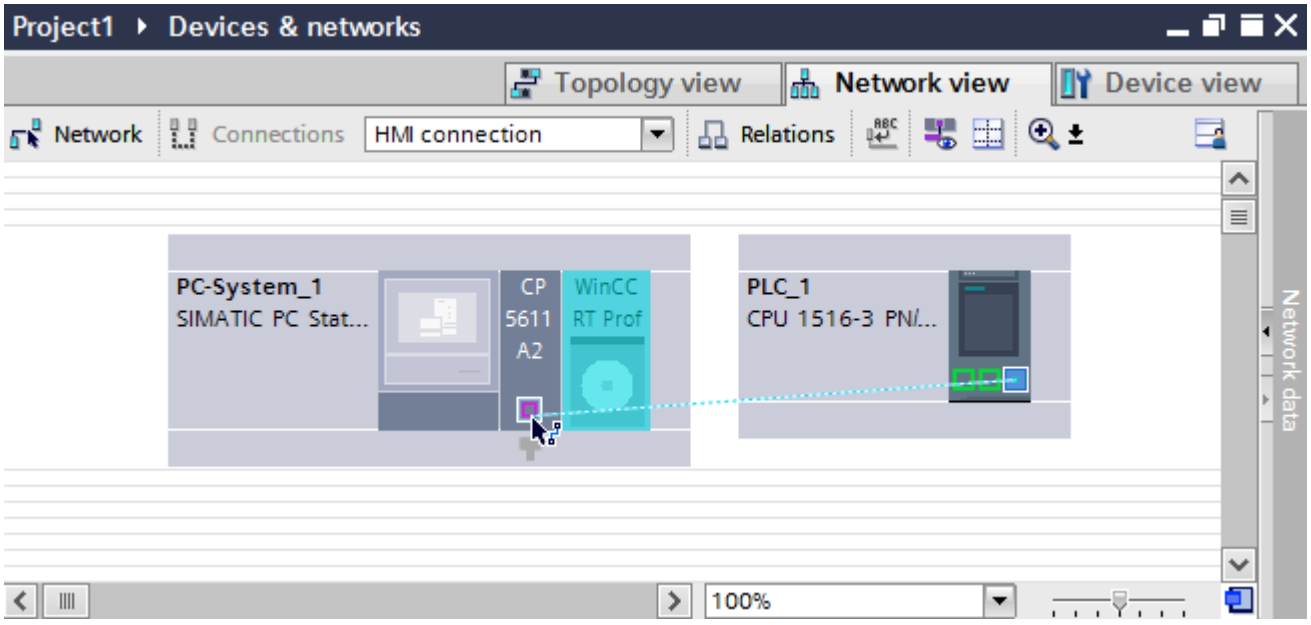
Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed in the network view.
2. Use a drag-and-drop operation to move a PROFIBUS-capable communication processor from the hardware catalog to the WinCC Runtime.



3. Click the "Connections" button and select "HMI connection" for the connection type.
   The devices available for connection are highlighted in color.

4. Click the PROFIBUS interface of the PLC and use a drag-and-drop operation to draw a connection to the PROFIBUS interface of the communication processor.



5. Click the connecting line.
6. Click "Highlight HMI connection" and select the HMI connection.
   The connection is displayed graphically in the Inspector window.
7. 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 613)" 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 1500 via PROFIBUS.

See also

PROFIBUS parameters (Page 613)

Configuring an HMI connection via PROFIBUS with a SIMATIC PC

Introduction

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

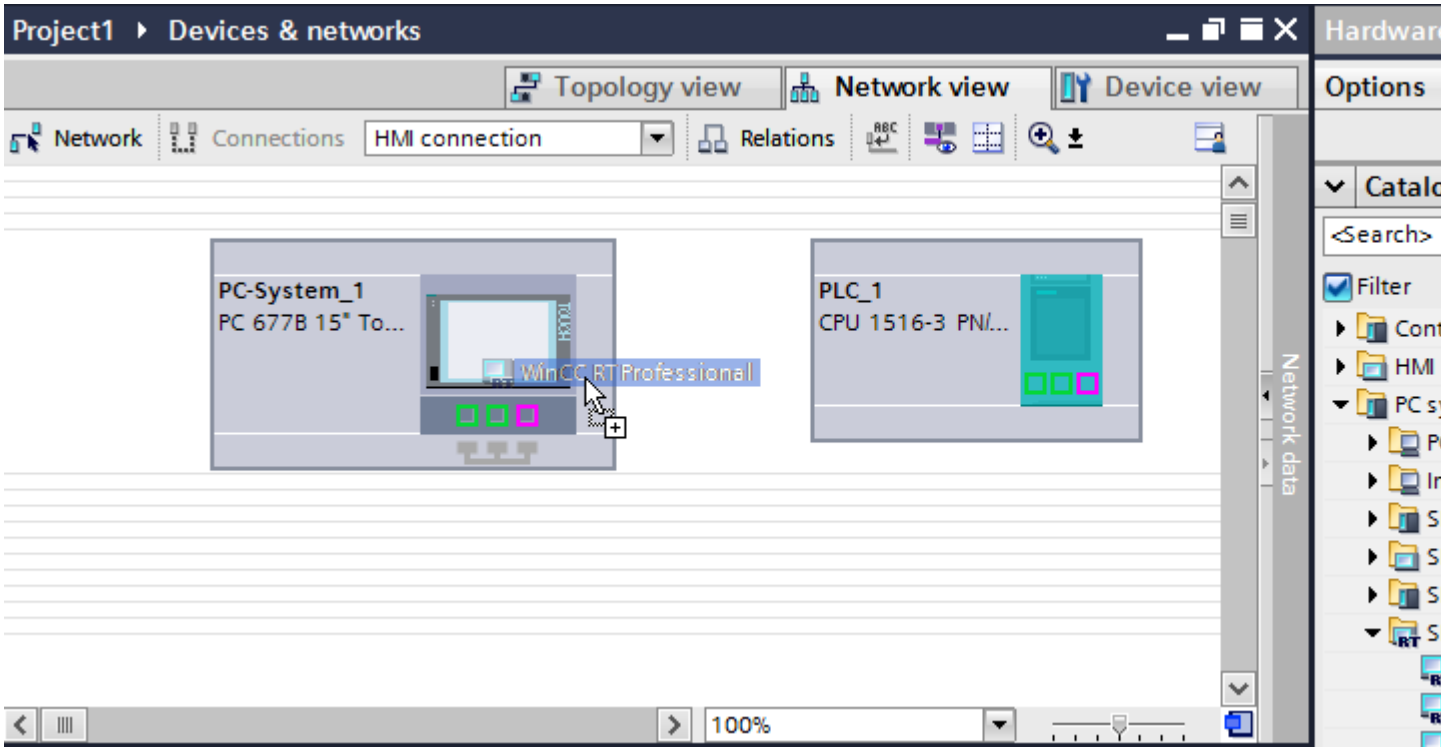
Requirements

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

- SIMATIC S7 1500 with PROFIBUS interface
- SIMATIC PC with PROFIBUS interface

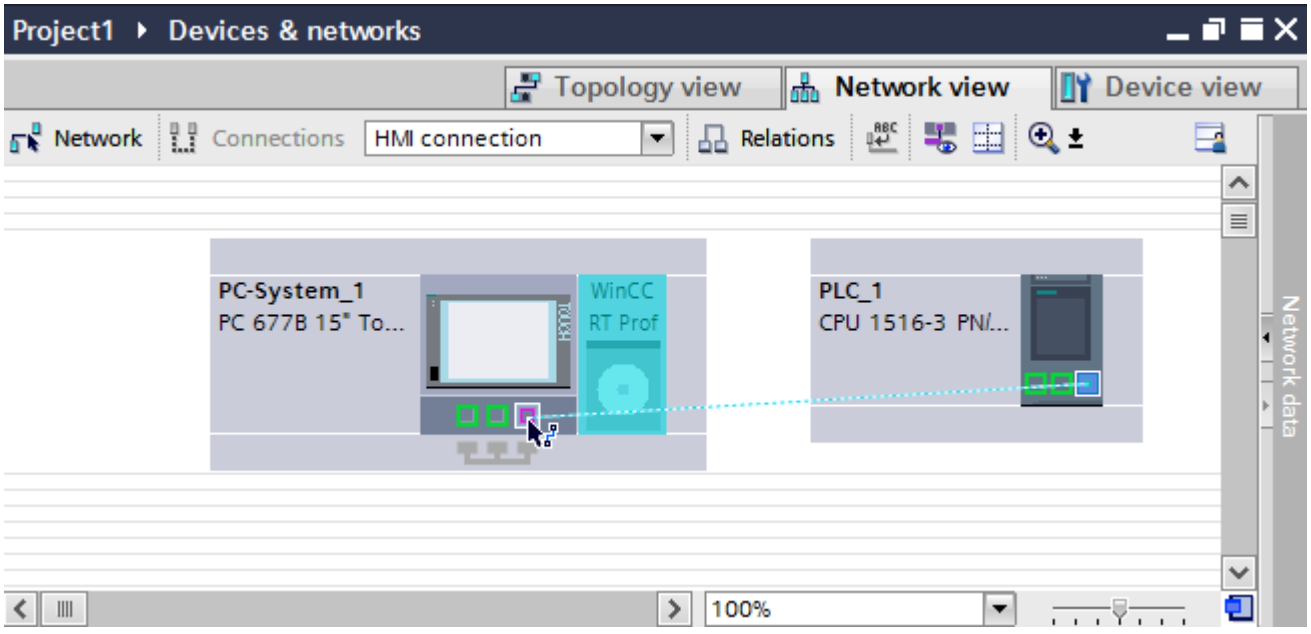
Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed in the network view.
2. Click the MPI interface of the PC and select "PROFIBUS" for the interface type in the Inspector window.
3. Use a drag-and-drop operation to move a WinCC Runtime from the hardware catalog to the PC.



4. Click the "Connections" button and select "HMI connection" for the connection type.
   The devices available for connection are highlighted in color.

5. Click the PROFIBUS interface of the PLC and use a drag-and-drop operation to draw a connection to the PROFIBUS interface of the PC.



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 613)".

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 1500 via PROFIBUS.

See also

PROFIBUS parameters (Page 613)

2.9.3.3 PROFIBUS parameters

PROFIBUS parameters for the HMI connection

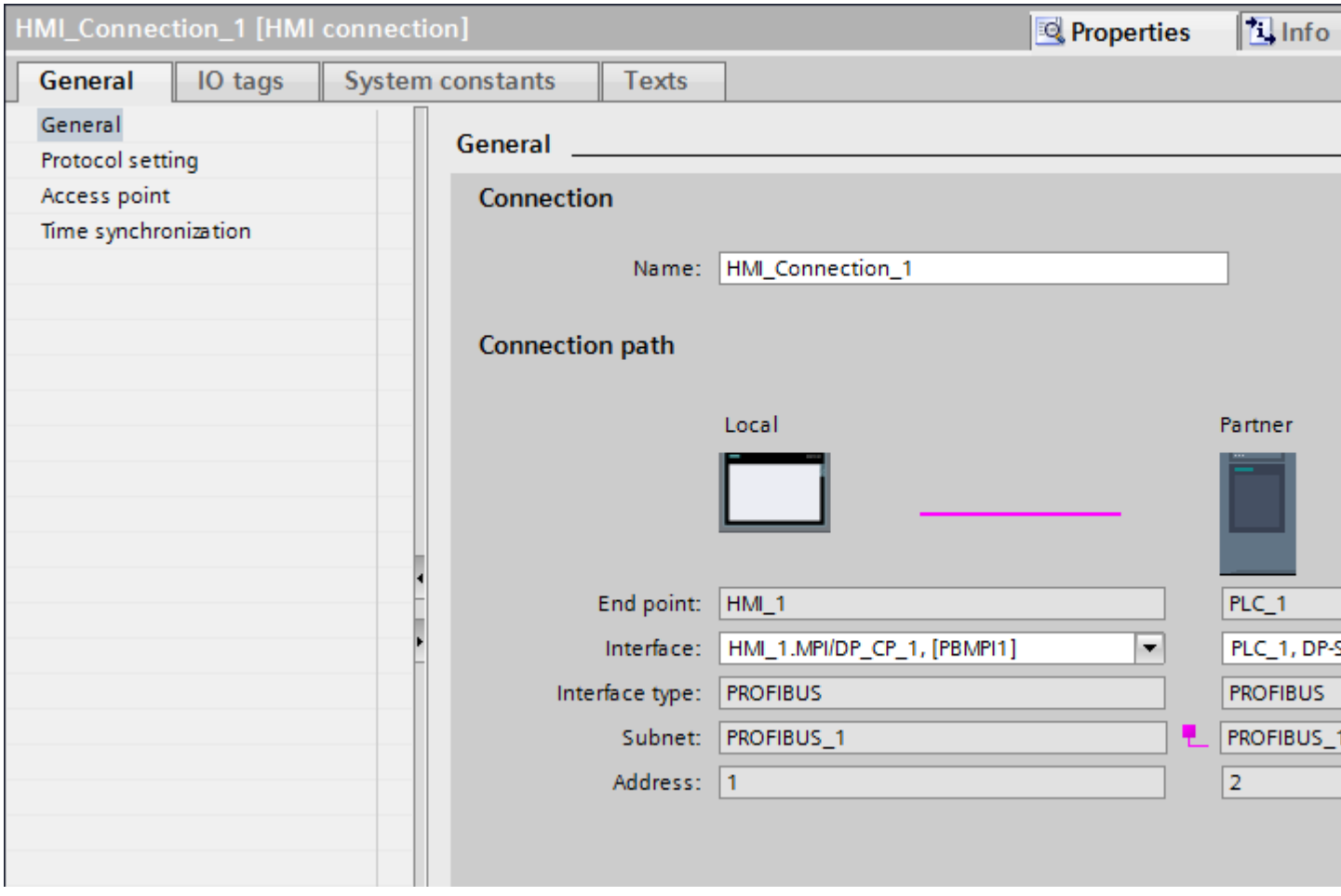
PROFIBUS parameters for the HMI connection

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.

Access password for the HMI connection

Introduction

Secure access to a PLC by assigning a password.

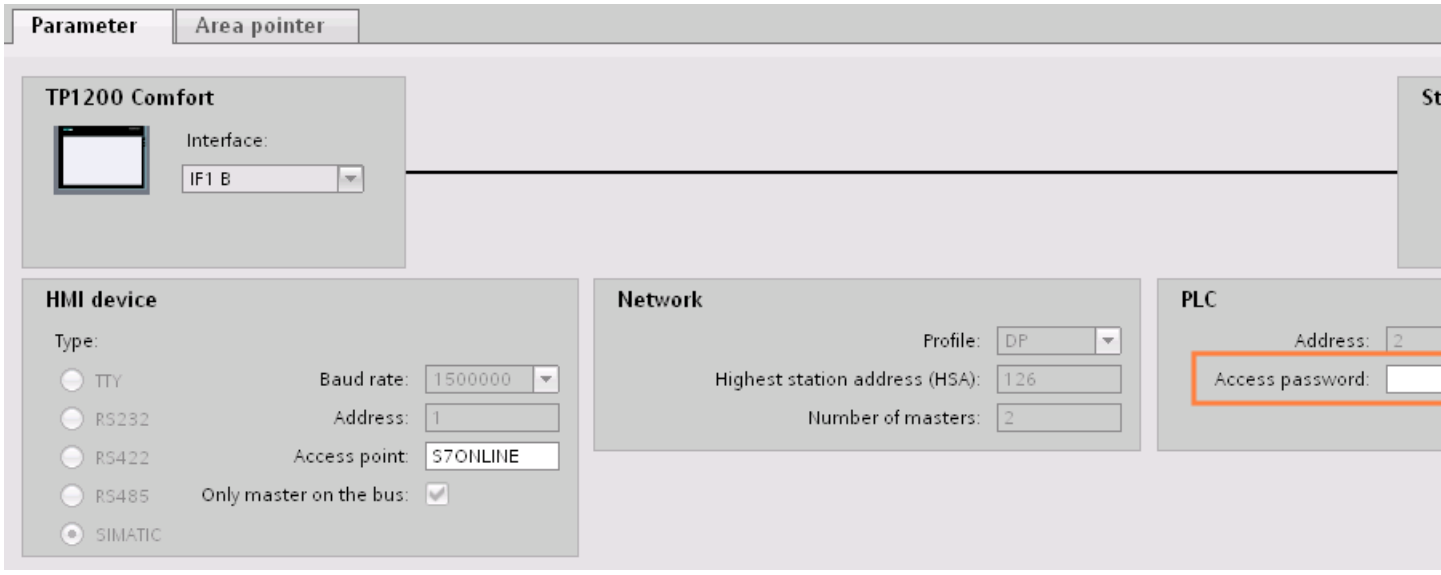
Assign the password when configuring the connection.

Input of the password of the PLC is mandatory as of "Complete protection" security level.

Communication to the PLC is denied if an incorrect password or no password is entered.

Assign password

Enter the "Access password" for the PLC in the "Connections" editor.



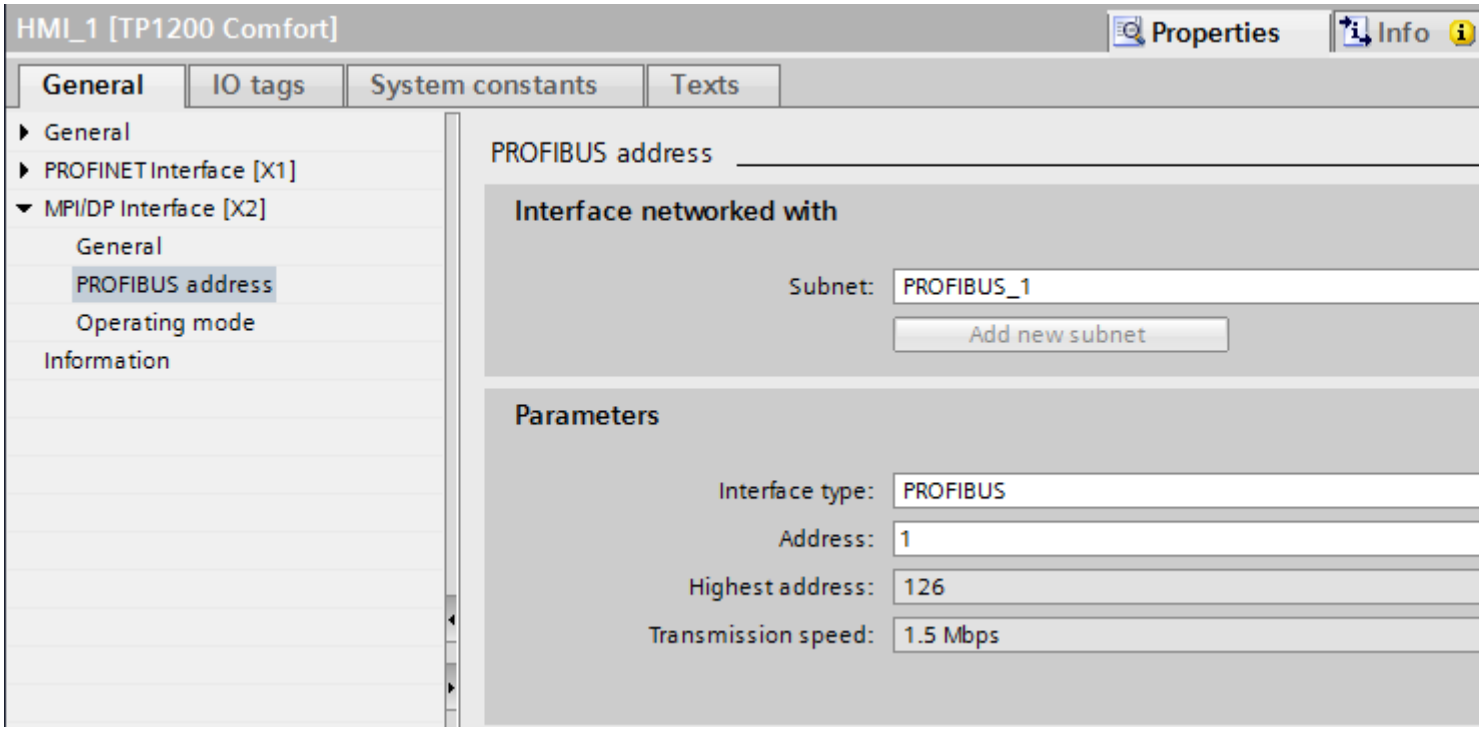
PROFIBUS parameters for the HMI device

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.

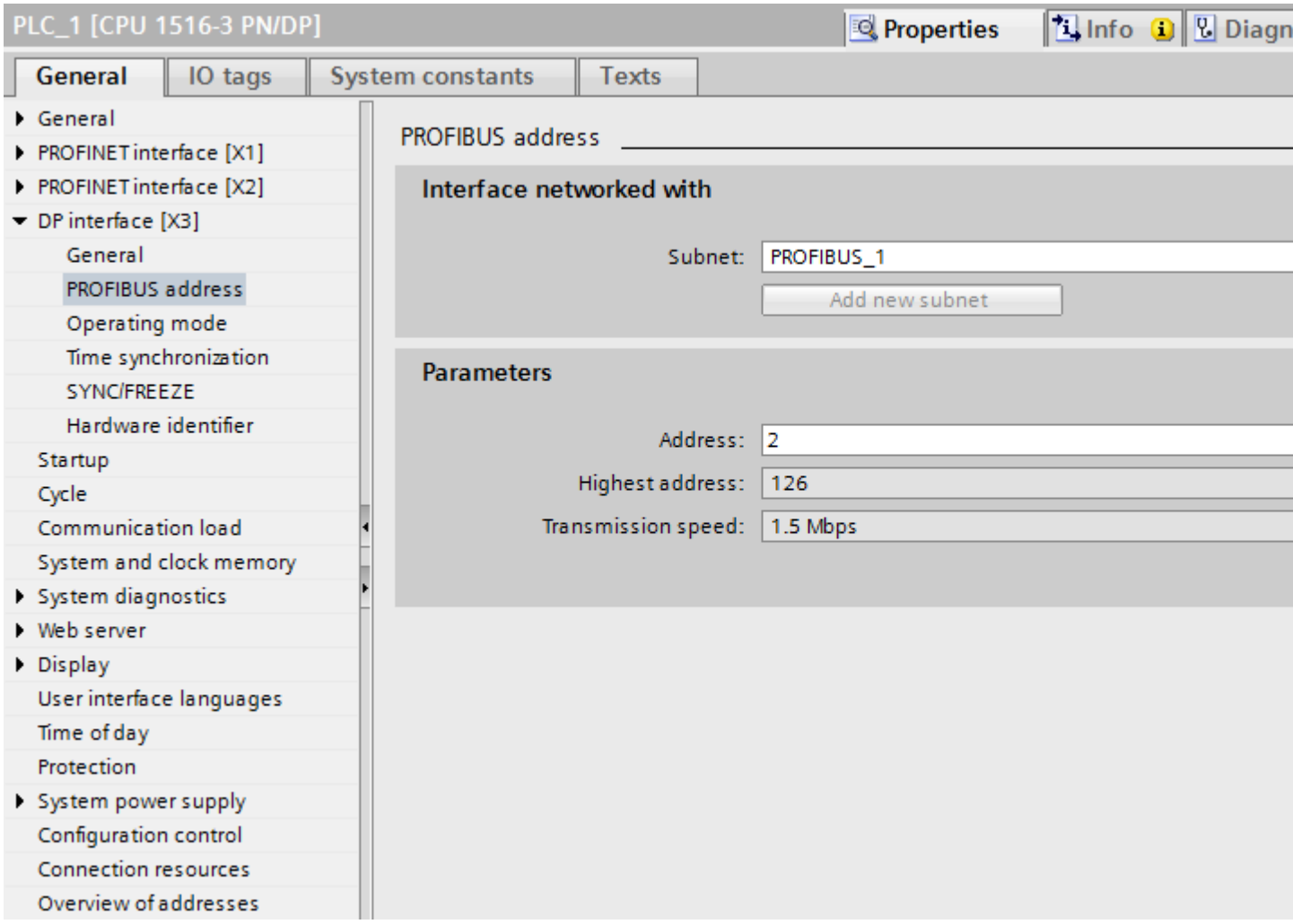
PROFIBUS parameters for the PLC

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.

Bus profiles with PROFIBUS

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 transfer speeds in Kbps
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.

Protection of communication

Security levels

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 620)

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 620)

Setting options for the protection

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.

Access password for the HMI connection

Introduction

Secure access to a PLC by assigning a password.

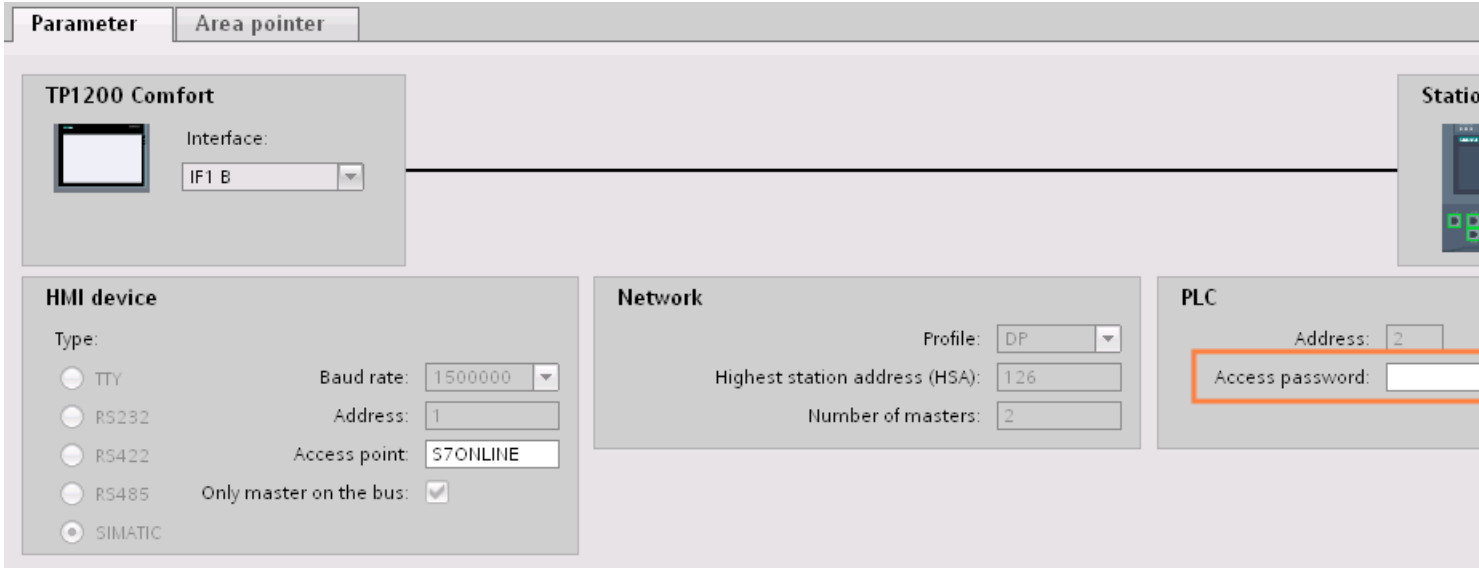
Assign the password when configuring the connection.

Input of the password of the PLC is mandatory as of "Complete protection" security level.

Communication to the PLC is denied if an incorrect password or no password is entered.

Assign password

Enter the "Access password" for the PLC in the "Connections" editor.



2.9.4 Data exchange

2.9.4.1 Data exchange using area pointers

General information on area pointers

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 153)

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

Communication with SIMATIC S7-1500 Software Controller (Page 582)

Configuring area pointers (Page 153)

"Screen number" area pointer (Page 625)

"Date/time" area pointer (Page 626)

"Date/time PLC" area pointer (Page 628)

"Coordination" area pointer (Page 630)

"Project ID" area pointer (Page 631)

"Job mailbox" area pointer (Page 632)

"Screen number" area pointer

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

Communication with SIMATIC S7-1500 Software Controller (Page 582)

General information on area pointers (Page 624)

"Date/time" area pointer (Page 626)

"Date/time PLC" area pointer (Page 628)

"Coordination" area pointer (Page 630)

"Project ID" area pointer (Page 631)

"Job mailbox" area pointer (Page 632)

"Date/time" area pointer

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

Communication with SIMATIC S7-1500 Software Controller (Page 582)

General information on area pointers (Page 624)

"Screen number" area pointer (Page 625)

"Date/time PLC" area pointer (Page 628)

"Coordination" area pointer (Page 630)

"Project ID" area pointer (Page 631)

"Job mailbox" area pointer (Page 632)

"Date/time PLC" area pointer

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

Communication with SIMATIC S7-1500 Software Controller (Page 582)

General information on area pointers (Page 624)

"Screen number" area pointer (Page 625)

"Date/time" area pointer (Page 626)

"Coordination" area pointer (Page 630)

"Project ID" area pointer (Page 631)

"Job mailbox" area pointer (Page 632)

"Coordination" area pointer

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

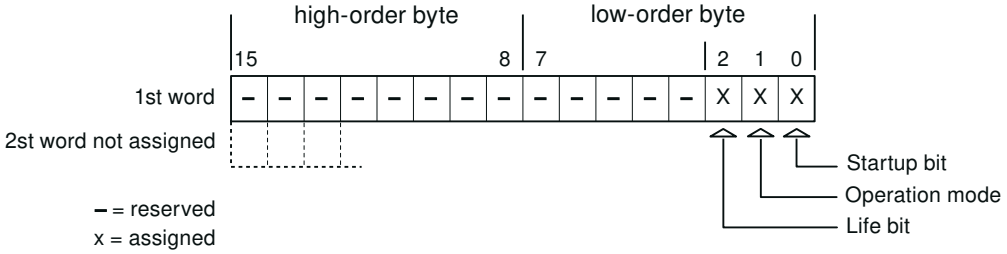
By default, the "Coordination" area pointer has the length of one word and cannot be changed.

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

Communication with SIMATIC S7-1500 Software Controller (Page 582)

General information on area pointers (Page 624)

"Screen number" area pointer (Page 625)

"Date/time" area pointer (Page 626)

"Date/time PLC" area pointer (Page 628)

"Project ID" area pointer (Page 631)

"Job mailbox" area pointer (Page 632)

"Project ID" area pointer

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

Communication with SIMATIC S7-1500 Software Controller (Page 582)

General information on area pointers (Page 624)

"Screen number" area pointer (Page 625)

"Date/time" area pointer (Page 626)

"Date/time PLC" area pointer (Page 628)

"Coordination" area pointer (Page 630)

"Job mailbox" area pointer (Page 632)

"Job mailbox" area pointer

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 to 23)
	Parameter 2	Left byte: Minutes (0-59) Right byte: Seconds (0 to 59)
	Parameter 3	-
15	**Setting the date (BCD coded)** [2) 3)]	
	Parameter 1	Left byte: - Right byte: Weekday (1-7: Sunday-Saturday)
	Parameter 2	Left byte: Day (1-31) Right byte: Month (1 to 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**	

No.	Function	
14	**Set time (BCD-coded)**	
	(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**	
	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	**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**	
	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]	The weekday is ignored on HMI device KTP 600 BASIC PN.
[3]	The weekday is ignored when you configure the "Date/Time PLC" area pointer.

See also

Communication with SIMATIC S7-1500 Software Controller (Page 582)
General information on area pointers (Page 624)

"Screen number" area pointer (Page 625)

"Date/time" area pointer (Page 626)

"Date/time PLC" area pointer (Page 628)

"Coordination" area pointer (Page 630)

"Project ID" area pointer (Page 631)

"Data record" area pointer

"Data record" area pointer

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

Communication with SIMATIC S7-1500 Software Controller (Page 582)

Transfer without synchronization (Page 636)

Transfer with synchronization (Page 637)

Transfer started by operator in the recipe view (Page 638)

Sequence of transfer triggered by a configured function (Page 639)

Transfer with job mailbox (Page 640)

Possible causes of error when transferring data records (Page 642)

Transfer without synchronization

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

Communication with SIMATIC S7-1500 Software Controller (Page 582)

"Data record" area pointer (Page 635)

Transfer with synchronization (Page 637)

Transfer started by operator in the recipe view (Page 638)

Sequence of transfer triggered by a configured function (Page 639)

Transfer with job mailbox (Page 640)

Possible causes of error when transferring data records (Page 642)

Transfer with synchronization

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

Value		Meaning
Decimal	Binary	
4	0000 0100	Transfer completed without error
12	0000 1100	Transfer completed with error

See also

Communication with SIMATIC S7-1500 Software Controller (Page 582)
"Data record" area pointer (Page 635)
Transfer without synchronization (Page 636)
Transfer started by operator in the recipe view (Page 638)
Sequence of transfer triggered by a configured function (Page 639)
Transfer with job mailbox (Page 640)
Possible causes of error when transferring data records (Page 642)

Transfer started by operator in the recipe view

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 must only ever be set by the HMI device. The PLC can only reset the status word to zero.

For Comfort Panel and Runtime Advanced, the data records are transferred even if the status word is set to 2.

Note

The PLC can only evaluate the recipe and data record number 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

Communication with SIMATIC S7-1500 Software Controller (Page 582)

"Data record" area pointer (Page 635)

Transfer without synchronization (Page 636)

Transfer with synchronization (Page 637)

Sequence of transfer triggered by a configured function (Page 639)

Transfer with job mailbox (Page 640)

Possible causes of error when transferring data records (Page 642)

Sequence of transfer triggered by a configured function

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

Communication with SIMATIC S7-1500 Software Controller (Page 582)

"Data record" area pointer (Page 635)

Transfer without synchronization (Page 636)

Transfer with synchronization (Page 637)

Transfer started by operator in the recipe view (Page 638)

Transfer with job mailbox (Page 640)

Possible causes of error when transferring data records (Page 642)

Transfer with job mailbox

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 (1-65,535)	
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

Step	Action	
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

Communication with SIMATIC S7-1500 Software Controller (Page 582)

"Data record" area pointer (Page 635)

Transfer without synchronization (Page 636)

Transfer with synchronization (Page 637)

Transfer started by operator in the recipe view (Page 638)

Sequence of transfer triggered by a configured function (Page 639)

Possible causes of error when transferring data records (Page 642)

Possible causes of error when transferring data records

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

Communication with SIMATIC S7-1500 Software Controller (Page 582)

"Data record" area pointer (Page 635)

Transfer without synchronization (Page 636)

Transfer with synchronization (Page 637)

Transfer started by operator in the recipe view (Page 638)

Sequence of transfer triggered by a configured function (Page 639)

Transfer with job mailbox (Page 640)

2.9.4.2 Trends

General information on trends

Trends

A trend is the graphical representation of one or more values from the PLC. The value is read out either time- or bit-triggered, depending on the configuration.

For additional information see:

AUTOHOTSPOT

Note

The value is read out time-triggered for Basic Panels.

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.

Bit-triggered trends

Through a trigger bit set in the trend request tag, the HMI device either reads in a trend value or an entire trend buffer. This setting is defined in the configuration. Bit-triggered trends are normally used to represent fast 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 "HMI tags" editor and connected to trend areas during configuration. The HMI device and PLC then communicate with each other via 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)

See also

Communication with SIMATIC S7-1500 Software Controller (Page 582)

Trend request and trend transfer (Page 644)

Permitted data types for trends (Page 645)

Trend request and trend transfer

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 closing the screen, the HMI device resets the relevant bits in the trend request area.

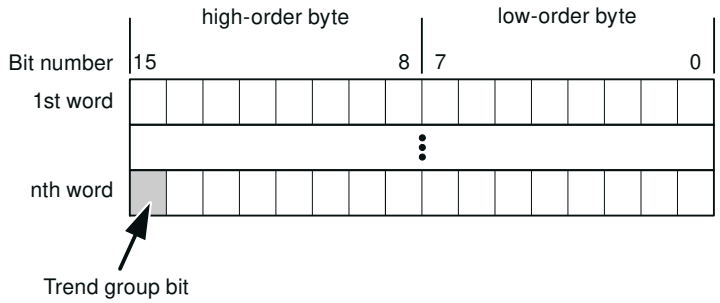
The trend request area can be used for evaluation purposes in the PLC to determine which trend is currently being displayed on the HMI device. Trends can also be triggered without evaluation of the trend request area.

Trend transfer area 1

This area is used to trigger trends. You must set the bit assigned to the trend in the trend transfer area and set the trend group bit in your control program. 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.

The PLC writes to Buffer 2 while the HMI device reads values from Buffer 1, and writes to Buffer 1 when the HMI device is reading Buffer 2. This prevents the PLC from overwriting trend values while the trend is being read by the HMI device.

See also

Communication with SIMATIC S7-1500 Software Controller (Page 582)

General information on trends (Page 643)

Permitted data types for trends (Page 645)

Permitted data types for trends

For SIMATIC S7

You assign one bit to each trend during configuration. Tags and array tags of the "Word" or "Int" data type are permitted.

See also

Communication with SIMATIC S7-1500 Software Controller (Page 582)

General information on trends (Page 643)

Trend request and trend transfer (Page 644)

2.9.4.3 Alarms

Configuring alarms

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

Communication with SIMATIC S7-1500 Software Controller (Page 582)

Acknowledgment of alarms (Page 647)

Acknowledgment of alarms

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

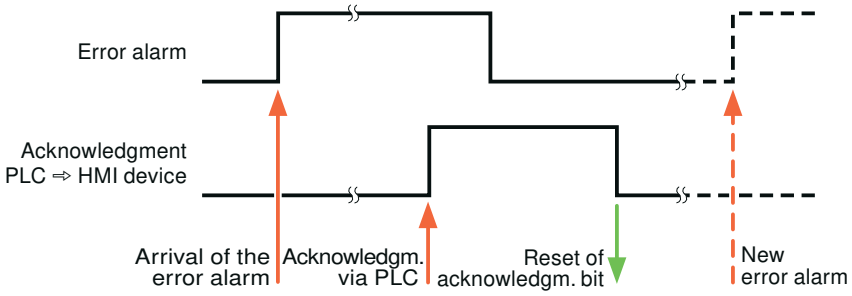
Acknowledgment by the PLC

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.

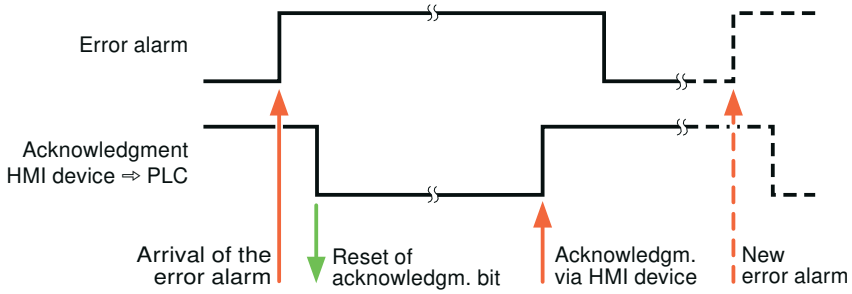
If a discrete alarm subject to acknowledgment is acknowledged by the HMI device, then the corresponding bit in the assigned acknowledgment tag will be set. The entire acknowledgment tag is then written to the PLC by the HMI device. This allows the PLC to recognize that a certain alarm message has been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

Communication with SIMATIC S7-1500 Software Controller (Page 582)

Configuring alarms (Page 646)

2.9.4.4 LED mapping

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 Comfort Panels	Panel PCs
0	0	Off	Off
0	1	Rapid flashing	Flashing
1	0	Slow flashing	Flashing
1	1	On permanently	On permanently

See also

Communication with SIMATIC S7-1500 Software Controller (Page 582)

2.9.5 Performance features of communication

2.9.5.1 Device dependencies of the SIMATIC S7-1500 software controller

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 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

HMI devices	SIMATIC S7-1500 Software Controller
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

Basic Panels V14.0.1

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 V15.0

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

Basic Panels V15.1

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

Mobile Panels V12.0

HMI devices	SIMATIC S7-1500 Software Controller
Mobile Panel 177 6" DP	No
Mobile Panel 177 6" PN	No
Mobile Panel 277 8"	No
Mobile Panel 277 8" IWLAN V2	No
Mobile Panel 277F 8" IWLAN V2	No
Mobile Panel 277F 8" IWLAN (RFID tag)	No
Mobile Panel 277 10"	No

Mobile Panels V13.0.1

HMI devices	SIMATIC S7-1500 Software Controller
KTP700 Mobile	Yes
KTP700F Mobile	Yes
KTP900 Mobile	Yes
KTP900F Mobile	Yes

Mobile Panels V14.0

HMI devices	SIMATIC S7-1500 Software Controller
KTP700 Mobile	Yes
KTP700F Mobile	Yes
KTP900 Mobile	Yes
KTP900F Mobile	Yes
KTP400F Mobile	Yes

Mobile Panels V14.0.1

HMI devices	SIMATIC S7-1500 Software Controller
KTP700 Mobile	Yes
KTP700F Mobile	Yes
KTP900 Mobile	Yes
KTP900F Mobile	Yes
KTP400F Mobile	Yes

Mobile Panels V15.0

HMI devices	SIMATIC S7-1500 Software Controller
KTP700 Mobile	Yes
KTP700F Mobile	Yes
KTP900 Mobile	Yes
KTP900F Mobile	Yes
KTP400F Mobile	Yes

Mobile Panels V15.1

HMI devices	SIMATIC S7-1500 Software Controller
KTP700 Mobile	Yes
KTP700F Mobile	Yes
KTP900 Mobile	Yes
KTP900F Mobile	Yes
KTP400F Mobile	Yes

Comfort Panels V12.0

HMI devices	SIMATIC S7-1500 Software Controller
KP400 Comfort	No
KTP400 Comfort	No
KTP400 Comfort Portrait	No

HMI devices	SIMATIC S7-1500 Software Controller
KP700 Comfort	No
TP700 Comfort	No
TP700 Comfort Portrait	No
KP900 Comfort	No
TP900 Comfort	No
TP900 Comfort Portrait	No
KP1200 Comfort	No
TP1200 Comfort	No
TP1200 Comfort Portrait	No
KP1500 Comfort	No
TP1500 Comfort	No
TP1500 Comfort Portrait	No
TP1900 Comfort	No
TP1900 Comfort Portrait	No
TP2200 Comfort	No
TP2200 Comfort Portrait	No

Comfort Panels V13.0

HMI devices	SIMATIC S7-1500 Software Controller
KP400 Comfort	No
KTP400 Comfort	No
KTP400 Comfort Portrait	No
KP700 Comfort	No
TP700 Comfort	No
TP700 Comfort Portrait	No
KP900 Comfort	No
TP900 Comfort	No
TP900 Comfort Portrait	No
KP1200 Comfort	No
TP1200 Comfort	No
TP1200 Comfort Portrait	No
KP1500 Comfort	No
TP1500 Comfort	No
TP1500 Comfort Portrait	No
TP1900 Comfort	No
TP1900 Comfort Portrait	No
TP2200 Comfort	No
TP2200 Comfort Portrait	No

Comfort Panels V13.0.1

HMI devices	SIMATIC S7-1500 Software Controller
KP400 Comfort	Yes
KTP400 Comfort	Yes
KTP400 Comfort Portrait	Yes
KP700 Comfort	Yes
TP700 Comfort	Yes
TP700 Comfort Outdoor	Yes
TP700 Comfort Outdoor Portrait	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
TP1500 Comfort Outdoor	Yes
TP1500 Comfort Outdoor Portrait	Yes
TP1900 Comfort	Yes
TP1900 Comfort Portrait	Yes
TP2200 Comfort	Yes
TP2200 Comfort Portrait	Yes

Comfort Panels V14.0

HMI devices	SIMATIC S7-1500 Software Controller
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

HMI devices	SIMATIC S7-1500 Software Controller
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

Comfort Panels V14.0.1

HMI devices	SIMATIC S7-1500 Software Controller
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

Comfort Panels V15.0

HMI devices	SIMATIC S7-1500 Software Controller
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

Comfort Panels V15.1

HMI devices	SIMATIC S7-1500 Software Controller
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

HMI devices	SIMATIC S7-1500 Software Controller
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 V12.0

HMI devices	SIMATIC S7-1500 Software Controller
WinCC RT Advanced	Yes

Runtime V13.0

HMI devices	SIMATIC S7-1500 Software Controller
WinCC RT Advanced	Yes

Runtime V13.0.1

HMI devices	SIMATIC S7-1500 Software Controller
WinCC RT Advanced	Yes

Runtime V14.0

HMI devices	SIMATIC S7-1500 Software Controller
WinCC RT Advanced	Yes

Runtime V14.0.1

HMI devices	SIMATIC S7-1500 Software Controller
WinCC RT Advanced	Yes

Runtime V15.0

HMI devices	SIMATIC S7-1500 Software Controller
WinCC RT Advanced	Yes

Runtime V15.1

HMI devices	SIMATIC S7-1500 Software Controller
WinCC RT Advanced	Yes

See also

Communication with SIMATIC S7-1500 Software Controller (Page 582)
Permitted data types (Page 658)

2.9.5.2 Permitted data types

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 Software Controller (Page 582)

Device dependencies of the SIMATIC S7-1500 software controller (Page 649)

2.9.6 Configuring connections in the "Connections" editor

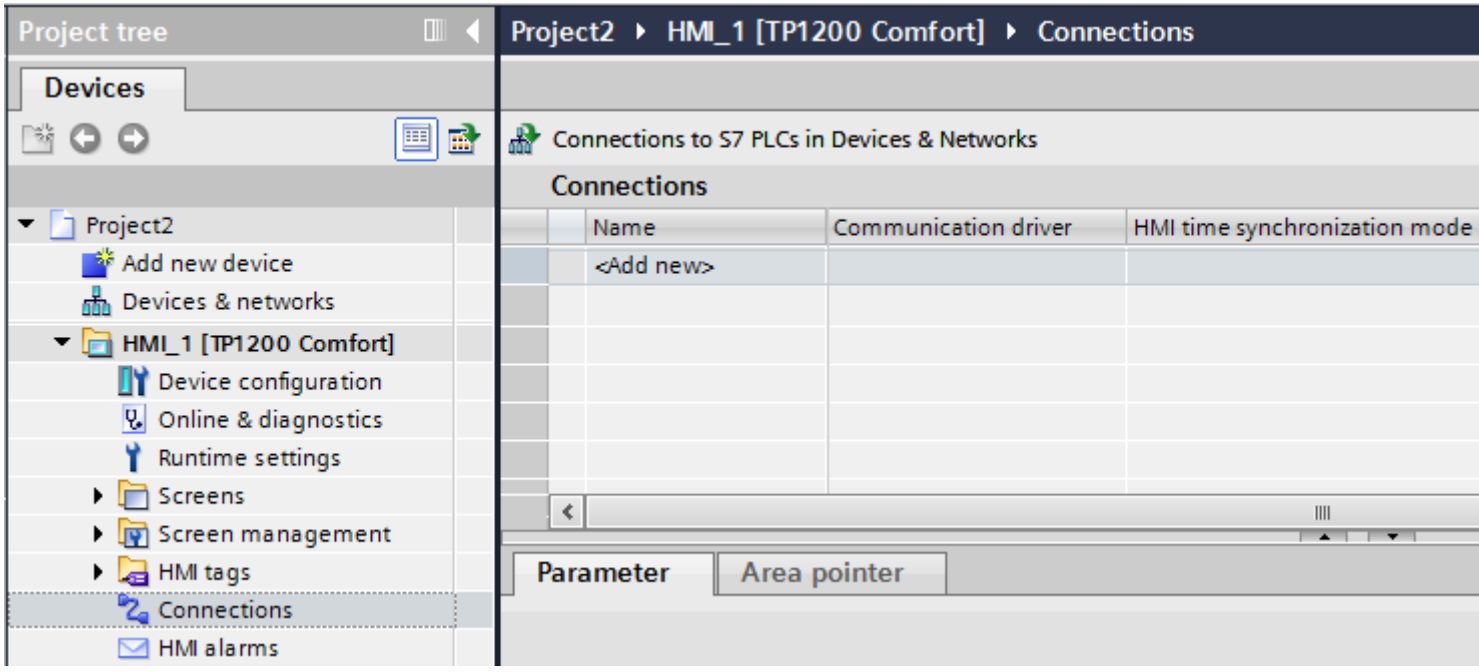
2.9.6.1 Creating a PROFINET connection

Requirements

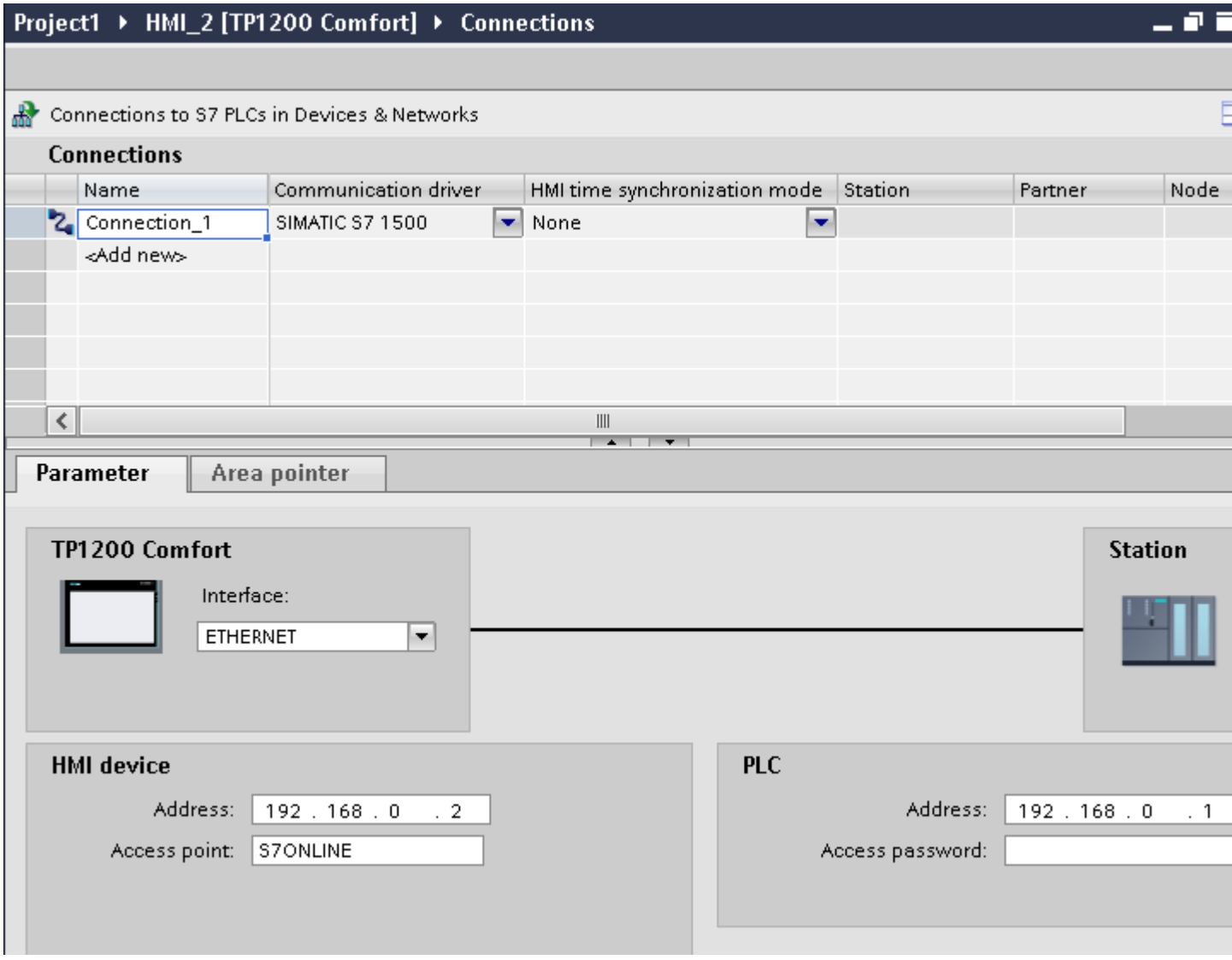
- A project is open.
- An HMI device with a PROFINET interface has been created.

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 with SIMATIC S7-1500 Software Controller (Page 582)

2.9.7 Configuring connections in the "Connections" editor

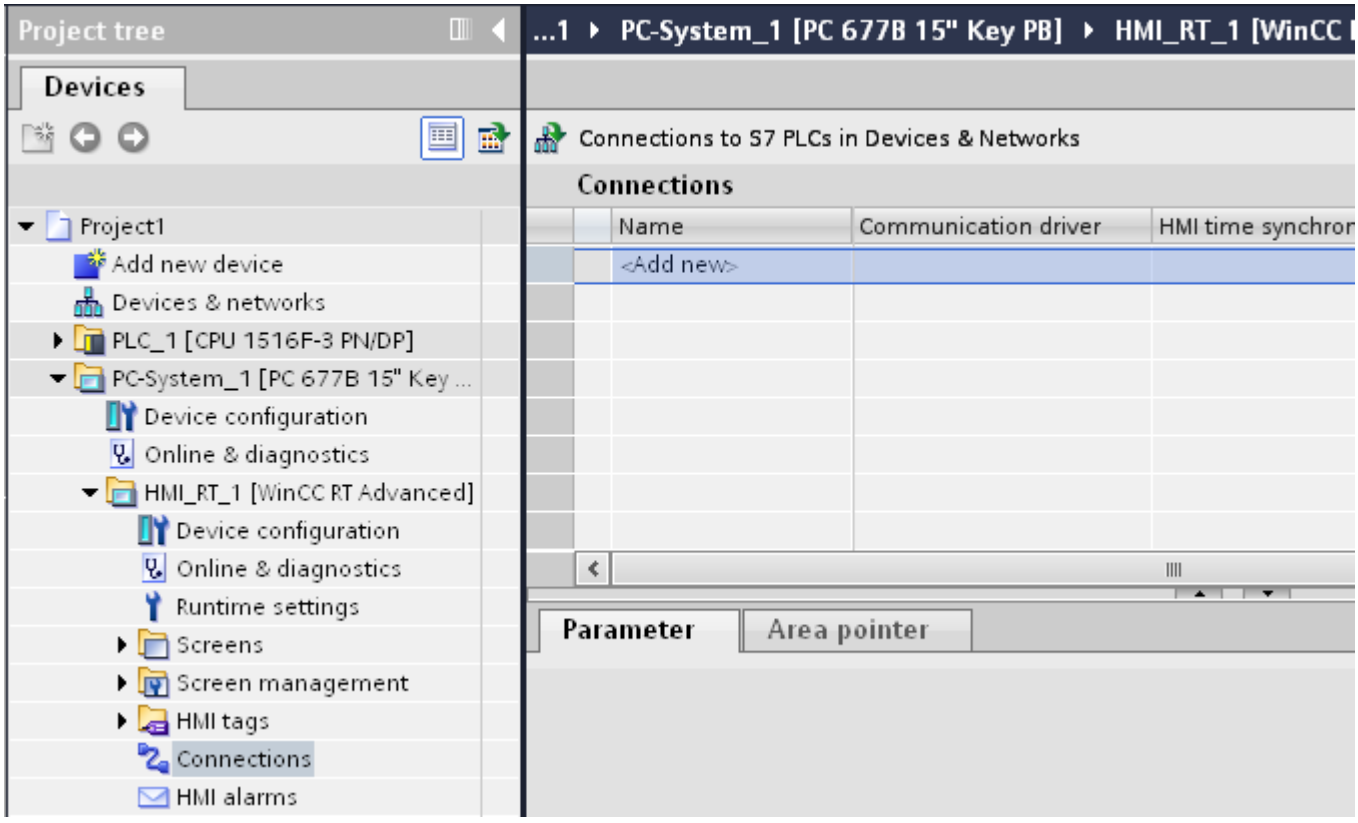
2.9.7.1 Creating a connection

Requirements

- A project is open.
- WinCC RT Advanced is 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 an interface of the HMI device in the Inspector window under "Parameters > Interface".
6. Set the parameters for the connection in the Inspector window.

Interfaces

Select one of the following interfaces in the Inspector window under "Parameters > WinCC RT Advanced > Interfaces".:

- Industrial Ethernet
- MPI
- PROFIBUS

For additional information on the parameters of the interfaces see:

Connection parameters (Page 663)

See also

Communication with SIMATIC S7-1500 Software Controller (Page 582)

Connection parameters (Page 663)

2.9.7.2 Connection parameters

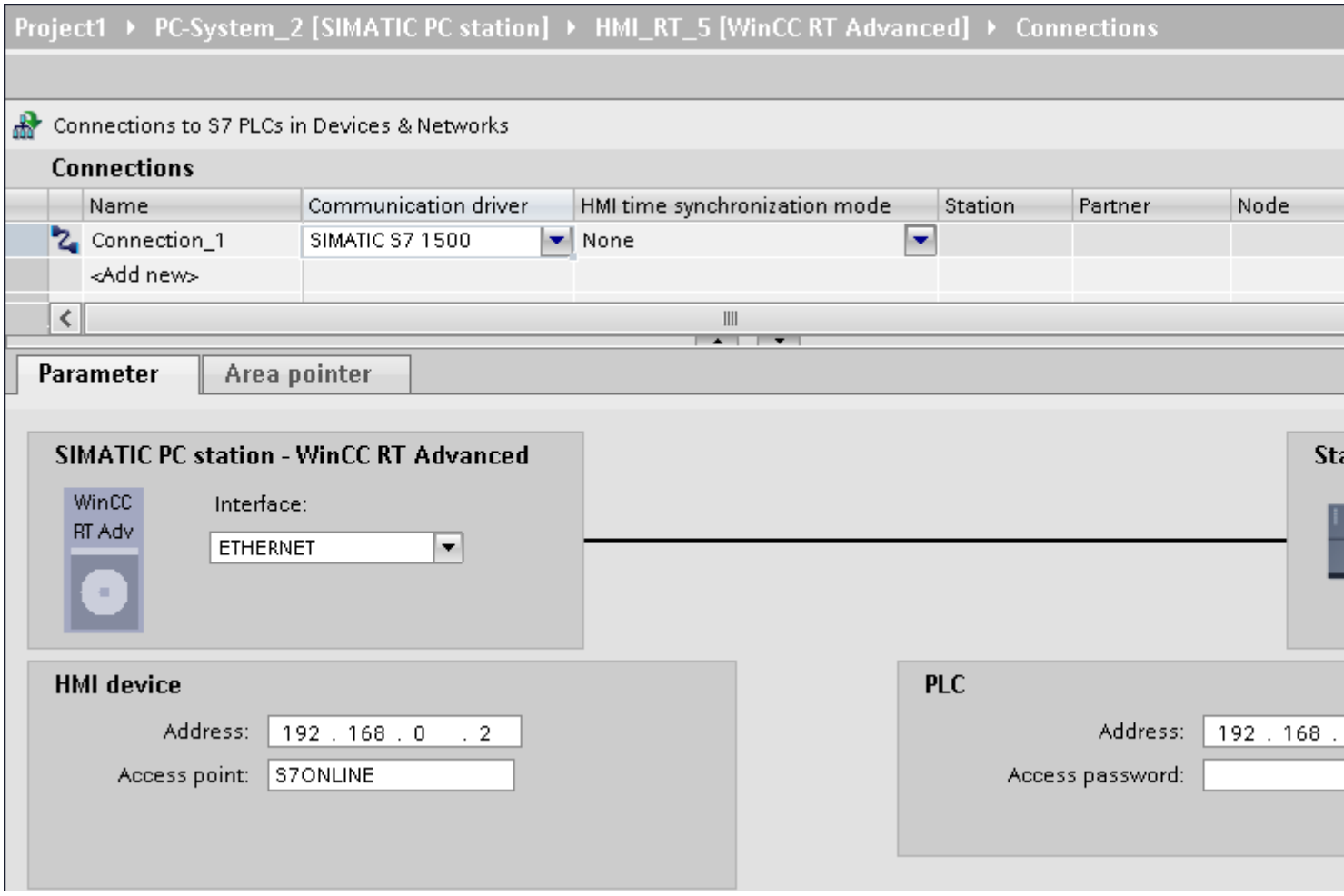
ETHERNET

Introduction

You assign the parameters for the HMI device and the controller in the network under "Parameters".

The parameters described in the following apply to the following interfaces:

- TCP/IP



HMI device

- "Address"
  Enter the IP address of the HMI device under "Address".
- "Access point"
  Specifies the access point for the PG/PC interface that can be used to reach the communication partner.

Controller

- "Address"
  Enter the IP address of the controller under "Address".
- "Access password"
  Enter a password in the "Access password" field. This password must match the one you saved to the controller.

Note

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

No connection to the controller is established if the "Complete protection" protection level is stored on the controller and you do not enter a password.

See also

Communication with SIMATIC S7-1500 Software Controller (Page 582)

Creating a connection (Page 662)

2.9.8 Configuring time synchronization

2.9.8.1 Time synchronization

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

Communication with SIMATIC S7-1500 Software Controller (Page 582)
Time synchronization restrictions (Page 666)
Configuring time synchronization for integrated connections (Page 667)
Configuring time synchronization for non-integrated connections (Page 668)

2.9.8.2 Time synchronization restrictions

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	-
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.

See also

Communication with SIMATIC S7-1500 Software Controller (Page 582)
Time synchronization (Page 665)

Configuring time synchronization for integrated connections (Page 667)
Configuring time synchronization for non-integrated connections (Page 668)

2.9.8.3 Configuring time synchronization for integrated connections

Introduction

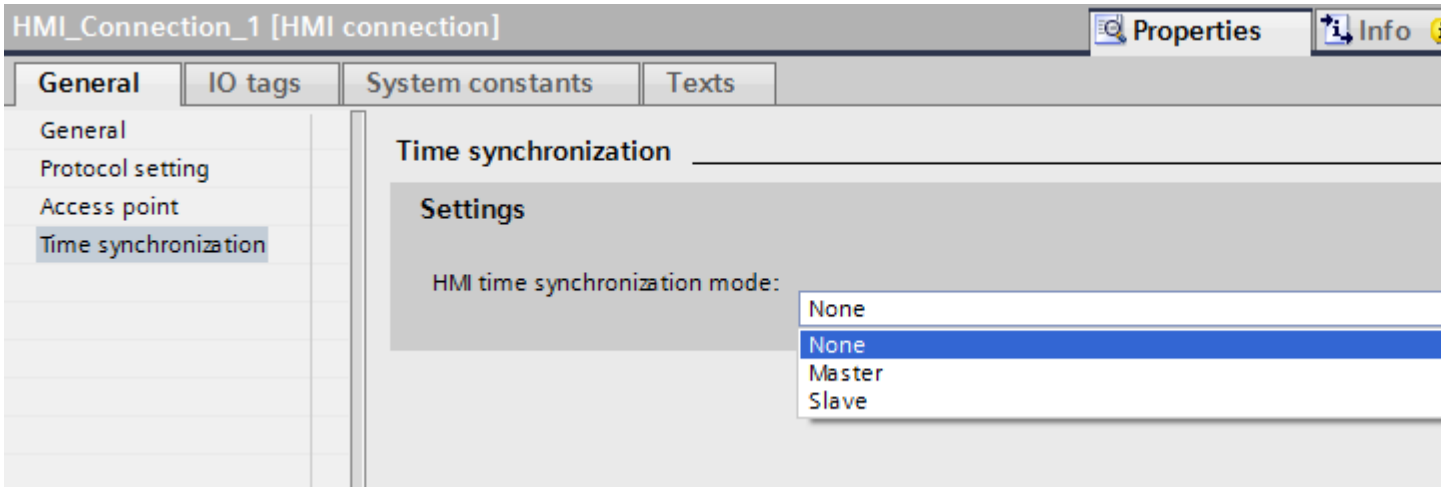
You configure time synchronization for an integrated connection in the "Devices & Networks" editor.

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

Communication with SIMATIC S7-1500 Software Controller (Page 582)
Time synchronization (Page 665)
Time synchronization restrictions (Page 666)
Configuring time synchronization for non-integrated connections (Page 668)

2.9.8.4 Configuring time synchronization for non-integrated connections

Introduction

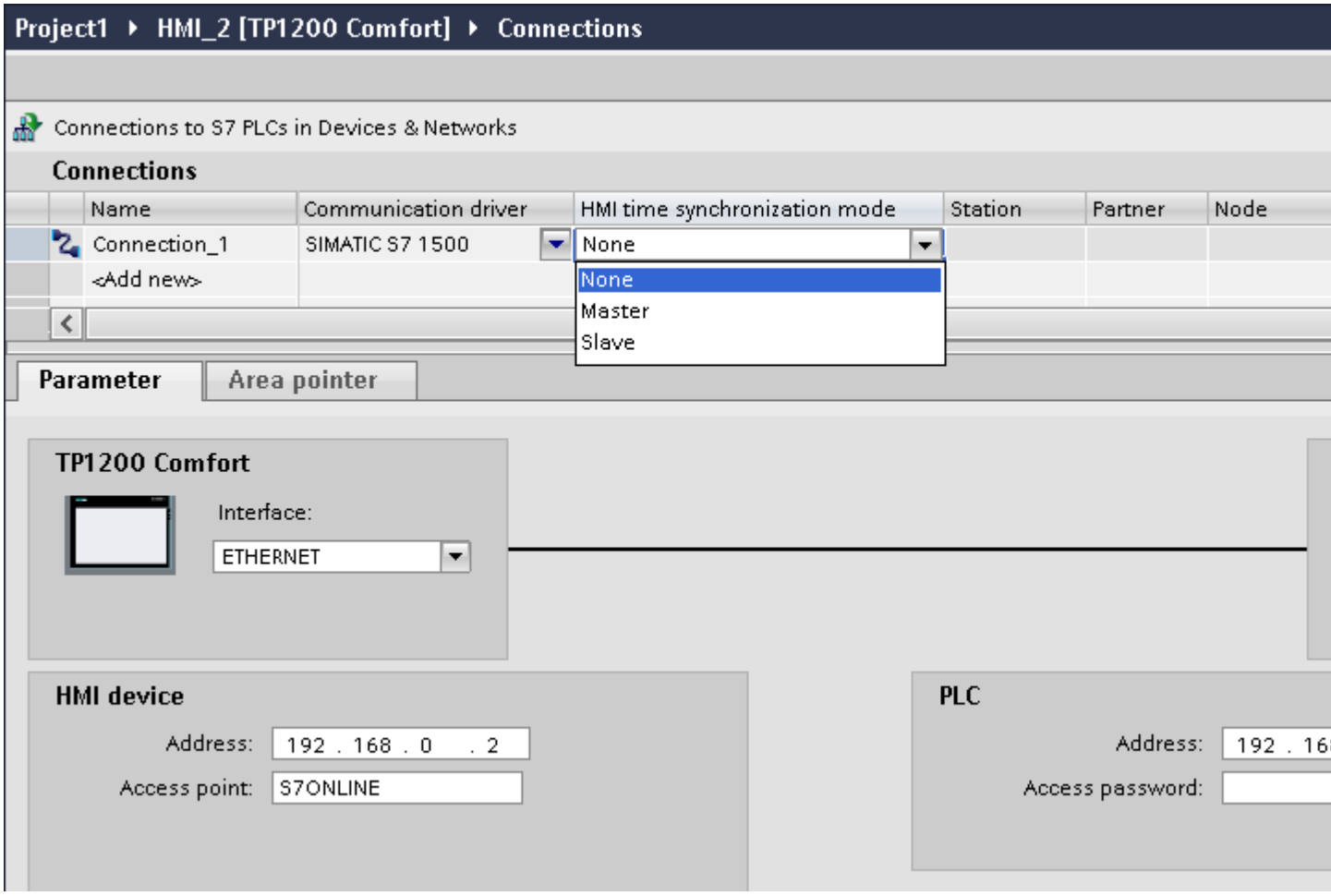
You configure time synchronization for a non-integrated connection in the "Connections" editor.

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

Communication with SIMATIC S7-1500 Software Controller (Page 582)

Time synchronization (Page 665)

Time synchronization restrictions (Page 666)

Configuring time synchronization for integrated connections (Page 667)

2.10 Communicating with the SIMATIC ET 200 CPU

2.10.1 Communication with SIMATIC ET 200 CPU

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.

See also

Creating a PROFINET connection (Page 737)

Creating a PROFIBUS DP connection (Page 740)

Ethernet parameters (Page 744)

PROFIBUS parameters (Page 742)

Parameters for the connection (SIMATIC S7 1500) (Page 745)

Creating a connection (Page 751)

Creating a connection (Page 746)

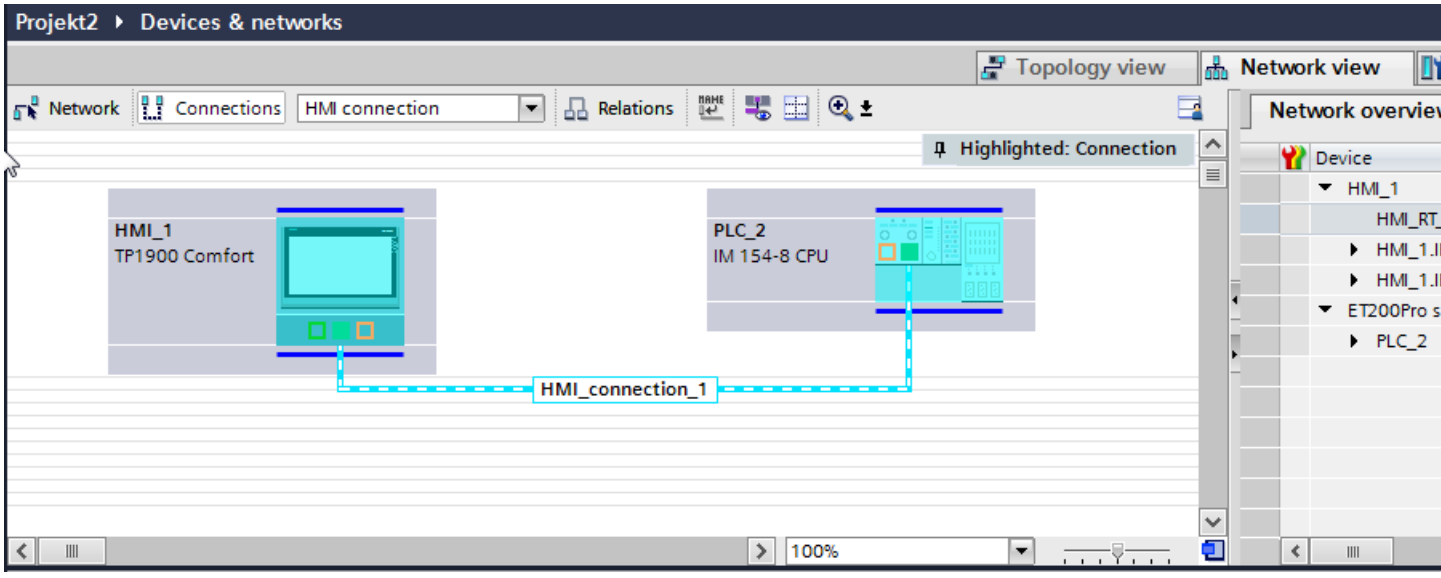
2.10.2 Communication via PROFINET

2.10.2.1 Configuring an HMI connection

Communication via PROFINET

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 670)

Configuring an HMI connection via PROFINET (Page 672)

Configuring an HMI connection via PROFINET

Introduction

You configure an HMI connection over PROFINET or Ethernet between HMI devices and a SIMATIC ET 200 CPU 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

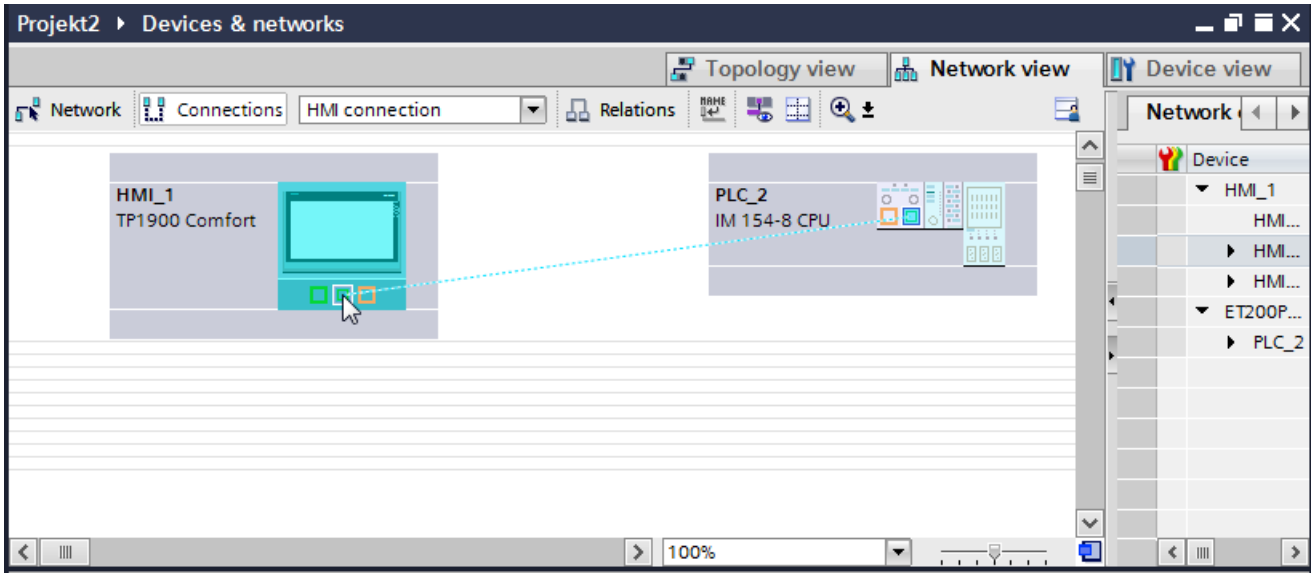
The following communication partners are created in the "Devices & Networks" editor:

- HMI device with PROFINET or Ethernet interface
- SIMATIC ET 200 CPU 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 681)".

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 670)

Communication via PROFINET (Page 671)

PROFINET parameters (Page 681)

2.10.2.2 Configuring an HMI connection

Communication via PROFINET

Communication via PROFINET

This section describes communication over PROFINET between a WinCC Runtime and the SIMATIC ET 200 CPU.

You can use the following WinCC Runtimes as an HMI device:

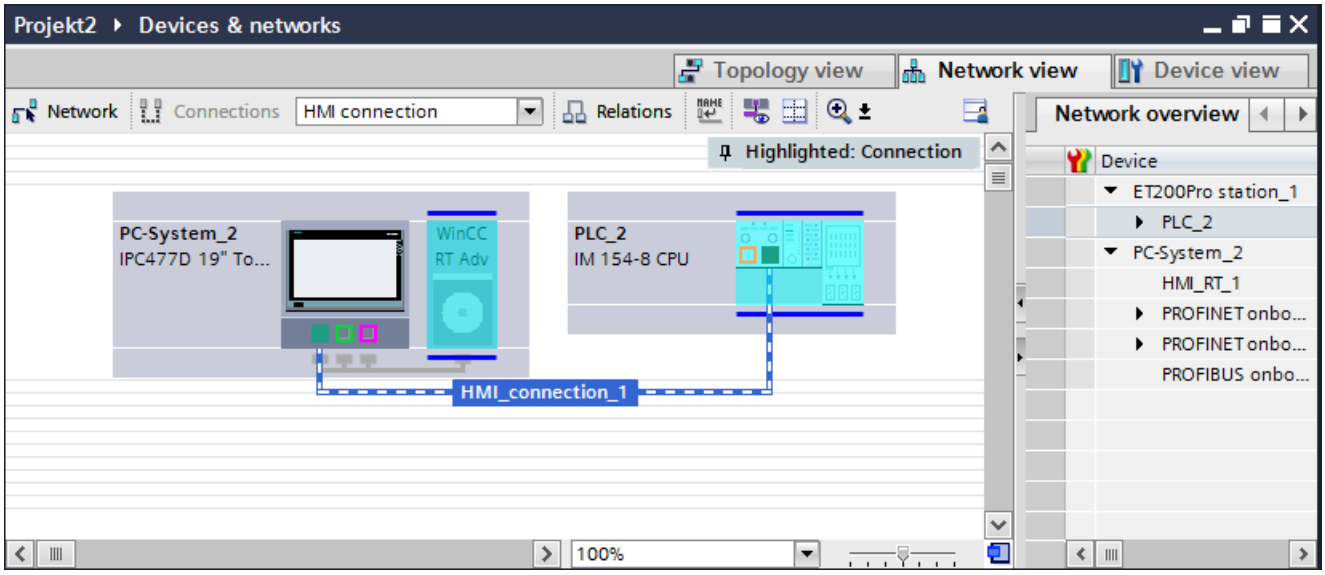
- WinCC RT Advanced
- WinCC RT Professional

WinCC Runtime as HMI device

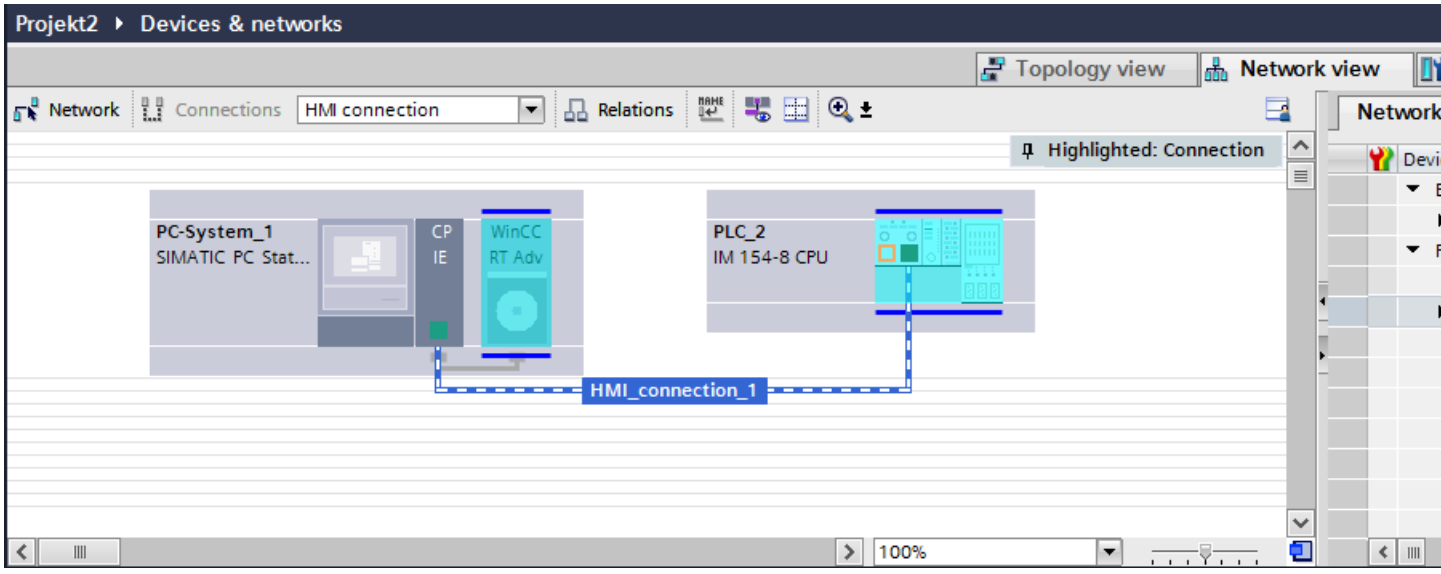
You configure the HMI connections between a WinCC Runtime and SIMATIC ET 200 CPU in the "Devices & Networks" editor.

If you want to use a WinCC Runtime as an HMI device, you have the following options:

1. You create a SIMATIC PC and configure a WinCC Runtime on this SIMATIC PC.
   In this way you use the SIMATIC PC with a WinCC Runtime as an HMI device.



2. You create a WinCC Runtime and configure a communication processor to the Runtime.
   In this way you use your configuration PC with a WinCC Runtime as an HMI device.



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 670)

Configuring an HMI connection via PROFINET with a PC (Page 676)

Configuring an HMI connection via PROFINET with a SIMATIC PC (Page 679)

Configuring an HMI connection via PROFINET with a PC

Introduction

You configure an HMI connection over PROFINET or Ethernet between a HMI device and a SIMATIC ET 200 CPU 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.

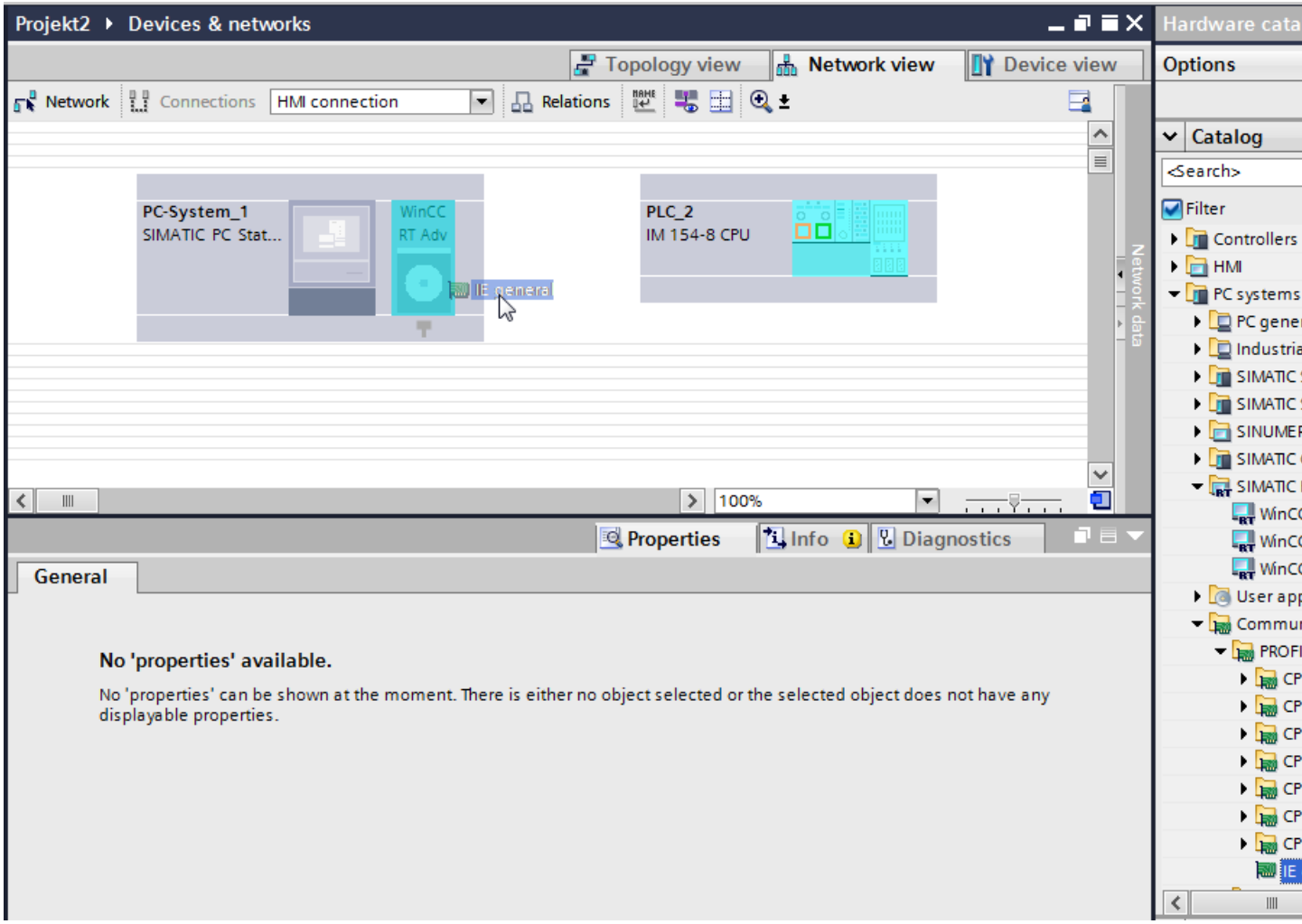
Requirements

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

- SIMATIC ET 200 CPU with PROFINET interface
- PC station with a "WinCC Runtime" application

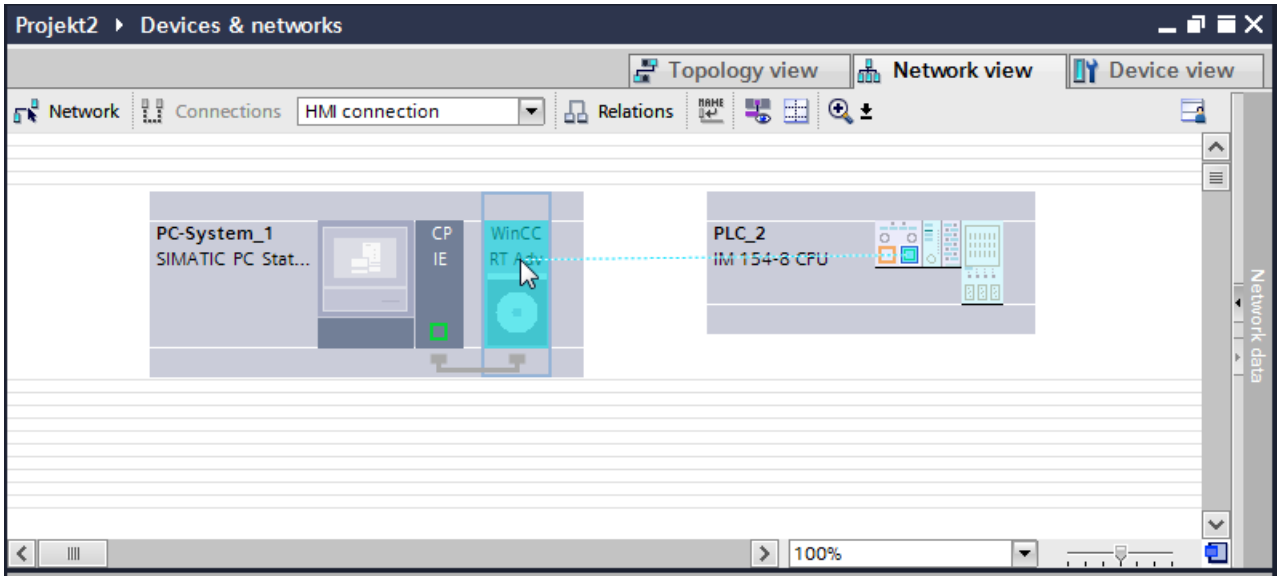
Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed in the network view.
2. Use a drag-and-drop operation to move a PROFINET-capable communication processor from the hardware catalog to the WinCC Runtime.



3. Click the "Connections" button and select "HMI connection" for the connection type.
   The devices available for connection are highlighted in color.

4. Click the PROFINET interface of the PLC and use a drag-and-drop operation to draw a connection to the PROFINET interface of the communication processor.



5. Click the connected interface on the PC station and select the entry "PC station" under "Interface assignment" in the Inspector window.
6. Click the connecting line.
   The connection is displayed graphically in the Inspector window.
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.
   For additional information on this, see section "PROFINET parameters (Page 681)".

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 670)

Communication via PROFINET (Page 674)

Configuring an HMI connection via PROFINET with a SIMATIC PC (Page 679)

PROFINET parameters (Page 681)

Configuring an HMI connection via PROFINET with a SIMATIC PC

Introduction

You configure an HMI connection over PROFINET or Ethernet between a HMI device and SIMATIC ET 200 CPU 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.

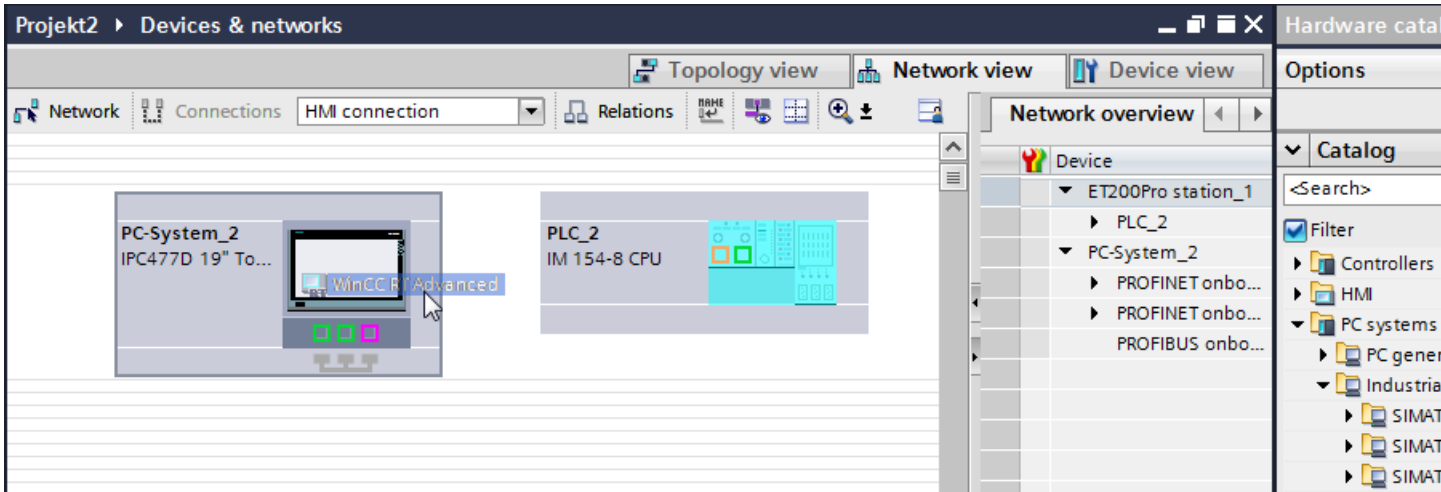
Requirements

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

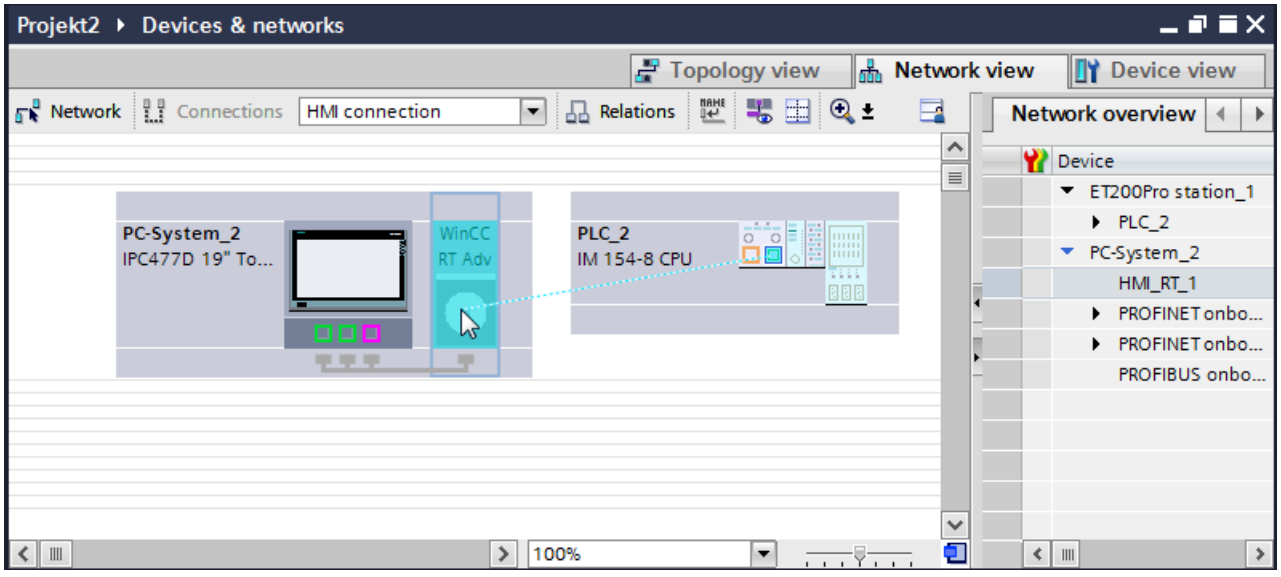
- SIMATIC ET 200 CPU with PROFINET interface
- SIMATIC PC with PROFINET interface

Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed in the network view.
2. Use a drag-and-drop operation to move a WinCC Runtime from the hardware catalog to the PC.



3. Click the "Connections" button and select "HMI connection" for the connection type.
   The devices available for connection are highlighted in color.
4. 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 PC.



5. Click the connected interface on the PC station and select the entry "PC station" under "Interface assignment" in the Inspector window.
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.
   For additional information on this, see section "PROFINET parameters (Page 681)".

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 670)

Communication via PROFINET (Page 674)

Configuring an HMI connection via PROFINET with a PC (Page 676)

PROFINET parameters (Page 681)

2.10.2.3 PROFINET parameters

PROFINET parameters for the HMI connection

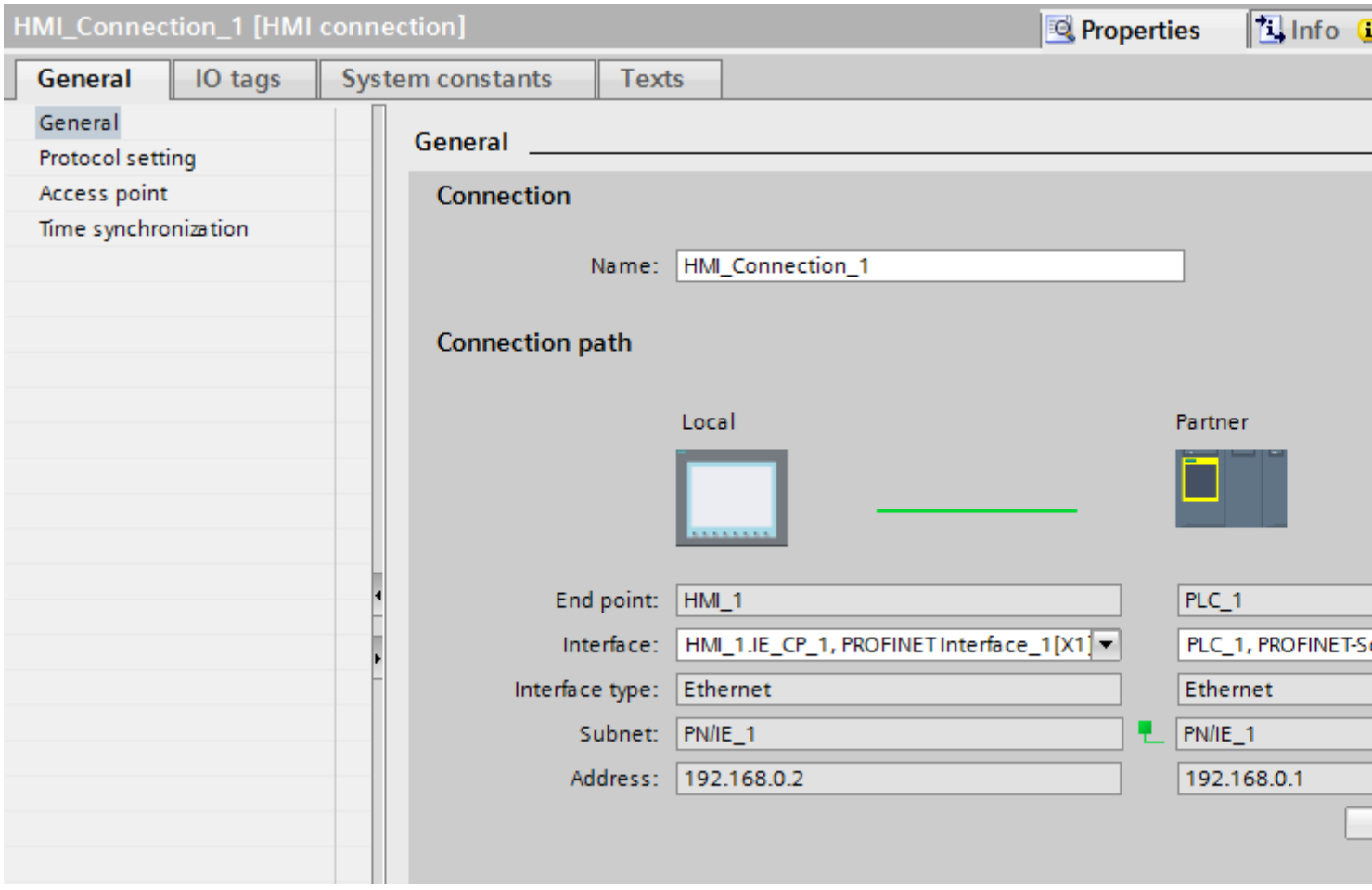
PROFINET parameters for the HMI connection

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 ET 200 CPU (Page 670)

PROFINET parameters for the HMI device (Page 683)

PROFINET parameters for the PLC (Page 685)

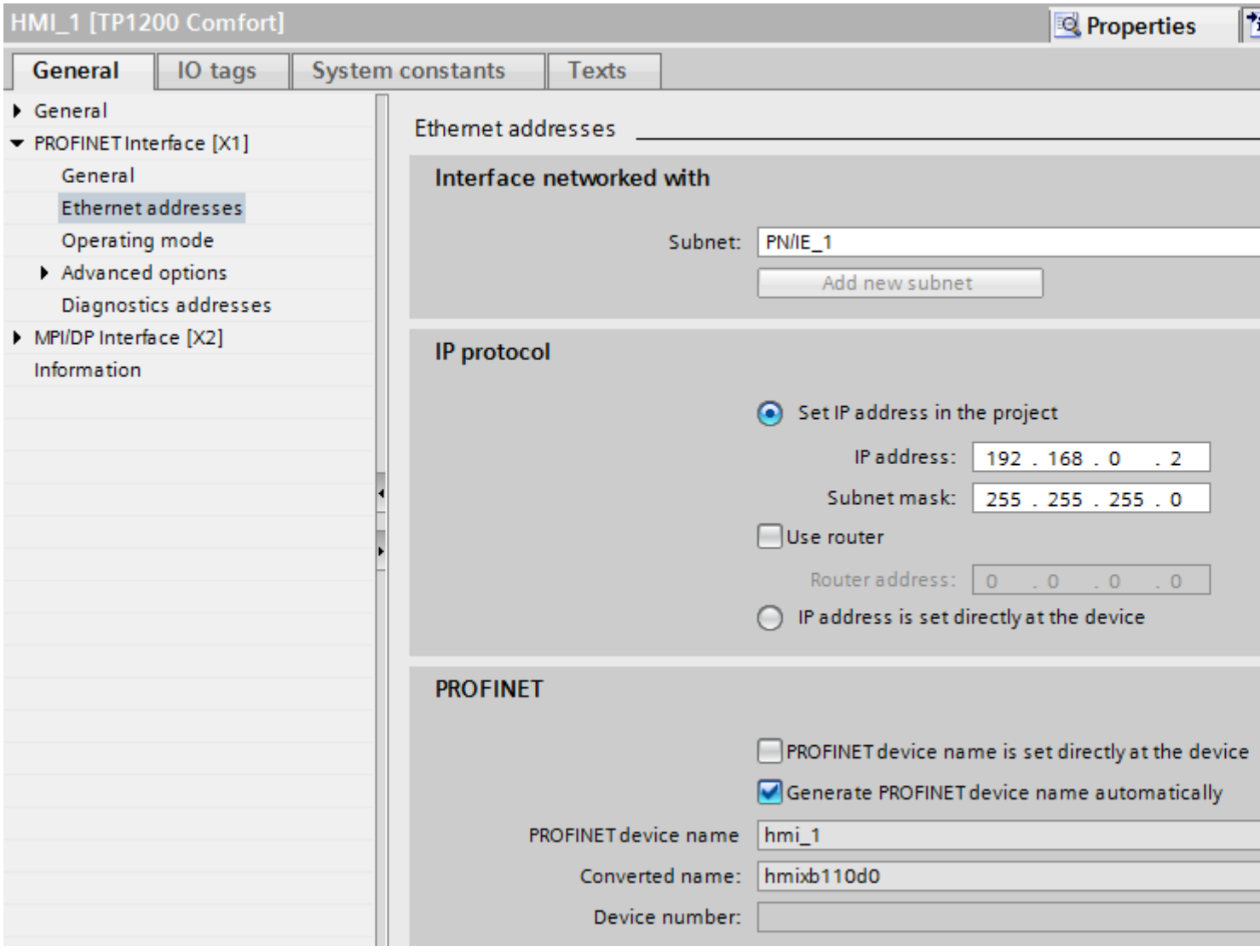
PROFINET parameters for the HMI device

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:
  - Mobile Panel 177 PN
  - Mobile Panel 177 DP

- "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 directly on the device"
  If the "Set IP address on the device" function is enabled, 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 ET 200 CPU (Page 670)

PROFINET parameters for the HMI connection (Page 681)

PROFINET parameters for the PLC (Page 685)

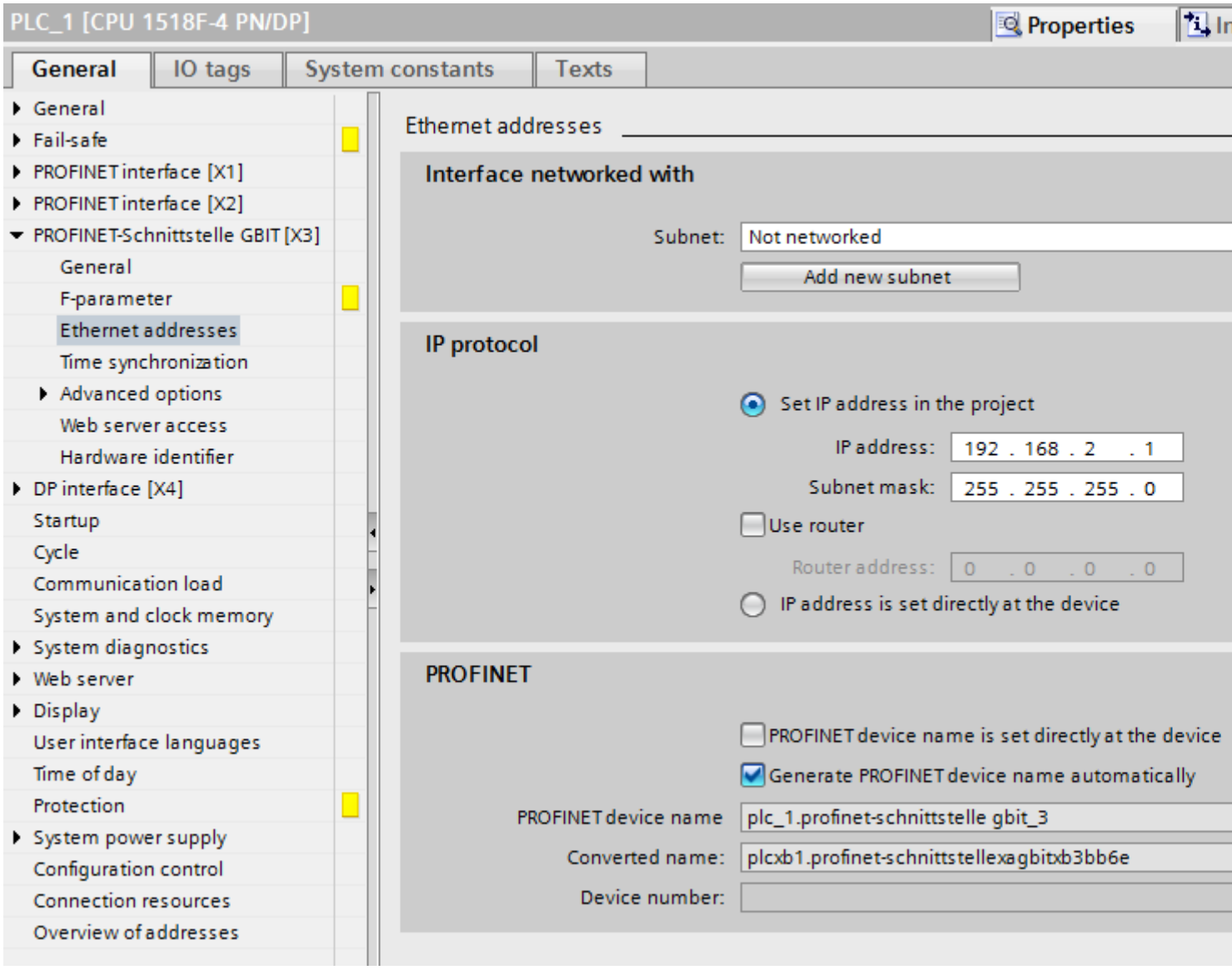
PROFINET parameters for the PLC

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 ET 200 CPU (Page 670)

PROFINET parameters for the HMI connection (Page 681)

PROFINET parameters for the HMI device (Page 683)

2.10.2.4 Setting port options

Setting the port options

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

Communication with SIMATIC ET 200 CPU (Page 670)

Wiring rules for disabled autonegotiation (Page 689)

Boundaries at the port (Page 690)

Wiring rules for disabled autonegotiation

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:

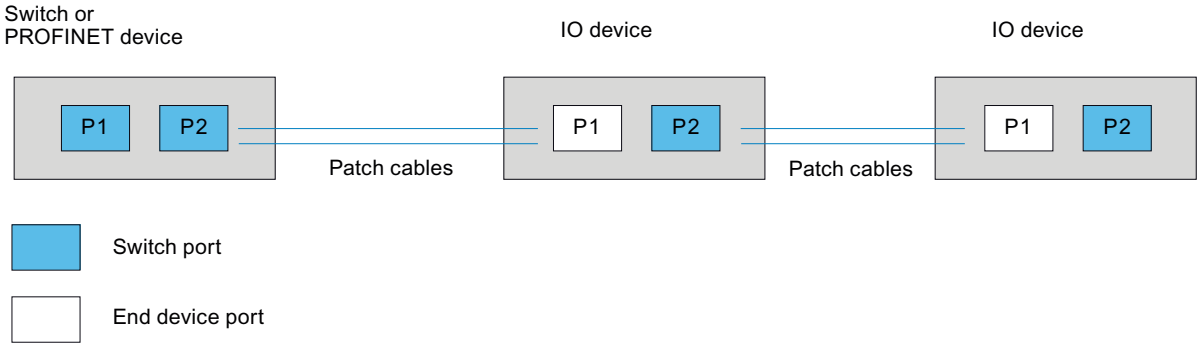
Type of port	PROFINET devices	Note
Switch port with crossed pin assignment	For IO devices: Port 2 For S7 CPUs with 2 ports: Ports 1 and 2	Crossed pin assignment means that the pin assignment for the ports for sending and receiving between the respective PROFINET devices is exchanged internally.
End device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

Communication with SIMATIC ET 200 CPU (Page 670)

Setting the port options (Page 687)

Boundaries at the port (Page 690)

Boundaries at the port

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

Communication with SIMATIC ET 200 CPU (Page 670)

Setting the port options (Page 687)

Wiring rules for disabled autonegotiation (Page 689)

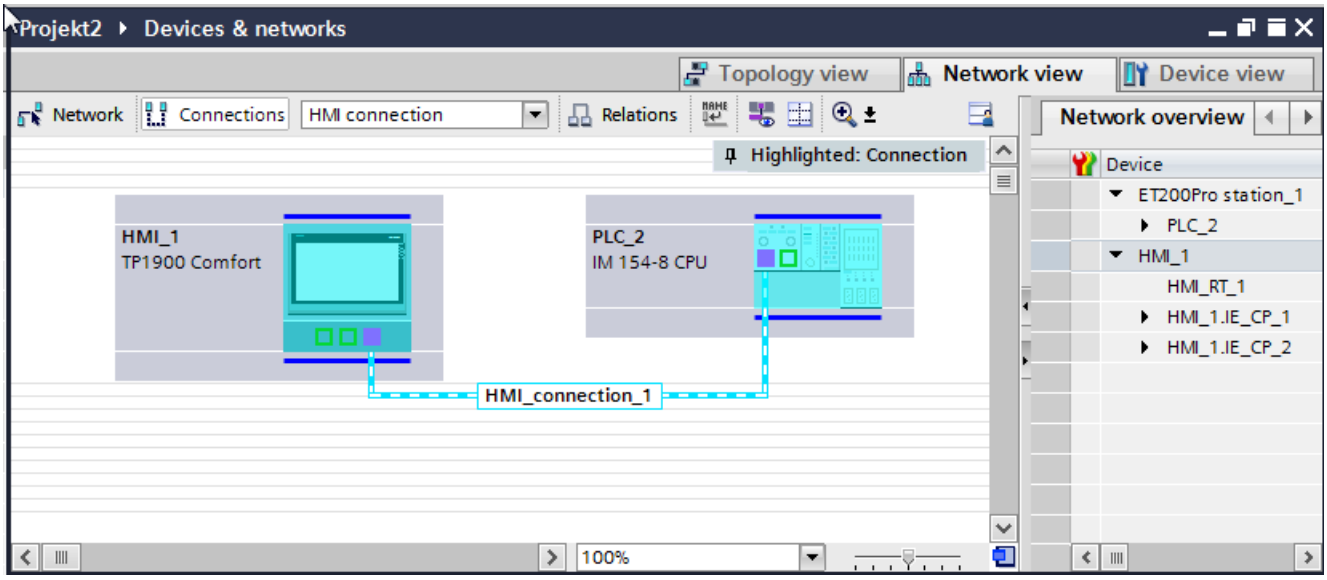
2.10.3 Communication via PROFIBUS

2.10.3.1 Configuring an HMI connection

Communication via PROFIBUS

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 670)

Configuring an HMI connection via PROFIBUS (Page 693)

Configuring an HMI connection via PROFIBUS

Introduction

You configure an HMI connection between HMI devices and a SIMATIC ET 200 CPU via PROFIBUS in the "Devices & Networks" editor.

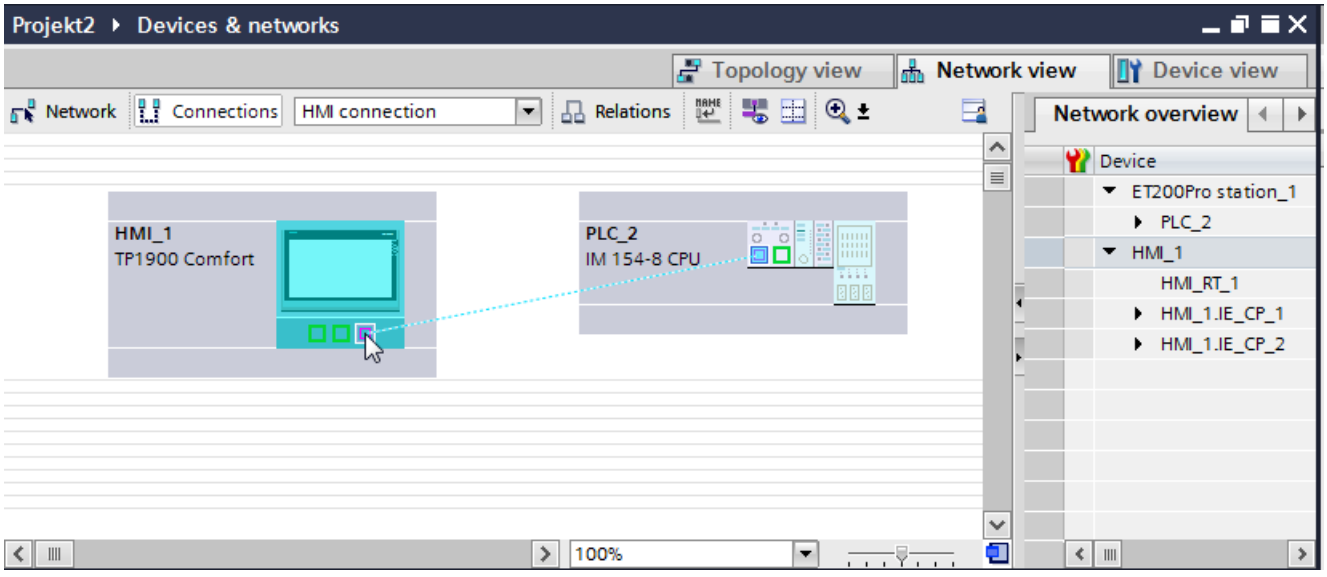
Requirements

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

- HMI device with MPI/DP interface
- SIMATIC ET 200 CPU

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. See the "PROFIBUS parameters (Page 700)" 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 ET 200 CPU via PROFIBUS.

See also

Communication with SIMATIC ET 200 CPU (Page 670)

Communication via PROFIBUS (Page 692)

PROFIBUS parameters (Page 700)

2.10.3.2 Configuring an HMI connection

Communication via PROFIBUS

Communication via PROFIBUS

This section describes the communication between a WinCC Runtime and the SIMATIC ET 200 CPU controller via PROFIBUS.

You can use the following WinCC Runtimes as an HMI device:

- WinCC RT Advanced
- WinCC RT Professional

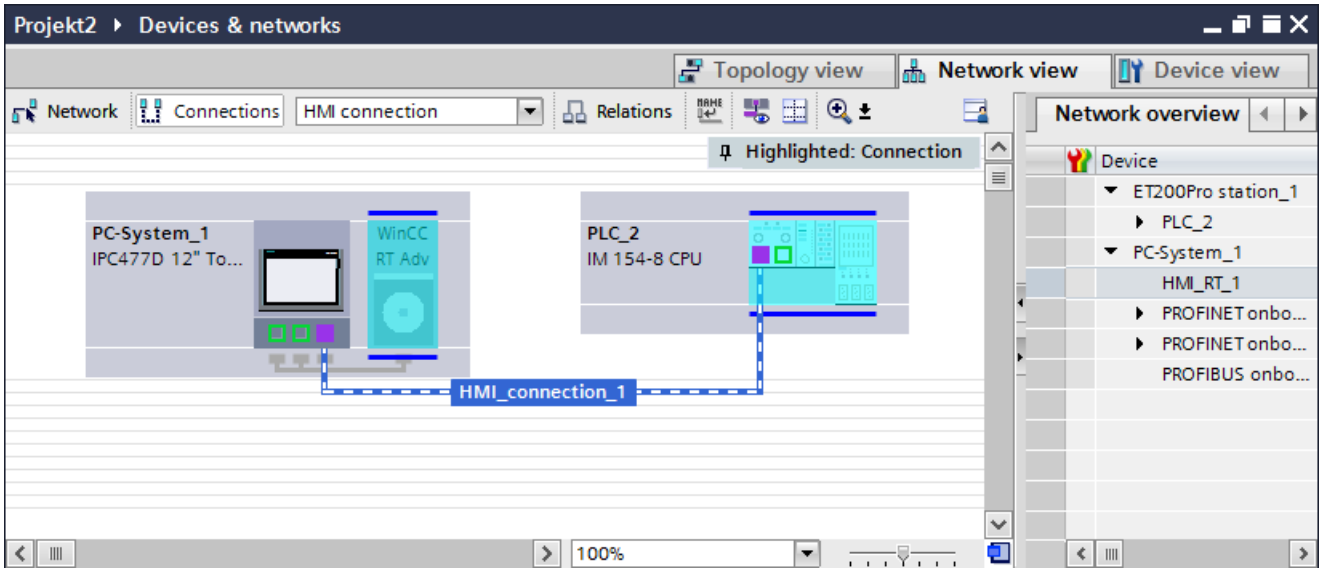
If you want to connect a SIMATIC ET 200 CPU to a HMI device via PROFIBUS, you must configure a PROFIBUS-capable communication module to a slot of the controller.

WinCC Runtime as HMI device

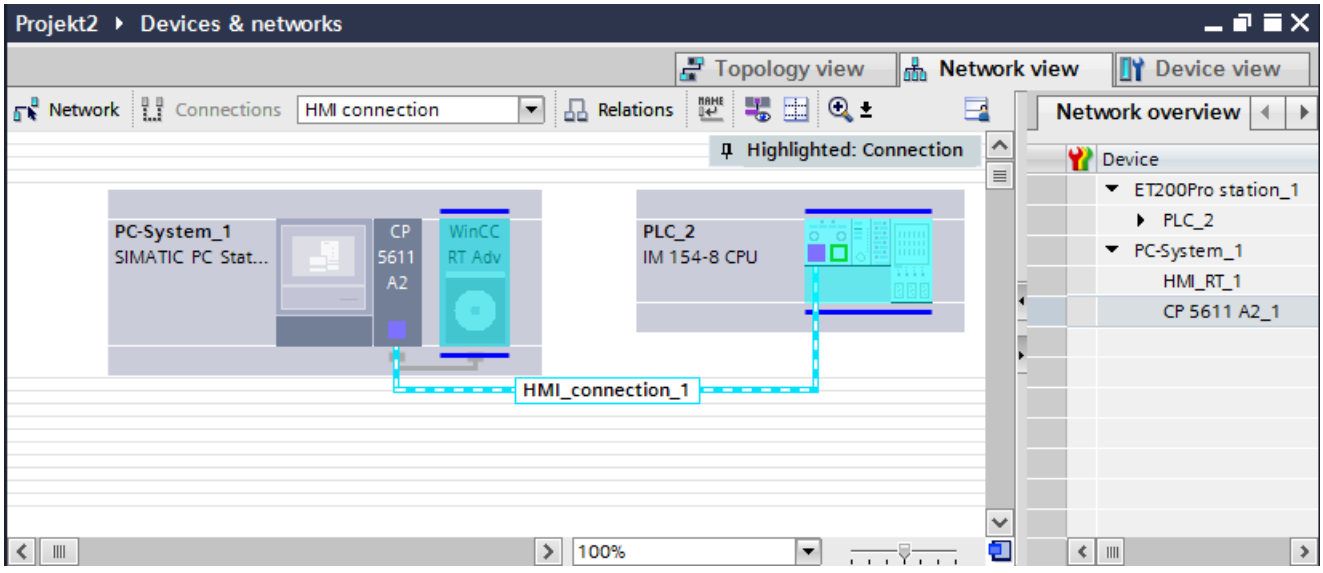
Configure the HMI connections between WinCC Runtime and SIMATIC ET 200 CPU in the "Devices & Networks" editor.

If you want to use a WinCC Runtime as an HMI device, you have the following options:

1. You create a SIMATIC PC and configure a WinCC Runtime on this SIMATIC PC.
   In this way you use the SIMATIC PC with a WinCC Runtime as an HMI device.



2. You create a WinCC Runtime and configure a communication processor to the Runtime.
   In this way you use your configuration PC with a WinCC Runtime as an HMI device.



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 PROFIBUS in the "Devices & Networks" editor.

Connection in the "Connections" editor

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

See also

Communication with SIMATIC ET 200 CPU (Page 670)

Configuring an HMI connection via PROFIBUS with a SIMATIC PC (Page 696)

Configuring an HMI connection via PROFIBUS with a PC (Page 698)

Configuring an HMI connection via PROFIBUS with a SIMATIC PC

Introduction

You configure an HMI connection between HMI devices and a SIMATIC ET 200 CPU via PROFIBUS in the "Devices & Networks" editor.

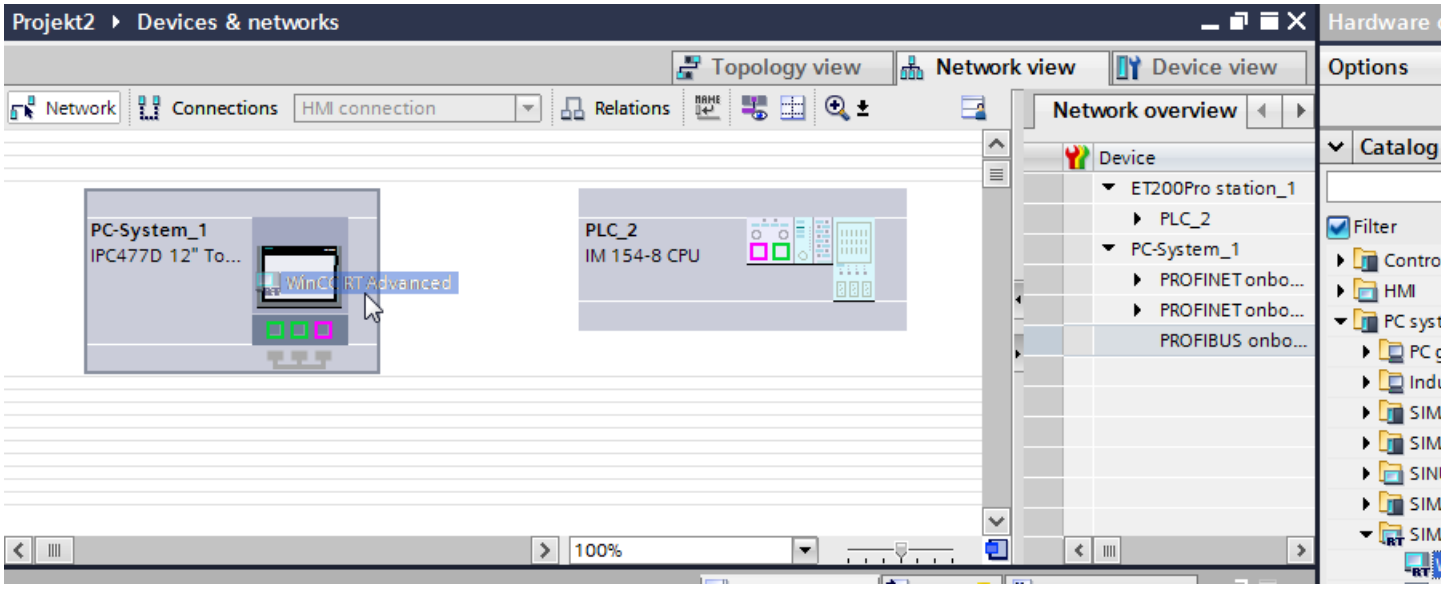
Requirements

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

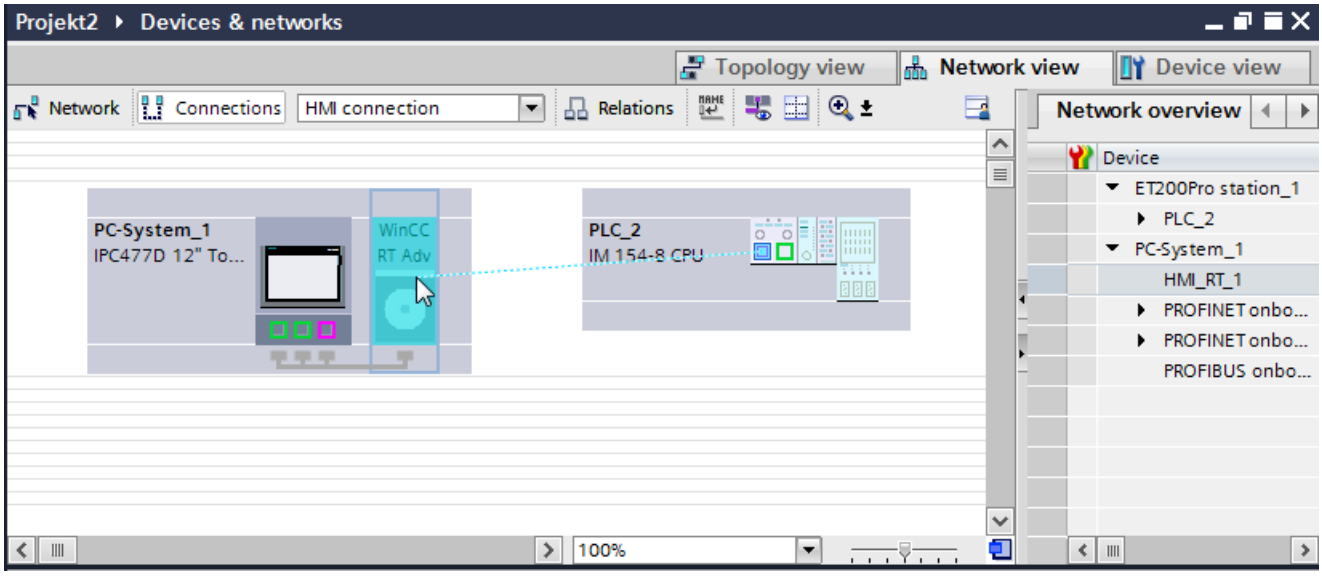
- SIMATIC ET 200 CPU
- SIMATIC PC with PROFIBUS interface

Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed in the network view.
2. Use a drag-and-drop operation to move a WinCC Runtime from the hardware catalog to the PC.



3. Click the "Connections" button and select "HMI connection" for the connection type.
   The devices available for connection are highlighted in color.
4. Click the PROFIBUS interface of the CPU and use a drag-and-drop operation to draw a connection to the PROFIBUS interface of the PC.



5. Click the connecting line.

6. Click "Highlight HMI connection" and select the HMI connection.
   The connection is displayed graphically in the Inspector window.
7. 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 700)" 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 ET 200 CPU via PROFIBUS.

See also

Communication with SIMATIC ET 200 CPU (Page 670)

Communication via PROFIBUS (Page 694)

Configuring an HMI connection via PROFIBUS with a PC (Page 698)

PROFIBUS parameters (Page 700)

Configuring an HMI connection via PROFIBUS with a PC

Introduction

You configure an HMI connection between HMI devices and a SIMATIC ET 200 CPU via PROFIBUS in the "Devices & Networks" editor.

Requirements

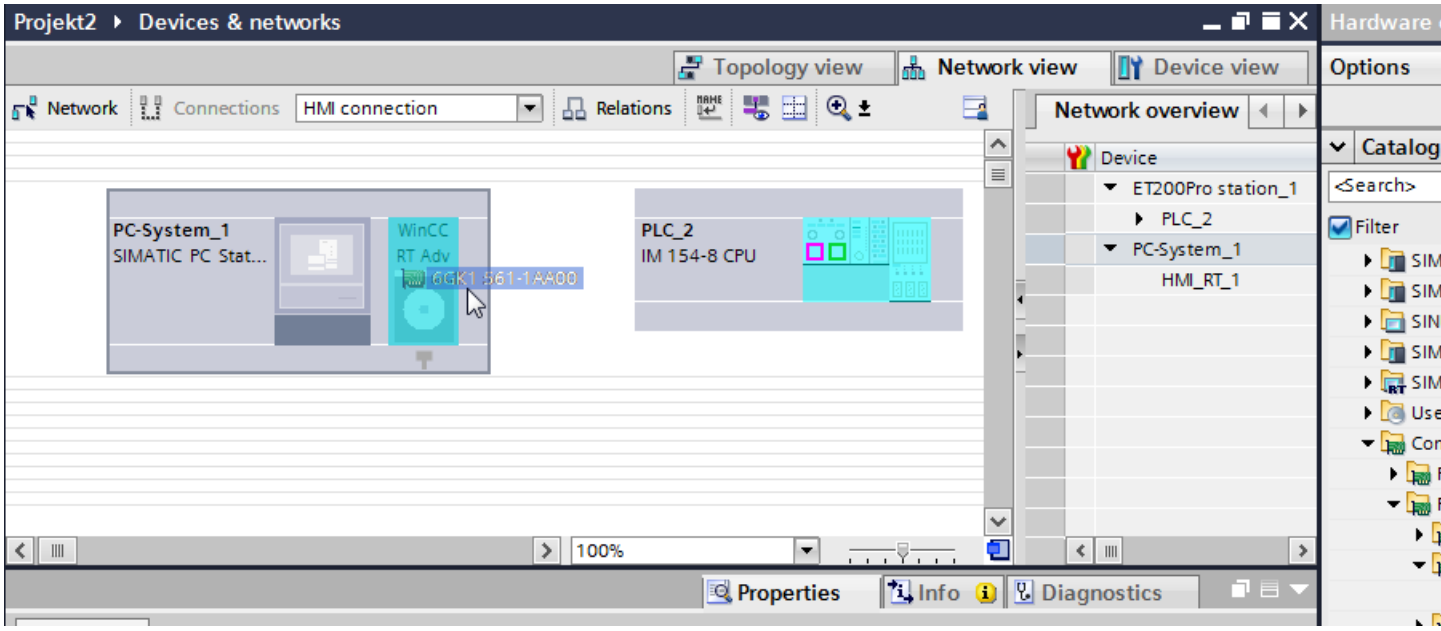
The following communication partners are created in the "Devices & Networks" editor:

- SIMATIC ET 200 CPU
- PC station with a WinCC RT Advanced or WinCC RT Professional

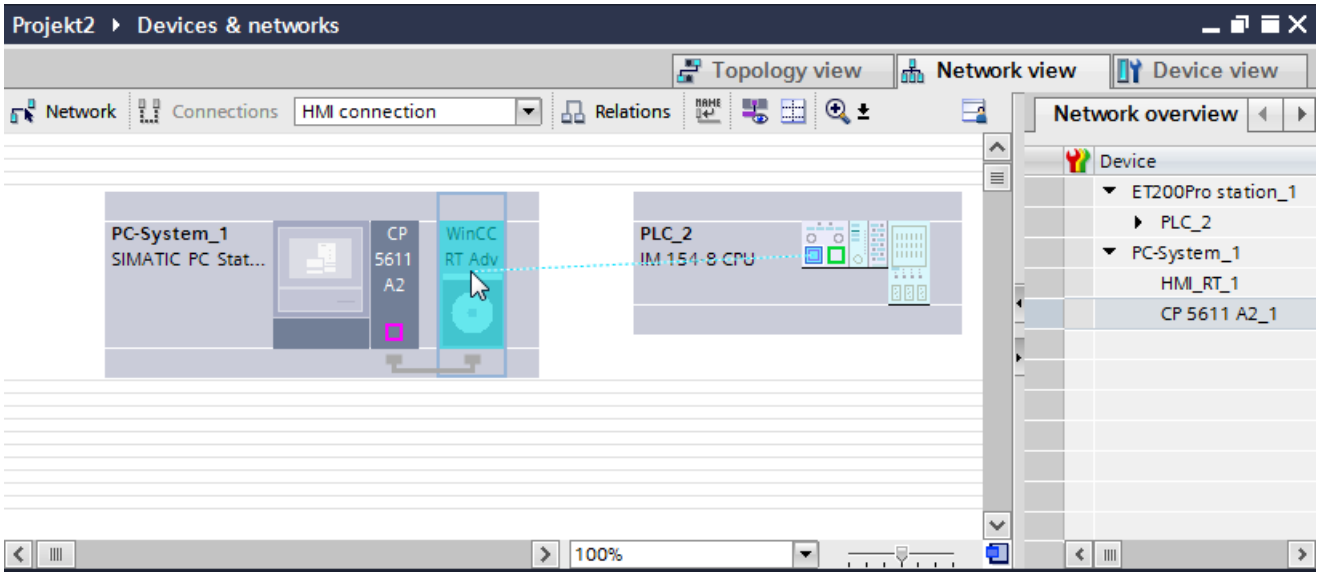
Procedure

1. Double-click the "Devices & Networks" item in the project tree.
   The available communication partners in the project are displayed in the network view.
2. Use a drag-and-drop operation to move a PROFIBUS-capable communication processor from the hardware catalog to the WinCC Runtime.

3. Use a drag-and-drop operation to move a PROFIBUS-capable communication module from the hardware catalog to the PLC.



4. Click the "Connections" button and select "HMI connection" for the connection type. The devices available for connection are highlighted in color.
5. Click the PROFIBUS interface of the communication module and use a drag-and-drop operation to draw a connection to the PROFIBUS interface of the communication processor.



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 700)".

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 670)

Communication via PROFIBUS (Page 694)

Configuring an HMI connection via PROFIBUS with a SIMATIC PC (Page 696)

PROFIBUS parameters (Page 700)

2.10.3.3 PROFIBUS parameters

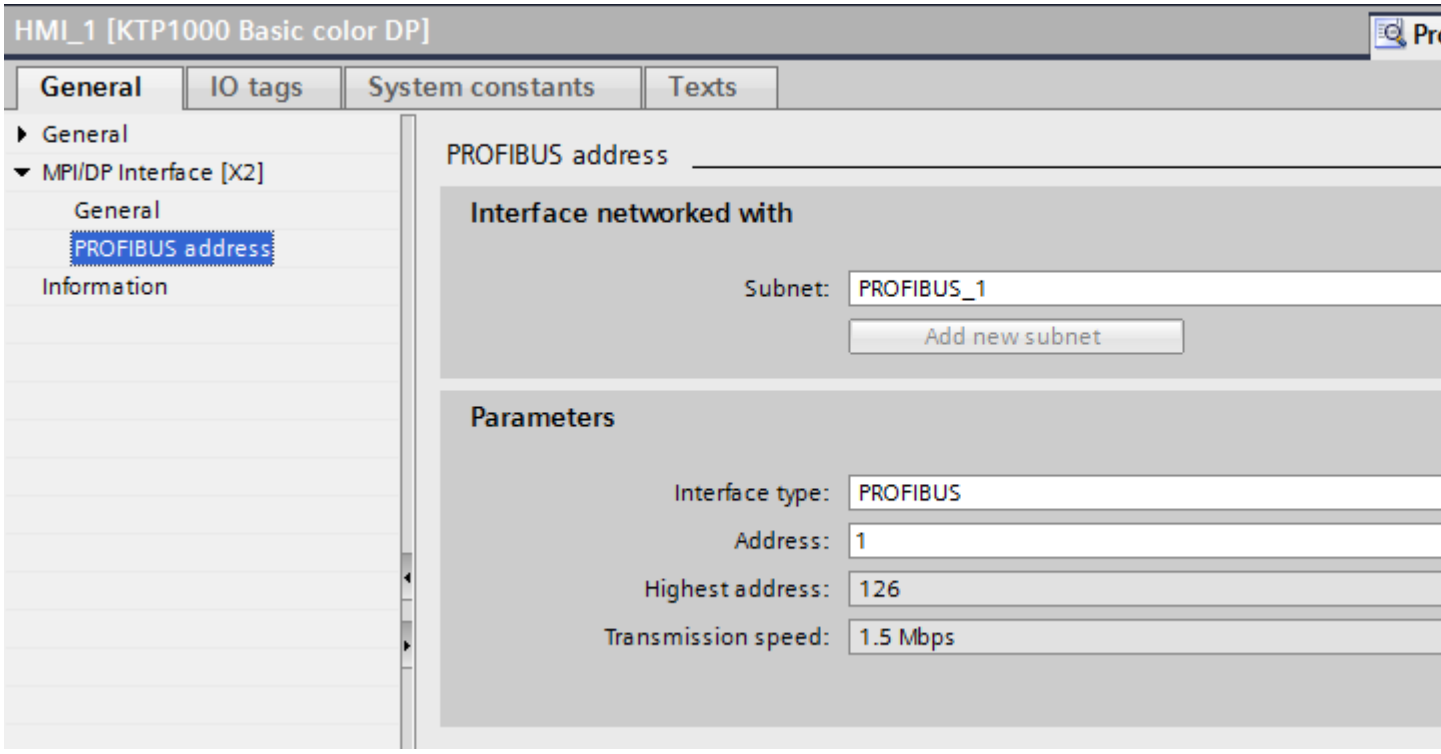
PROFIBUS parameters for the HMI device

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 with SIMATIC ET 200 CPU (Page 670)

PROFIBUS parameters for the PLC (Page 702)

PROFIBUS parameters for the HMI connection (Page 703)

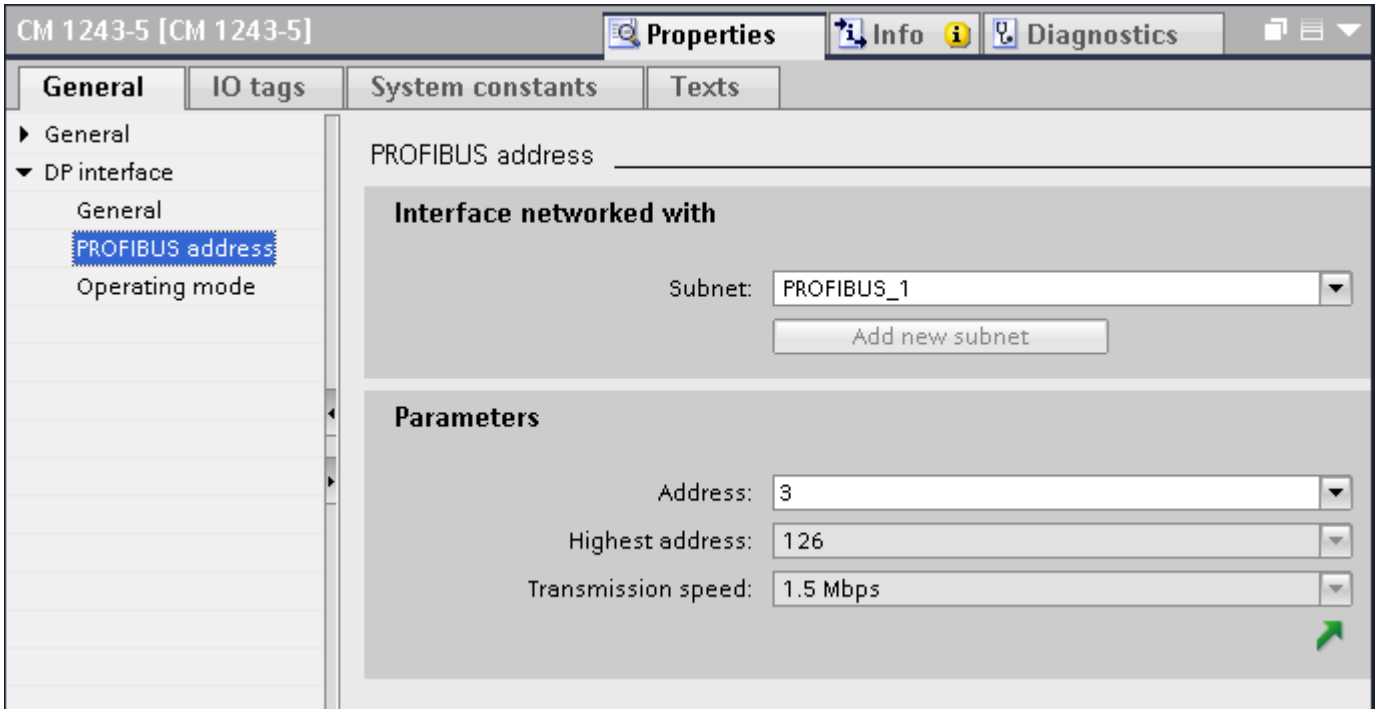
PROFIBUS parameters for the PLC

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 ET 200 CPU (Page 670)

PROFIBUS parameters for the HMI device (Page 700)

PROFIBUS parameters for the HMI connection (Page 703)

PROFIBUS parameters for the HMI connection

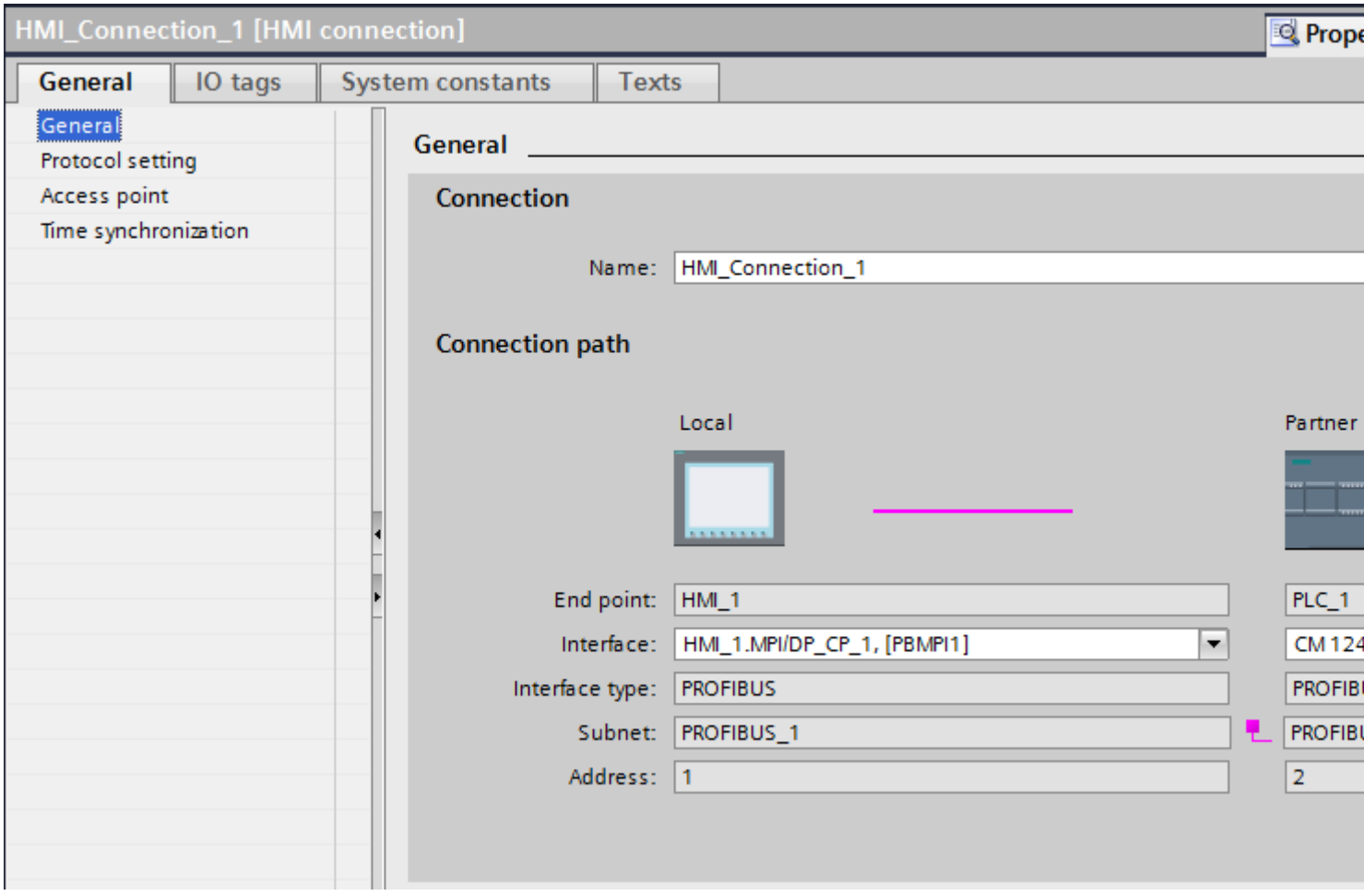
PROFIBUS parameters for the HMI connection

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 with SIMATIC ET 200 CPU (Page 670)

PROFIBUS parameters for the HMI device (Page 700)

PROFIBUS parameters for the PLC (Page 702)

2.10.4 Data exchange

2.10.4.1 Data exchange using area pointers

General information on area pointers

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 153)

See also

Configuration of area pointers (Page 153)
Communication with SIMATIC ET 200 CPU (Page 670)
"Screen number" area pointer (Page 706)
"Coordination" area pointer (Page 710)
"Project ID" area pointer (Page 711)
"Job mailbox" area pointer (Page 712)
Area pointer "Date/time" (Page 707)
"Date/time PLC" area pointer (Page 708)

"Screen number" area pointer

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

Communication with SIMATIC ET 200 CPU (Page 670)

General information on area pointers (Page 705)

"Coordination" area pointer (Page 710)

"Project ID" area pointer (Page 711)

"Job mailbox" area pointer (Page 712)

Area pointer "Date/time" (Page 707)

"Date/time PLC" area pointer (Page 708)

Area pointer "Date/time"

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 705)

"Screen number" area pointer (Page 706)

"Coordination" area pointer (Page 710)

"Project ID" area pointer (Page 711)

"Job mailbox" area pointer (Page 712)

"Date/time PLC" area pointer

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 705)

"Screen number" area pointer (Page 706)

"Coordination" area pointer (Page 710)

"Project ID" area pointer (Page 711)

"Job mailbox" area pointer (Page 712)

"Coordination" area pointer

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

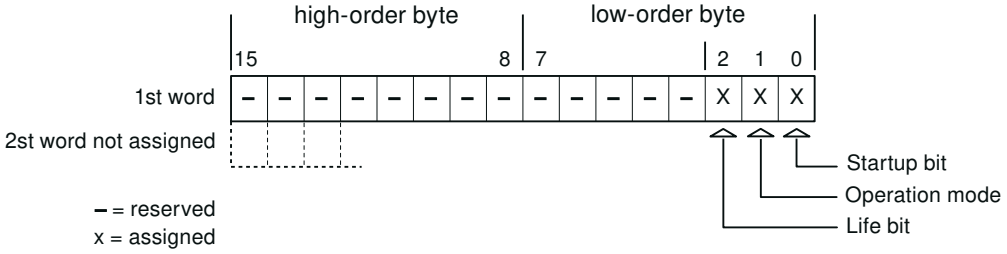
By default, the "Coordination" area pointer has the length of one word and cannot be changed.

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

Communication with SIMATIC ET 200 CPU (Page 670)

General information on area pointers (Page 705)

"Screen number" area pointer (Page 706)

"Project ID" area pointer (Page 711)

"Job mailbox" area pointer (Page 712)

Area pointer "Date/time" (Page 707)

"Date/time PLC" area pointer (Page 708)

"Project ID" area pointer

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

Communication with SIMATIC ET 200 CPU (Page 670)

General information on area pointers (Page 705)

"Screen number" area pointer (Page 706)

"Coordination" area pointer (Page 710)

"Job mailbox" area pointer (Page 712)

Area pointer "Date/time" (Page 707)

"Date/time PLC" area pointer (Page 708)

"Job mailbox" area pointer

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 to 23)
	Parameter 2	Left byte: Minutes (0-59) Right byte: Seconds (0 to 59)
	Parameter 3	-
15	**Setting the date (BCD coded)** [2) 3)]	
	Parameter 1	Left byte: - Right byte: Weekday (1-7: Sunday-Saturday)
	Parameter 2	Left byte: Day (1-31) Right byte: Month (1 to 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**	
	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**	

No.	Function	
14	**Set time (BCD-coded)**	
	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**	
	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]	The weekday is ignored on HMI device KTP 600 BASIC PN.
[3]	The weekday is ignored when you configure the "Date/Time PLC" area pointer.

See also

Communication with SIMATIC ET 200 CPU (Page 670)

General information on area pointers (Page 705)

"Screen number" area pointer (Page 706)

"Coordination" area pointer (Page 710)

"Project ID" area pointer (Page 711)

Area pointer "Date/time" (Page 707)

"Date/time PLC" area pointer (Page 708)

"Data record" area pointer

"Data record" area pointer

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

Communication with SIMATIC ET 200 CPU (Page 670)

Transfer without synchronization (Page 716)

Transfer with synchronization (Page 716)

Transfer started by operator in the recipe view (Page 718)

Transfer with job mailbox (Page 719)

Sequence of transfer triggered by a configured function (Page 721)

Possible causes of error when transferring data records (Page 722)

Transfer without synchronization

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

Communication with SIMATIC ET 200 CPU (Page 670)

"Data record" area pointer (Page 715)

Transfer with synchronization (Page 716)

Transfer started by operator in the recipe view (Page 718)

Transfer with job mailbox (Page 719)

Sequence of transfer triggered by a configured function (Page 721)

Possible causes of error when transferring data records (Page 722)

Transfer with synchronization

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

Communication with SIMATIC ET 200 CPU (Page 670)

"Data record" area pointer (Page 715)

Transfer without synchronization (Page 716)

Transfer started by operator in the recipe view (Page 718)
Transfer with job mailbox (Page 719)
Sequence of transfer triggered by a configured function (Page 721)
Possible causes of error when transferring data records (Page 722)

Transfer started by operator in the recipe view

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

Communication with SIMATIC ET 200 CPU (Page 670)

"Data record" area pointer (Page 715)

Transfer without synchronization (Page 716)

Transfer with synchronization (Page 716)

Transfer with job mailbox (Page 719)

Sequence of transfer triggered by a configured function (Page 721)

Possible causes of error when transferring data records (Page 722)

Transfer with job mailbox

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)	

	Left byte (LB)	Right byte (RB)
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

Communication with SIMATIC ET 200 CPU (Page 670)

"Data record" area pointer (Page 715)

Transfer without synchronization (Page 716)

Transfer with synchronization (Page 716)

Transfer started by operator in the recipe view (Page 718)

Sequence of transfer triggered by a configured function (Page 721)
Possible causes of error when transferring data records (Page 722)

Sequence of transfer triggered by a configured function

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

Communication with SIMATIC ET 200 CPU (Page 670)
"Data record" area pointer (Page 715)

Transfer without synchronization (Page 716)
Transfer with synchronization (Page 716)
Transfer started by operator in the recipe view (Page 718)
Transfer with job mailbox (Page 719)
Possible causes of error when transferring data records (Page 722)

Possible causes of error when transferring data records

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

Communication with SIMATIC ET 200 CPU (Page 670)
"Data record" area pointer (Page 715)
Transfer without synchronization (Page 716)
Transfer with synchronization (Page 716)
Transfer started by operator in the recipe view (Page 718)
Transfer with job mailbox (Page 719)
Sequence of transfer triggered by a configured function (Page 721)

2.10.4.2 Trends

General information on trends

Trends

A trend is the graphical representation of one or more values from the PLC. The value is read out either time- or bit-triggered, depending on the configuration.

For additional information see:

AUTOHOTSPOT

Note

The value is read out time-triggered for Basic Panels.

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.

Bit-triggered trends

Through a trigger bit set in the trend request tag, the HMI device either reads in a trend value or an entire trend buffer. This setting is defined in the configuration. Bit-triggered trends are normally used to represent fast 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 "HMI tags" editor and connected to trend areas during configuration. The HMI device and PLC then communicate with each other via 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)

See also

Communication with SIMATIC ET 200 CPU (Page 670)

Trend request and trend transfer (Page 724)

Permitted data types for trends (Page 725)

Trend request and trend transfer

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 closing the screen, the HMI device resets the relevant bits in the trend request area.

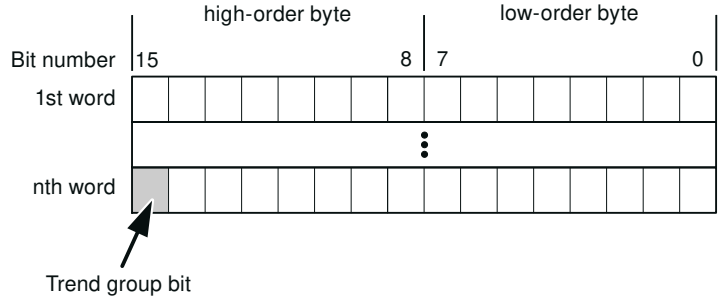
The trend request area can be used for evaluation purposes in the PLC to determine which trend is currently being displayed on the HMI device. Trends can also be triggered without evaluation of the trend request area.

Trend transfer area 1

This area is used to trigger trends. You must set the bit assigned to the trend in the trend transfer area and set the trend group bit in your control program. 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.

The PLC writes to Buffer 2 while the HMI device reads values from Buffer 1, and writes to Buffer 1 when the HMI device is reading Buffer 2. This prevents the PLC from overwriting trend values while the trend is being read by the HMI device.

See also

Communication with SIMATIC ET 200 CPU (Page 670)

General information on trends (Page 723)

Permitted data types for trends (Page 725)

Permitted data types for trends

For SIMATIC S7

You assign one bit to each trend during configuration. Tags and array tags of the "Word" or "Int" data type are permitted.

See also

Communication with SIMATIC ET 200 CPU (Page 670)

General information on trends (Page 723)

Trend request and trend transfer (Page 724)

2.10.4.3 Alarms

Configuring alarms

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

Communication with SIMATIC ET 200 CPU (Page 670)

Acknowledgment of alarms (Page 726)

Acknowledgment of alarms

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

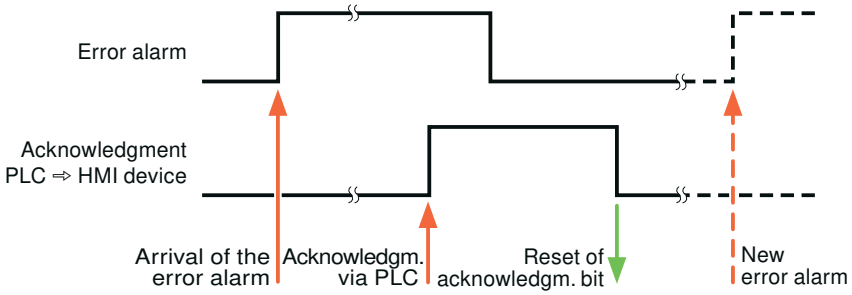
Acknowledgment by the PLC

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.

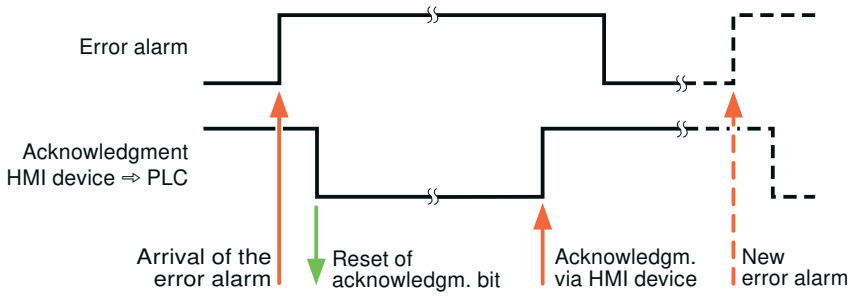
If a discrete alarm subject to acknowledgment is acknowledged by the HMI device, then the corresponding bit in the assigned acknowledgment tag will be set. The entire acknowledgment tag is then written to the PLC by the HMI device. This allows the PLC to recognize that a certain alarm message has been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

Communication with SIMATIC ET 200 CPU (Page 670)

Configuring alarms (Page 725)

2.10.4.4 LED mapping

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 Comfort Panels	Panel PCs
0	0	Off	Off
0	1	Rapid flashing	Flashing

Bit n+ 1	Bit n	LED function	
		all Mobile Panels, all Comfort Panels	**Panel PCs**
1	0	Slow flashing	Flashing
1	1	On permanently	On permanently

See also

Communication with SIMATIC ET 200 CPU (Page 670)

2.10.5 Performance features of communication

2.10.5.1 Valid data types for SIMATIC ET 200 CPU

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	
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	

Data type	Length
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 670)

Device dependencies of SIMATIC ET 200 CPU (Page 730)

2.10.5.2 Device dependencies of SIMATIC ET 200 CPU

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 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

Basic Panels V14.0.1

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

Basic Panels V15.0

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

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

Mobile Panels V14.0.1

HMI devices	SIMATIC ET 200 CPU
KTP700 Mobile	Yes
KTP700F Mobile	Yes
KTP900 Mobile	Yes
KTP900F Mobile	Yes
KTP400F Mobile	Yes

Mobile Panels V15.0

HMI devices	SIMATIC ET 200 CPU
KTP700 Mobile	Yes
KTP700F Mobile	Yes

HMI devices	SIMATIC ET 200 CPU
KTP900 Mobile	Yes
KTP900F Mobile	Yes
KTP400F Mobile	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
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

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 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
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

HMI devices	SIMATIC ET 200 CPU
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

Comfort Panels V14.0.1

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
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

HMI devices	SIMATIC ET 200 CPU
TP1900 Comfort Portrait	Yes
TP2200 Comfort	Yes
TP2200 Comfort Portrait	Yes

Comfort Panels V15.0

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 V12.0

HMI devices	SIMATIC ET 200 CPU
WinCC RT Advanced	Yes

Runtime V13.0

HMI devices	SIMATIC ET 200 CPU
WinCC RT Advanced	Yes

Runtime V13.0.1

HMI devices	SIMATIC ET 200 CPU
WinCC RT Advanced	Yes

Runtime V14.0

HMI devices	SIMATIC ET 200 CPU
WinCC RT Advanced	Yes
WinCC RT Professional	Yes

Runtime V14.0.1

HMI devices	SIMATIC ET 200 CPU
WinCC RT Advanced	Yes
WinCC RT Professional	Yes

Runtime V15.0

HMI devices	SIMATIC ET 200 CPU
WinCC RT Advanced	Yes
WinCC RT Professional	Yes

See also

Communication with SIMATIC ET 200 CPU (Page 670)
Valid data types for SIMATIC ET 200 CPU (Page 729)

2.10.6 Configuring connections in the "Connections" editor

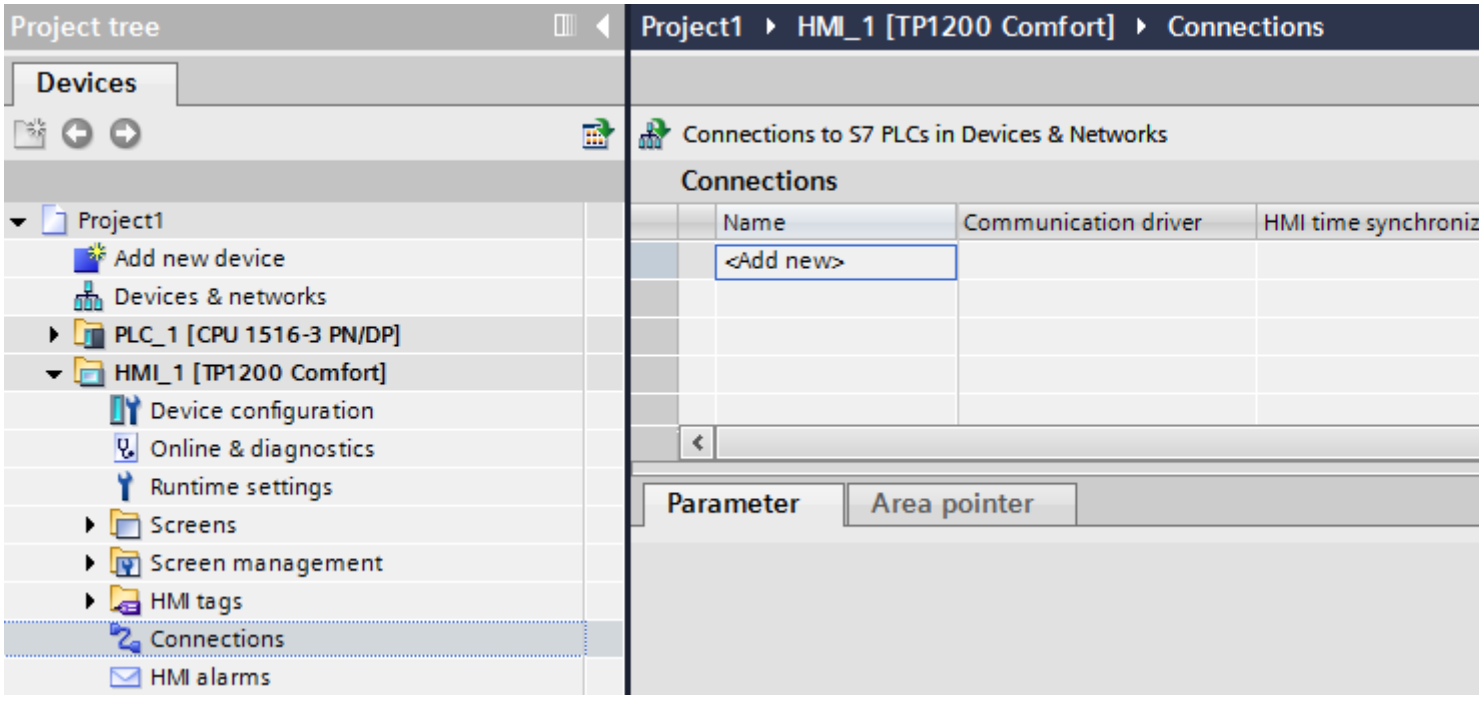
2.10.6.1 Creating a PROFINET connection

Requirements

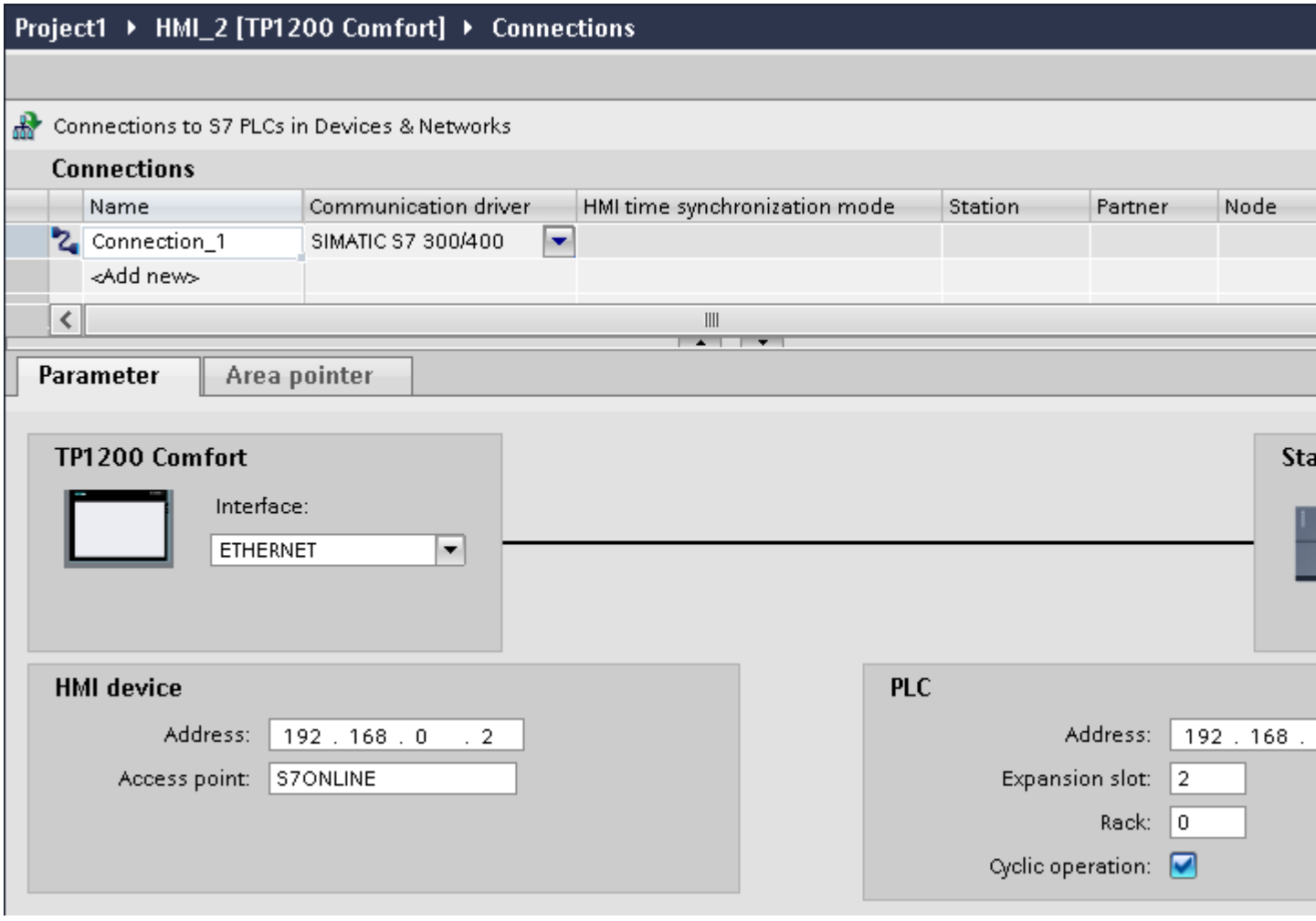
- A project is open.
- An HMI device with a PROFINET interface has been created.

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 with SIMATIC ET 200 CPU (Page 670)

Creating a PROFIBUS DP connection (Page 740)

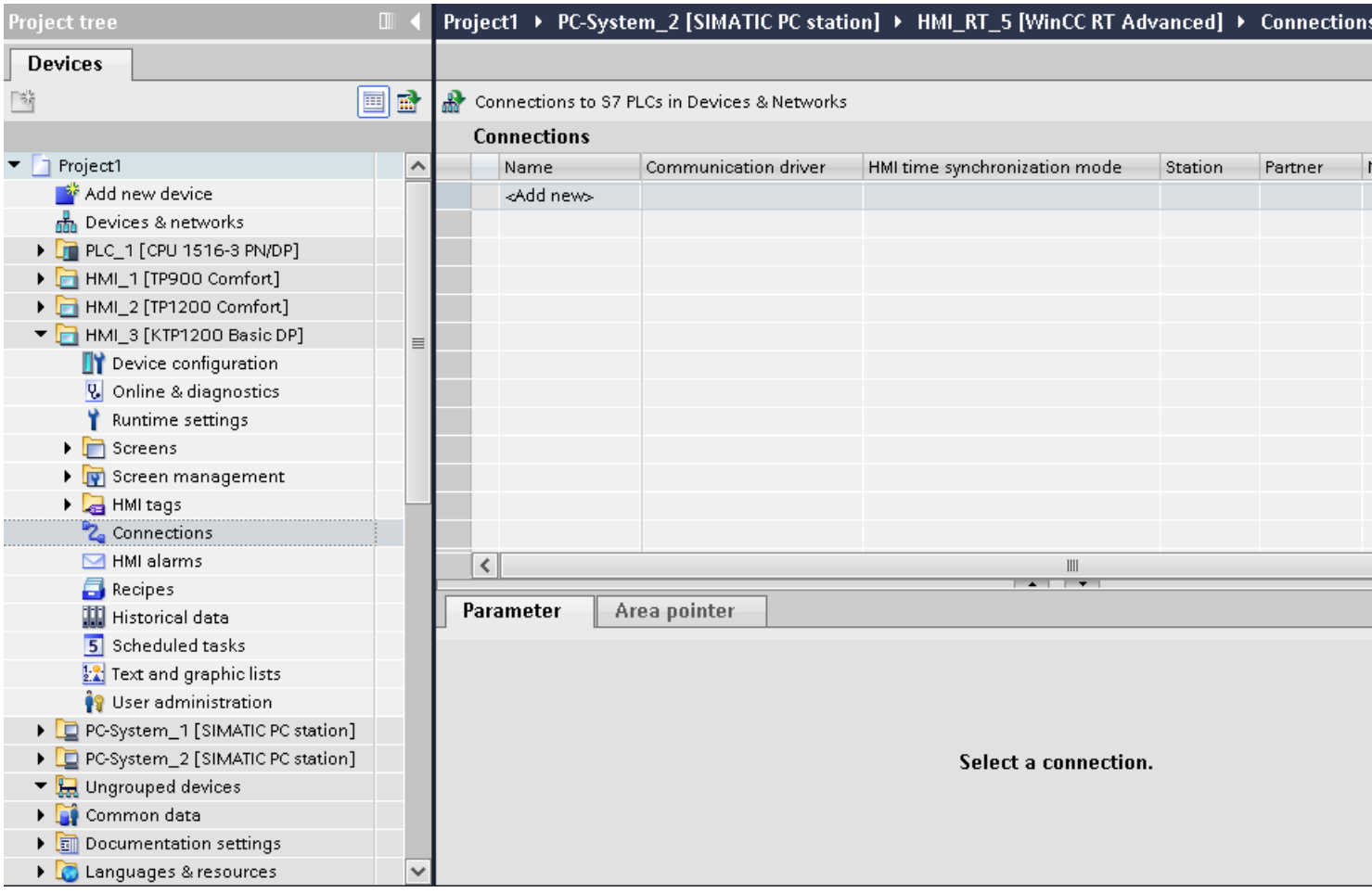
2.10.6.2 Creating a PROFIBUS DP connection

Requirements

- A project is open.
- An HMI device with a PROFIBUS interface has been created.

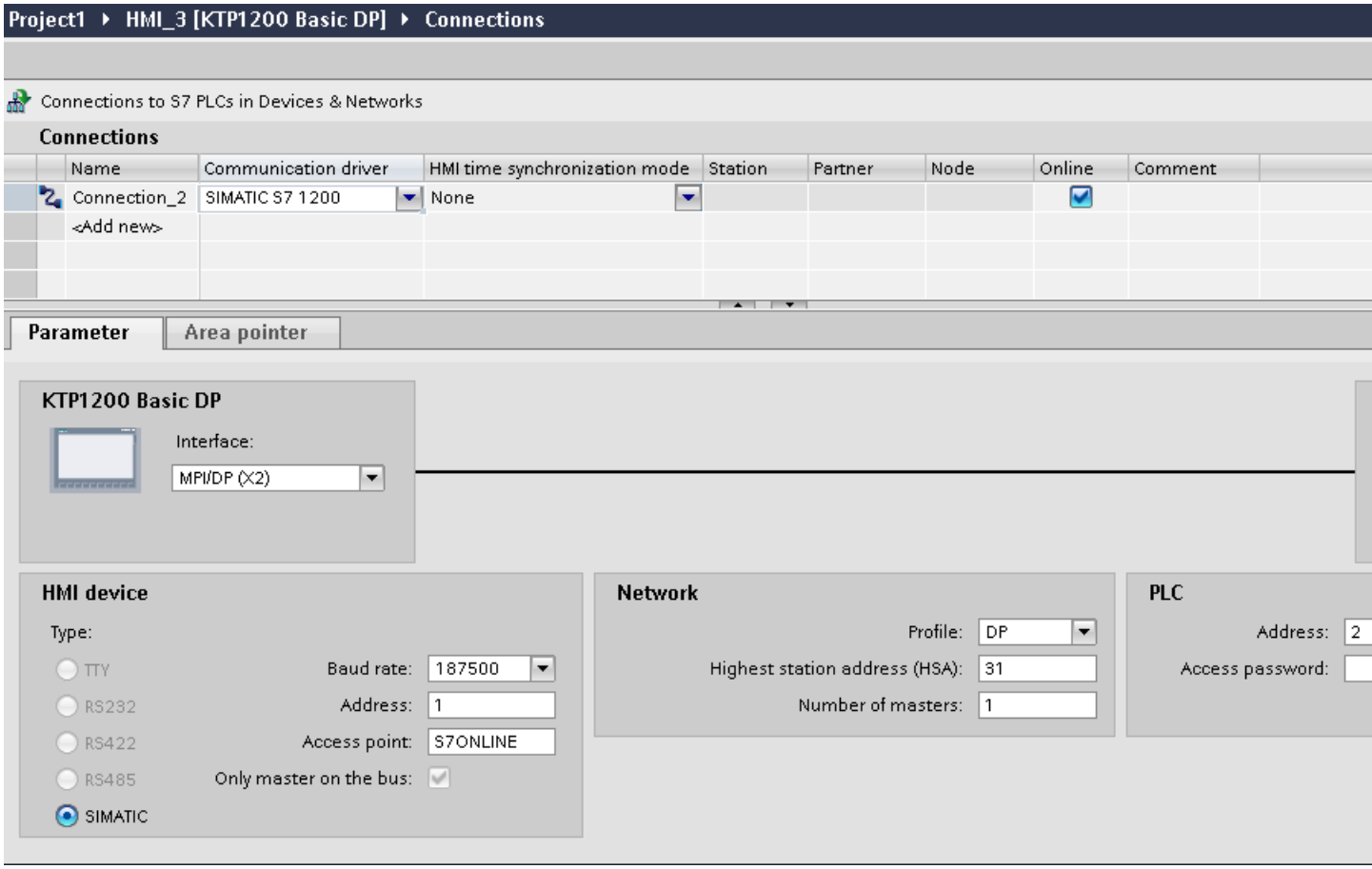
Procedure

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

Creating a PROFINET connection (Page 737)

Communication with SIMATIC ET 200 CPU (Page 670)

2.10.6.3 Connection parameters

PROFIBUS parameters

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.
- "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.
- "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

Ethernet parameters (Page 744)

Communication with SIMATIC ET 200 CPU (Page 670)

Parameters for the connection (SIMATIC S7 1500) (Page 745)

Ethernet parameters

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 ET 200 CPU (Page 670)

PROFIBUS parameters (Page 742)

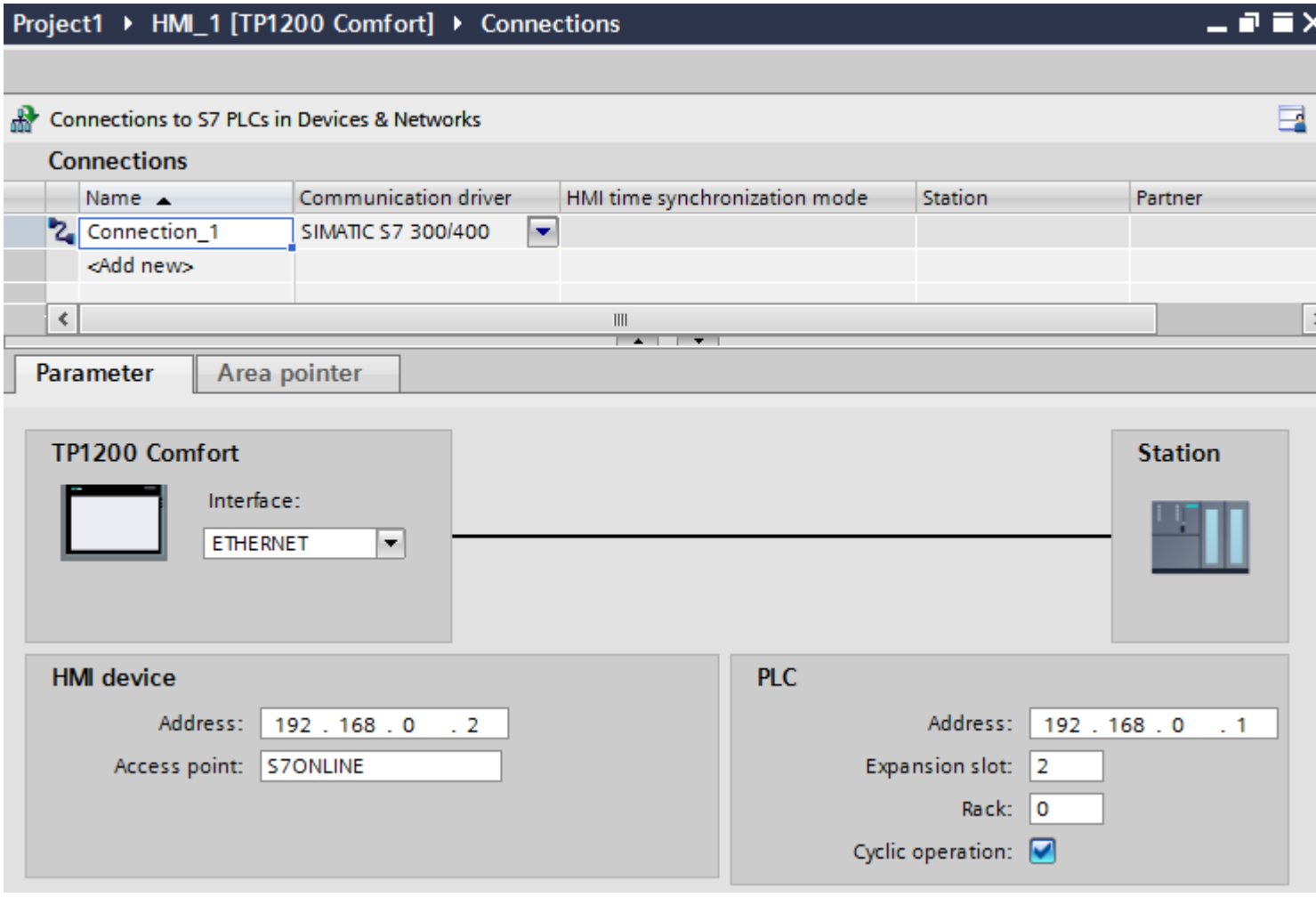
Parameters for the connection (SIMATIC S7 1500) (Page 745)

Parameters for the connection (SIMATIC S7 1500)

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

Ethernet parameters (Page 744)

PROFIBUS parameters (Page 742)

Communication with SIMATIC ET 200 CPU (Page 670)

2.10.7 Configuring connections in the "Connections" editor

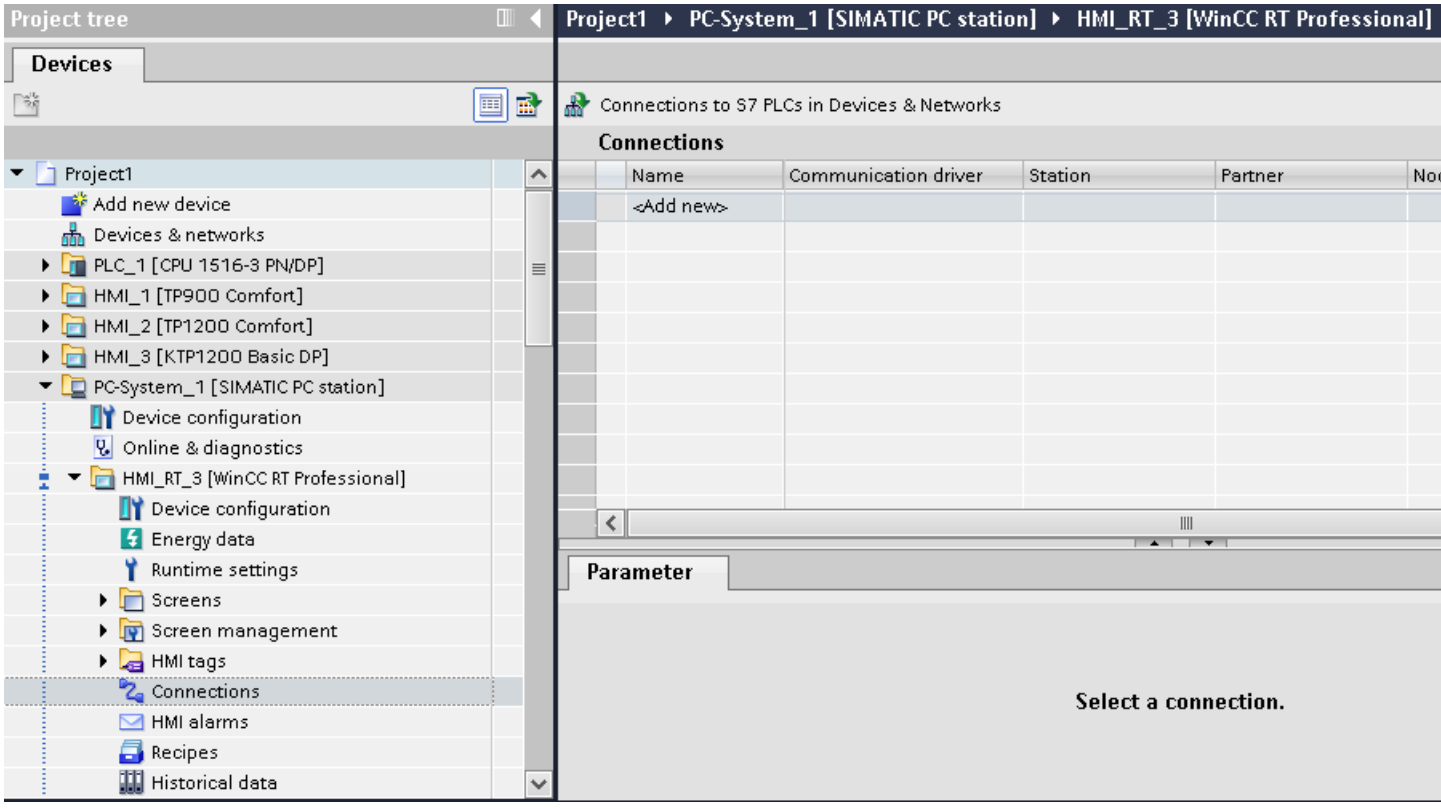
2.10.7.1 Creating a connection

Requirements

- A project is open.
- WinCC RT Advanced is 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 an interface of the HMI device in the Inspector window under "Parameters > Interface".
6. Set the parameters for the connection in the Inspector window.

Interfaces

Select one of the following interfaces in the Inspector window under "Parameters > WinCC RT Advanced > Interfaces".:

- Industrial Ethernet
- MPI
- PROFIBUS

For additional information on the parameters of the interfaces see:

Connection parameters (Page 748)

See also

Communication with SIMATIC ET 200 CPU (Page 670)

Connection parameters (Page 748)

2.10.7.2 Connection parameters

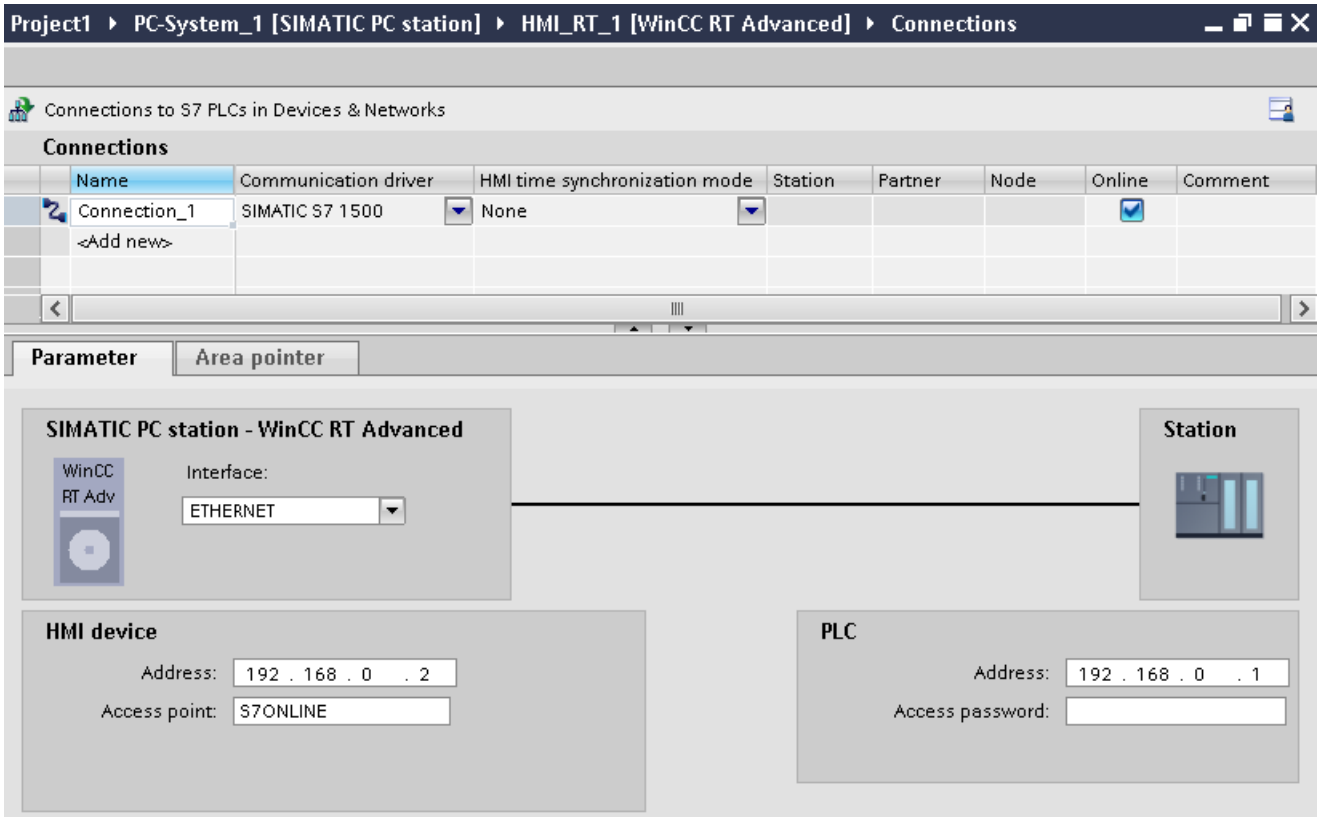
ETHERNET

Introduction

You assign the parameters for the HMI device and the controller in the network under "Parameters".

The parameters described in the following apply to the following interfaces:

- TCP/IP



HMI device

- "Address"
  Enter the IP address of the HMI device under "Address".
- "Access point"
  Specifies the access point for the PG/PC interface that can be used to reach the communication partner.

Controller

- "Address"
  Enter the IP address of the controller under "Address".
- "Access password"
  Enter a password in the "Access password" field. This password must match the one you saved to the controller.

Note

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

No connection to the controller is established if the "Complete protection" protection level is stored on the controller and you do not enter a password.

See also

Creating a connection (Page 746)

MPI/DP (Page 749)

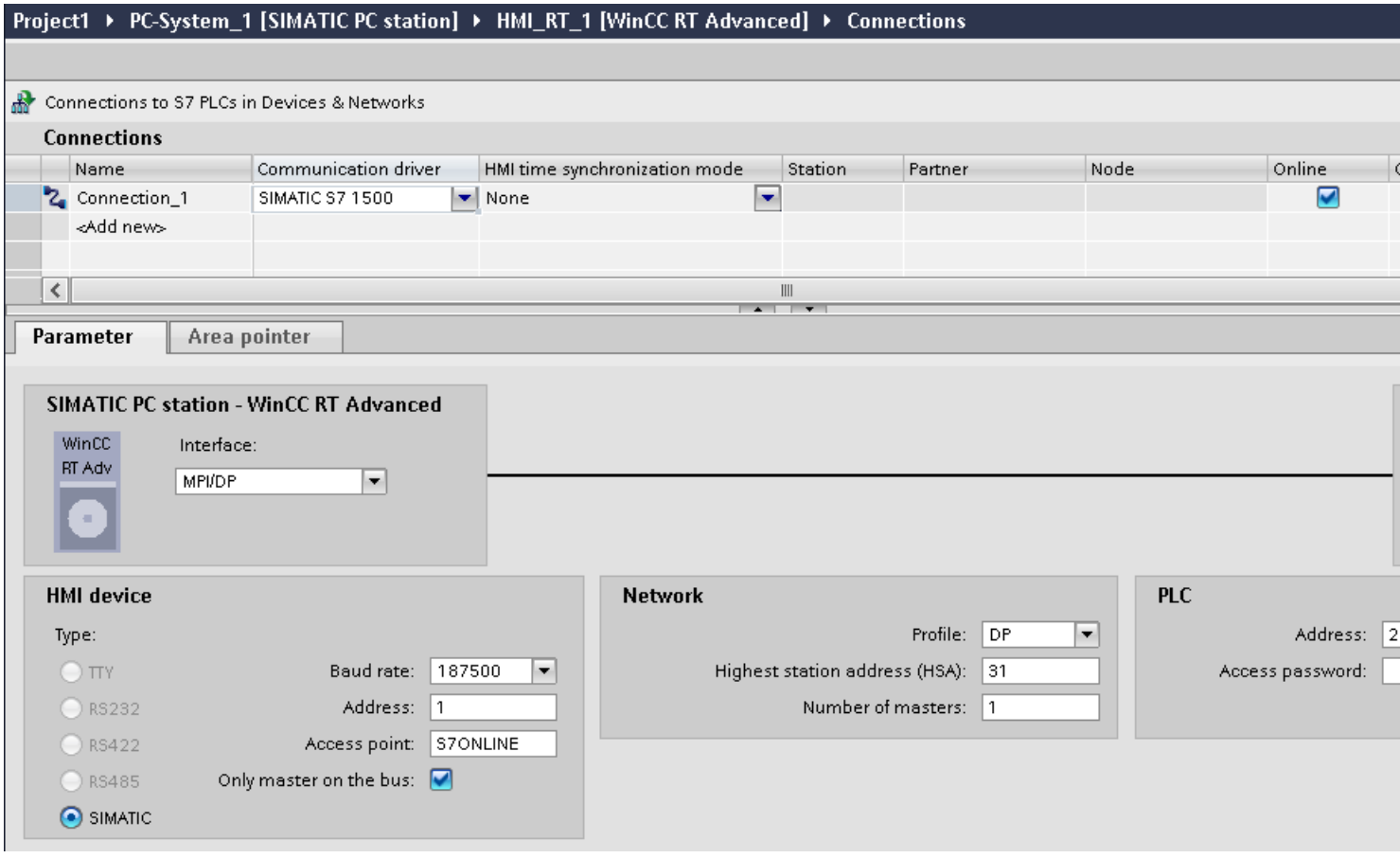
MPI/DP

Introduction

You assign the parameters for the HMI device and the controller in the network under "Parameters".

The parameters described in the following apply to the following interfaces:

- PROFIBUS



HMI device

- "Baud rate"
  Select the baud rate of the HMI device under "Baud rate".
- "Address"
  Enter the address of the HMI device under "Address".
- "Access point"
  Enter the logical device name under "Access point".

Controller

- "Address"
  Enter the address of the controller under "Address".
- "Access password"
  If you have configured the degree of protection "Complete protection" for the controller, enter the password for the controller under "Access password".

See also

Creating a connection (Page 746)
ETHERNET (Page 748)

2.10.8 Configuring connections in the "Connections" editor

2.10.8.1 Creating a connection

Requirements

- A project is open.
- WinCC RT Professional is 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 an interface of the HMI device in the Inspector window under "Parameters > Interface".
6. Set the parameters for the connection in the Inspector window.

Interfaces

Select one of the following interfaces in the Inspector window under "Parameters > WinCC RT Professional > Interfaces":

- TCP/IP
- PROFIBUS

For additional information on the parameters of the interfaces see:

Connection parameters (Page 752)

See also

Communication with SIMATIC ET 200 CPU (Page 670)
Connection parameters (Page 752)

2.10.8.2 Connection parameters

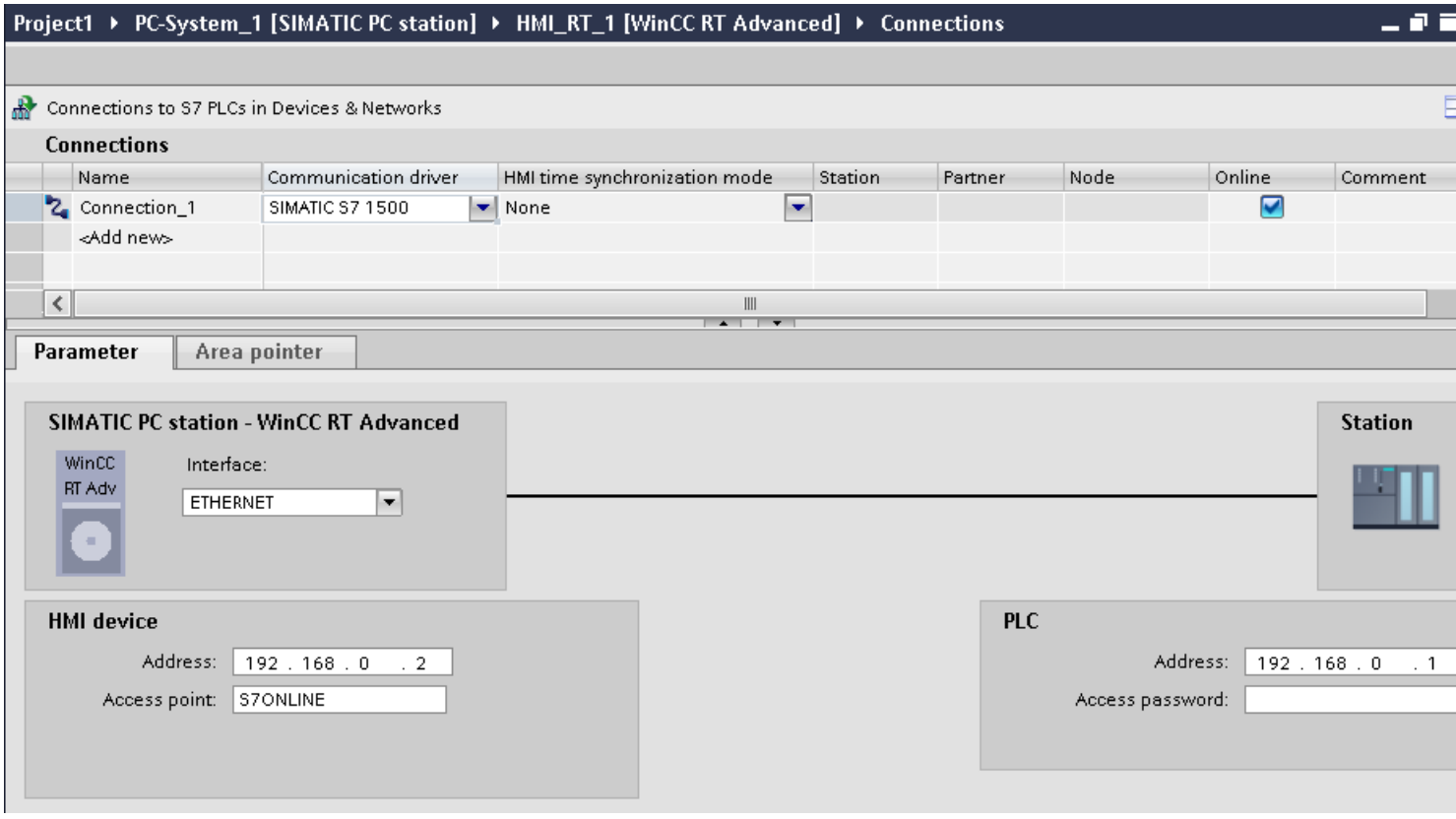
ETHERNET

Introduction

You assign the parameters for the HMI device and the controller in the network under "Parameters".

The parameters described in the following apply to the following interfaces:

- TCP/IP



HMI device

- "Address"
  Enter the IP address of the HMI device under "Address".
- "Access point"
  Specifies the access point for the PG/PC interface that can be used to reach the communication partner.

Controller

- "Address"
  Enter the IP address of the controller under "Address".
- "Access password"
  Enter a password in the "Access password" field. This password must match the one you saved to the controller.

Note

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

No connection to the controller is established if the "Complete protection" protection level is stored on the controller and you do not enter a password.

See also

Communication with SIMATIC ET 200 CPU (Page 670)

MPI/DP (Page 753)

PROFIBUS parameters (Page 755)

Ethernet parameters (Page 757)

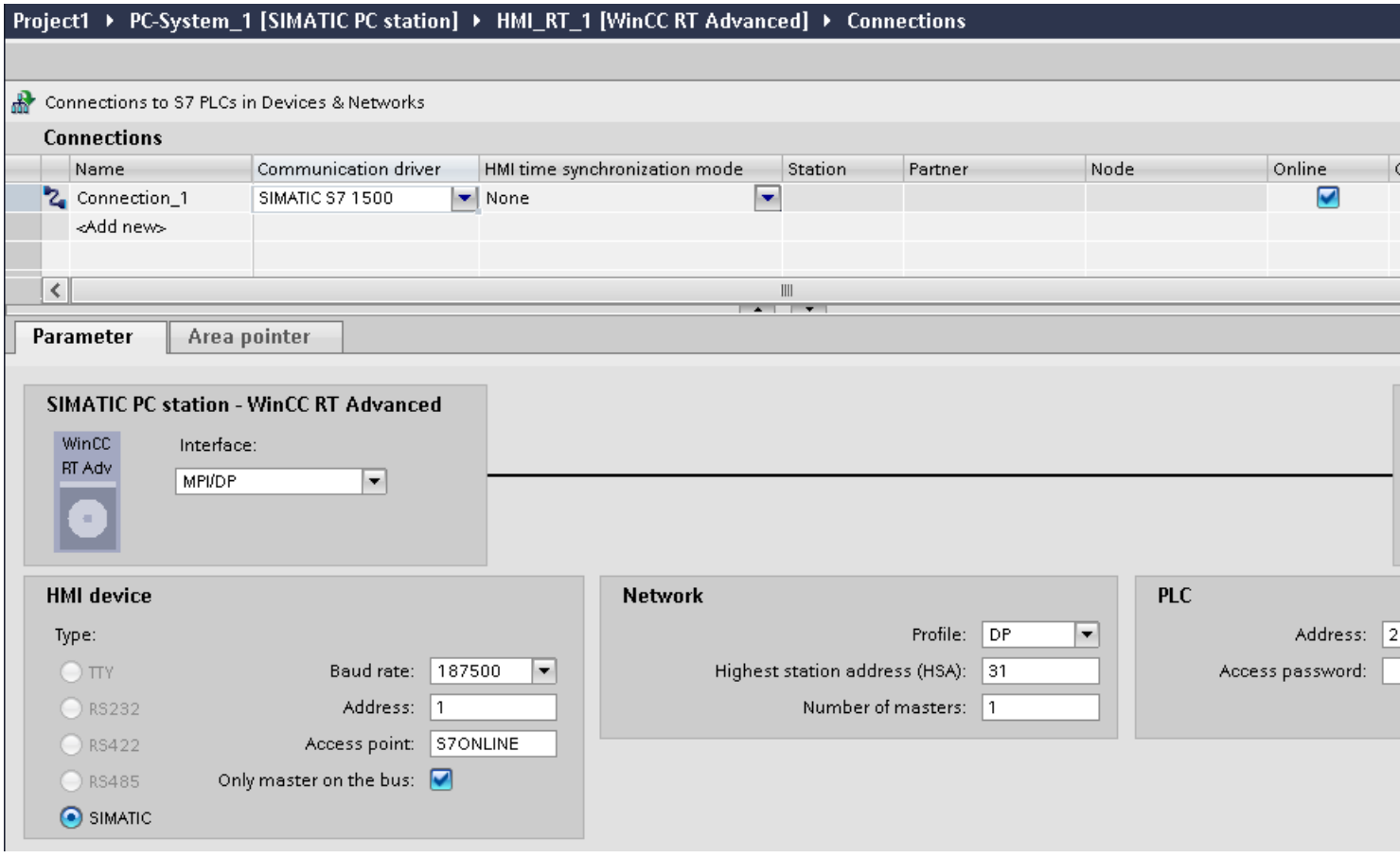
MPI/DP

Introduction

You assign the parameters for the HMI device and the controller in the network under "Parameters".

The parameters described in the following apply to the following interfaces:

- PROFIBUS



HMI device

- "Baud rate"
  Select the baud rate of the HMI device under "Baud rate".
- "Address"
  Enter the address of the HMI device under "Address".
- "Access point"
  Enter the logical device name under "Access point".

Controller

- "Address"
  Enter the address of the controller under "Address".
- "Access password"
  If you have configured the degree of protection "Complete protection" for the controller, enter the password for the controller under "Access password".

See also

Communication with SIMATIC ET 200 CPU (Page 670)

ETHERNET (Page 752)

PROFIBUS parameters (Page 755)

Ethernet parameters (Page 757)

PROFIBUS parameters

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.
- "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.
- "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 ET 200 CPU (Page 670)

ETHERNET (Page 752)

MPI/DP (Page 753)

Ethernet parameters (Page 757)

Ethernet parameters

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 ET 200 CPU (Page 670)

ETHERNET (Page 752)

MPI/DP (Page 753)

PROFIBUS parameters (Page 755)

2.10.9 Configuring time synchronization

2.10.9.1 Time synchronization

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

Communication with SIMATIC ET 200 CPU (Page 670)
Time synchronization restrictions (Page 759)
Configuring time synchronization for integrated connections (Page 760)
Configuring time synchronization for non-integrated connections (Page 761)

2.10.9.2 Time synchronization restrictions

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	-
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.

See also

Communication with SIMATIC ET 200 CPU (Page 670)

Time synchronization (Page 758)

Configuring time synchronization for integrated connections (Page 760)

Configuring time synchronization for non-integrated connections (Page 761)

2.10.9.3 Configuring time synchronization for integrated connections

Introduction

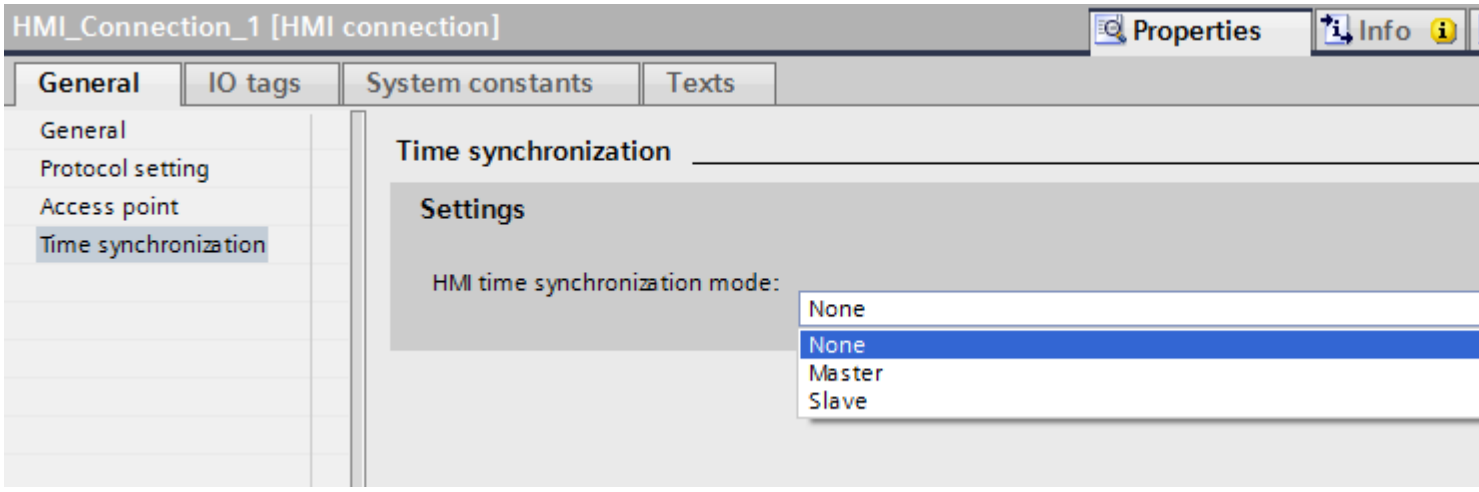
You configure time synchronization for an integrated connection in the "Devices & Networks" editor.

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

Communication with SIMATIC ET 200 CPU (Page 670)
Time synchronization (Page 758)
Time synchronization restrictions (Page 759)
Configuring time synchronization for non-integrated connections (Page 761)

2.10.9.4 Configuring time synchronization for non-integrated connections

Introduction

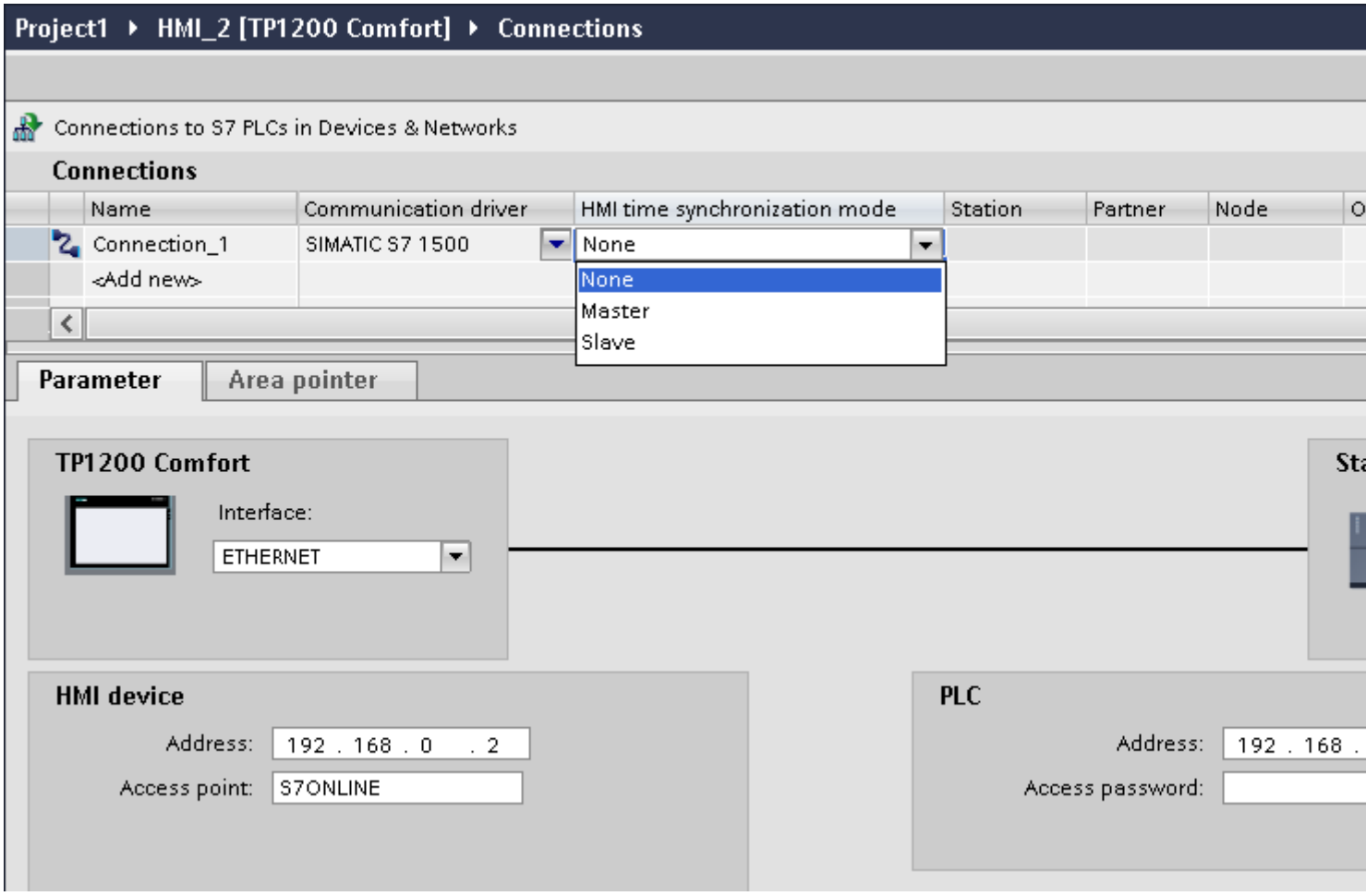
You configure time synchronization for a non-integrated connection in the "Connections" editor.

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

Communication with SIMATIC ET 200 CPU (Page 670)

Time synchronization (Page 758)

Time synchronization restrictions (Page 759)

Configuring time synchronization for integrated connections (Page 760)

2.11 Communicating with SIMATIC S7 200

2.11.1 Communication with SIMATIC S7 200

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 763)

2.11.2 Creating a connection to SIMATIC S7 200

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.

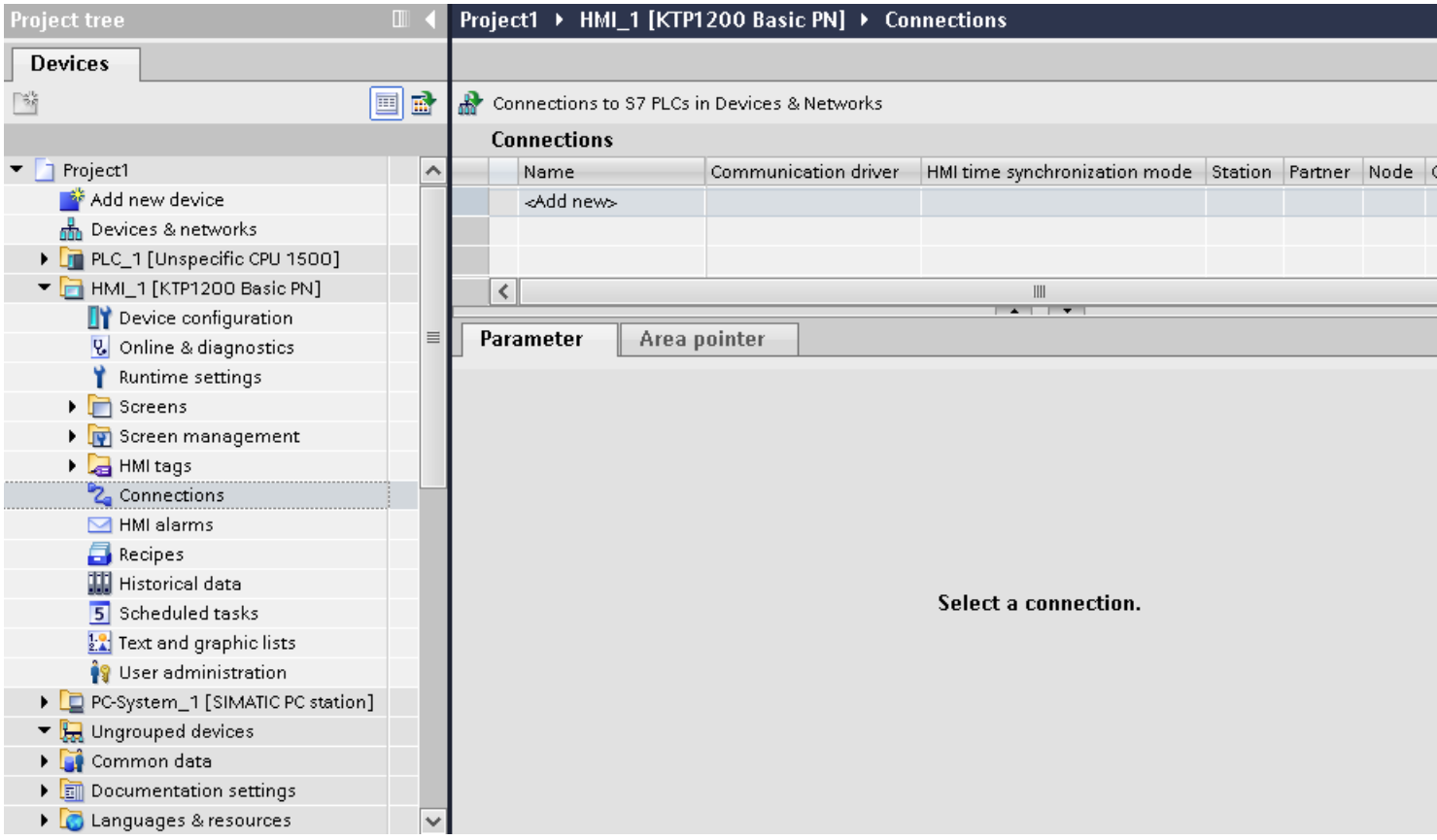
Requirements

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

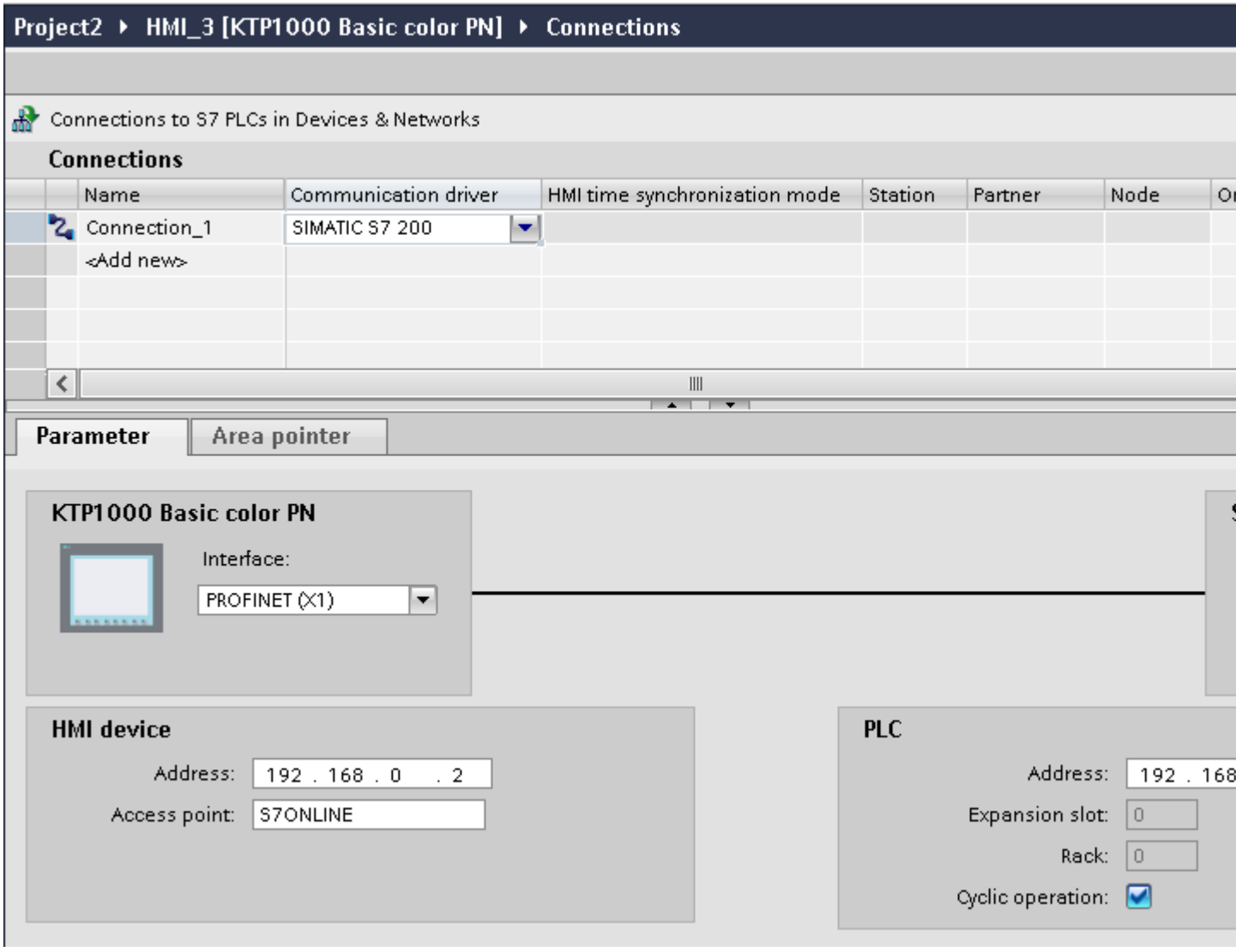
Procedure

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 766)" section for more detailed information.

See also

Communication with SIMATIC S7 200 (Page 763)

Parameters for the connection (Page 766)

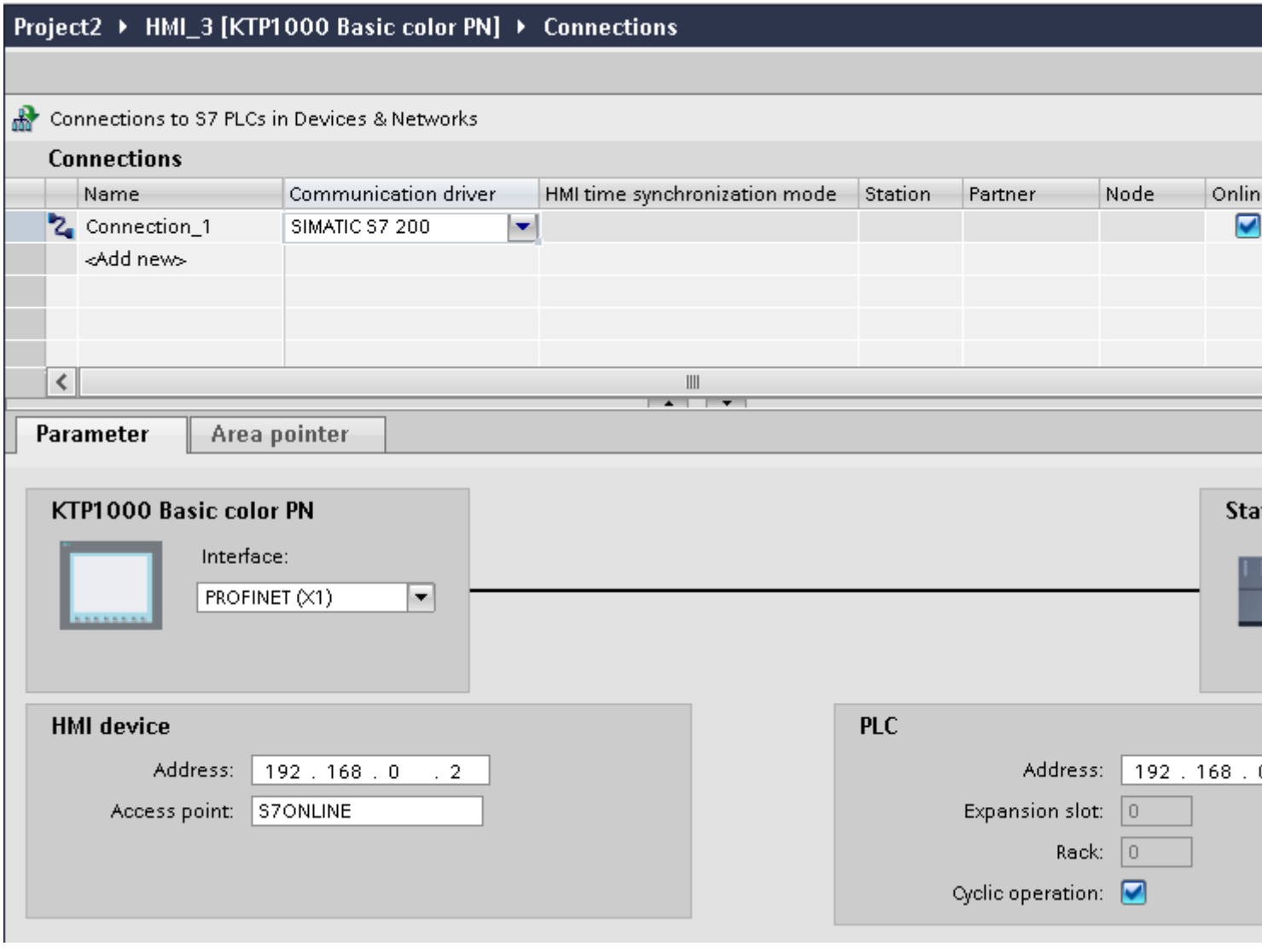
2.11.3 Parameters for the connection

2.11.3.1 Parameters for the connection (SIMATIC S7 200)

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 763)
Ethernet parameters (Page 767)
PROFIBUS parameters (Page 768)
MPI parameters (Page 770)
PPI parameters (Page 771)
Cyclic operation (Page 773)

2.11.3.2 Ethernet parameters

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 763)

Parameters for the connection (SIMATIC S7 200) (Page 766)

PROFIBUS parameters (Page 768)

MPI parameters (Page 770)

PPI parameters (Page 771)

Cyclic operation (Page 773)

2.11.3.3 PROFIBUS parameters

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.

- "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.
- "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 763)

Parameters for the connection (SIMATIC S7 200) (Page 766)

Ethernet parameters (Page 767)
MPI parameters (Page 770)
PPI parameters (Page 771)
Cyclic operation (Page 773)

2.11.3.4 MPI parameters

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.
- "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 763)

Parameters for the connection (SIMATIC S7 200) (Page 766)

Ethernet parameters (Page 767)

PROFIBUS parameters (Page 768)

PPI parameters (Page 771)

Cyclic operation (Page 773)

2.11.3.5 PPI parameters

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.
- "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 763)

Parameters for the connection (SIMATIC S7 200) (Page 766)

Ethernet parameters (Page 767)

PROFIBUS parameters (Page 768)

MPI parameters (Page 770)

Cyclic operation (Page 773)

2.11.3.6 Cyclic operation

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 763)

Parameters for the connection (SIMATIC S7 200) (Page 766)

Ethernet parameters (Page 767)

PROFIBUS parameters (Page 768)

MPI parameters (Page 770)

PPI parameters (Page 771)

2.11.4 Data exchange

2.11.4.1 Data exchange using area pointers

General information on area pointers

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 153)

See also

Communication with SIMATIC S7 200 (Page 763)

Configuration of area pointers (Page 153)

"Screen number" area pointer (Page 774)

"Date/time" area pointer (Page 775)

"Date/time PLC" area pointer (Page 777)

"Coordination" area pointer (Page 778)

"Project ID" area pointer (Page 779)

"Job mailbox" area pointer (Page 781)

"Screen number" area pointer

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

Communication with SIMATIC S7 200 (Page 763)

General information on area pointers (Page 774)

"Date/time" area pointer (Page 775)

"Date/time PLC" area pointer (Page 777)

"Coordination" area pointer (Page 778)

"Project ID" area pointer (Page 779)

"Job mailbox" area pointer (Page 781)

"Date/time" area pointer

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

Communication with SIMATIC S7 200 (Page 763)

General information on area pointers (Page 774)

"Screen number" area pointer (Page 774)

"Date/time PLC" area pointer (Page 777)

"Coordination" area pointer (Page 778)

"Project ID" area pointer (Page 779)

"Job mailbox" area pointer (Page 781)

"Date/time PLC" area pointer

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

Communication with SIMATIC S7 200 (Page 763)

General information on area pointers (Page 774)

"Screen number" area pointer (Page 774)

"Date/time" area pointer (Page 775)

"Coordination" area pointer (Page 778)

"Project ID" area pointer (Page 779)

"Job mailbox" area pointer (Page 781)

"Coordination" area pointer

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

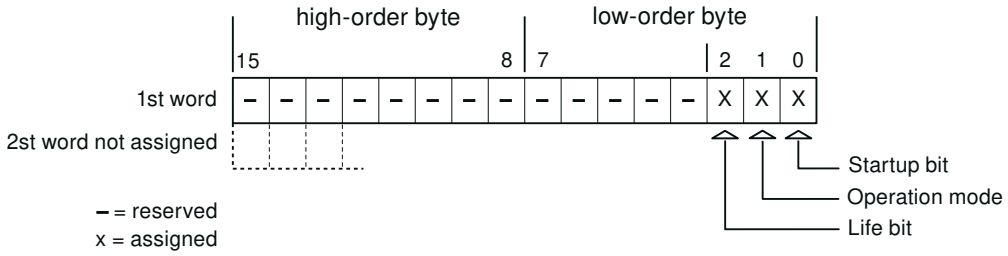
By default, the "Coordination" area pointer has the length of one word and cannot be changed.

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

Communication with SIMATIC S7 200 (Page 763)

General information on area pointers (Page 774)

"Screen number" area pointer (Page 774)

"Date/time" area pointer (Page 775)

"Date/time PLC" area pointer (Page 777)

"Project ID" area pointer (Page 779)

"Job mailbox" area pointer (Page 781)

"Project ID" area pointer

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

Communication with SIMATIC S7 200 (Page 763)

General information on area pointers (Page 774)

"Screen number" area pointer (Page 774)

"Date/time" area pointer (Page 775)

"Date/time PLC" area pointer (Page 777)

"Coordination" area pointer (Page 778)

"Job mailbox" area pointer (Page 781)

"Job mailbox" area pointer

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 to 23)
	Parameter 2	Left byte: Minutes (0-59) Right byte: Seconds (0 to 59)
	Parameter 3	-
15	**Setting the date (BCD coded)** [2)]	
	Parameter 1	Left byte: - Right byte: Weekday (1-7: Sunday-Saturday)
	Parameter 2	Left byte: Day (1-31) Right byte: Month (1 to 12)
	Parameter 3	Left byte: Year

No.	Function	
14	**Set time (BCD-coded)**	
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**	
	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	**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**	
	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
3)	The weekday is ignored on HMI device KTP 600 BASIC PN.

See also

Communication with SIMATIC S7 200 (Page 763)

General information on area pointers (Page 774)

"Screen number" area pointer (Page 774)

"Date/time" area pointer (Page 775)

"Date/time PLC" area pointer (Page 777)

"Coordination" area pointer (Page 778)

"Project ID" area pointer (Page 779)

"Data record" area pointer

"Data record" area pointer

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

Communication with SIMATIC S7 200 (Page 763)

Sequence of a transfer started by the operator in the recipe view (Page 784)

Sequence of the transfer triggered by a job mailbox (Page 785)

Sequence of the transfer when triggered by a configured function (Page 787)

Possible causes of error when transferring data records (Page 788)

Transfer without synchronization (Page 789)

Transfer with synchronization (Page 790)

Sequence of a transfer started by the operator in the recipe view

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.	

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

Communication with SIMATIC S7 200 (Page 763)

"Data record" area pointer (Page 783)

Sequence of the transfer triggered by a job mailbox (Page 785)

Sequence of the transfer when triggered by a configured function (Page 787)

Possible causes of error when transferring data records (Page 788)

Transfer without synchronization (Page 789)

Transfer with synchronization (Page 790)

Sequence of the transfer triggered by a job mailbox

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)	

	Left byte (LB)	Right byte (RB)
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.	

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

Communication with SIMATIC S7 200 (Page 763)
"Data record" area pointer (Page 783)
Sequence of a transfer started by the operator in the recipe view (Page 784)
Sequence of the transfer when triggered by a configured function (Page 787)
Possible causes of error when transferring data records (Page 788)
Transfer without synchronization (Page 789)
Transfer with synchronization (Page 790)

Sequence of the transfer when triggered by a configured function

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

Step	Action	
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

Communication with SIMATIC S7 200 (Page 763)

"Data record" area pointer (Page 783)

Sequence of a transfer started by the operator in the recipe view (Page 784)

Sequence of the transfer triggered by a job mailbox (Page 785)

Possible causes of error when transferring data records (Page 788)

Transfer without synchronization (Page 789)

Transfer with synchronization (Page 790)

Possible causes of error when transferring data records

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

Communication with SIMATIC S7 200 (Page 763)

"Data record" area pointer (Page 783)

Sequence of a transfer started by the operator in the recipe view (Page 784)

Sequence of the transfer triggered by a job mailbox (Page 785)

Sequence of the transfer when triggered by a configured function (Page 787)

Transfer without synchronization (Page 789)

Transfer with synchronization (Page 790)

Transfer without synchronization

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

Communication with SIMATIC S7 200 (Page 763)

"Data record" area pointer (Page 783)

Sequence of a transfer started by the operator in the recipe view (Page 784)

Sequence of the transfer triggered by a job mailbox (Page 785)

Sequence of the transfer when triggered by a configured function (Page 787)

Possible causes of error when transferring data records (Page 788)

Transfer with synchronization (Page 790)

Transfer with synchronization

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

Communication with SIMATIC S7 200 (Page 763)

"Data record" area pointer (Page 783)

Sequence of a transfer started by the operator in the recipe view (Page 784)

Sequence of the transfer triggered by a job mailbox (Page 785)

Sequence of the transfer when triggered by a configured function (Page 787)

Possible causes of error when transferring data records (Page 788)

Transfer without synchronization (Page 789)

2.11.4.2 Trends

General information on trends

Trends

A trend is the graphical representation of one or more values from the PLC. The value is read out either time- or bit-triggered, depending on the configuration.

For additional information see:

AUTOHOTSPOT

Note

The value is read out time-triggered for Basic Panels.

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.

Bit-triggered trends

Through a trigger bit set in the trend request tag, the HMI device either reads in a trend value or an entire trend buffer. This setting is defined in the configuration. Bit-triggered trends are normally used to represent fast 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 "HMI tags" editor and connected to trend areas during configuration. The HMI device and PLC then communicate with each other via 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)

See also

Communication with SIMATIC S7 200 (Page 763)

Trend request and trend transfer (Page 792)

Permitted data types for trends (Page 793)

Trend request and trend transfer

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 closing the screen, the HMI device resets the relevant bits in the trend request area.

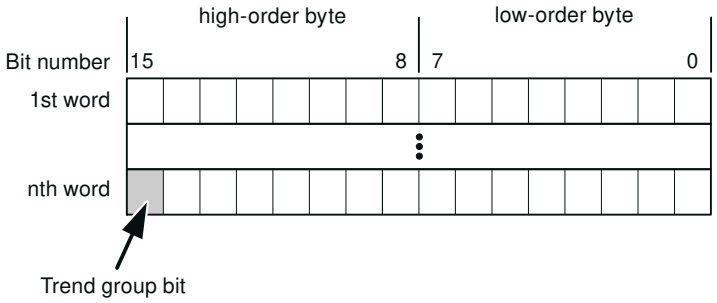
The trend request area can be used for evaluation purposes in the PLC to determine which trend is currently being displayed on the HMI device. Trends can also be triggered without evaluation of the trend request area.

Trend transfer area 1

This area is used to trigger trends. You must set the bit assigned to the trend in the trend transfer area and set the trend group bit in your control program. 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.

The PLC writes to Buffer 2 while the HMI device reads values from Buffer 1, and writes to Buffer 1 when the HMI device is reading Buffer 2. This prevents the PLC from overwriting trend values while the trend is being read by the HMI device.

See also

Communication with SIMATIC S7 200 (Page 763)

General information on trends (Page 791)

Permitted data types for trends (Page 793)

Permitted data types for trends

For SIMATIC S7

You assign one bit to each trend during configuration. Tags and array tags of the "Word" or "Int" data type are permitted.

See also

Communication with SIMATIC S7 200 (Page 763)

General information on trends (Page 791)

Trend request and trend transfer (Page 792)

2.11.4.3 Alarms

Configuring alarms

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

Communication with SIMATIC S7 200 (Page 763)

Acknowledgment of alarms (Page 795)

Acknowledgment of alarms

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

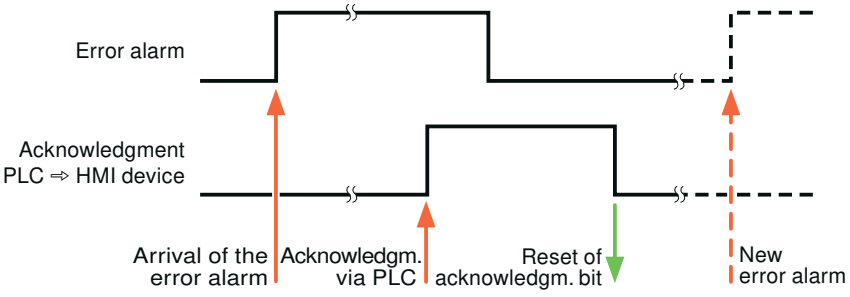
Acknowledgment by the PLC

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.

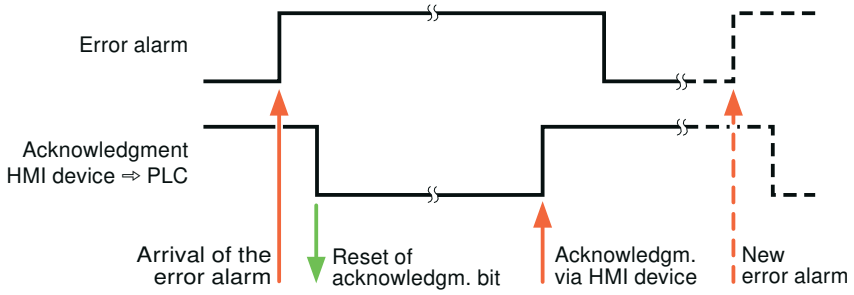
If a discrete alarm subject to acknowledgment is acknowledged by the HMI device, then the corresponding bit in the assigned acknowledgment tag will be set. The entire acknowledgment tag is then written to the PLC by the HMI device. This allows the PLC to recognize that a certain alarm message has been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

Communication with SIMATIC S7 200 (Page 763)

Configuring alarms (Page 794)

2.11.5 Performance features of communication

2.11.5.1 Device dependency S7 200

Communication with the SIMATIC S7-200 PLC

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 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
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

Basic Panels V14.0.1

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 V15.0

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

Mobile Panels V12.0

HMI devices	SIMATIC S7-200
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 S7-200
KTP700 Mobile	Yes
KTP700F Mobile	Yes
KTP900 Mobile	Yes
KTP900F Mobile	Yes

Mobile Panels V14.0

HMI devices	SIMATIC S7-200
KTP700 Mobile	Yes
KTP700F Mobile	Yes
KTP900 Mobile	Yes
KTP900F Mobile	Yes
KTP400F Mobile	Yes

Mobile Panels V14.0.1

HMI devices	SIMATIC S7-200
KTP700 Mobile	Yes
KTP700F Mobile	Yes
KTP900 Mobile	Yes
KTP900F Mobile	Yes
KTP400F Mobile	Yes

Mobile Panels V15.0

HMI devices	SIMATIC S7-200
KTP700 Mobile	Yes
KTP700F Mobile	Yes
KTP900 Mobile	Yes
KTP900F Mobile	Yes
KTP400F Mobile	Yes

Comfort Panels V12.0

HMI devices	SIMATIC S7-200
KP400 Comfort	Yes
KTP400 Comfort	Yes
KTP400 Comfort Portrait	Yes
KP700 Comfort	Yes

HMI devices	SIMATIC S7-200
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

HMI devices	SIMATIC S7-200
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 S7-200
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 V14.0

HMI devices	SIMATIC S7-200
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

HMI devices	SIMATIC S7-200
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

Comfort Panels V14.0.1

HMI devices	SIMATIC S7-200
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

Comfort Panels V15.0

HMI devices	SIMATIC S7-200
KP400 Comfort	Yes
KTP400 Comfort	Yes
KTP400 Comfort Portrait	Yes

HMI devices	SIMATIC S7-200
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 V12.0

HMI devices	SIMATIC S7-200
WinCC RT Advanced	Yes

Runtime V13.0

HMI devices	SIMATIC S7-200
WinCC RT Advanced	Yes

Runtime V13.0.1

HMI devices	SIMATIC S7-200
WinCC RT Advanced	Yes

Runtime V14.0

HMI devices	SIMATIC S7-200
WinCC RT Advanced	Yes
WinCC RT Professional	No

Runtime V14.0.1

HMI devices	SIMATIC S7-200
WinCC RT Advanced	Yes
WinCC RT Professional	No

Runtime V15.0

HMI devices	SIMATIC S7-200
WinCC RT Advanced	Yes
WinCC RT Professional	No

See also

Communication with SIMATIC S7 200 (Page 763)
Permitted data types for SIMATIC S7 200 (Page 804)

2.11.5.2 Permitted data types for SIMATIC S7 200

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 763)

Device dependency S7 200 (Page 797)

2.12 Communicating with SIMATIC LOGO!

2.12.1 Communication with SIMATIC LOGO!

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 806)

2.12.2 Creating a connection to SIMATIC LOGO!

Introduction

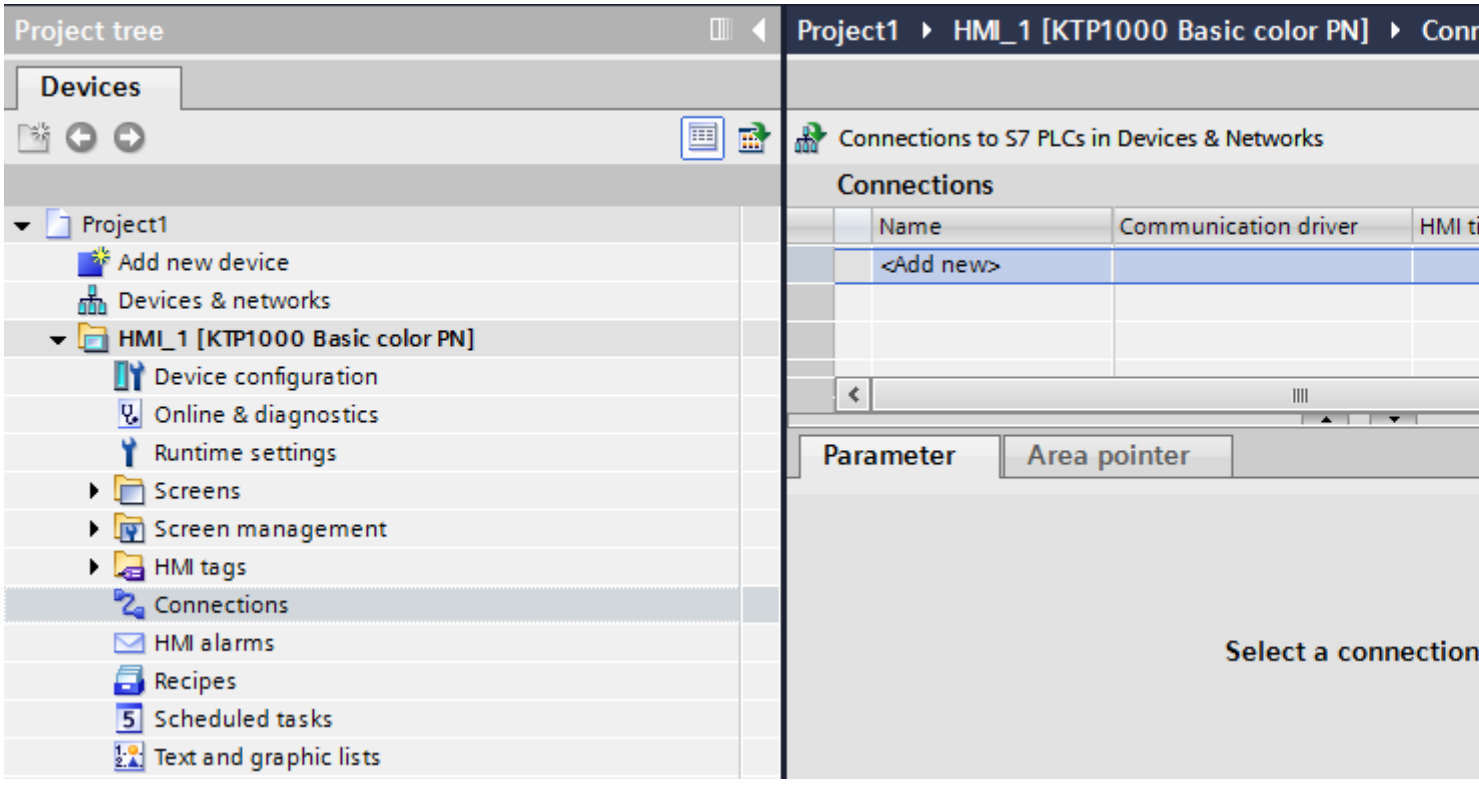
You configure a connection to the SIMATIC LOGO! controller in the "Connections" editor of the HMI device. The interfaces are named differently depending on the HMI device.

Requirements

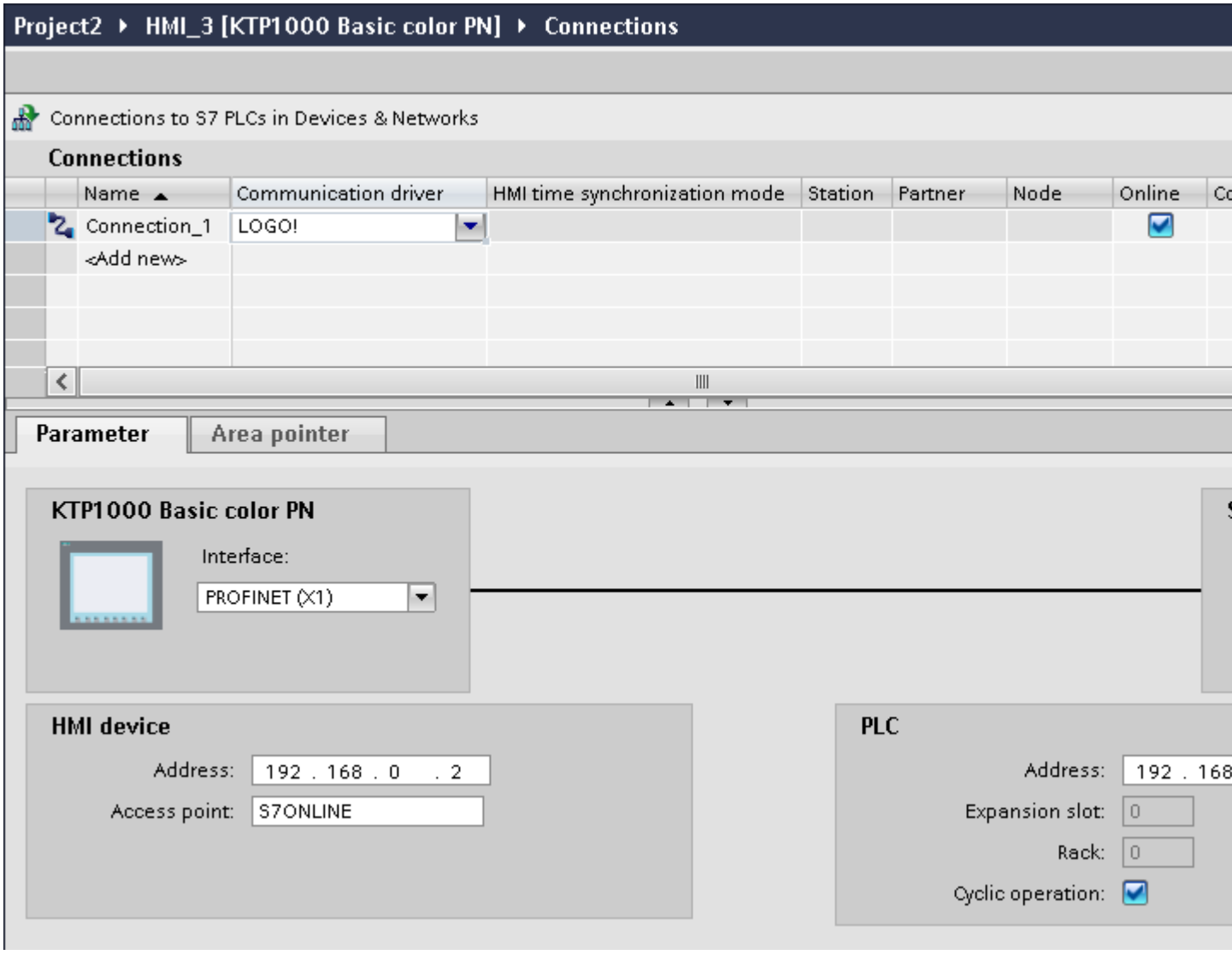
- A project is open.
- An HMI device has been created.

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 "Connection parameters (Page 808)" section for more detailed information.

See also

Communication with SIMATIC LOGO! (Page 805)

Connection parameters (Page 808)

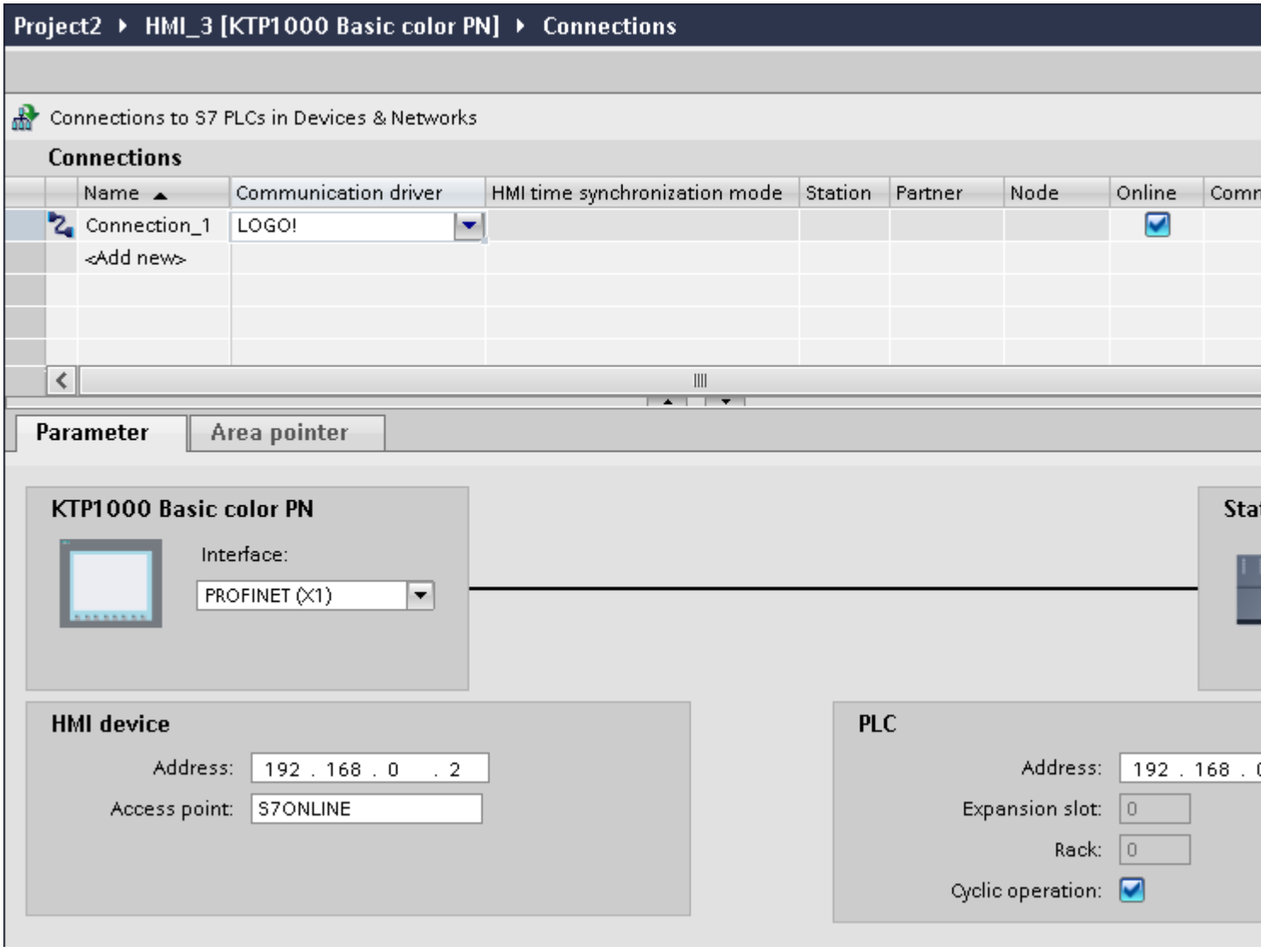
2.12.3 Connection parameters

2.12.3.1 Connection parameters

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 805)

Ethernet parameters (Page 809)

Cyclic operation (Page 810)

2.12.3.2 Ethernet parameters

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 805)

Connection parameters (Page 808)

Cyclic operation (Page 810)

2.12.3.3 Cyclic operation

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 805)
Connection parameters (Page 808)
Ethernet parameters (Page 809)

2.12.4 Data exchange

2.12.4.1 Trends

General information on trends

Trends

A trend is the graphical representation of one or more values from the PLC. The value is read out either time- or bit-triggered, depending on the configuration.

For additional information see:

AUTOHOTSPOT

Note

The value is read out time-triggered for Basic Panels.

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.

Bit-triggered trends

Through a trigger bit set in the trend request tag, the HMI device either reads in a trend value or an entire trend buffer. This setting is defined in the configuration. Bit-triggered trends are normally used to represent fast 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 "HMI tags" editor and connected to trend areas during configuration. The HMI device and PLC then communicate with each other via 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)

See also

Communication with SIMATIC LOGO! (Page 805)

Trend request and trend transfer (Page 812)

Permitted data types for trends (Page 813)

Trend request and trend transfer

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 closing the screen, the HMI device resets the relevant bits in the trend request area.

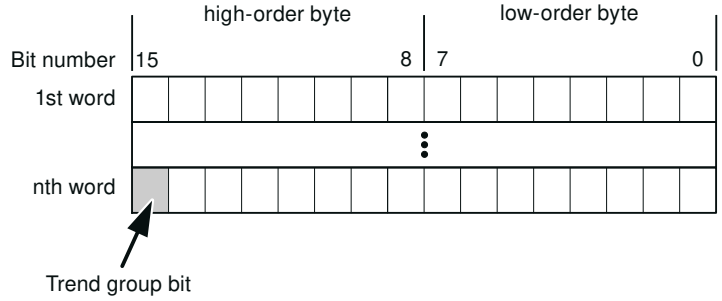
The trend request area can be used for evaluation purposes in the PLC to determine which trend is currently being displayed on the HMI device. Trends can also be triggered without evaluation of the trend request area.

Trend transfer area 1

This area is used to trigger trends. You must set the bit assigned to the trend in the trend transfer area and set the trend group bit in your control program. 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.

The PLC writes to Buffer 2 while the HMI device reads values from Buffer 1, and writes to Buffer 1 when the HMI device is reading Buffer 2. This prevents the PLC from overwriting trend values while the trend is being read by the HMI device.

See also

Communication with SIMATIC LOGO! (Page 805)

General information on trends (Page 811)

Permitted data types for trends (Page 813)

Permitted data types for trends

For SIMATIC S7

You assign one bit to each trend during configuration. Tags and array tags of the "Word" or "Int" data type are permitted.

See also

Communication with SIMATIC LOGO! (Page 805)

General information on trends (Page 811)

Trend request and trend transfer (Page 812)

2.12.4.2 Alarms

Configuring alarms

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

Communication with SIMATIC LOGO! (Page 805)

Acknowledgment of alarms (Page 815)

Acknowledgment of alarms

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

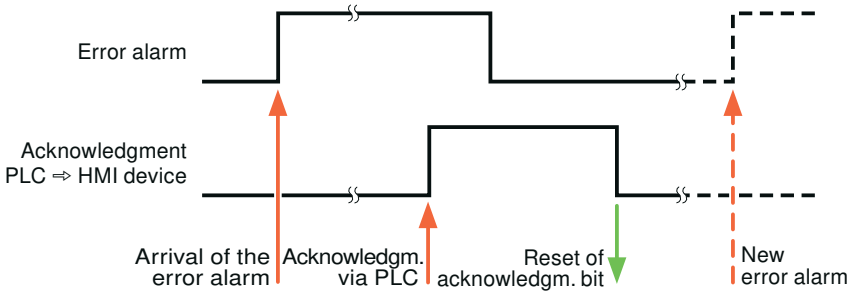
Acknowledgment by the PLC

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.

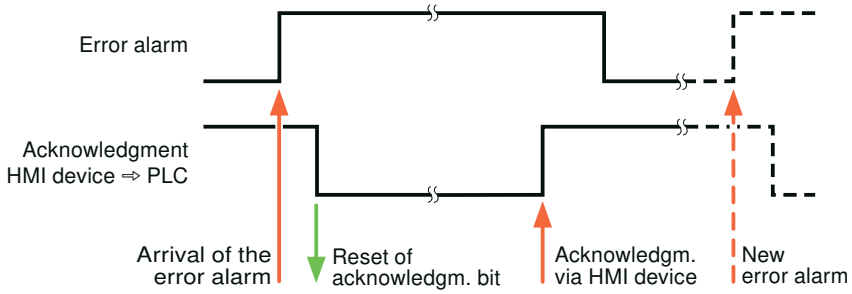
If a discrete alarm subject to acknowledgment is acknowledged by the HMI device, then the corresponding bit in the assigned acknowledgment tag will be set. The entire acknowledgment tag is then written to the PLC by the HMI device. This allows the PLC to recognize that a certain alarm message has been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

Communication with SIMATIC LOGO! (Page 805)

Configuring alarms (Page 814)

2.12.5 Performance features of communication

2.12.5.1 SIMATIC LOGO! device dependency

SIMATIC LOGO! 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 integrated connections to certain HMI devices.

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

Basic Panels V14.0.1

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 V15.0

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

Mobile Panels V12.0

HMI devices	SIMATIC LOGO!
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 (RFID tag)	Yes
Mobile Panel 277 10"	Yes

Mobile Panels V13.0.1

HMI devices	SIMATIC LOGO!
KTP700 Mobile	Yes
KTP700F Mobile	Yes
KTP900 Mobile	Yes
KTP900F Mobile	Yes

Mobile Panels V14.0

HMI devices	SIMATIC LOGO!
KTP700 Mobile	Yes
KTP700F Mobile	Yes
KTP900 Mobile	Yes
KTP900F Mobile	Yes
KTP400F Mobile	Yes

Mobile Panels V14.0.1

HMI devices	SIMATIC LOGO!
KTP700 Mobile	Yes
KTP700F Mobile	Yes
KTP900 Mobile	Yes
KTP900F Mobile	Yes
KTP400F Mobile	Yes

Mobile Panels V15.0

HMI devices	SIMATIC LOGO!
KTP700 Mobile	Yes
KTP700F Mobile	Yes
KTP900 Mobile	Yes
KTP900F Mobile	Yes
KTP400F Mobile	Yes

Comfort Panels V12.0

HMI devices	SIMATIC LOGO!
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

HMI devices	SIMATIC LOGO!
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

HMI devices	SIMATIC LOGO!
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 LOGO!
KP400 Comfort	Yes
KTP400 Comfort	Yes
KTP400 Comfort Portrait	Yes
KP700 Comfort	Yes
TP700 Comfort	Yes
TP700 Comfort Portrait	Yes
TP700 Comfort Outdoor	Yes

HMI devices	SIMATIC LOGO!
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

Comfort Panels V14.0

HMI devices	SIMATIC LOGO!
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

HMI devices	SIMATIC LOGO!
TP2200 Comfort	Yes
TP2200 Comfort Portrait	Yes

Comfort Panels V14.0.1

HMI devices	SIMATIC LOGO!
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

Comfort Panels V15.0

HMI devices	SIMATIC LOGO!
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

HMI devices	SIMATIC LOGO!
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 V12.0

HMI devices	SIMATIC LOGO!
WinCC RT Advanced	Yes

Runtime V13.0

HMI devices	SIMATIC LOGO!
WinCC RT Advanced	Yes

Runtime V13.0.1

HMI devices	SIMATIC LOGO!
WinCC RT Advanced	Yes

Runtime V14.0

HMI devices	SIMATIC LOGO!
WinCC RT Advanced	Yes
WinCC RT Professional	No

Runtime V14.0.1

HMI devices	SIMATIC LOGO!
WinCC RT Advanced	Yes
WinCC RT Professional	No

Runtime V15.0

HMI devices	SIMATIC LOGO!
WinCC RT Advanced	Yes
WinCC RT Professional	No

See also

Communication with SIMATIC LOGO! (Page 805)

Valid data types for SIMATIC LOGO! (Page 824)

2.12.5.2 Valid data types for SIMATIC LOGO!

Valid data types for connections with SIMATIC LOGO!

Data type	Length
Bool	1 bit
Byte	1 byte
Int	2 bytes
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 805)

SIMATIC LOGO! device dependency (Page 816)

2.13 Configuring direct keys

2.13.1 Direct keys

Introduction

In addition to their standard use, the F, K and S keys of an HMI device with key operation can be used as direct keys.

With HMI devices with touch operation, you configure the system function "Direct key" to a button

You can configure the following direct keys:

- PROFINET connection: PROFINET IO direct keys
- PROFIBUS connection: PROFIBUS DP direct keys

Operating mode of HMI devices with direct keys

Before accessing the PLC from the HMI device using the direct keys you have to change the operating mode of the HMI device.

See also

KTP400F Mobile (Page 833)

2.13.2 Changing the operating mode of the HMI device

2.13.2.1 Changing the operating mode for a PROFINET connection

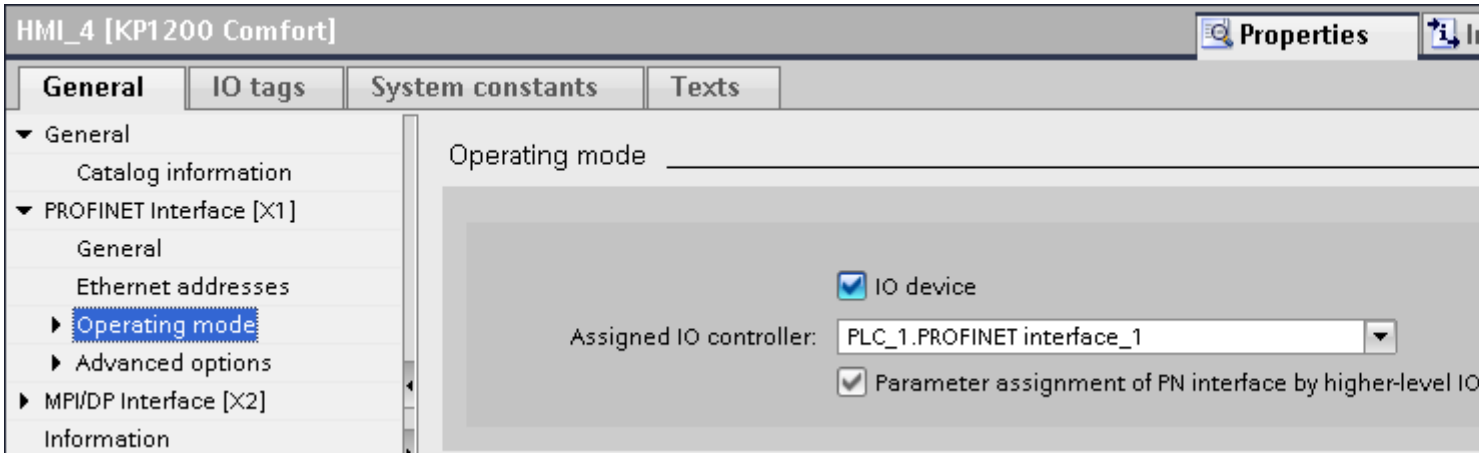
Requirements

The HMI device is networked with a PLC via PROFINET.

Procedure

1. Click the PROFINET interface of the HMI device in the "Devices & Networks" editor.
2. Click "Operating mode" under "Properties > General" in the Inspector window.

3. Select the function "IO device" in the "Operating mode" area.
4. Select the PLC which is networked with the HMI device under "Assigned IO controller".



See also

Direct keys (Page 825)

Changing the operating mode for a PROFIBUS connection (Page 826)

2.13.2.2 Changing the operating mode for a PROFIBUS connection

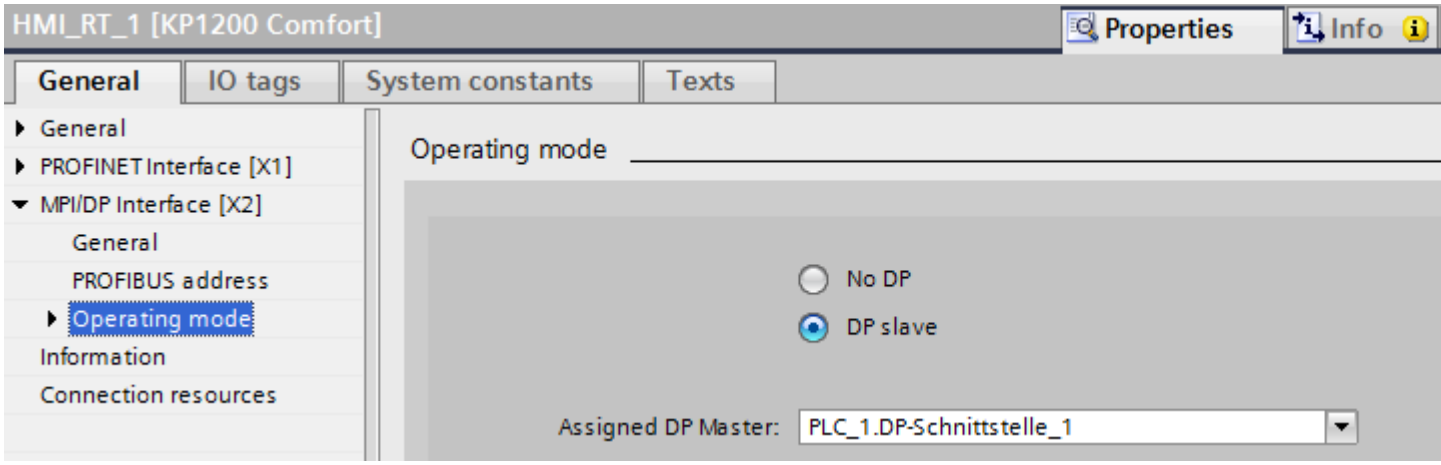
Requirements

The HMI device is networked with a PLC via PROFIBUS.

Procedure

1. Click the PROFIBUS interface of the HMI device in the "Devices & Networks" editor.
2. Click "Operating mode" under "Properties > General" in the Inspector window.

3. Select the function "DP slave" in the "Operating mode" area.
4. Select the PLC which is networked with the HMI device under "Assigned DP master".



See also

Direct keys (Page 825)

Changing the operating mode for a PROFINET connection (Page 825)

2.13.3 Configuring direct keys

2.13.3.1 Configuring direct keys for an HMI device with touch operation

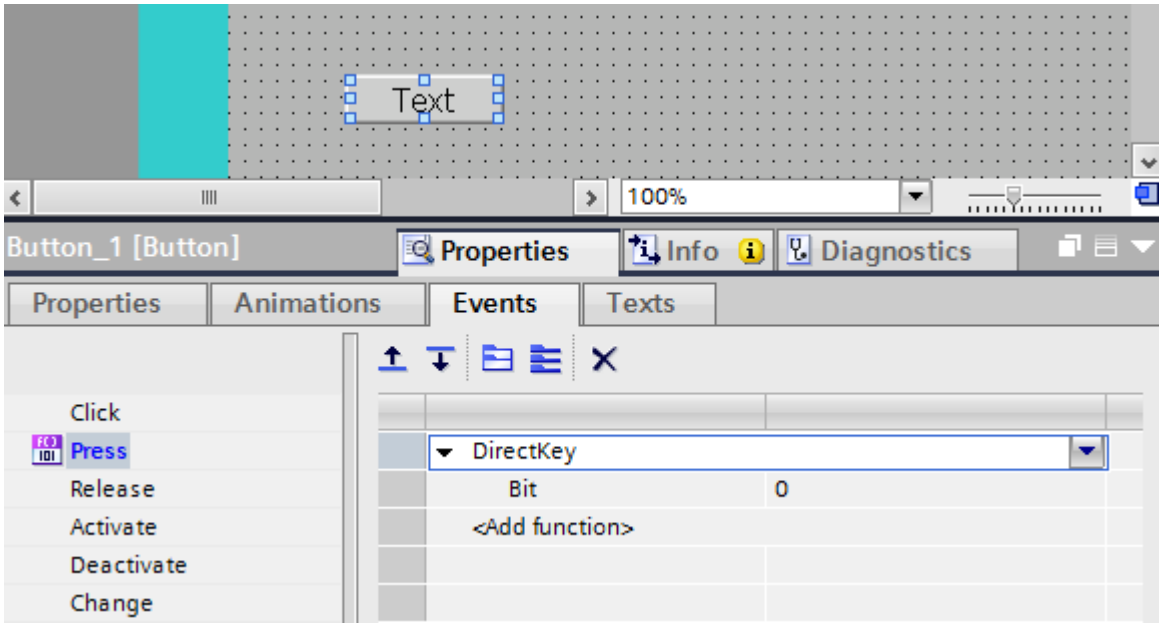
Requirements

- An HMI device with key operation has been created.
- You have created an LED tag.

Procedure

1. Create a new screen.
2. Drag-and-drop a button from the "Tools" task card into the screen.
3. Click the button.
4. In the Inspector window, click "Properties > Events > Press".
5. Click "<Add function>".

6. Select the "Direct key" system function.



7. Under "Bit" enter the correct bit number.
   The correct bit number depends on the HMI device and the input and output assignments on the HMI device.

Assignment of inputs and outputs

The exact assignment of inputs and outputs can be found under:

- PROFINET IO direct keys: Inputs and outputs of HMI devices (Page 832)
- PROFIBUS DP direct keys: Inputs and outputs of HMI devices (Page 847)

See also

Direct keys (Page 825)

Inputs and outputs of HMI devices (Page 832)

Inputs and outputs of HMI devices (Page 847)

2.13.4 PROFINET IO direct keys

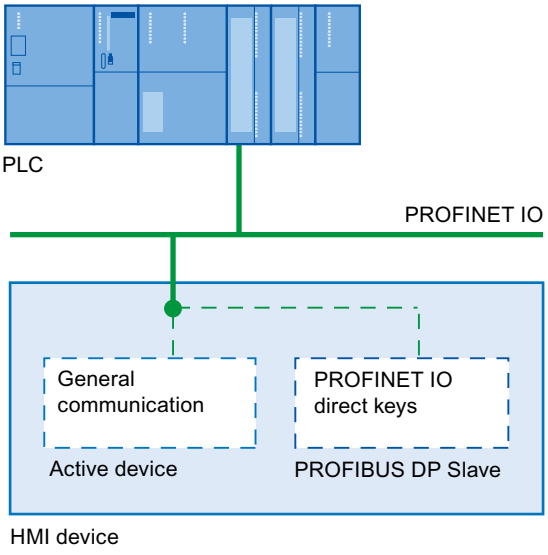
2.13.4.1 PROFINET IO direct keys

PROFINET IO direct keys

You configure the HMI device created in WinCC as an active communication partner in the automation network.

For the PROFINET IO direct keys, the HMI device can also be configured as a slave in the PROFINET IO network.

The following figure shows the basic configuration based on an automation network with an HMI device and a PLC.



Mode of operation of the PROFINET IO direct keys

The cycle time of the Ethernet bus can be set between 8 ms and 512 ms.

Thus, the reaction time of the PROFINET IO direct keys can also be determined. For a typical configuration with a cycle time of 64 ms, the response time of the PROFINET IO direct keys is < 100 ms.

A response time of < 100 ms to the CPU is usually ensured when using the PROFINET IO direct keys. This time can be exceeded significantly in the following cases:

- Complex functions run in the background, e.g. transmitting recipes, printing reports.
- At the same time, multiple connections can be maintained to the CPU.

See also

Direct keys (Page 825)

HMI devices for the configuration of PROFINET IO direct keys (Page 830)

Restrictions for PROFINET IO direct keys (Page 830)

2.13.4.2 HMI devices for the configuration of PROFINET IO direct keys

HMI devices

You can configure PROFINET IO direct keys with the following HMI devices:

HMI device class	HMI device
Mobile Panel	KTP400F Mobile
	KTP700 Mobile
	KTP700F Mobile
	KTP700F Mobile HW
	KTP900 Mobile
	KTP900F Mobile
	TP1000F Mobile
Comfort Panel	KTP400 Comfort
	KP400 Comfort
	KP700 Comfort
	TP700 Comfort
	KP900 Comfort
	TP900 Comfort
	KP1200 Comfort
	TP1200 Comfort
	KP1500 Comfort
	TP1500 Comfort
	TP1900 Comfort
	TP2200 Comfort

See also

Direct keys (Page 825)
PROFINET IO direct keys (Page 828)
Restrictions for PROFINET IO direct keys (Page 830)

2.13.4.3 Restrictions for PROFINET IO direct keys

Note

Direct keys are still active when the HMI device is in "offline" mode.

Note

If an external application such as Pocket Internet Explorer or Control Panel is started, it will run in the foreground and Runtime is placed in the background. The bit for the function "DirectKeyScreenNumber" is no longer set and the keys or buttons with the configured function "DirectKey" no longer trigger the associated bit in the PLC.

Restrictions

- It is not permitted to operate PROFIBUS IO direct keys and PROFINET DP direct keys simultaneously.
- Establish the following via the "PROFINET IO enabled" option in the control panel of the HMI device:
  - Option disabled = PROFIBUS DP direct keys enabled
  - Option enabled = PROFINET IO direct keys enabled
- If communication is enabled with PROFINET IO, you are not permitted to use the serial interface.
- You can operate direct keys only on local HMI devices. On the Sm@rtClient it is possible to operate the key / button for the direct key. However, No bit is set in the I/O area of the CPU.
- Direct keys which are assigned to a button are only triggered through touch operation. Triggering with a mouse click, for example with a connected USB mouse, is not possible.
- With touch operation, direct keys are triggered independent of any configured password protection.
- The buttons used as direct keys on HMI devices with touch screen functionality may not be modified as follows by means of scripting:
  - Moving
  - Resizing
  - Hiding
  - Disabling for operation
- The LEDs are addressed either via the PROFINET IO direct keys or the HMI Runtime application. Avoid concurrent addressing via the PROFINET IO direct keys and the HMI Runtime application. The LEDs "ACK", "A-Z l", "A-Z r" and "HELP" are reserved for system functions and cannot be configured. It is not advisable to control the "ACK", "A-Z l", "A-Z r" and "HELP" LEDs by means of PROFINET IO direct key functionality.

⚠ WARNING

Triggering the "DirectKey" system function by mistake will endanger personnel or result in damage to the machine.

In order to avoid this danger, the following must be observed:

- When the process screen is configured, the button with the "DirectKey" system function must not be covered by a screen object.
- The dynamic positioning or display (release) of a screen object in relation to process values must not cover the button with the "DirectKey" system function in runtime.

Note

Please observe this guideline during configuration. Check also existing configurations and adjust them immediately.

See also

Direct keys (Page 825)

PROFINET IO direct keys (Page 828)

HMI devices for the configuration of PROFINET IO direct keys (Page 830)

2.13.4.4 Inputs and outputs of HMI devices

Assignment of inputs and outputs

Assigning the inputs/outputs

The keys or buttons of the device are assigning bytes in the input area. The LEDs are assigning bytes in the output area. The number of bytes used depends on the HMI device.

The touch panels do not have any permanently assigned keys. They only have user-configurable buttons. Via the "DirectKey system function you can assign a bit to a button in the input area. The counting direction of the bit in the input direction is from right to left. In contrast to the operator panels which have permanently assigned keys, the touch panel buttons can be assigned freely.

Assigning direct keys to screen numbers (only for touch devices)

If a PROFINET IO direct key is using the same bit for different functions in different screens, the SIMATIC PLC must differentiate between the respective functionality by means of the screen number. You can use the "DirectKeyScreenNumber" system function to avoid the delayed updating of the screen number following a screen change.

Using the "DirectKeyScreenNumber"system function you can set within the input area any number of bits in order to identify the screen and simultaneously transfer the direct key bits to the PLC. This ensures unambiguous allocation of a control bit to screen number at all times.

See also

Direct keys (Page 825)

KTP400F Mobile (Page 833)

Mobile Panel (Page 833)

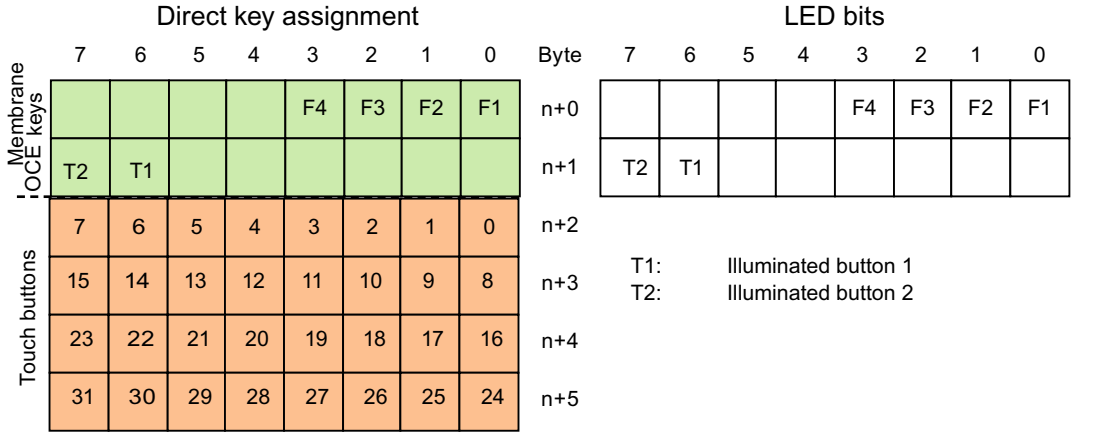
Comfort Panel (Page 837)

Mobile Panel

KTP400F Mobile

Direct keys KTP400F Mobile

Inputs	Outputs
6 bytes	2 bytes



See also

Direct keys (Page 825)

Assignment of inputs and outputs (Page 832)

KTP700 Mobile, KTP900 Mobile

Evaluating operator controls as direct keys

You can configure the operator controls of the HMI device as direct keys. The states of the following operator controls are available directly in the I/O area of the PLC:

- The switching state of the function keys
- The switching state of the key-operated switch
- The switching state of the illuminated pushbuttons

Byte assignment

The following table shows the assignment of the keys (inputs) and LEDs (outputs) to the bytes in the PLC process image. For more information, see your plant documentation.

- KTP700 Mobile and KTP700F Mobile

Direct key bit							
7	6	5	4	3	2	1	0
F8	F7	F6	F5	F4	F3	F2	F1
T2	T1	S1	S0				
7	6	5	4	3	2	1	0
15	14	13	12	11	10	9	8
23	22	21	20	19	18	17	16
31	30	29	28	27	26	25	24

Byte
n
n + 1
n + 2
n + 3
n + 4
n + 5

LED bit							
7	6	5	4	3	2	1	0
F8	F7	F6	F5	F4	F3	F2	F1
T2	T1						

F Bit for function key

S Bit for key-operated switch

T1 Bit for left illuminated pushbutton

T2 Bit for right illuminated pushbutton

The bytes "n+2" to "n+5" contain the direct key bits for the touch buttons.

- KTP900 Mobile and KTP900F Mobile

Direct key bit							
7	6	5	4	3	2	1	0
F8	F7	F6	F5	F4	F3	F2	F1
T2	T1	S1	S0			F10	F9
7	6	5	4	3	2	1	0
15	14	13	12	11	10	9	8
23	22	21	20	19	18	17	16
31	30	29	28	27	26	25	24

Byte
n
n + 1
n + 2
n + 3
n + 4
n + 5

LED bit							
7	6	5	4	3	2	1	0
F8	F7	F6	F5	F4	F3	F2	F1
T2	T1					F10	F9

F Bit for function key

S Bit for key-operated switch

T1 Bit for left illuminated pushbutton

T2 Bit for right illuminated pushbutton

The bytes "n+2" to "n+5" contain the direct key bits for the touch buttons.

Bit coding

The following tables show the bit coding for function keys, key-operated switch and illuminated pushbutton:

- Bit coding of function keys

State	F1 to F8 or F10
Not pressed	0
Pressed	1

- Bit coding of function key LEDs

State	F1 to F8 or F10
LED not illuminated	0
LED is illuminated	1

- Bit coding of key-operated switch

State	S1	S0	Key position
Position 0	0	0	In middle position
Position I	0	1	Turned in clockwise direction up to stop
Position II	1	0	Turned counter-clockwise up to stop

- Bit coding of illuminated pushbuttons

State	K1	K2
Not pressed	0	0
Pressed	1	1

- Bit coding of illuminated pushbutton LEDs

State	K1	K2
Off	0	0
On permanently	1	1

See also

Direct keys (Page 825)

PROFINET IO direct keys (Page 828)

Assignment of inputs and outputs (Page 832)

Controlling the LEDs of the function keys by means of system functions

F1 to F8 or F10 to LEDs are integrated in the function keys of the HMI device. The connected controller can control the integrated LEDs. The LED signals the operator that he should use the corresponding function key while the project is running.

The following table shows the possible states of the LEDs and the corresponding entries in bit n+1 and bit n of the LED tags:

Bit n + 1	Bit n	LED status
0	0	Off
0	1	Flashing quickly
1	0	Flashing slowly
1	1	ON, continuous

See also

Direct keys (Page 825)

PROFINET IO direct keys (Page 828)

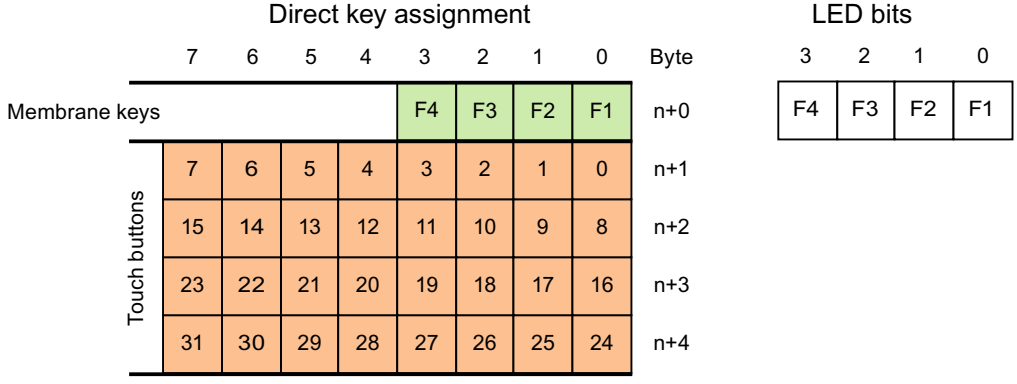
Assignment of inputs and outputs (Page 832)

Comfort Panel

KTP400 Comfort

Direct keys KTP400 Comfort

HMI device	Inputs	Outputs
Touch operation	5 bytes	1 byte



See also

Direct keys (Page 825)

PROFINET IO direct keys (Page 828)

Assignment of inputs and outputs (Page 832)

KP400 Comfort

Direct keys KP400 Comfort

Inputs	Outputs
1 byte	1 byte

Direct keys assignment:

	7	6	5	4	3	2	1	0	Byte
Keys	F8	F7	F6	F5	F4	F3	F2	F1	n+0

LED

7	6	5	4	3	2	1	0
F8	F7	F6	F5	F4	F3	F2	F1

See also

Direct keys (Page 825)

PROFINET IO direct keys (Page 828)

Assignment of inputs and outputs (Page 832)

KP700 Comfort

Direct keys KP700 Comfort

Inputs	Outputs
3 bytes	3 bytes

Direct keys assignment:

	7	6	5	4	3	2	1	0	Byte
Keys	F8	F7	F6	F5	F4	F3	F2	F1	n+0
	F16	F15	F14	F13	F12	F11	F10	F9	n+1
	F24	F23	F22	F21	F20	F19	F18	F17	n+2

LED

7	6	5	4	3	2	1	0
F8	F7	F6	F5	F4	F3	F2	F1
F16	F15	F14	F13	F12	F11	F10	F9
F24	F23	F22	F21	F20	F19	F18	F17

See also

Direct keys (Page 825)

PROFINET IO direct keys (Page 828)

Assignment of inputs and outputs (Page 832)

TP700 Comfort

Direct keys TP700 Comfort

Inputs	Outputs
4 bytes	--

Direct keys assignment:

	7	6	5	4	3	2	1	0	Byte
Touch buttons	7	6	5	4	3	2	1	0	n+0
	15	14	13	12	11	10	9	8	n+1
	23	22	21	20	19	18	17	16	n+2
	31	30	29	28	27	26	25	24	n+3

LED

No output area

See also

Direct keys (Page 825)

PROFINET IO direct keys (Page 828)

Assignment of inputs and outputs (Page 832)

KP900 Comfort

Direct keys KP900 Comfort

Inputs	Outputs
4 bytes	4 bytes

Direct keys assignment:

	7	6	5	4	3	2	1	0	Byte
Keys	F8	F7	F6	F5	F4	F3	F2	F1	n+0
	F16	F15	F14	F13	F12	F11	F10	F9	n+1
	F24	F23	F22	F21	F20	F19	F18	F17	n+2
							F26	F25	n+3

LED

7	6	5	4	3	2	1	0
F8	F7	F6	F5	F4	F3	F2	F1
F16	F15	F14	F13	F12	F11	F10	F9
F24	F23	F22	F21	F20	F19	F18	F17
						F18	F17

See also

Direct keys (Page 825)

PROFINET IO direct keys (Page 828)

Assignment of inputs and outputs (Page 832)

TP900 Comfort

Direct keys TP900 Comfort

Inputs	Outputs
5 bytes	--

Direct keys assignment:

	7	6	5	4	3	2	1	0	Byte
Touch buttons	7	6	5	4	3	2	1	0	n+0
	15	14	13	12	11	10	9	8	n+1
	23	22	21	20	19	18	17	16	n+2
	31	30	29	28	27	26	25	24	n+3
	39	38	37	36	35	34	33	32	n+4

LED

No output area

See also

Direct keys (Page 825)

PROFINET IO direct keys (Page 828)

Assignment of inputs and outputs (Page 832)

KP1200 Comfort

Direct keys KP1200 Comfort

Inputs	Outputs
5 bytes	5 bytes

Direct keys assignment

	7	6	5	4	3	2	1	0	Byte
Keys	F8	F7	F6	F5	F4	F3	F2	F1	n+0
	F16	F15	F14	F13	F12	F11	F10	F9	n+1
	F24	F23	F22	F21	F20	F19	F18	F17	n+2
	F32	F31	F30	F29	F28	F27	F26	F25	n+3
							F34	F33	n+4

LED

7	6	5	4	3	2	1	0
F8	F7	F6	F5	F4	F3	F2	F1
F16	F15	F14	F13	F12	F11	F10	F9
F24	F23	F22	F21	F20	F19	F18	F17
F32	F31	F30	F29	F28	F27	F26	F25
						F34	F33

See also

Direct keys (Page 825)

PROFINET IO direct keys (Page 828)

Assignment of inputs and outputs (Page 832)

TP1200 Comfort

Direct keys TP1200 Comfort

Inputs	Outputs
5 bytes	--

Direct keys assignment:

	7	6	5	4	3	2	1	0	Byte
Touch buttons	7	6	5	4	3	2	1	0	n+0
	15	14	13	12	11	10	9	8	n+1
	23	22	21	20	19	18	17	16	n+2
	31	30	29	28	27	26	25	24	n+3
	39	38	37	36	35	34	33	32	n+4

LED

No output area

See also

Direct keys (Page 825)

PROFINET IO direct keys (Page 828)

Assignment of inputs and outputs (Page 832)

TP1500, TP1900 and TP2200 Comfort

Inputs	Outputs
5 bytes	--

Direct key assignment

	7	6	5	4	3	2	1	0	Byte
Touch buttons	7	6	5	4	3	2	1	0	n+0
	15	14	13	12	11	10	9	8	n+1
	23	22	21	20	19	18	17	16	n+2
	31	30	29	28	27	26	25	24	n+3
	39	38	37	36	35	34	33	32	n+4

LED

No output area

See also

Direct keys (Page 825)

PROFINET IO direct keys (Page 828)

Assignment of inputs and outputs (Page 832)

KP1500 Comfort

Inputs	Outputs
5 bytes	5 bytes

Direct key assignment

	7	6	5	4	3	2	1	0	Byte
Keys	F8	F7	F6	F5	F4	F3	F2	F1	n+0
	F16	F15	F14	F13	F12	F11	F10	F9	n+1
	F24	F23	F22	F21	F20	F19	F18	F17	n+2
	F32	F31	F30	F29	F28	F27	F26	F25	n+3
					F36	F35	F34	F33	n+4

LED

7	6	5	4	3	2	1	0
F8	F7	F6	F5	F4	F3	F2	F1
F16	F15	F14	F13	F12	F11	F10	F9
F24	F23	F22	F21	F20	F19	F18	F17
F32	F31	F30	F29	F28	F27	F26	F25
				F36	F35	F34	F33

See also

Direct keys (Page 825)

PROFINET IO direct keys (Page 828)

Assignment of inputs and outputs (Page 832)

2.13.5 PROFIBUS DP direct keys

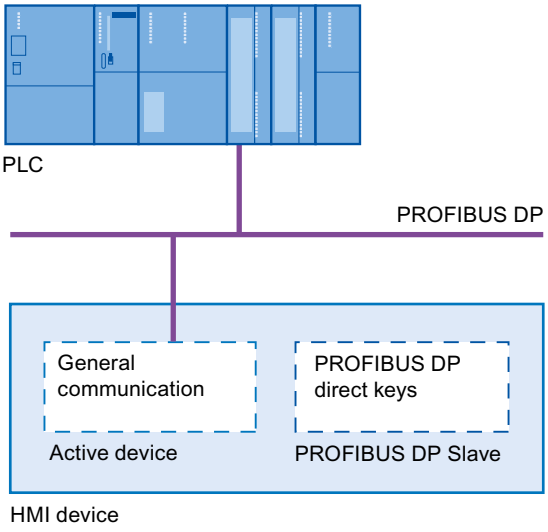
2.13.5.1 PROFIBUS DP direct keys

PROFIBUS DP direct keys

You configure the HMI device created in WinCC as an active communication partner in the automation network.

For the PROFIBUS DP direct keys, also configure the HMI device as a slave in the PROFIBUS DP network.

The following figure shows the basic configuration based on an automation network with an HMI device and a PLC.



Mode of operation of the PROFIBUS DP direct keys

With PROFIBUS DP direct keys, when the key or button is pressed, a bit is set in the I/O area of the CPU.

The cycle time of the PROFIBUS DP bus is calculated from the total of all configured inputs and outputs.

The number of configured inputs and outputs can influence the reaction time of the PROFIBUS DP direct key. For a typical configuration the response time of the PROFIBUS DP direct key is < 100 ms.

See also

Direct keys (Page 825)

HMI devices for the configuration of PROFIBUS DP direct keys (Page 845)

Restrictions for PROFIBUS DP direct keys (Page 845)

2.13.5.2 HMI devices for the configuration of PROFIBUS DP direct keys

HMI devices

You can configure PROFIBUS DP direct keys with the following HMI devices:

HMI device class	HMI device
Comfort Panel	KTP400 Comfort
	KP400 Comfort
	KP700 Comfort
	TP700 Comfort
	KP900 Comfort
	TP900 Comfort
	KP1200 Comfort
	TP1200 Comfort
	KP1500 Comfort
	TP1500 Comfort
	TP1900 Comfort
	TP2200 Comfort

See also

Direct keys (Page 825)

PROFIBUS DP direct keys (Page 843)

Restrictions for PROFIBUS DP direct keys (Page 845)

2.13.5.3 Restrictions for PROFIBUS DP direct keys

Note

If an external application such as Pocket Internet Explorer or Control Panel is started, it will run in the foreground and Runtime is placed in the background. The bit for the function "DirectKeyScreenNumber" is no longer set and the keys or buttons with the configured function "DirectKey" no longer trigger the associated bit in the PLC.

Restrictions

- It is not possible to operate PROFIBUS DP direct keys and PROFINET IO direct keys simultaneously.
  Establish the following via the "PROFINET IO enabled" option in the control panel of the HMI device:
  - Option disabled = PROFIBUS DP direct keys enabled
  - Option enabled = PROFINET IO direct keys enabled
- You can operate direct keys only on local HMI devices. On the Sm@rtClient it is possible to operate the key / button for the direct key. However, No bit is set in the I/O area of the CPU.
- The buttons used as direct keys on HMI devices with touch screen functionality may not be modified as follows by means of scripting:
  - Moving
  - Resizing
  - Hiding
  - Disabling for operation
- The LEDs are addressed either via the PROFIBUS DP direct keys or the HMI Runtime application. Avoid concurrent addressing via the PROFIBUS DP direct keys and the HMI Runtime application. The LEDs "ACK", "A-Z l", "A-Z r" and "HELP" are reserved for system functions and cannot be configured. We do not recommend addressing the LEDs "ACK", "A-Z l", "A-Z r" and "HELP" with the PROFIBUS DP direct keys.
- The PROFIBUS direct keys on an HMI device are triggered independent of any configured password protection.

  ⚠ WARNING

  Triggering the "DirectKey" system function by mistake will endanger personnel or result in damage to the machine.

  In order to avoid this danger, the following must be observed:
  - When the process screen is configured, the button with the "DirectKey" system function must not be covered by a screen object.
  - The dynamic positioning or display (release) of a screen object in relation to process values must not cover the button with the "DirectKey" system function in runtime.

  Note

  Please observe this guideline during configuration. Check also existing configurations and adjust them immediately.

See also

Direct keys (Page 825)

PROFIBUS DP direct keys (Page 843)

HMI devices for the configuration of PROFIBUS DP direct keys (Page 845)

2.13.5.4 Inputs and outputs of HMI devices

Assignment of inputs and outputs

Assigning the inputs/outputs

The keys or buttons of the device are assigning bytes in the input area. The LEDs occupy bytes in the DP output area. The following table shows the number of bytes used in the various HMI devices. The precise assignment is shown in the following screens.

The touch panels do not have any permanently assigned keys. They only have user-configurable buttons. Via the direct keys function you can assign to a button a bit in the input area. The counting direction of the bit in the input direction is from right to left. In contrast to the operator panels which have permanently assigned keys, the touch panel buttons can be assigned freely. For more detailed information, refer to the "WinCC flexible - Configuring WinCC Windows-based Systems" user manual.

Assigning direct keys to screen numbers (only for touch devices)

If a PROFIBUS DP direct key is using the same bit for different functions in different screens, the S7 must differentiate between the respective functionality by means of the screen number. To combat the delayed updating of the screen number in the PLC after a change of screen you use the "DirectKeyScreenNumber" screen function.

Using the "DirectKeyScreenNumber" you can set within the input area any number of bits in order to identify the screen and simultaneously transfer the direct key bits to the PLC. This ensures unambiguous allocation of a control bit to screen number at all times.

See also

Direct keys (Page 825)

PROFIBUS DP direct keys (Page 843)

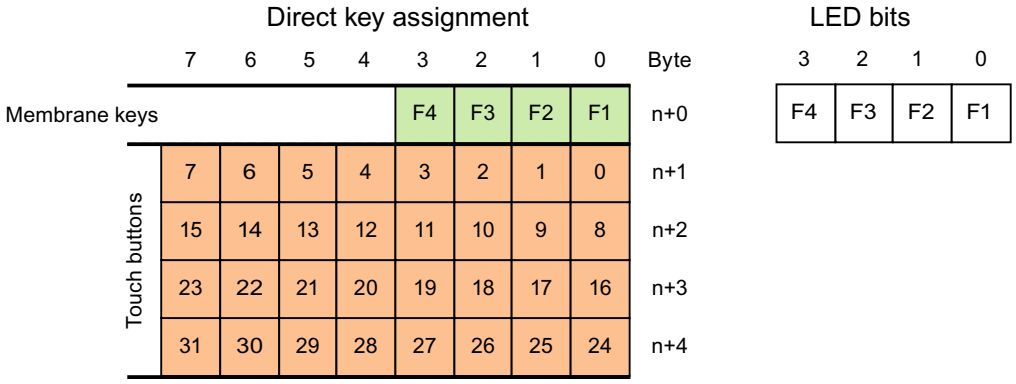
Comfort Panel (Page 847)

Comfort Panel

KTP400 Comfort

Direct keys KTP400 Comfort

HMI device	Inputs	Outputs
Touch operation	5 bytes	1 byte



See also

Direct keys (Page 825)

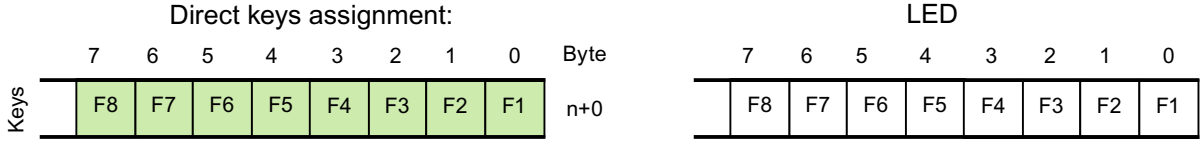
PROFIBUS DP direct keys (Page 843)

Assignment of inputs and outputs (Page 847)

KP400 Comfort

Direct keys KP400 Comfort

Inputs	Outputs
1 byte	1 byte



See also

Direct keys (Page 825)

PROFIBUS DP direct keys (Page 843)

Assignment of inputs and outputs (Page 847)

KP700 Comfort

Direct keys KP700 Comfort

Inputs	Outputs
3 bytes	3 bytes

Direct keys assignment:

	7	6	5	4	3	2	1	0	Byte
Keys	F8	F7	F6	F5	F4	F3	F2	F1	n+0
	F16	F15	F14	F13	F12	F11	F10	F9	n+1
	F24	F23	F22	F21	F20	F19	F18	F17	n+2

LED

7	6	5	4	3	2	1	0
F8	F7	F6	F5	F4	F3	F2	F1
F16	F15	F14	F13	F12	F11	F10	F9
F24	F23	F22	F21	F20	F19	F18	F17

See also

Direct keys (Page 825)

PROFIBUS DP direct keys (Page 843)

Assignment of inputs and outputs (Page 847)

TP700 Comfort

Direct keys TP700 Comfort

Inputs	Outputs
4 bytes	--

Direct keys assignment:

	7	6	5	4	3	2	1	0	Byte
Touch buttons	7	6	5	4	3	2	1	0	n+0
	15	14	13	12	11	10	9	8	n+1
	23	22	21	20	19	18	17	16	n+2
	31	30	29	28	27	26	25	24	n+3

LED

No output area

See also

Direct keys (Page 825)

PROFIBUS DP direct keys (Page 843)

Assignment of inputs and outputs (Page 847)

KP900 Comfort

Direct keys KP900 Comfort

Inputs	Outputs
4 bytes	4 bytes

Direct keys assignment:

	7	6	5	4	3	2	1	0	Byte
Keys	F8	F7	F6	F5	F4	F3	F2	F1	n+0
	F16	F15	F14	F13	F12	F11	F10	F9	n+1
	F24	F23	F22	F21	F20	F19	F18	F17	n+2
							F26	F25	n+3

LED

7	6	5	4	3	2	1	0
F8	F7	F6	F5	F4	F3	F2	F1
F16	F15	F14	F13	F12	F11	F10	F9
F24	F23	F22	F21	F20	F19	F18	F17
						F18	F17

See also

Direct keys (Page 825)

PROFIBUS DP direct keys (Page 843)

Assignment of inputs and outputs (Page 847)

TP900 Comfort

Direct keys TP900 Comfort

Inputs	Outputs
5 bytes	--

Direct keys assignment:

	7	6	5	4	3	2	1	0	Byte
Touch buttons	7	6	5	4	3	2	1	0	n+0
	15	14	13	12	11	10	9	8	n+1
	23	22	21	20	19	18	17	16	n+2
	31	30	29	28	27	26	25	24	n+3
	39	38	37	36	35	34	33	32	n+4

LED

No output area

See also

Direct keys (Page 825)

PROFIBUS DP direct keys (Page 843)

Assignment of inputs and outputs (Page 847)

KP1200 Comfort

Direct keys KP1200 Comfort

Inputs	Outputs
5 bytes	5 bytes

Direct keys assignment

	7	6	5	4	3	2	1	0	Byte
Keys	F8	F7	F6	F5	F4	F3	F2	F1	n+0
	F16	F15	F14	F13	F12	F11	F10	F9	n+1
	F24	F23	F22	F21	F20	F19	F18	F17	n+2
	F32	F31	F30	F29	F28	F27	F26	F25	n+3
							F34	F33	n+4

LED

7	6	5	4	3	2	1	0
F8	F7	F6	F5	F4	F3	F2	F1
F16	F15	F14	F13	F12	F11	F10	F9
F24	F23	F22	F21	F20	F19	F18	F17
F32	F31	F30	F29	F28	F27	F26	F25
						F34	F33

See also

Direct keys (Page 825)

PROFIBUS DP direct keys (Page 843)

Assignment of inputs and outputs (Page 847)

TP1200 Comfort

Direct keys TP1200 Comfort

Inputs	Outputs
5 bytes	--

Direct keys assignment:

	7	6	5	4	3	2	1	0	Byte
Touch buttons	7	6	5	4	3	2	1	0	n+0
	15	14	13	12	11	10	9	8	n+1
	23	22	21	20	19	18	17	16	n+2
	31	30	29	28	27	26	25	24	n+3
	39	38	37	36	35	34	33	32	n+4

LED

No output area

See also

Direct keys (Page 825)

PROFIBUS DP direct keys (Page 843)

Assignment of inputs and outputs (Page 847)

KP1500 Comfort

Inputs	Outputs
5 bytes	5 bytes

Direct key assignment

	7	6	5	4	3	2	1	0	Byte
Keys	F8	F7	F6	F5	F4	F3	F2	F1	n+0
	F16	F15	F14	F13	F12	F11	F10	F9	n+1
	F24	F23	F22	F21	F20	F19	F18	F17	n+2
	F32	F31	F30	F29	F28	F27	F26	F25	n+3
					F36	F35	F34	F33	n+4

LED

7	6	5	4	3	2	1	0
F8	F7	F6	F5	F4	F3	F2	F1
F16	F15	F14	F13	F12	F11	F10	F9
F24	F23	F22	F21	F20	F19	F18	F17
F32	F31	F30	F29	F28	F27	F26	F25
				F36	F35	F34	F33

See also

Direct keys (Page 825)

PROFIBUS DP direct keys (Page 843)

Assignment of inputs and outputs (Page 847)

TP1500, TP1900 and TP2200 Comfort

Inputs	Outputs
5 bytes	--

Direct key assignment

	7	6	5	4	3	2	1	0	Byte
Touch buttons	7	6	5	4	3	2	1	0	n+0
	15	14	13	12	11	10	9	8	n+1
	23	22	21	20	19	18	17	16	n+2
	31	30	29	28	27	26	25	24	n+3
	39	38	37	36	35	34	33	32	n+4

LED

No output area

See also

Direct keys (Page 825)

PROFIBUS DP direct keys (Page 843)

Assignment of inputs and outputs (Page 847)

2.14 Communication via SIMATIC HMI HTTP

2.14.1 Basic information on SIMATIC HMI HTTP

Introduction

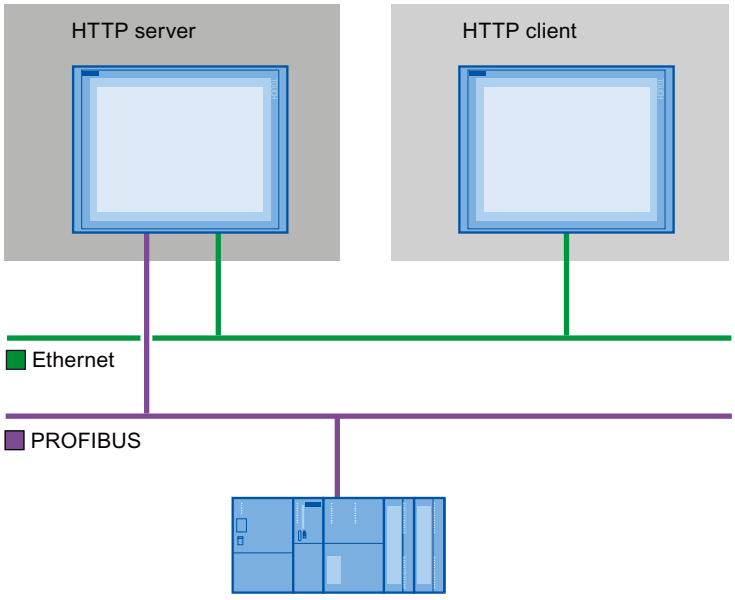
You use the SIMATIC HMI HTTP Protocol for data exchange between HMI devices.

The SIMATIC HMI HTTP Protocol is not suited for exchange of bulk data.

Data exchange occurs according to the Request-Response method. The HTTP client sends its request to the HTTP server, which processes it and returns a response.

A device can be configured as an HTTP client and HTTP server simultaneously.

The client and server establish a connection via the Ethernet interface for data exchange.



HTTP components

The components of the SIMATIC HMI HTTP Protocol are transferred to the HMI device in "Load to device".

HTTP components are:

- HTTP server
- HTTP client

If you use a Standard-PC or SIMATIC IPC, you must install WinCC RT Advanced or WinCC RT Professional first.

SMTP protocol

The SMTP protocol is not a requirement for communication via SIMATIC HMI HTTP Protocol .

HTTP / HTTPS

TheSIMATIC HMI HTTP Protocol offers two standards:

- HTTP
  Is implemented in local networks for a fast, uncoded transfer of uncritical data.
- HTTPS
  Enables a reliable HTTP connection between devices. Initially there is an exchange of secret keys. Then, in a further step, the respective public keys can be securely exchanged in the form of digital certificates. The public keys are used to encode the utility data in order to guarantee a bug-proof communication.

Note

Due to the encryption, the HTTPS protocol has a lower transmission performance than the HTTP protocol.

CAUTION

End users are responsible for the security of their own networks.

2.14.2 Configuring a connection via SIMATIC HMI HTTP

2.14.2.1 Configuring a connection via SIMATIC HMI HTTP

Communication between HMI devices via SIMATIC HMI HTTP protocol

Several steps are needed to configure communication between HMI devices via the SIMATIC HMI HTTP protocol:

1. Configure HMI device as a server.
2. Configure HMI device as a client.
3. Assign tag parameters in the client.
4. Commission connection.

Access to tags via SIMATIC HMI HTTP protocol

If access to tags via the SIMATIC HMI HTTP protocol is to be enabled, these tags must be defined for the relevant HMI devices and interconnected.

See also

Basic information on SIMATIC HMI HTTP (Page 853)

2.14.2.2 Configuring an HMI device as a server

Configuring a server

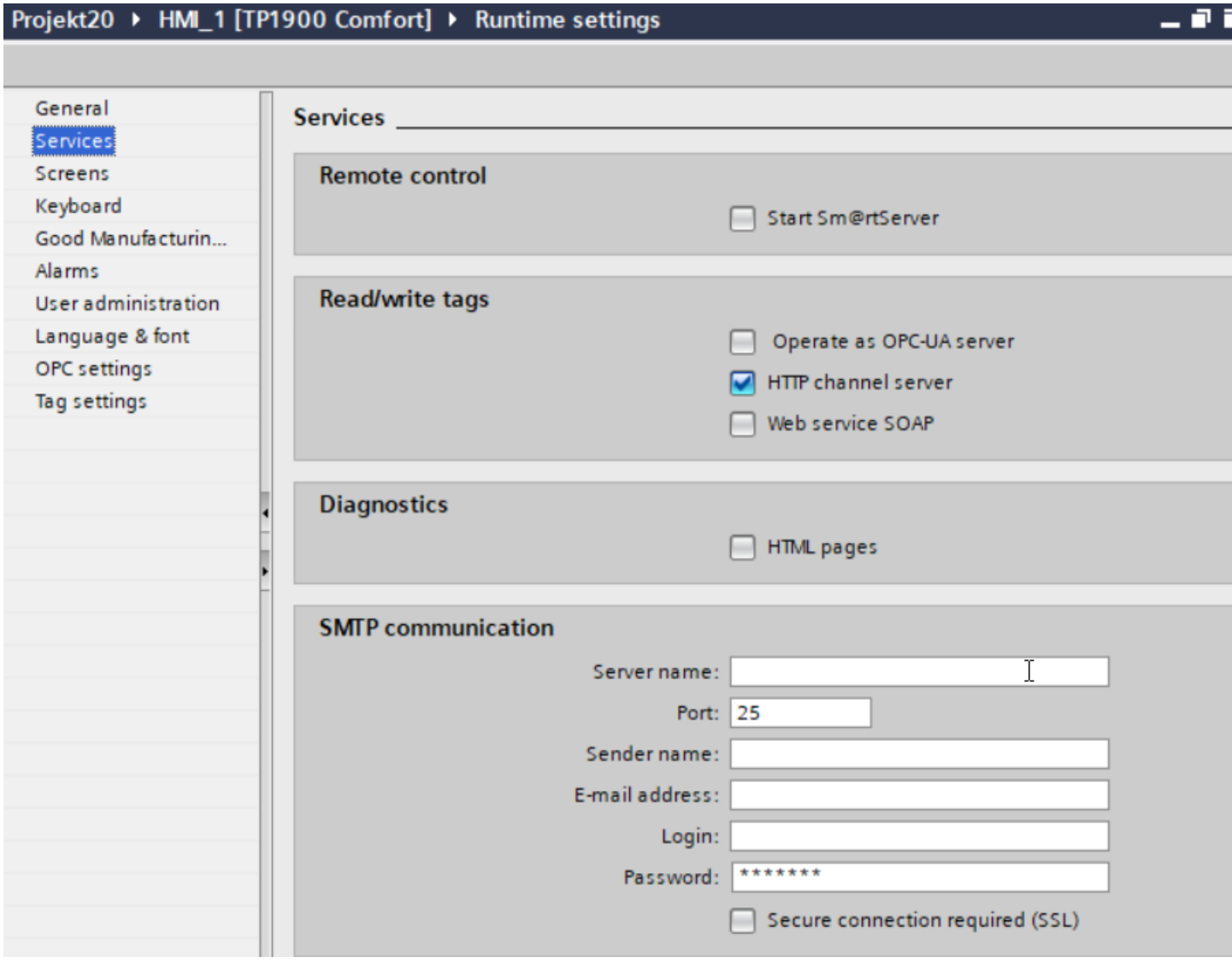
Requirements

- The HTTP communication channel must be set in the Control Panel of the HMI device.

Procedure

1. Double-click the HMI device under "Devices" in the project tree.
2. Double-click "Runtime settings".
   The "Runtime settings" editor opens.

3. Click "Services".
4. Select the "HTTP channel server" option.



If you use the SMTP protocol, you assign the following values under "SMTP settings" according to your project specifications:

- Server name
- Login
- Password

Result

The HMI device has been configured as a server.

See also

Basic information on SIMATIC HMI HTTP (Page 853)

Configuring a connection via SIMATIC HMI HTTP (Page 855)

Tags in the server (Page 858)

Tags in the server

Tags used

The client can use the HTTP protocol for read and write access to tags configured on the server in runtime. This means that it is not necessary to configure additional tags for an HTTP communication.

However, the following aspects must be taken into account to ensure correct data exchange:

- The data type of the server tags must match the data type in the client.
- The tag name configured in the HTTP server must be identical to the name of the address tag of the HTTP client.

See also

Basic information on SIMATIC HMI HTTP (Page 853)

Configuring a connection via SIMATIC HMI HTTP (Page 855)

Configuring a server (Page 856)

2.14.2.3 Configuring an HMI device as a client

Configuring a client

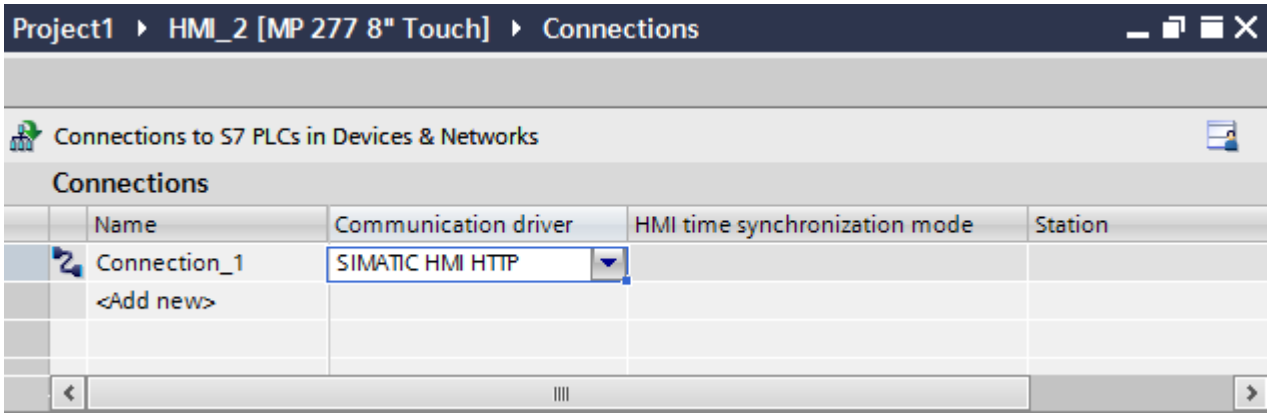
Requirements

- The HTTP communication channel must be set in the Control Panel of the HMI device.
- An HMI device has been configured as a server.

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 HMI HTTP Protocol" driver.



5. Select all necessary connection parameters for the interface in the Inspector window under "Parameters".
6. Set the parameters in the Property window.
   See "Setting parameters for the client" for more information about the parameters.

See also

Basic information on SIMATIC HMI HTTP (Page 853)

Configuring a connection via SIMATIC HMI HTTP (Page 855)

Setting parameters for the client (Page 859)

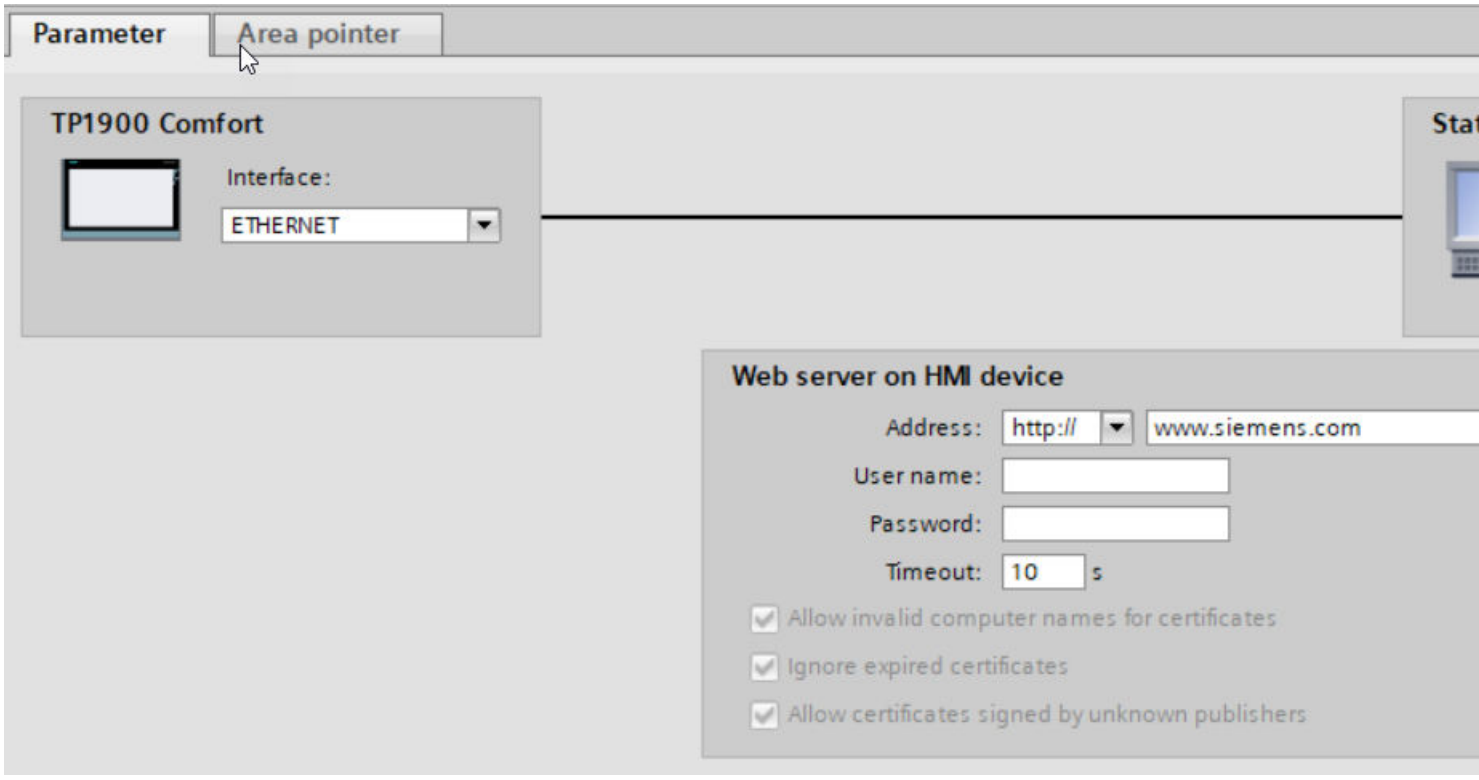
Setting parameters for the client

Requirements

- Connection has been created via SIMATIC HMI HTTP protocol.
- "Connections" editor is open.

Procedure

1. Select "Ethernet" in the Inspector window under "Parameters > Interface".
2. Select protocol http:// or https://.
3. Enter the name of the server or its IP address.



User name and password

If the "Authentication required" check box was selected in the server in the "Control Panel > WinCC Internet Settings > Web Server" dialog, a user name and password must be entered here in the client.

Timeout

Assign a time after which a connection break is recognized.

HTTPS protocol

If the HTTPS protocol is selected, with the following settings you can establish how the HTTPS client should check the properties of the server certificate and how it should react in the event of error:

- "Allow invalid computer names for certificates"
- "Allow expired certificates"
- "Allow certificates signed by unknown publishers"

Reading out an IP address

If the server has already been commissioned, you can read out the IP address on the server as well:

- For Panel
  Click "Start > Programs > Command Prompt" on the server and enter the "ipconfig" command using the screen keyboard. The IP address is displayed after pressing <Ret>.
- For PC / Panel PC
  Click "Start > Run" on the server, enter "Cmd", and press <Ret>: The command interpreter is displayed. Enter the "ipconfig" command. The IP address is displayed after pressing <Ret>.

See also

Basic information on SIMATIC HMI HTTP (Page 853)

Configuring a connection via SIMATIC HMI HTTP (Page 855)

Configuring a client (Page 858)

2.14.2.4 Assigning tag parameters in the client

Assigning tag parameters in the client

Introduction

In order to be able to access tags on the server, they must be configured in the client as tag addresses.

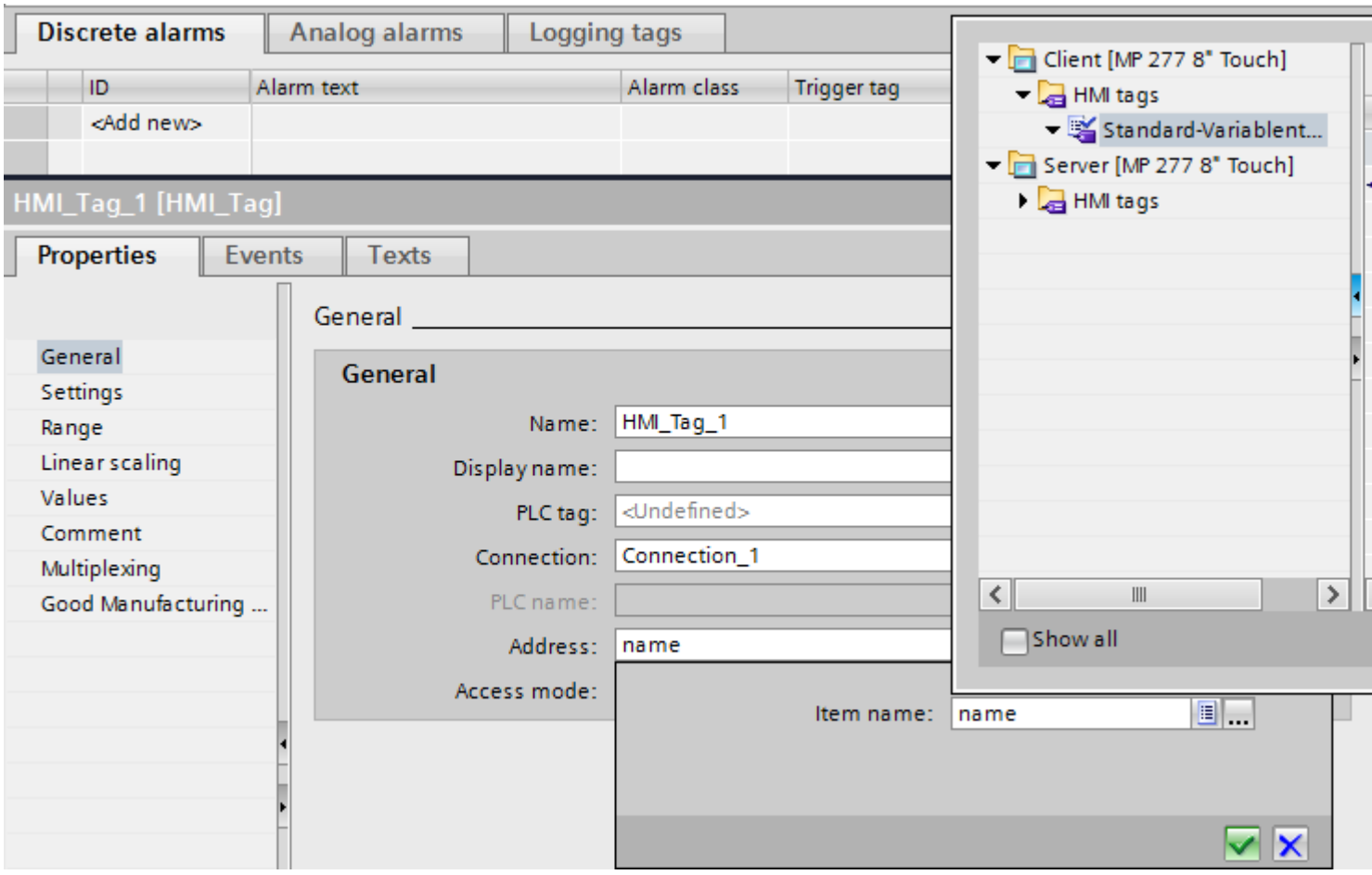
Requirements

- Tags have been created in the "HMI tags" editor of the server.

Procedure

1. Double-click the client under "Devices" in the project tree.
2. Double-click "HMI tags".
3. Create a new tag.
4. Select a data type.

5. Open the selection dialog in the "Addresses" column.



6. Define the parameters in the working area:
   See "Parameters for the client tags" for more information about the parameters.

See also

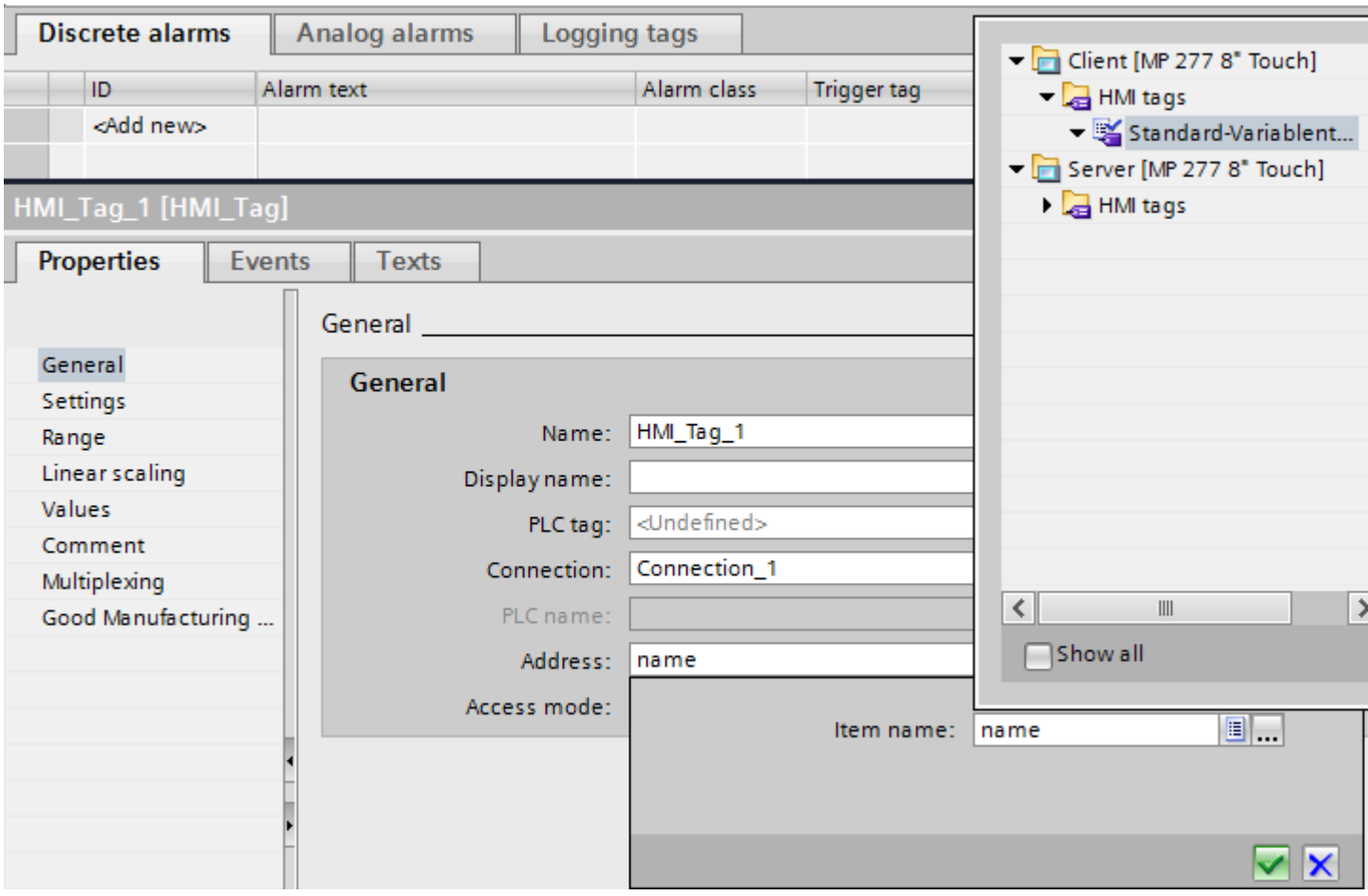
Basic information on SIMATIC HMI HTTP (Page 853)

Configuring a connection via SIMATIC HMI HTTP (Page 855)

Parameters for the tags of the client (Page 863)

Parameters for the tags of the client

Parameters for the tags



Data type

- The client does not check the data type. Therefore, pay attention that the data type selected here matches the data type of the tags on the server.
- Array tags are not permitted.

Addressing

If the server and client are not within the same project, note the following:

- Enter the exact name of the tag that is to be communicated with on the server.

See also

Basic information on SIMATIC HMI HTTP (Page 853)

Configuring a connection via SIMATIC HMI HTTP (Page 855)

Assigning tag parameters in the client (Page 861)

2.14.2.5 Commissioning a connection

Setting the server's Internet Settings

Introduction

To commission an HTTP/HTTPS connection, the WinCC Internet Settings must be set up in the Control Panel of both the server and client, in addition to the configuration in WinCC.

The name and number of tabs contained in the "WinCC Internet Settings" dialog depend on the software installed.

WinCC Internet Settings, Proxy tab

The valid network settings are specified by the network administrator.

Make the following settings:

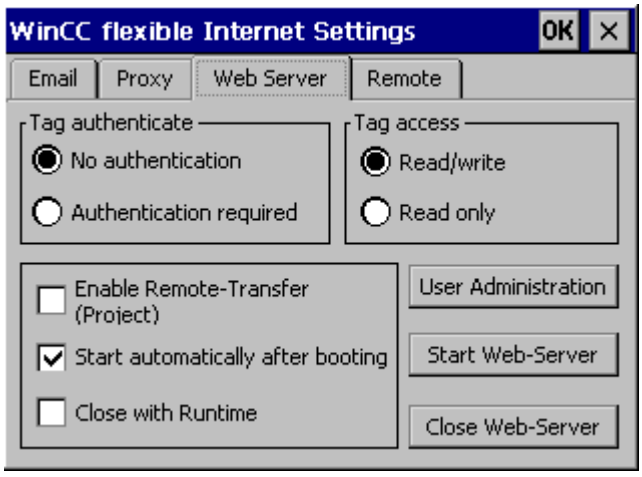
- Internet settings for HMI devices with Windows CE
  In this case, there are several options for calling the Control Panel:
  - In the startup phase, click on the "Control Panel" button in the Windows CE Loader menu.
  - Select "Start > Settings > Control Panel" in the toolbar.
  - During normal operation, pressing the key combination <CTRL + ESC> opens the toolbar. Start the Control Panel by selecting "Start > Settings > Control Panel".
  - Using the RT function "OpenControlPanel"

  The language in the Windows CE based devices is always English.

WinCC Internet Settings, Web Server tab

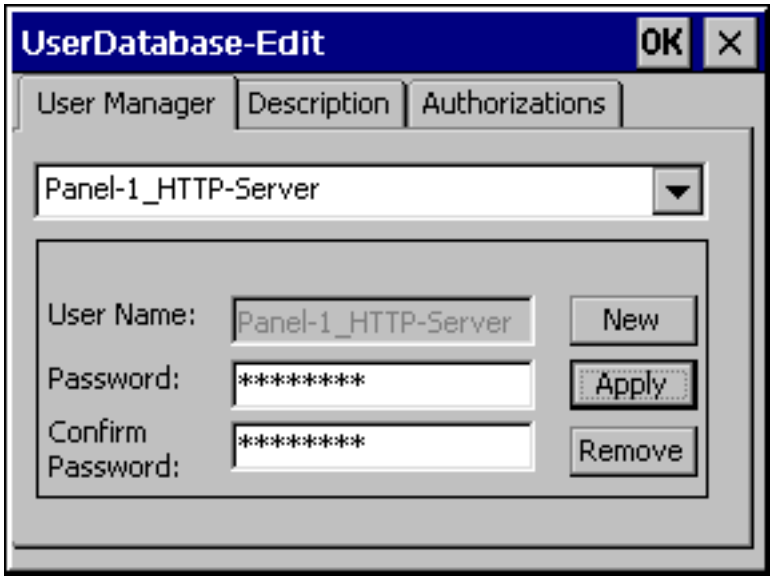
Settings for using the integrated Web Server are made in this tab.

The "Web Server" tab is only available in the WinCC Internet Settings if the Web Server is installed on the HMI device.



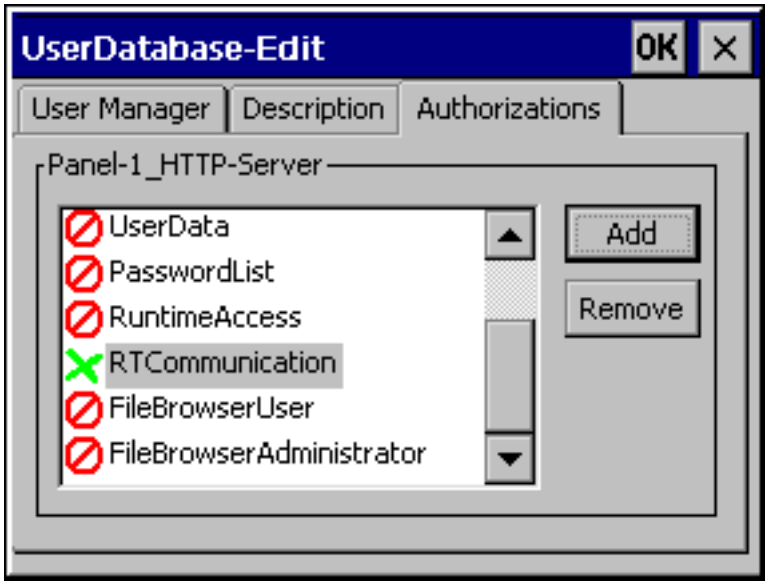
- Tag authenticate
  Controls authorization for tag access:
  - "No authentication": Authorization (user name and password) is not required for access.
  - "Authentication required": Authorization (user name and password) is required for access. The user name and password for the client are specified during configuration.
- Tag access
  Controls access to tags:
  - "Read/write": Tags can be read and rewritten.
  - "Read only": Tags can only be read.
- Group for defining operational performance
  The following check boxes are not selected in conjunction with tag exchange via the HTTP/ HTTPS protocol:
  - Enable remote transfer (project)
    Select this check box to enable an HTTP transfer from the configuration computer to the HMI device.
  - Start automatically after booting
    Specifies when the HTTP server should be started:
    Enabled: The HTTP server starts immediately after the HMI device is booted, irrespective of the runtime software
    Disabled: The HTTP server starts at the same time as the runtime software.
  - Close with Runtime
    Enabling this check box causes the HTTP server to be closed at the same time as the runtime software.

- User Administration
  This button opens the "UserDatabase-Edit" dialog box.



  In the "User Manager" tab of this dialog box, you can select, delete, or create a new user. The "Description" and "Authorizations" tabs always apply to the user currently selected.
  (This example dialog box shows a user with the name 'Panel-1_HTTP-Server'.)

- User Manager
  If a user name and password have been entered for the HTTP server under "Connections" in your WinCC project, the same user name and password must be specified on the server.
- Description
  Enables input of a user description.
- Authorizations
  The newly set up user must be assigned the authorizations for "RT Communication on the "Authorizations" tab control.



See also

Basic information on SIMATIC HMI HTTP (Page 853)

Configuring a connection via SIMATIC HMI HTTP (Page 855)

Establishing an HTTP connection (Page 867)

Establishing an HTTP connection

Introduction

To establish an HTTPS connection, you must perform the following actions:

- In the "Connections" editor of WinCC, configure the connection as an "https://" protocol type and define how the HTTPS client should verify the properties of the server certificate and respond to errors.
- Install a valid certificate on the HTTPS client.
  Certificates are necessary for server authentication. Using certificates you can ensure that the server with which the connection is to be developed is actually the server for which it is outputting.

Principle of an HTTPS connection

After runtime start, the HTTPS client establishes a connection to the HTTPS server. The HTTPS server presents its certificate, which the client verifies for authenticity. The session code that can only be read by the HTTPS server is then transmitted. The session code is now available on both sides and enables a symmetrical data encryption.

Note

The certificate contains the current time. The current time can lead to problems if the time zones of the server and client are different. For example, a certificate generated on a server with an Asian time zone only becomes valid on a client with European time zone in the future (8 hours).

Preparation for installing a certificate on the client

With the first HTTPS client access, the HTTPS server generates the certificate itself. The HTTPS server then saves it in the "Cert.cer" file. The file is stored in the following directory:

- On a PC/Panel PC (with Windows) in the directory "<Runtime Directory>\SystemRoot\SSL"
- On Windows CE-based devices in the directory "Flash\Simatic\SystemRoot\SSL"
  The certificate must be stored on the HTTPS client on a storage medium from which it can be launched with a double click.

Server	Client	Possible file transfer
with Windows (PC, Panel PC)	with Windows (PC, Panel PC)	• Disk • USB stick • LAN (Ethernet) • Internet Explorer (via TCP/IP, if Service is already running)
with Windows CE (Comfort Panels)	with Windows (PC, Panel PC)	• Memory card • ActiveSync (serial)
with Windows (PC, Panel PC)	with Windows CE (Comfort Panels)	
with Windows CE (Comfort Panels)	with Windows CE (Comfort Panels)	• Memory card

Installing a certificate on a client with Windows

Insert the storage medium on which you have saved the "Cert.cer" file into the HTTPS client or open the directory in which the file is located. Double click on the file and follow the instructions in the Windows dialog.

Tip: The Internet Explorer provides an easy way to install a certificate. Connect to this device via HTTPS (e.g.: https://<my device>). The browser establishes if a certificate has not yet been imported. In this case, the browser asks if you want to install the certificate. Any errors in the certificate will be displayed.

Installing a certificate on a client with Windows CE

Insert the memory card on which you have saved the converted "Cert.cer" file into the HTTPS client. WinCC provides the "InstallCert.exe" tool for importing certificates with Windows CE.

You can implement the installation as follows:

- In Explorer:
  Double click the "Cert.cer" file to install the certificate.
- At the command prompt:
  Enter "InstallCert /[command parameter] [filename]".
  - command parameters:
    /r parameter must be specified because the certificate used in WinCC Runtime is a root certificate.
    A root certificate is the main certificate and is used to verify the authenticity of all other certificates transferred.
  - filename
    You must specify the certificate file with its complete path (e.g. "\Storage Card\Cert.cer")

A status alarm is output when you completed the installation. Runtime has to be restarted after the installation of a certificate on Windows CE- HMI devices with HTTPS clients. It is necessary to restart Runtime so that an HTTPS connection can be established.

The file "Cert.cer" cannot be opened.

If the "Cert.cer" file generated on the HTTPS server cannot be opened on HMI devices based on Windows CE 5.0 by double-clicking the client, follow these steps:

1. Open the Control Panel.
2. Select "Certificates > My Certificates."
3. Click the "Import" button.
   A dialog box opens.
4. Select the "From a File" menu in the file browser and select the "Cert.cer" file.

See also

Basic information on SIMATIC HMI HTTP (Page 853)

Configuring a connection via SIMATIC HMI HTTP (Page 855)

Setting the server's Internet Settings (Page 864)

2.14.3 Performance features of communication

2.14.3.1 Permitted data types

Permitted data types

When configuring tags, the data types listed below can be used.

Data type in the SIMATIC HMI HTTP protocol	Length	Sign	Range of values
Bool	0	No	true (-1) or false (0)
SInt	1 Byte	Yes	-128 to 127
USint	1 Byte	No	0 to 255
Int	2 Byte	Yes	-32768 to 32767
UInt	2 Byte	No	0 to 65535
DInt	4 Byte	Yes	-2,147,483,648 to 2,147,483,647
UDInt	4 Byte	No	0 to 4,294,967,295
Real	4 Byte	Yes	-3.402823E38 to -1.401298E-45 for negative values and 1.401298E-45 to 3.402823E38 for positive values
LReal	8 Byte	Yes	-1.79769313486231E308 to -4.94065645841247E-324 for negative values and 4.94065645841247E-324 to 1.79769313486232E308 for positive values

Data type in the SIMATIC HMI HTTP protocol	Length	Sign	Range of values
String	1 to 255 byte	–––	
DateTime	8 Byte	–––	1.1.1970 00:00:00 up to 31.12.2037 23:59:59

Note

Data types for third-party PLCs

Data types in third-party PLCs may differ from those in WinCC.

Note the the tag definition of the third-party PLC for the correct assignment.

Note

It is not possible to access array tags from an HTTP client.

Multiplex tags are not supported by the HTTP protocol. If you need a multiplex tag, you must configure it on the HTTP client. You will have to create the tags for the multiplex list on the server and the client and then connect them. You will then add these tags to the multiplex list on the client.

Address multiplex tags are not supported by the HTTP protocol.

See also

Basic information on SIMATIC HMI HTTP (Page 853)

2.15 Communication via OPC

2.15.1 Communication via OPC

Introduction

This section describes the communication between an HMI device and a device via OPC.

In this case, the HMI device is the OPC client and the device is the OPC server.

For additional information on OPC and on configuring OPC servers, see section "OPC (Page 18)".

See also

Configuring a connection via OPC (Page 871)

Configuring an OPC server (Page 18)

2.15.2 Configuring a connection via OPC

Introduction

You configure a connection to a device via OPC in the "Connections" editor of the HMI device.

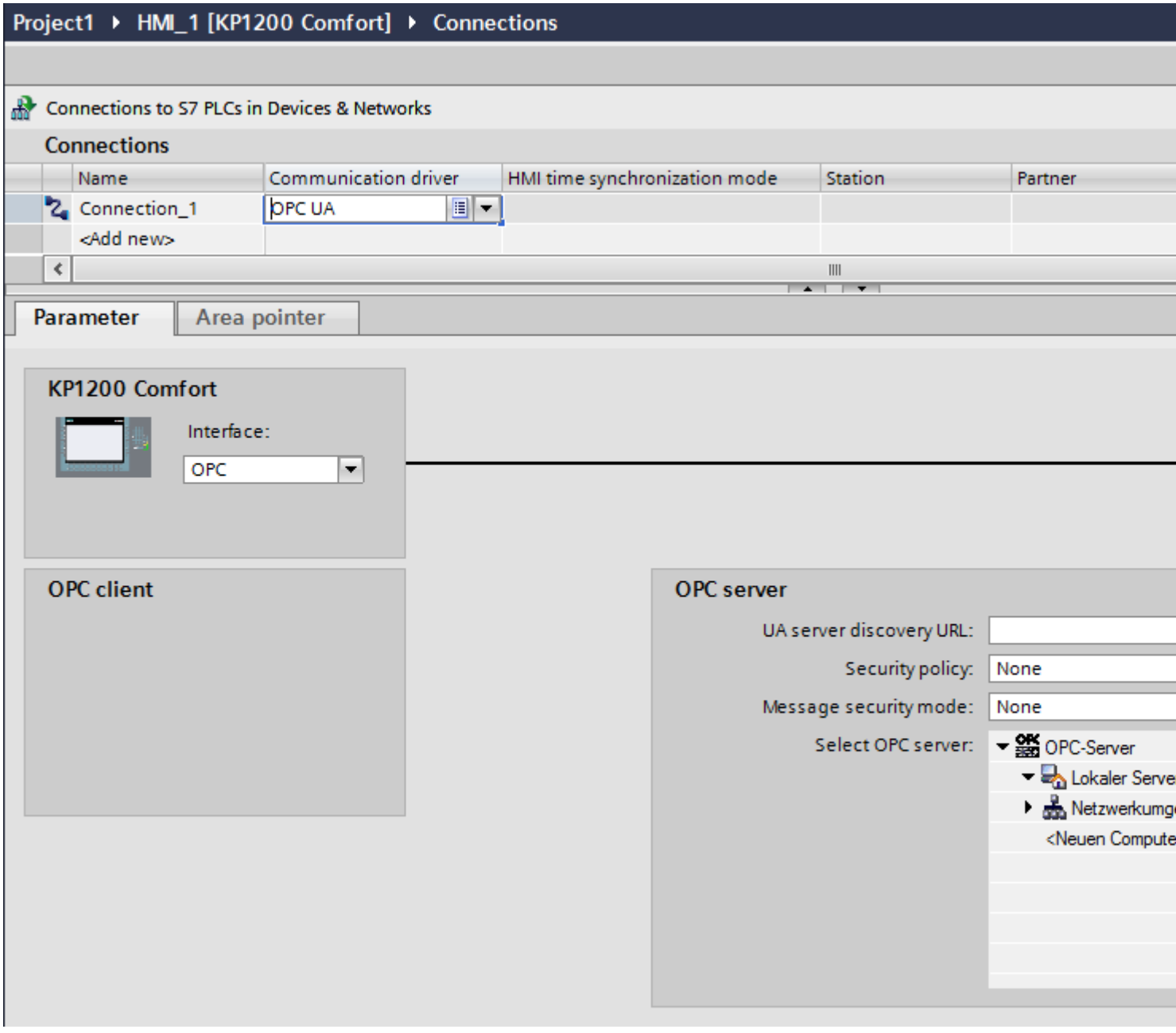
Requirements

- A project is open.
- An HMI device has been created.
- An entry has been stored for the HMI device in the OPC XML Manager.

Procedure

1. Double-click the HMI device under "Devices" in the project tree.
2. Double-click "Runtime settings".
3. Select the function "Operate as OPC server" under "Services > Services in runtime".
4. Double-click on the "Connections" entry in the project tree.
5. Double-click "<Add>" in the "Connections" editor.
6. In the "Communication drivers" column, select the "OPC" driver.

7. Select the "OPC" interface for the HMI device under "Parameters > Interface".



8. Select an OPC server from the tree view under "Parameters > OPC servers".
   The OPC server is entered in the "Name of OPC server" field.

"Coordination" area pointer in an OPC connection

In principle, the "Coordination" area pointer can be used eight times in an OPC connection.

If you have configured an OPC connection and you automatically create another OPC connection using "Add", the "Coordination" area pointer is only displayed once in the newly created connection. Change the communication driver of the connector and then set OPC

as communication driver again. After this, the "Coordination" area pointer can again be used eight times.

See also

Communication via OPC (Page 870)

2.15.3 Performance features of communication

2.15.3.1 Permitted data types for OPC DA

Permitted data types for connections with OPC DA

The table lists the user data types that can be used when configuring tags.

Data type	Length
VT_BOOL	1-bit
VT_I1	1 byte
VT_UI1	1 bytes
VT_I2	2 bytes
VT_UI2	2 bytes
VT_I4	4 bytes
VT_UI4	4 bytes
VT_R4	4 bytes
VT_R8	8 bytes
VT_BSTR	--
VT_DATE	8 bytes
VT_DATE|VT_ARRAY	--
VT_I1|VT_ARRAY	--
VT_UI1|VT_ARRAY	--
VT_I2|VT_ARRAY	--
VT_UI2|VT_ARRAY	--
VT_I4|VT_ARRAY	--
VT_UI4|VT_ARRAY	--
VT_R4|VT_ARRAY	--
VT_R8|VT_ARRAY	--

See also

Communication via OPC (Page 870)
Permitted data types for OPC XML DA (Page 874)
Permitted data types for OPC UA (Page 875)

2.15.3.2 Permitted data types for OPC XML DA

Permitted data types for connections with OPC DA XML

The table lists the user data types that can be used when configuring tags.

Data type	Length
boolean	1-bit
byte	1 byte
unsignedByte	1 byte
short	2 bytes
unsignedShort	2 bytes
int	4 bytes
unsignedInt	4 bytes
float	4 bytes
double	8 bytes
string	--
dateTime	--
time	--
date	--

See also

Communication via OPC (Page 870)
Permitted data types for OPC DA (Page 873)
Permitted data types for OPC UA (Page 875)

2.15.3.3 Permitted data types for OPC UA

Permitted data types for connections with OPC UA

The table lists the user data types that can be used when configuring tags.

Data type	Length
Boolean	1-bit
SByte	1 byte
Byte	1 byte
Int16	2 bytes
UInt16	2 bytes
Int32	4 bytes
UInt32	4 bytes
Float	4 bytes
Double	8 bytes
String	--
DateTime	--

See also

Communication via OPC (Page 870)
Permitted data types for OPC DA (Page 873)
Permitted data types for OPC XML DA (Page 874)

2.15.4 Performance features of communication

2.15.4.1 Permitted data types for OPC DA

Permitted data types for connections with OPC DA

The table lists the user data types that can be used when configuring tags.

Data type	Length
VT_EMPTY	--
VT_BOOL	1-bit
VT_I1	1 byte
VT_UI1	1 byte
VT_I2	2 bytes

Data type	Length
VT_UI2	2 bytes
VT_I4	4 bytes
VT_UI4	4 bytes
VT_R4	4 bytes
VT_R8	8 bytes
VT_BSTR	--
VT_DATE|VT_ARRAY	--
VT_I1|VT_ARRAY	--
VT_UI1|VT_ARRAY	--
VT_I2|VT_ARRAY	--
VT_UI2|VT_ARRAY	--
VT_I4|VT_ARRAY	--
VT_UI4|VT_ARRAY	--
VT_R4|VT_ARRAY	--
VT_R8|VT_ARRAY	--

See also

Communication via OPC (Page 870)
Permitted data types for OPC XML DA (Page 876)

2.15.4.2 Permitted data types for OPC XML DA

Permitted data types for connections with OPC XML DA

The table lists the user data types that can be used when configuring tags.

Data type	Length
boolean	1-bit
byte	1 byte
unsignedByte	1 byte
short	2 bytes
unsignedShort	2 bytes
int	4 bytes
unsignedInt	4 bytes
float	4 bytes
double	8 bytes
string	--
base64Binary	--

Data type	Length
ArrayOfByte	--
ArrayOfShort	--
ArrayOfInt	--
ArrayOfUnsignedInt	--
ArrayOfLong	--
ArrayOfUnsignedLong	--
ArrayOfFloat	--
ArrayOfDecimal	--
ArrayOfDouble	--
ArrayOfBoolean	--
ArrayOfString	--
ArrayOfDateTime	--
ArrayOfAnyType	--

See also

Communication via OPC (Page 870)

Permitted data types for OPC DA (Page 875)

2.16 Communication via routing

2.16.1 Communication via routing

Introduction

If all stations in an automation system are not connected to the same subnet, these stations cannot be accessed directly online.

A connection between partners in different subnets is only possible via routing.

You configure the routing settings with the properties of the interfaces.

Communication of connection partners in different subnets is possible via routing with the following connections:

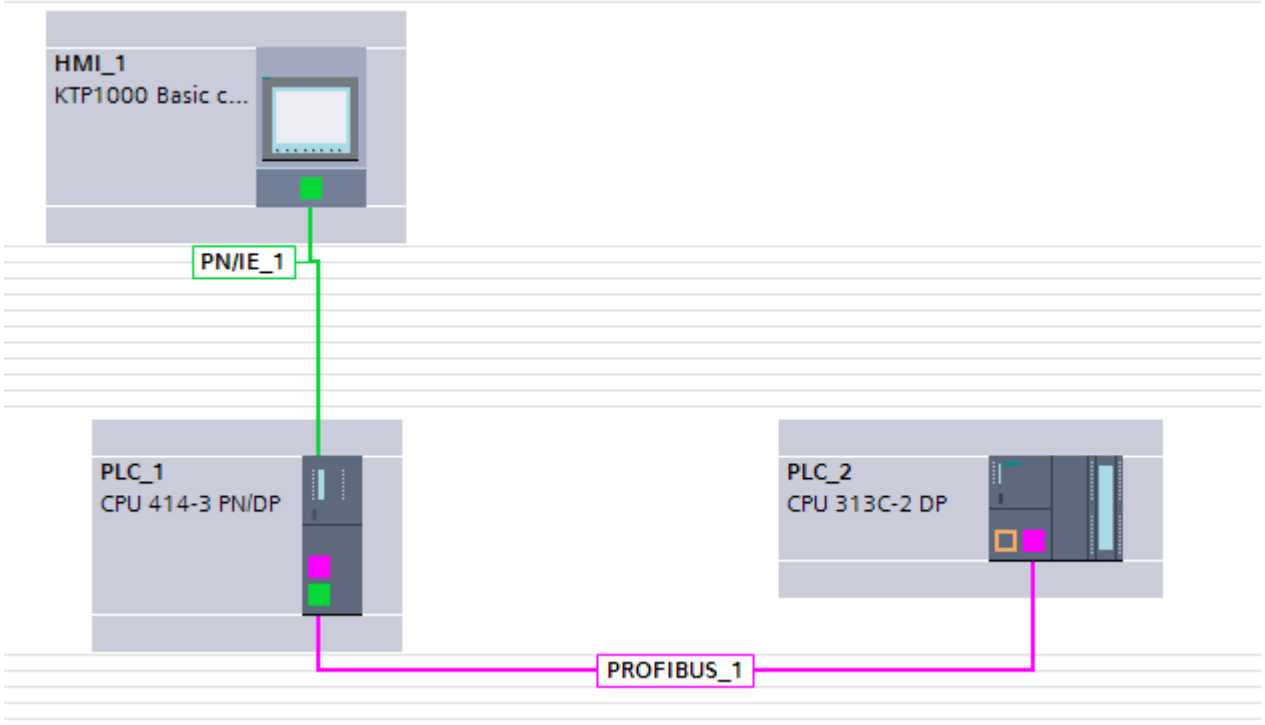
- PROFINET
- PROFIBUS
- MPI

Requirements for routing

To establish a connection to these devices, a router must be interposed. In this case, a SIMATIC station can also act as a router if it has appropriate interfaces with the different subnets.

Modules with communication capability (CPUs or CPs) used to establish gateways between the subnetworks must have routing capability.

A connection between partners in different subnets is only possible with IP routing. The routing settings can be edited in the corresponding interface properties.



Note

"Named Connection"

With routed HMI connections between WinCC RT Professional and the S7-300/400 controllers, depending on the router, the use of "Named Connection" is mandatory.

Routing path

The routing path is determined in Runtime by the system and cannot be influenced by the user. During configuration, it is not possible to generate information about a faulty connection.

HMI devices as routers

Only SIMATIC PCs or PC stations can be used as router.

All other HMI devices from the "HMI" area cannot be used as a router.

See also

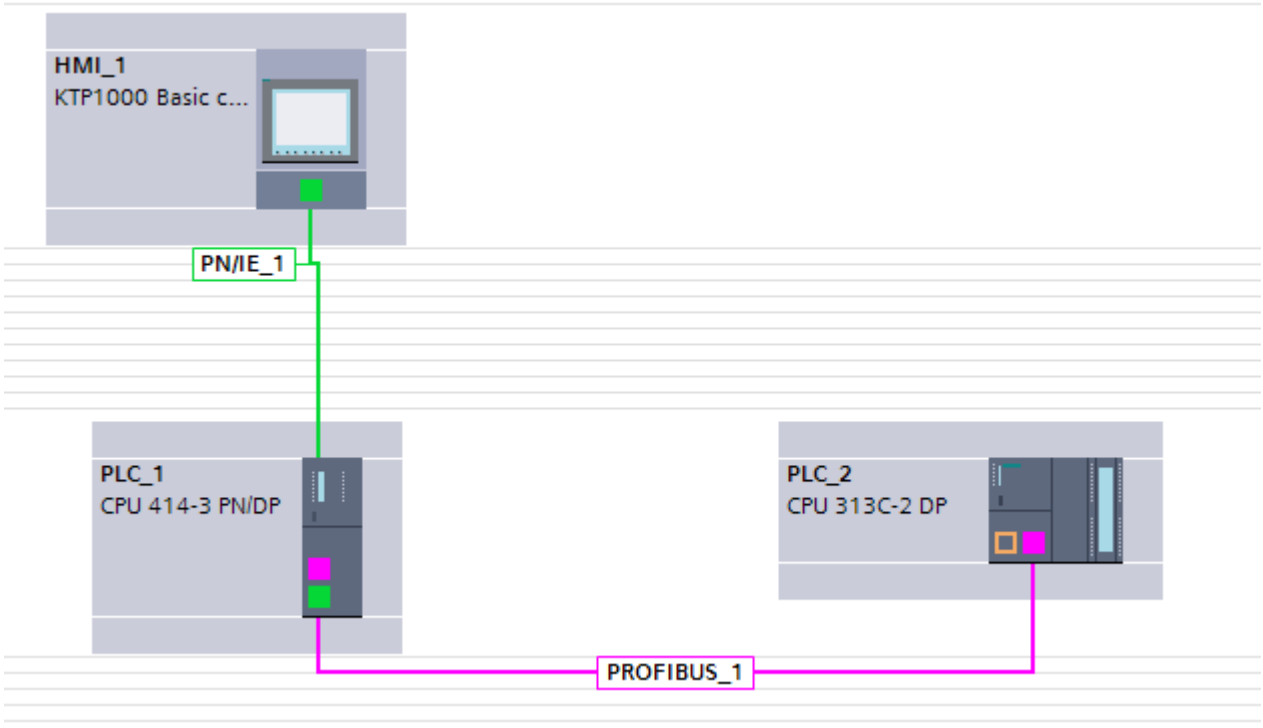
Example for communication via routing (Page 879)

Configuring communication via routing (Page 881)

2.16.2 Example for communication via routing

Communication via routing

Illustration of a hardware configuration with a routing connection

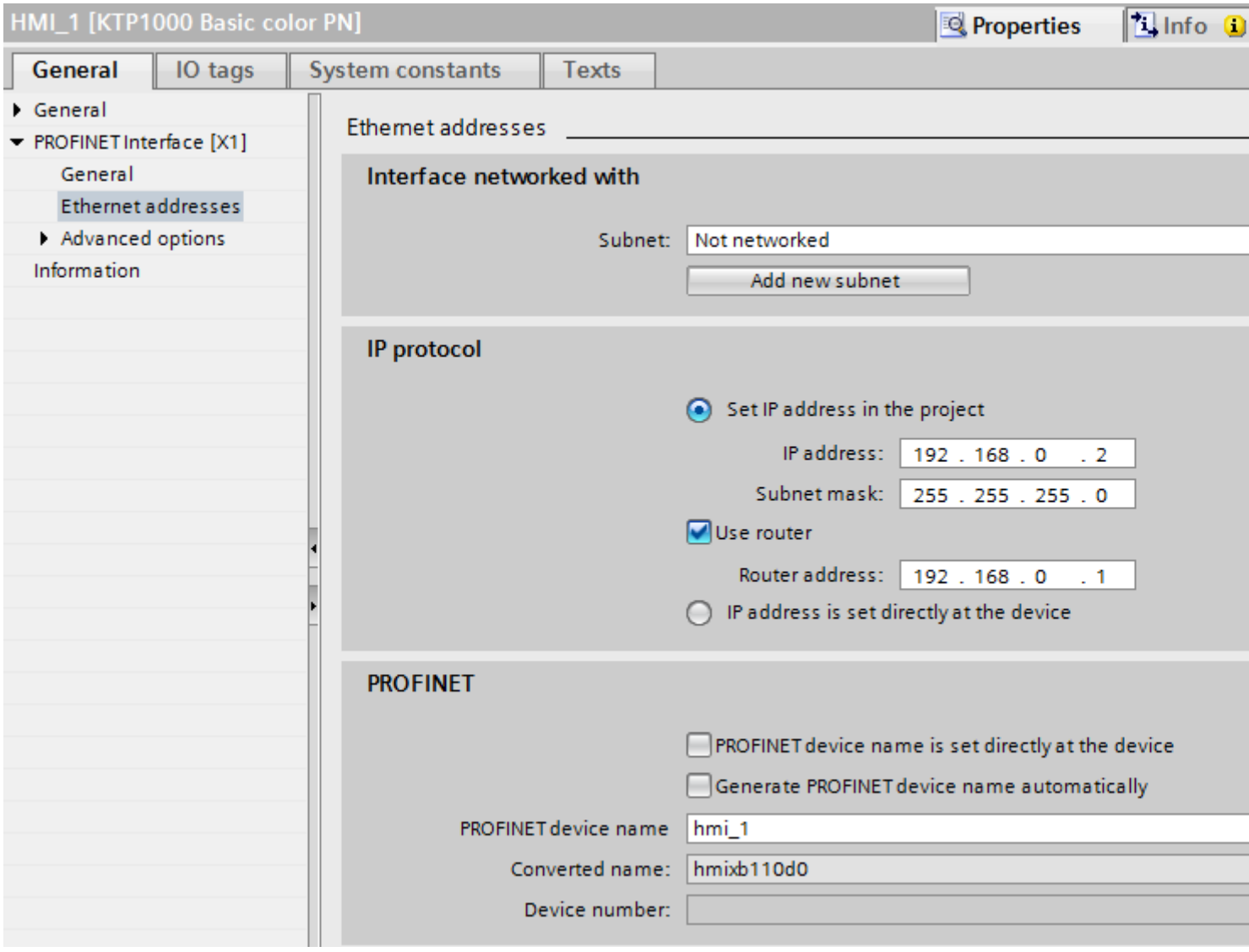


In the figure above, a routing connection has been established between an HMI device and the SIMATIC S7-300 controller.

The SIMATIC S7-400 automation device serves as a router.

A routing connection can be made directly in integrated projects. The communication partners are connected in the "Devices and Networks" editor.

You configure the parameters of the connection and interface in the Inspector window.



See also

Communication via routing (Page 877)

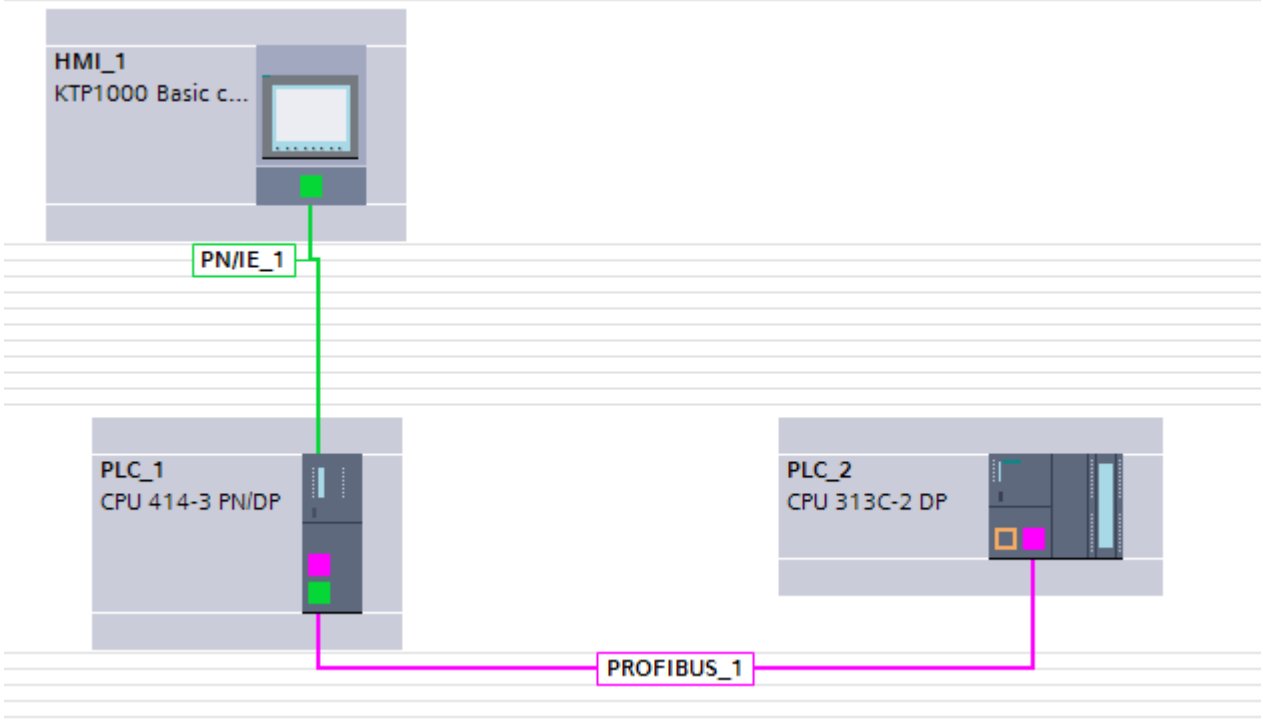
Configuring communication via routing (Page 881)

2.16.3 Configuring communication via routing

Introduction

The following configuration describes a network for the following communication partners:

- KTP1000 Basic PN
- SIMATIC S7-400 with PROFINET and PROFIBUS interface
- SIMATIC S7-300 with PROFIBUS interface



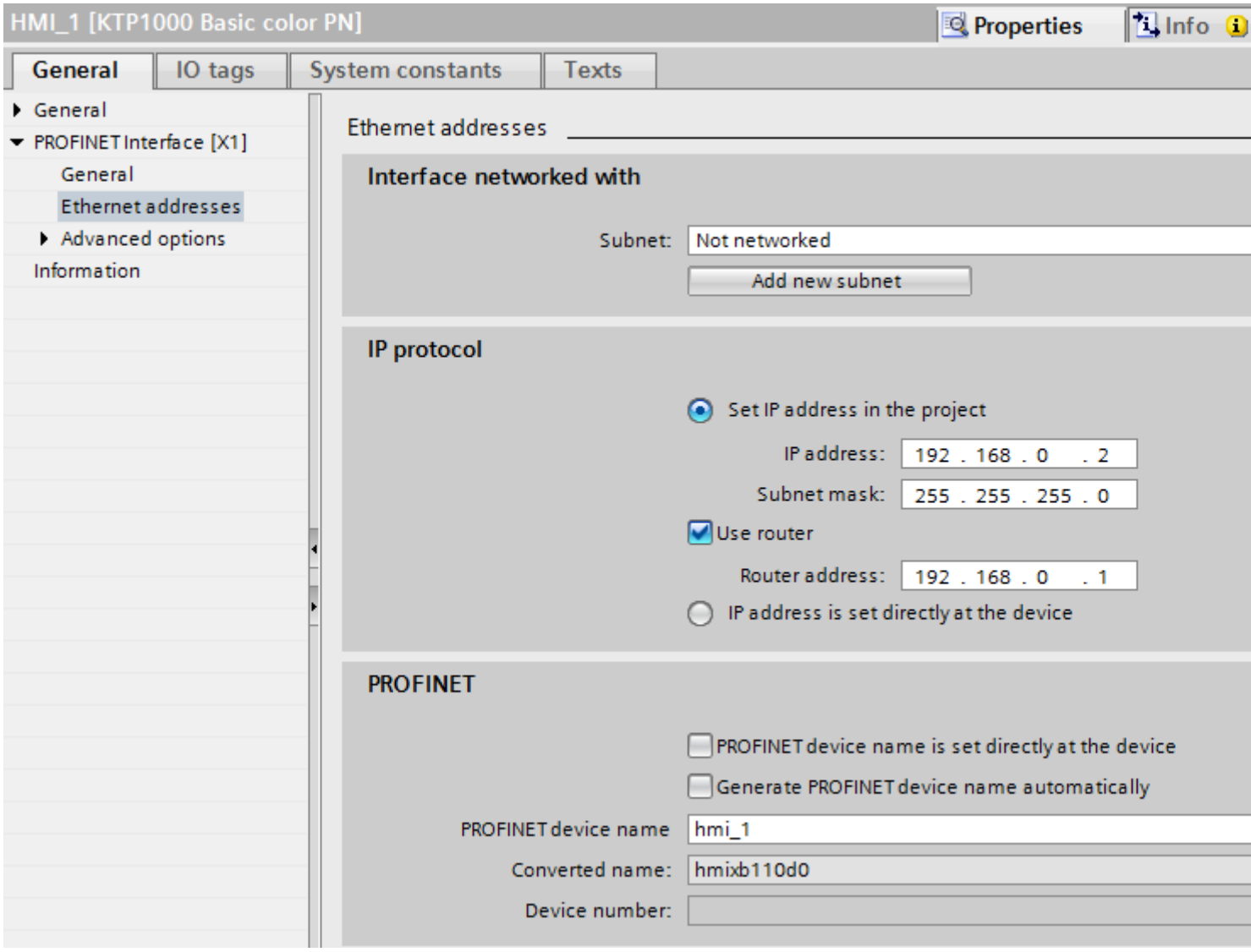
Requirements

- KTP1000 Basic PN is connected via PROFINET to the SIMATIC S7-400.
- SIMATIC S7-400 is connected via PROFIBUS to the SIMATIC S7-300.

Procedure

1. Click on the PROFINET interface of the HMI device.
2. In the Inspector window, click "Properties > General > Ethernet addresses".

3. Click "Use router" in the "IP protocol" area.
4. Enter the IP address of the router in the "Router address" field.



See also

Communication via routing (Page 877)

Example for communication via routing (Page 879)

2.17 PROFINET IO and IRT

2.17.1 PROFINET IO

2.17.1.1 What is PROFINET IO?

PROFINET IO

PROFINET is an Ethernet-based automation standard of PROFIBUS Nutzerorganisation e.V. (PNO) which defines a manufacturer-neutral communication, automation and engineering model.

Objective

The objective of PROFINET is:

- Integrated communication via field bus and Ethernet
- Open, distributed automation
- Use of open standards

Architecture

The PROFIBUS User Organisation e.V. (PNO) has designated the following aspects for PROFINET architecture:

- Communication between controllers as components within distributed systems.
- Communication between field devices, such as I/O devices and drives.

Implementation by Siemens

The demand for "Communication between controllers as components within distributed systems" is implemented by "Component Based Automation" (CBA). Component Based Automation is used to create a distributed automation solution based on prefabricated components and partial solutions.

The demand for "Communication between field devices" is implemented by Siemens with "PROFINET IO". Just as with PROFIBUS DP, the complete configuration and programming of the components involved is possible using the Totally Integrated Automation Portal.

The following sections deal with the configuration of communication between field devices using PROFINET IO.

See also

PROFINET IO systems (Page 884)

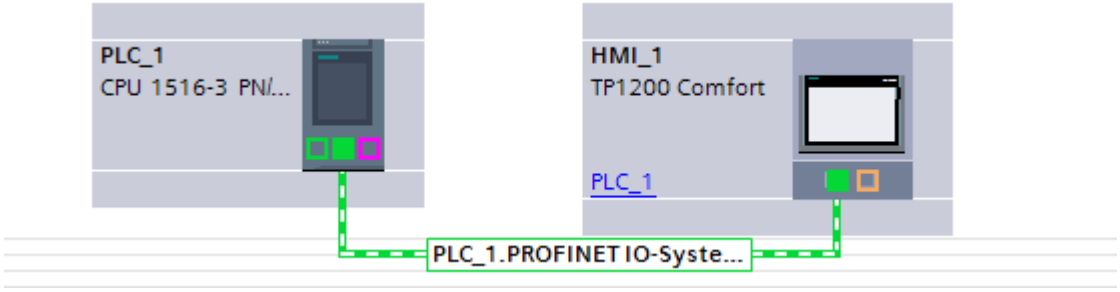
2.17.1.2 PROFINET IO systems

PROFINET IO system

A PROFINET IO system is comprised of a PROFINET IO controller and its assigned PROFINET IO devices.

- A PROFINET IO controller is a PLC that controls the automation task.
  Example: SIMATIC S7 1500 PLC
- A PROFINET IO device is a device that is controlled by an IO controller.
  Examples: HMI device TP 1200 Comfort, a head module of the distributed IO family ET 200S

As soon as you connect an IO controller to an IO device, a controller-device link is established.



See also

What is PROFINET IO? (Page 883)

2.17.2 IRT communication

2.17.2.1 Introduction: Isochronous Realtime Ethernet

Introduction: Isochronous Realtime Ethernet

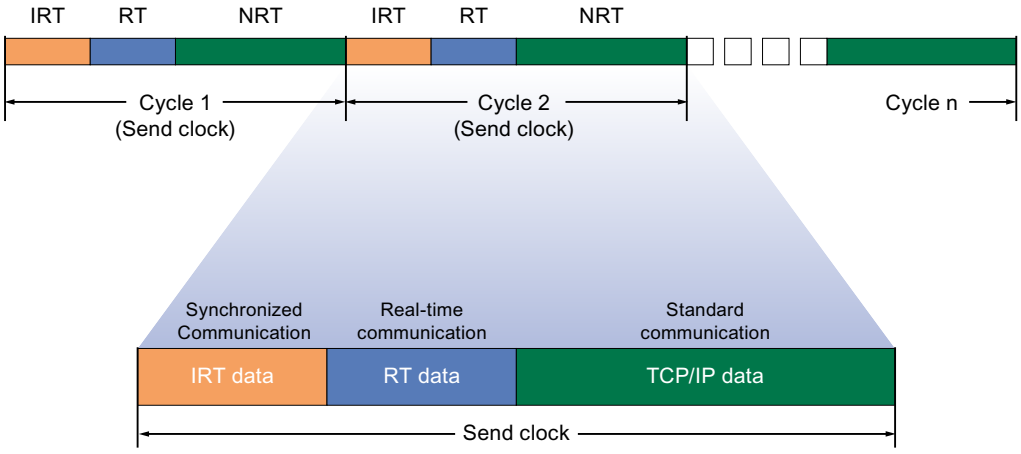
IRT is a transmission method by which PROFINET devices are synchronized with very high accuracy.

A sync master specifies the clock to which sync slaves are synchronized. An IO controller or an IO device can have the role of sync master.

Sync master and sync slaves are always devices in a sync domain. Bandwidth is reserved within the sync domain for IRT communication. Real-time and non-real-time communication (TCP/IP communication) is possible outside of the reserved bandwidth.

Time ranges of communication cycles

The communication cycle is divided into three time ranges, which are represented in the following chart:



- IRT data (synchronized communication)
  This area can be reserved in specific steps, depending on the send clock. Within this time range the IRT data is transmitted.
- RT data (real-time communication)
  In this time range the cyclic RT date is transmitted. RT data has a higher priority than "normal" TCP/IP data. TCP/IP data or Ethernet frames can have a priority between 1 and 7. RT data has a priority of 6.
- TCP/IP data (standard communication)
  Standard communication (TCP/IP, etc.) is transmitted in the remaining interval of the communication cycle.

Applications of IRT

PROFINET with IRT is especially suited for:

- High performance and deterministics for large quantity structures with regard to user data communications (productive data)
- High performance even with many devices in the line topology with regard to user data communications (productive data).
- Parallel transmission of production and TCP/IP data over wire, and securing the forwarding of production data when data volume is high by reserving transmission bandwidth.

See also

Overview of RT classes (Page 886)

Basic procedures for configuring IRT (Page 886)

Sync domain (Page 887)

2.17.2.2 Overview of RT classes

RT classes in PROFINET IO

PROFINET IO is a scalable, real-time communication system based on Ethernet technology. The scalable approach is reflected in several real-time classes:

- **RT:** Transmission of data in prioritized, non-isochronous Ethernet frames. The required bandwidth is within the free bandwidth area for TCP/IP communication.
- **IRT:** Isochronous transmission of data with high stability for time-critical applications (for example, Motion Control). The required bandwidth is from the area of bandwidth reserved for cyclic data.

Device dependency

Depending on the device, not all real-time classes are supported.

You can find further information on device dependency here: Device dependency (Page 891)

See also

Introduction: Isochronous Realtime Ethernet (Page 884)

Basic procedures for configuring IRT (Page 886)

Sync domain (Page 887)

Device dependency (Page 891)

2.17.2.3 Basic procedures for configuring IRT

Basic procedures for configuring IRT

If you want to upgrade equipment with PROFINET IO to IRT, follow these three steps:

1. Configure the PROFINET IO device. The PROFINET device must support IRT.
2. Set which device acts as the sync master and synchronizes the other devices. For this you must configure a sync domain with a sync master and at least one sync slave.
3. Download the configuration to all involved devices.

Creating PROFINET IO configuration for IRT

As a prerequisite for IRT configuration, there must be an existing PROFINET IO configuration, i.e., one or more stations must be configured with an IO controller and IO devices.

You can read which components are suitable for IRT communication in the task card "Information", once you have selected the corresponding components in the hardware catalog.

The requirements for functioning IRT communication are met simply by configuring a PROFINET IO system with components that support IRT.

See also

Introduction: Isochronous Realtime Ethernet (Page 884)

Overview of RT classes (Page 886)

Sync domain (Page 887)

2.17.2.4 Sync domain

Sync domain

A sync domain is required for synchronization of PROFINET IO devices. The sync domain assures that all devices belonging to it communicate synchronously.

Prerequisite for IRT communication is a synchronization cycle to distribute a common time base for all PROFINET devices in a sync domain. With this base synchronization, a synchronous transmission cycle of PROFINET devices within a sync domain is achieved. The sync master (usually an IO controller) generates the common synchronization clock and specifies the time basis on which all other sync slaves (usually IO devices) are synchronized.

If the sync master fails, the communication between the IRT devices reverts to RT communication.

Background processes during configuration of an IO system

If you configure an IO controller and network it with an Ethernet subnet, it will automatically be added to the default sync domain of the Ethernet subnet. The default sync domain is always available. The IO controller operates unsynchronized.

If you assign an additional IO device to the IO system of the IO controller, the IO device automatically assigns the IO controller's sync domain.

See also

Introduction: Isochronous Realtime Ethernet (Page 884)

Overview of RT classes (Page 886)

Basic procedures for configuring IRT (Page 886)

2.17.3 Configuring the HMI device as IO device

Introduction

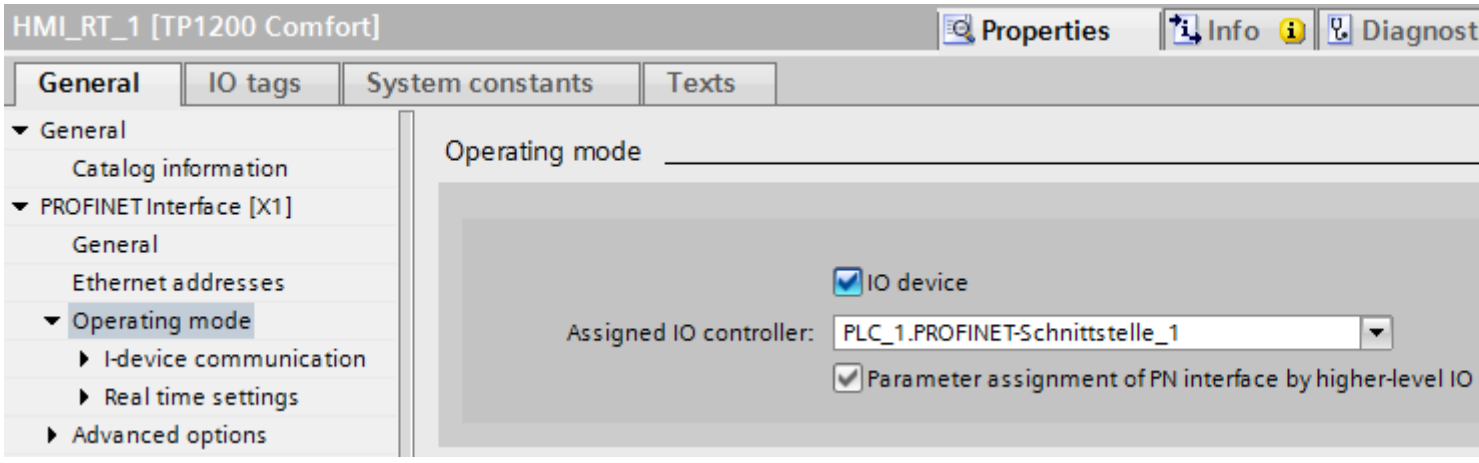
An IO controller and at least one PROFINET IO device are necessary for a PROFINET IO system.

Requirements

- A SIMATIC S7 1200 or SIMATIC S7 1500 PLC has been created.
- An HMI device from the Comfort Panels device family has been created.
- The "Devices & Networks" editor is open.

Procedure

1. You network the PLC with the HMI device:
2. Establish an HMI connection between PLC and HMI device.
3. Click the HMI device in the "Devices & Networks" editor.
4. In the inspector window, select the following:
   "General > PROFINET Interface (X1) > Operating mode"
5. Enable the "IO device" option.



Note

IP address independent of the higher-level IO controller

If the IO device is an HMI device, the IP address is set by the IO device, even if the "Parameter assignment of PN interface by higher-level IO controller" option is set.

6. You assign the parameters required for your PROFINET IO system

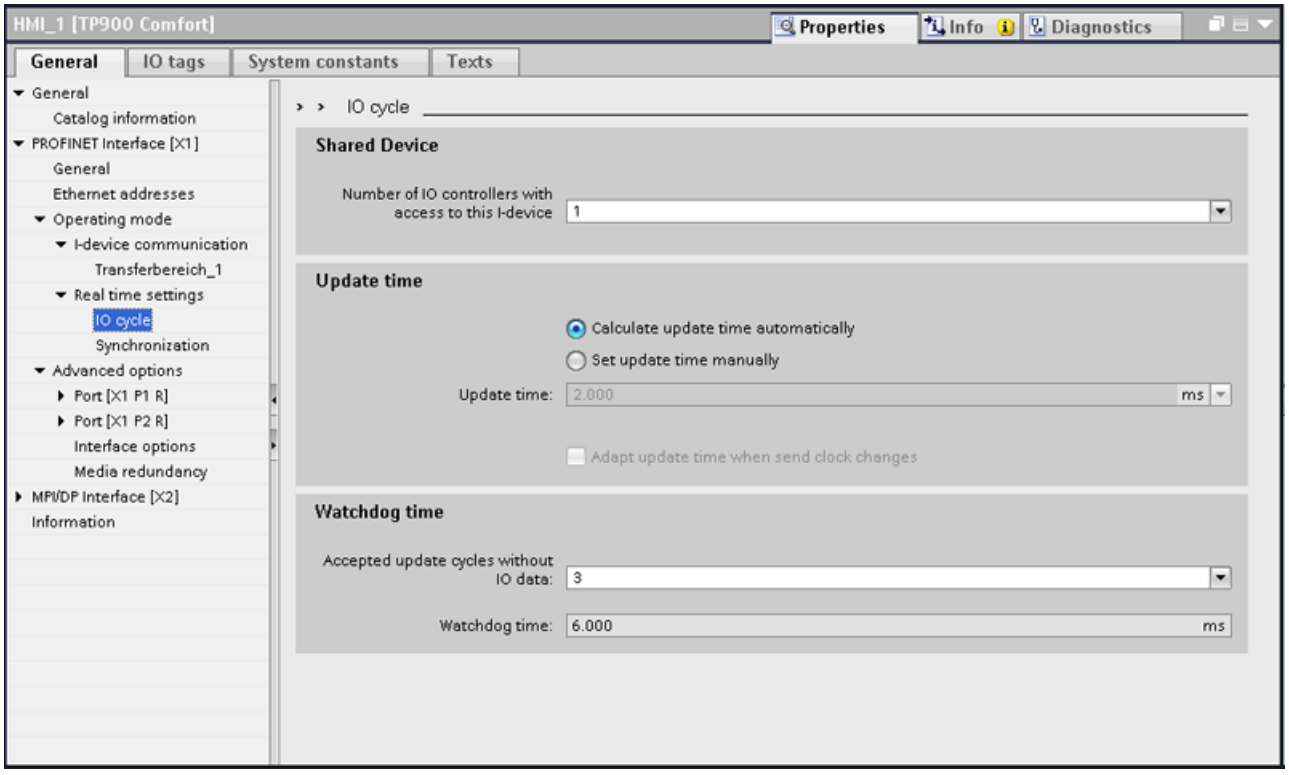
See also

Parameters for PROFINET IO and IRT (Page 889)

2.17.4 Parameters for PROFINET IO and IRT

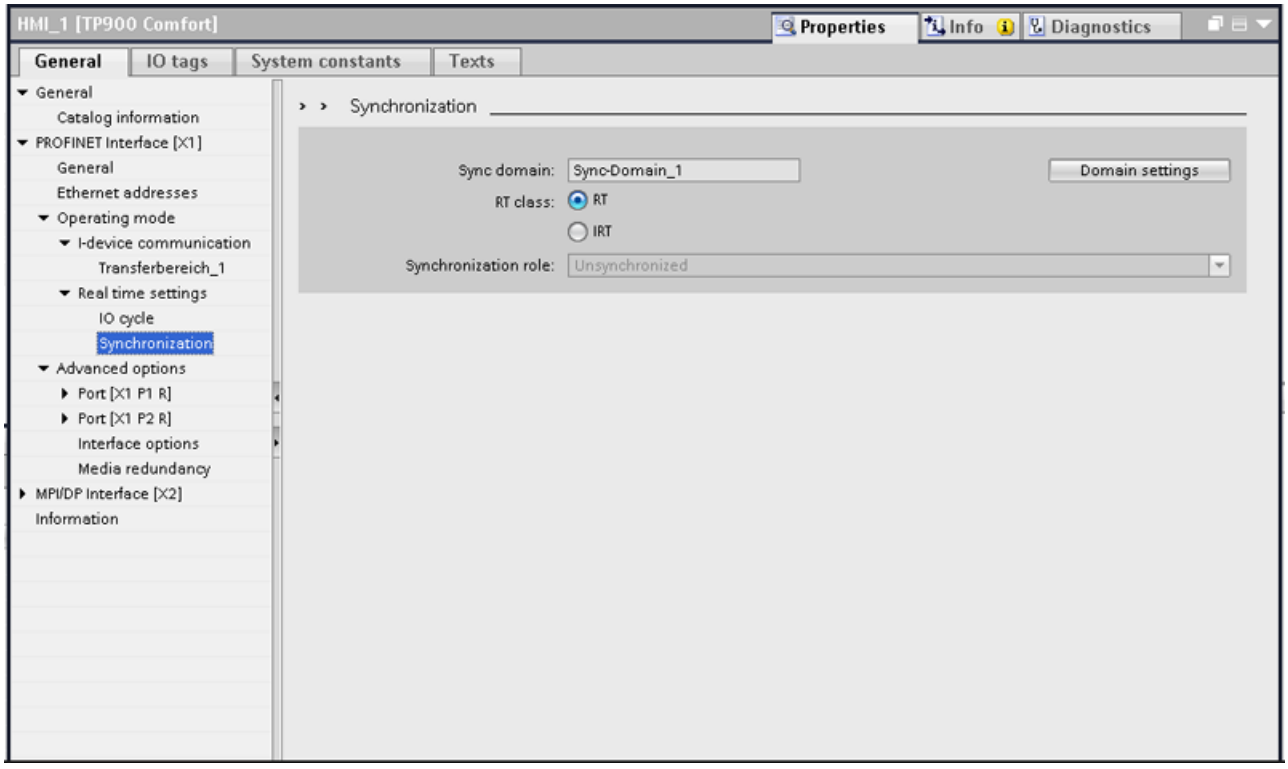
2.17.4.1 Real-time settings

Parameters for real-time settings



Under "Real time settings > IO cycle", set the following parameters:

- Shared Device: Number of IO controllers with access to this I-device
- Update time
  - Display of update time used in the network
  - Selection of update times available in the network
- Watchdog time
  - Selection of possible update cycles
  - Display of watchdog time used in the network



Under "Real time settings > Synchronization", set the following parameters:

- Display of sync domain used
- Selection of RT class
- Selection of role in the sync domain

2.17.5 Performance features of PROFINET IO and IRT

2.17.5.1 Configuration of PROFINET and IRT

Setting up PROFINET with IRT

Keep in mind the following rules for the set up and operation of a PROFINET IO system in IRT operation. They serve to insure an optimal operation of your PROFINET IO system.

- When using IRT, you must configure the topology. This will enable exact calculation of the update time, bandwidth, and additional parameters.
- If you would like to use multiple sync domains, configure a sync domain boundary for the port which is currently connected to the PROFINET device of another sync domain.
- In a sync domain you can only configure one sync master at a time.
- A PROFINET IO system may only be part of a single sync domain.

- If you configure a PROFINET device in a sync domain and want to synchronize with IRT, the PROFINET devices concerned must support IRT communication.
- Wherever possible, use the same PROFINET device as the PROFINET IO controller and sync master.
- If only some of the PROFINET devices in a PROFINET IO system are synchronized, please keep in mind the following: PROFINET devices which are not part of IRT communication are placed outside of the sync domain.

See also

Device dependency (Page 891)

2.17.5.2 Device dependency

IRT communication

The following HMI devices support IRT communication:

- TP700 Comfort and KP700 Comfort
- TP900 Comfort and KP900 Comfort
- TP1200 Comfort and KP1200 Comfort
- TP1500 Comfort and KP1500 Comfort
- TP1900 Comfort
- TP2200 Comfort

The following connection boxes for Mobile Panels support IRT communication:

- Connection Box Advanced

Communication options

The following configuration options are available for Comfort Panels:

- Comfort Panel with PROFINET Basis functionality without PROFINET IO
- Comfort Panel with PROFINET RT in an RT network networked with an RT controller
- Comfort Panel with PROFINET RT as end point in an IRT network networked with an RT controller
- Comfort Panel with PROFINET IRT in an IRT network networked with an IRT controller

The following configuration options are available for Mobile Panels:

- Mobile Panel with PROFINET Basis functionality without PROFINET IO
- Mobile Panel with PROFINET IO as IO device in a PROFINET network connected to an IO controller
- Mobile Panel with PROFINET basic funtionality or with PROFINET IO as the terminal point in a IRT network (via Connection Box Advanced)

See also

Configuration of PROFINET and IRT (Page 890)

2.18 Media redundancy

2.18.1 Media redundancy

Media redundancy options

To increase the network availability of an industrial Ethernet network with optical or electrical line topology, the following options are available:

- Meshing networks
- Parallel switching of transmission modes
- Consolidation of a line topology to a ring topology

Media redundancy in ring topologies

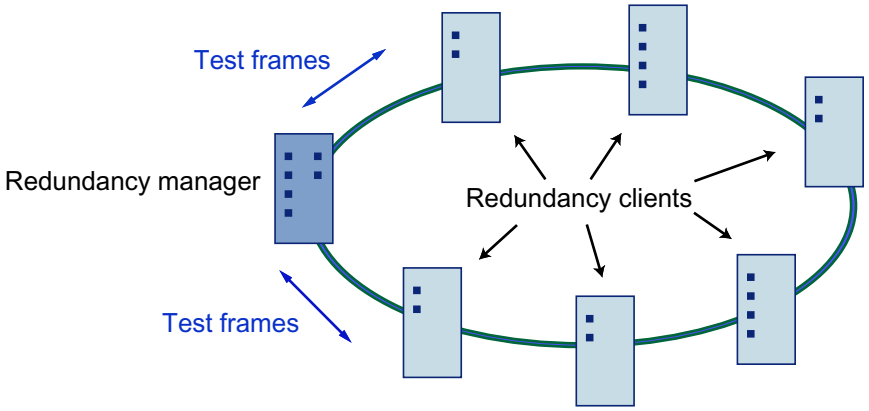
Devices in ring topologies can be the external switches and/or integrated switches of communication modules.

To set up a ring topology with media redundancy you must join both free ends of a line-shaped network topology to one device.

Consolidation of a line topology to a ring takes place over 2 ports (ring ports) in a device in the ring.

This device is the redundancy manager. All other devices in the ring are the redundancy clients.

The following figure shows the ring type interconnection:



The two ring ports in a device are the ports which create the connection to both of its neighboring devices in a ring topology.

The selection and set up of the ring ports takes place during configuration of the respective device.

Download the configuration in the individual devices before physically connecting the ring.

Functions of media redundancy in a ring topology

If a part of the ring is disconnected, then the data paths between the individual devices is automatically reconfigured. After the reconfiguration the devices are once again accessible in the newly created topology.

In the redundancy manager, 2 ring ports will be separated from each other during interruption-free network operation, so that no data frames circulate.

The ring topology is a line when viewed from the standpoint of data transfer. The redundancy manager monitors the ring topology. To do so it sends a test frame not only from ring port 1 but also from ring port 2. The test frame passes through the ring in both directions until it arrives at the other ring port on the redundancy manager.

An interruption in the ring can occur from connection failure between two devices or device failure in the ring.

If the redundancy manager's test telegram does not reach the other ring port because of interruption in the ring, the redundancy manager connects via both ports.

With this alternate route a functioning connection is again created among all remaining devices in a line-shaped topology.

The time between ring interruption and re-establishment of a functional line topology is called the reconfiguration time.

As soon as the interruption is eliminated, the original transmission route is again created, both ring ports in the redundancy manager are separated from each other, and the redundancy clients are notified of the change.

The redundancy clients then use the new route to the other devices.

If the redundancy manager fails, then the ring is changed to a functional line.

See also

Media Redundancy Protocol (MRP) (Page 894)
Configuring media redundancy for HMI devices (Page 895)
Parameters for media redundancy (Page 897)
Managing MRP domains (Page 898)
Restrictions with media redundancy (Page 900)

2.18.2 Media Redundancy Protocol (MRP)

Media Redundancy Protocol (MRP)

The MRP process works in conformity with Media Redundancy Protocol (MRP), which is specified in IEC 61158 Type 10 "PROFINET".

The reconfiguration time following an interruption of the ring amounts to a maximum of 0.2 seconds.

Requirements for trouble-free operation

Requirements for trouble-free operation with media redundancy process MRP are:

- MRP is supported in ring topologies of up to 50 devices. Exceeding the device limit can lead to data traffic failure.
- All devices must be connected to each other via their ring ports.
- "MRP" must be activated on all devices in the ring.
- The connection settings (transfer medium / duplex) must be set to full duplex and for at least 100 Mbit/s for all ring ports. Otherwise data traffic failure can occur.
- In STEP 7 configuration in the properties dialog for all ports in the ring, set the connection in the register "Options" to "Automatic settings".

HMI devices for media redundancy

The following HMI devices support "Media Redundancy":

- KP700 Comfort
- TP700 Comfort
- KP900 Comfort
- TP900 Comfort
- KP1200 Comfort
- TP1200 Comfort
- KP1500 Comfort
- TP1500 Comfort

- TP1900 Comfort
- TP2200 Comfort
- KTP700 Mobile
- KTP700F Mobile
- KTP900 Mobile
- KTP900F Mobile

Connection boxes for Mobile Panels for media redundancy

The following connection boxes for Mobile Panels support "media redundancy":

- Connection Box Standard
- Connection Box Advanced

See also

Media redundancy (Page 892)
Configuring media redundancy for HMI devices (Page 895)
Parameters for media redundancy (Page 897)
Managing MRP domains (Page 898)
Restrictions with media redundancy (Page 900)

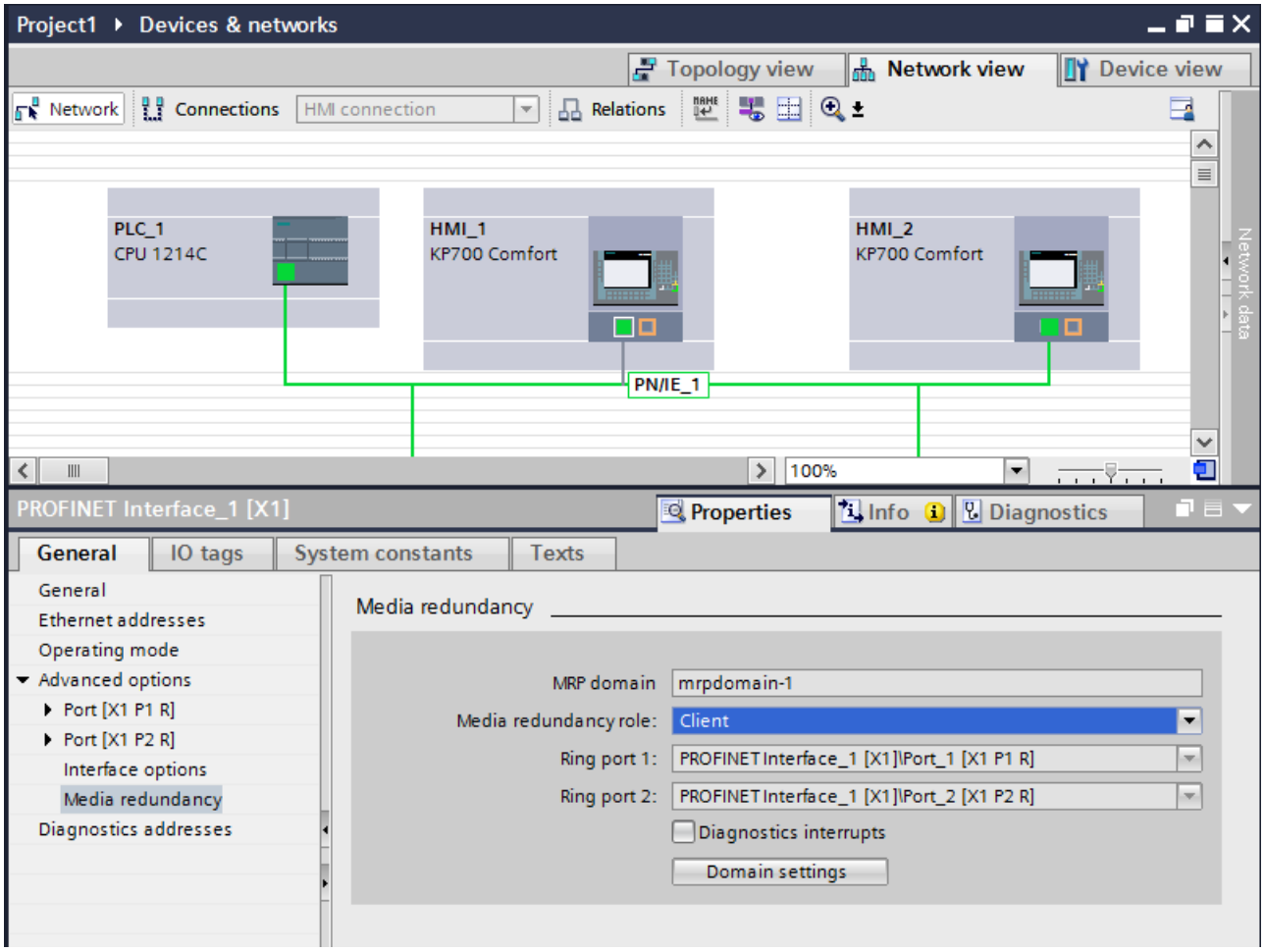
2.18.3 Configuring media redundancy for HMI devices

Requirements

- The participating components must support "media redundancy".
- IRT communication is not configured
- One device on the network must be configured as "Redundancy manager".
- A ring topology is set up by means of port interconnections.

Procedure

1. Click on the PROFINET interface of an HMI device.
2. Select "General > Advanced settings > Media redundancy" in the properties.
3. Select "Client" in the "Media redundancy role" area.



Result

The HMI device is now operating as client on the MRP network.

The default ring ports will automatically be displayed in the subjacent fields.

See also

Media redundancy (Page 892)

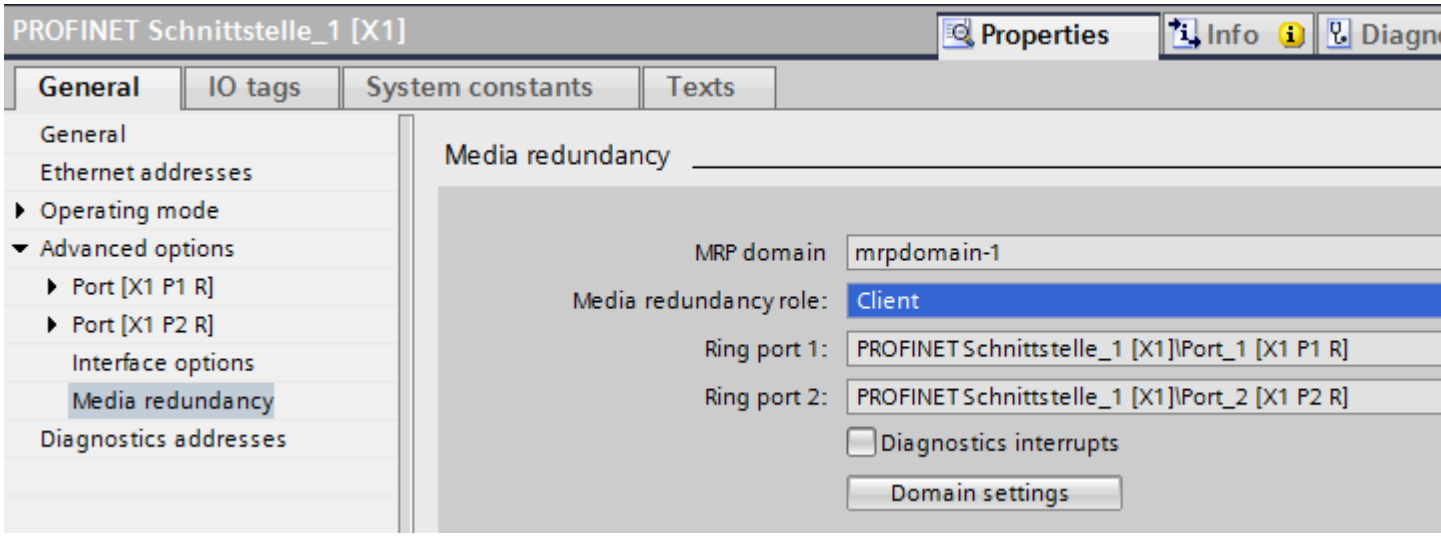
Media Redundancy Protocol (MRP) (Page 894)

Parameters for media redundancy (Page 897)

Managing MRP domains (Page 898)

Restrictions with media redundancy (Page 900)

2.18.4 Parameters for media redundancy



MRP domain

This area displays the MRP domain.

You can rename the MRP domain in the "Domain settings" dialog.

Domain settings

Click "Domain settings" to open the "Domain settings" editor.

You also manage the MRP domains in the MRP domain area of the "Domain settings" editor.

Media redundancy role

Assign the role of the HMI device in the "Media redundancy role" area.

You have the following options:

- Client
  The HIM device is operated as "Client" in an MRP domain.
- Not a node of the ring
  The HIM device is not operated within an MRP domain.

Ring port 1 and Ring port 2

Display of available ring ports

Diagnostic interrupts

The following diagnostic interrupts can be generated if "Diagnostic interrupts" is activated:

- Wiring or port error
  Diagnostics interrupts will be generated for the following errors in the ring ports:
  - A neighbor of the ring port does not support MRP
  - A ring port is connected to a non-ring port

See also

Media redundancy (Page 892)
Media Redundancy Protocol (MRP) (Page 894)
Configuring media redundancy for HMI devices (Page 895)
Managing MRP domains (Page 898)
Restrictions with media redundancy (Page 900)

2.18.5 Managing MRP domains

You can monitor and adjust the media redundancy settings centrally.

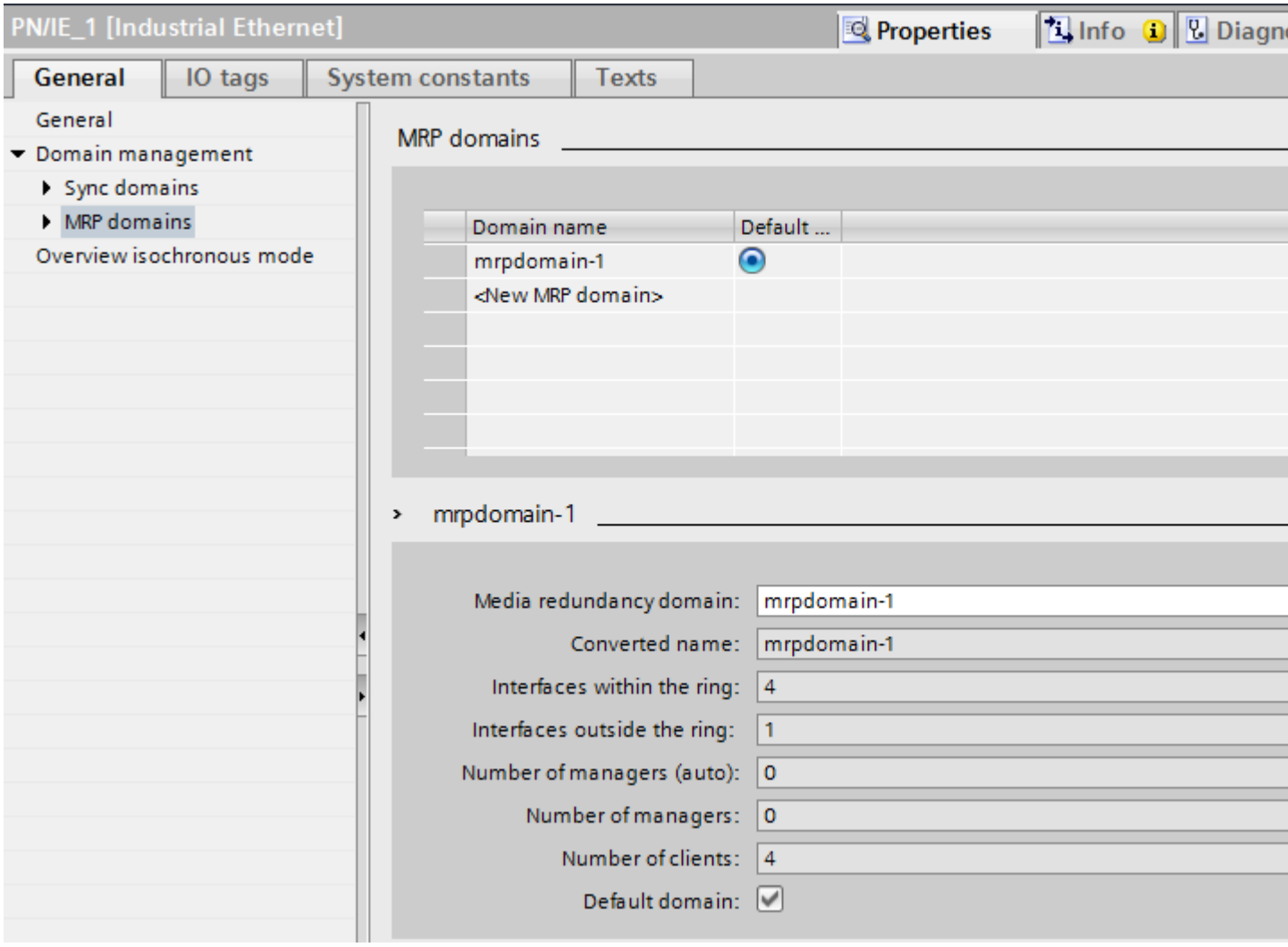
Requirements

The devices are configured and networked.

Procedure

1. Select the PROFINET interface of the HMI device.
2. Select the "Advanced options > Media redundancy" area in the Inspector window.

3. Click the "Domain settings" button.
4. Navigate to the "Domain management > MRP domains" area in the Inspector window.



MRP domains

You can specify the default MRP domain in the "MRP domains" area.

You can rename all domains and check the redundancy roles and ring ports.

For each MRP domain, you receive an overview of how many PROFINET interfaces are interconnected in the ring, how many PROFINET interfaces are located outside the ring, and the number of redundancy managers and redundancy clients.

You can have the devices in the ring, including device name, associated MRP domain, media redundancy role, and utilized ring ports, displayed in groups:

- IO systems
  The list shows you all configured devices of the selected IO system.
- Rings
  The list shows you all ring interconnections of the selected IO system. The identified rings are numbered consecutively. If all devices of this ring belong to one MRP domain, this is also displayed.
- MRP domain
  The list shows you the existing MRP domains of the selected IO system.

See also

Media redundancy (Page 892)

Media Redundancy Protocol (MRP) (Page 894)

Configuring media redundancy for HMI devices (Page 895)

Parameters for media redundancy (Page 897)

Restrictions with media redundancy (Page 900)

2.18.6 Restrictions with media redundancy

Diagnostic interrupts

If you have activated "Diagnostic interrupts", diagnostic interrupts pertaining to "Media redundancy" are only processed by PROFINET-compatible devices and displayed on the "IO Controller".

If you do not use a PROFINET-compatible device, the "Diagnostic interrupts" selection does not have any function.

Diagnostic interrupts are only detected by the IO controller. There is no separate diagnostic interrupts for HMI devices that are detected and displayed by HMI devices.

See also

Media redundancy (Page 892)

Media Redundancy Protocol (MRP) (Page 894)

Configuring media redundancy for HMI devices (Page 895)

Parameters for media redundancy (Page 897)

Managing MRP domains (Page 898)

2.19 Communication with other controllers

2.19.1 Communication with other PLCs

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
  - Omron Ethernet IP

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

Allen-Bradley communication drivers (Page 905)

2.19.2 Distinctive features when configuring

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

Parallel communication (Page 903)

2.19.3 Parallel communication

Parallel communication of communication drivers

The following table shows an overview of which communication drivers you can use simultaneously on one HMI device.

Note

Note the parallel communication has not been released 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.

The following table shows the special features of the parallel communication for WinCC Runtime Advanced. For WinCC Runtime Professional there are no restrictions regarding parallel communication between different communication drivers.

	Allen-Bradley EtherNet/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 EtherNet/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

	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 (DA/XML DA)	Yes	Yes	Yes	--	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
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	Yes	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.

See also

Distinctive features when configuring (Page 902)

2.19.4 Communication drivers

2.19.4.1 Allen-Bradley

Allen-Bradley communication drivers

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.

Allen-Bradley EtherNet/IP

Configuring a connection via Allen-Bradley EtherNet/IP

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.

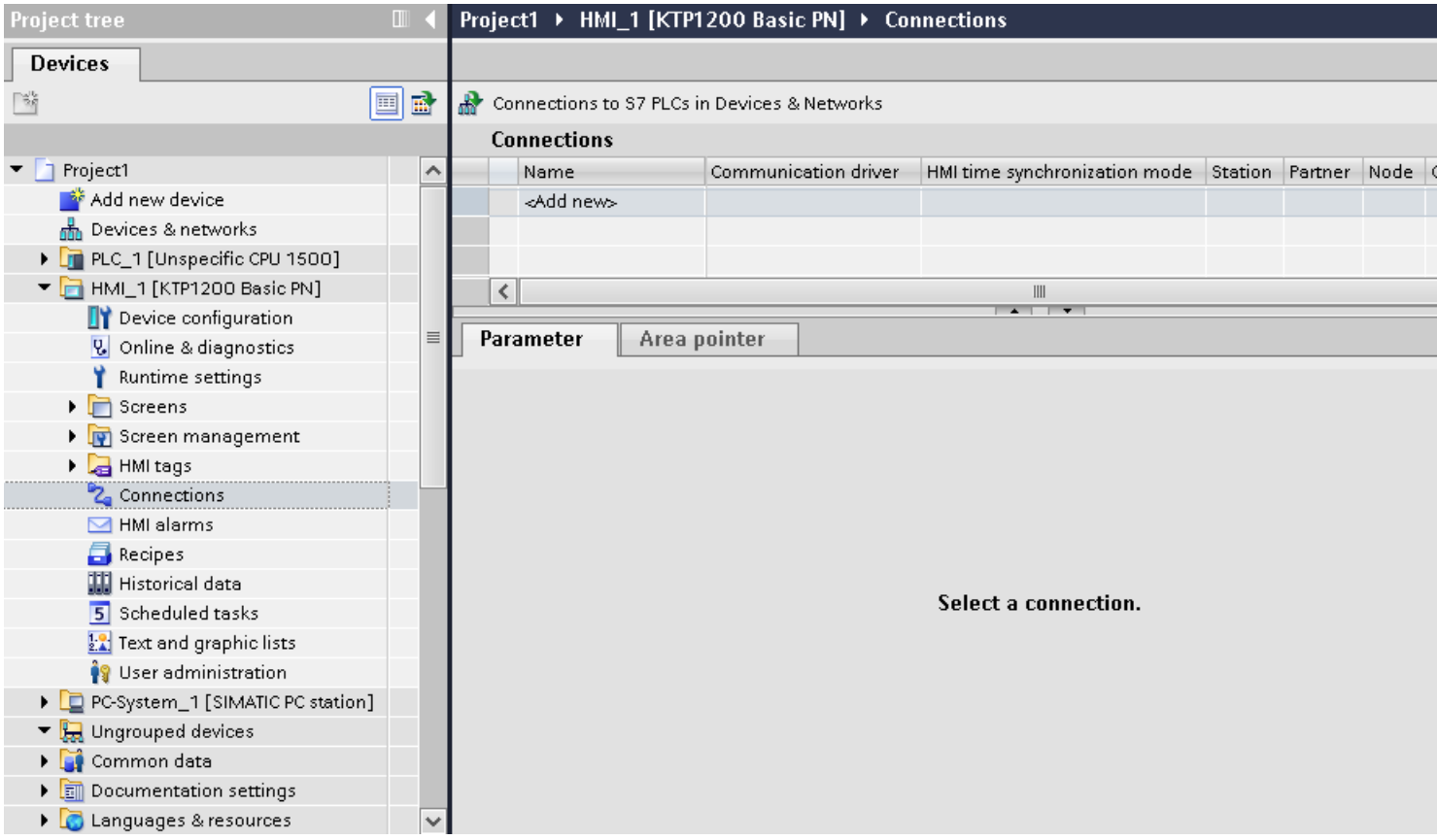
Example: PROFINET interface corresponds to the Ethernet interface

Requirements

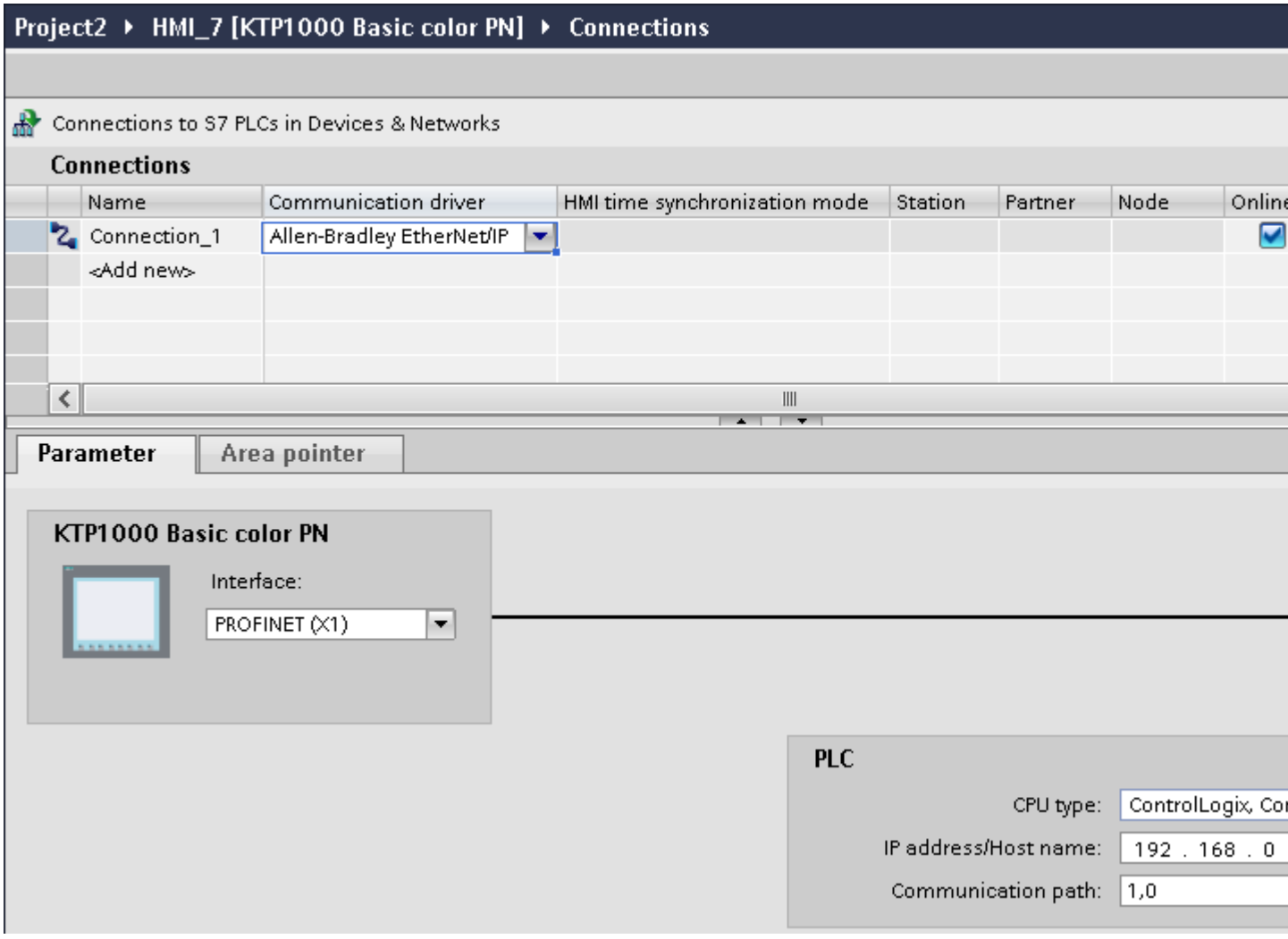
- A project is open.
- An HMI device has been created.

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 905)

Parameters for the connection (Allen-Bradley EtherNet/IP) (Page 907)

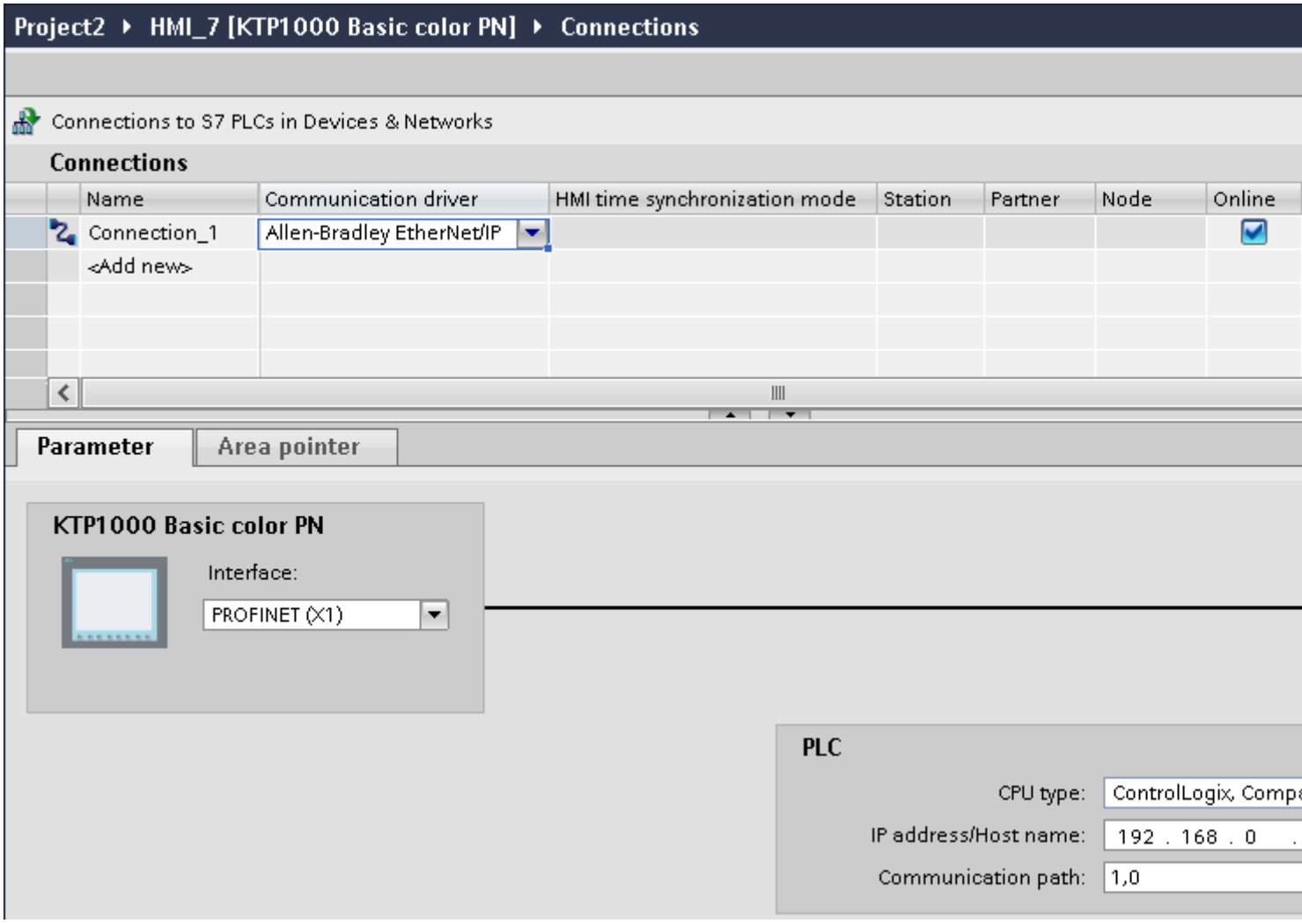
Optimizing the configuration (Page 927)

Parameters for the connection (Allen-Bradley EtherNet/IP)

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.

See also

Allen-Bradley communication drivers (Page 905)

Configuring a connection via Allen-Bradley EtherNet/IP (Page 905)

Connecting HMI device to PLC

Connections via Allen-Bradley EtherNet/IP

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

Allen-Bradley communication drivers (Page 905)

Communication types (Page 910)

Communication types

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 905)

Connections via Allen-Bradley EtherNet/IP (Page 910)

Performance features of communication

Permitted data types for Allen-Bradley EtherNet/IP

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

Data type	Length
UDInt	4 bytes
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

Allen-Bradley communication drivers (Page 905)

Supported CPU types for Allen-Bradley EtherNet/IP (Page 913)

Supported CPU types for Allen-Bradley EtherNet/IP

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

Allen-Bradley communication drivers (Page 905)

Permitted data types for Allen-Bradley EtherNet/IP (Page 911)

Addressing in the CompactLogix, ControlLogix CPU type

Addressing

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

Allen-Bradley communication drivers (Page 905)

Addressing syntax (Page 915)

Addressing types (Page 916)

Address multiplexing (Page 918)

Examples for addressing (Page 919)

Examples: Communication path (Page 920)

Addressing syntax

Notation of addresses

The tables below define the notation for the individual addressing options for Allen-Bradley EtherNet/IP.

Table 2-7 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

Data types	Type	Address
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

Allen-Bradley communication drivers (Page 905)

Addressing (Page 914)

Addressing types (Page 916)

Address multiplexing (Page 918)

Examples for addressing (Page 919)

Examples: Communication path (Page 920)

Addressing types

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

Allen-Bradley communication drivers (Page 905)

Addressing (Page 914)

Addressing syntax (Page 915)

Address multiplexing (Page 918)

Examples for addressing (Page 919)

Examples: Communication path (Page 920)

Address multiplexing

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

Allen-Bradley communication drivers (Page 905)

Addressing (Page 914)

Addressing syntax (Page 915)

Addressing types (Page 916)

Examples for addressing (Page 919)

Examples: Communication path (Page 920)

Examples for addressing

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

Allen-Bradley communication drivers (Page 905)
Addressing (Page 914)
Addressing syntax (Page 915)
Addressing types (Page 916)
Address multiplexing (Page 918)
Examples: Communication path (Page 920)

Examples: Communication path

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

Allen-Bradley communication drivers (Page 905)

Addressing (Page 914)

Addressing syntax (Page 915)

Addressing types (Page 916)

Address multiplexing (Page 918)

Examples for addressing (Page 919)

Addressing in the SLC, MicroLogix CPU type

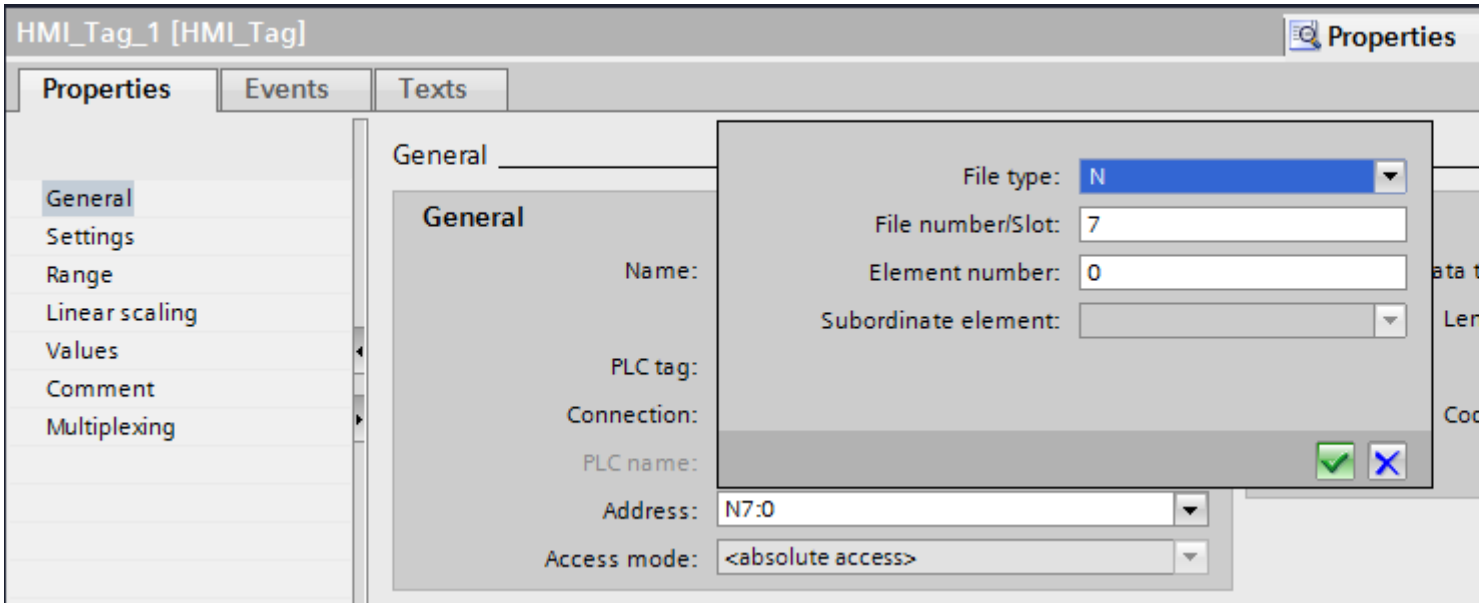
Addressing

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 905)

Address areas for PLC, SLC, MicroLogix (Page 923)

Address areas for PLC, SLC, MicroLogix

SLCMicro

Address areas	Bool	SInt	USInt	Int	UInt	DInt	UDInt	Real	String
N	N3:0/0 - N999:1999/15	--	--	N3:0 - N999:1999	N3:0 - N999:1999	N3:0 - N999:1999	N3:0 - N999:1999	N3:0 - N999:1999	N3:0 - N999:1999
R	R3:0.EN - R999:1999.ER - R999:1999.DN - R999:1999.FD - R999:1999.IN - R999:1999.EU - R999:1999.EM - R999:1999.UL	--	--	R3:0.LEN - R999:1999.POS	R3:0.LEN - R999:1999.POS	--	--	--	R3:0.LEN - R999:1999.POS

Address areas	Bool	SInt	USInt	Int	UInt	DInt	UDInt	Real	String
C	C3:0.DN - C999:1999.CU - C999:1999.CD - C999:1999.OV - C999:1999.UN	--	--	C3:0.PRE - C999:1999.ACC	C3:0.PRE - C999:1999.ACC	--	--	--	C3:0.PRE - C999:1999.ACC
T	T3:0.EN - T999:1999.TT - T999:1999.DN	--	--	T3:0.PRE - T999:1999.ACC	T3:0.PRE - T999:1999.ACC	--	--	--	T3:0.PRE - T999:1999.ACC
B	B3:0/0 - B999:1999/15	--	--	B3:0 - B999:1999	B3:0 - B999:1999	--	--	--	B3:0 - B999:1999
S	S2:0/0 - S2:127/65535	--	--	S2:0 - S2:127	S2:0 - S2:127	--	--	--	S2:0 - S2:127
I	I1:0/0 - I999:255/15	--	--	I1:0 - I999:255	I1:0 - I999:255	--	--	--	I1:0 - I999:255
O	O0:0/0 - O999:255/15	--	--	O0:0 - O999:255	O0:0 - O999:255	--	--	--	O0:0 - O999:255
F	--	--	--	--	--	F3:0 - F999:1999	F3:0 - F999:1999	F3:0 - F999:1999	F3:0 - F999:1999
D	D3:0/0 - D999:1999/15	--	--	D3:0 - D999:1999	D3:0 - D999:1999	--	--	--	D3:0 - D999:1999
A	A3:0/0 - A999:1999/15	A3:0 - A999:1999	A3:0 - A999:1999	A3:0 - A999:1999	A3:0 - A999:1999	A3:0 - A999:1999	A3:0 - A999:1999	A3:0 - A999:1999	A3:0 - A999:1999
ST	ST3:0/0 - ST999:1999/15	ST3:0 - ST999:1999	ST3:0 - ST999:1999	ST3:0 - ST999:1999	ST3:0 - ST999:1999	ST3:0 - ST999:1999	ST3:0 - ST999:1999	ST3:0 - ST999:1999	ST3:0 - ST999:1999

PLC5

Address areas	Bool	SInt	USInt	Int	UInt	DInt	UDInt	Real	String
N	N3:0/0 - N999:1999/15	--	--	N3:0/0 - N999:1999/15	N3:0/0 - N999:1999/15	N3:0/0 - N999:1999/15	N3:0/0 - N999:1999/15	N3:0/0 - N999:1999/15	N3:0/0 - N999:1999/15
R	R3:0.EN - R999:1999.ER - R999:1999.DN - R999:1999.FD - R999:1999.IN - R999:1999.EU - R999:1999.EM - R999:1999.UL	--	--	R3:0.LEN - R999:1999.POS	R3:0.LEN - R999:1999.POS	--	--	--	R3:0.LEN - R999:1999.POS
C	C3:0.DN - C999:1999.CU - C999:1999.CD - C999:1999.OV - C999:1999.UN	--	--	C3:0.PRE - C999:1999.ACC	C3:0.PRE - C999:1999.ACC	--	--	--	C3:0.PRE - C999:1999.ACC
T	T3:0.EN - T999:1999.TT - T999:1999.DN	--	--	T3:0.PRE - T999:1999.ACC	T3:0.PRE - T999:1999.ACC	--	--	--	T3:0.PRE - T999:1999.ACC
B	B3:0/0 - B999:1999/15	--	--	B3:0 - B999:1999	B3:0 - B999:1999	--	--	--	B3:0 - B999:1999
S	S2:0/0 - S2:127/65535	--	--	S2:0 - S2:127	S2:0 - S2:127	--	--	--	S2:0 - S2:127
I	I1:0/0 - I999:277/17	--	--	I1:0 - I999:277	I1:0 - I999:277	--	--	--	I1:0 - I999:277
O	O0:0/0 - O999:277/17	--	--	O0:0 - O999:277	O0:0 - O999:277	--	--	--	O0:0 - O999:277
F	--	--	--	--	--	F3:0 - F999:1999	F3:0 - F999:1999	F3:0 - F999:1999	F3:0 - F999:1999

Address areas	Bool	SInt	USInt	Int	UInt	DInt	UDInt	Real	String
D	D3:0/0 - D999:1999/15	--	--	D3:0/0 - D999:1999/15	D3:0/0 - D999:1999/15	--	--	--	D3:0/0 - D999:1999/15
A	A3:0/0 - A999:1999/15	A3:0/0 - A999:1999/15	A3:0/0 - A999:1999/15	A3:0/0 - A999:1999/15	A3:0/0 - A999:1999/15	A3:0/0 - A999:1999/15	A3:0/0 - A999:1999/15	A3:0/0 - A999:1999/15	A3:0/0 - A999:1999/15
ST	ST3:0/0 - ST999:1999/15	ST3:0/0 - ST999:1999/15	ST3:0/0 - ST999:1999/15	ST3:0/0 - ST999:1999/15	ST3:0/0 - ST999:1999/15	ST3:0/0 - ST999:1999/15	ST3:0/0 - ST999:1999/15	ST3:0/0 - ST999:1999/15	ST3:0/0 - ST999:1999/15

See also

Allen-Bradley communication drivers (Page 905)

Addressing (Page 921)

Commissioning components

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

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.

See also

Configuring a connection via Allen-Bradley EtherNet/IP (Page 905)

Allen-Bradley DF1

Configuring a connection via Allen-Bradley DF1

Introduction

You configure a connection to a PLC with an Allen-Bradley DF1 communication driver in the "Connections" editor of the HMI device.

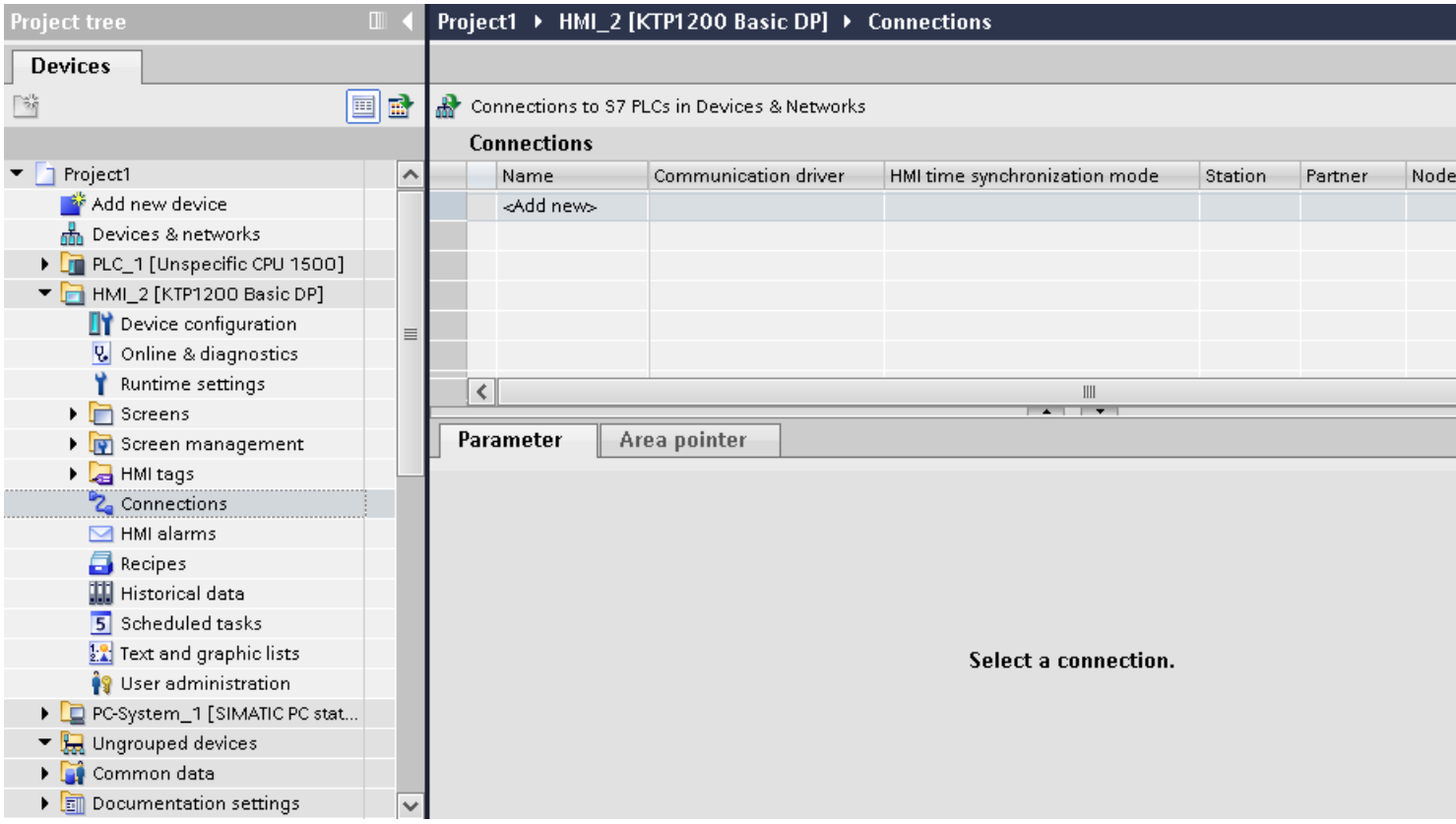
The interfaces are named differently depending on the HMI device.

Requirements

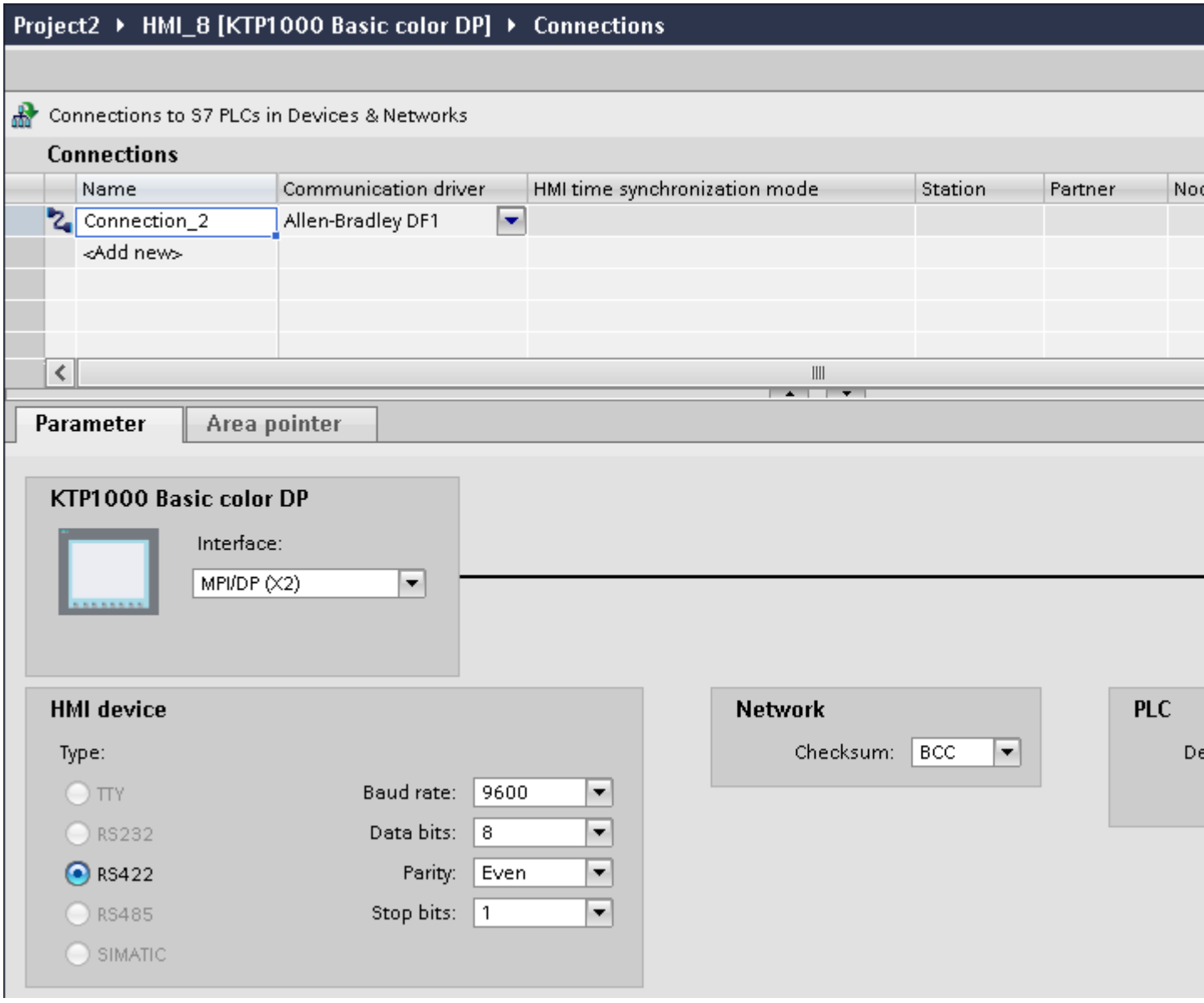
- A project is open.
- An HMI device has been created.

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 905)

Parameters for the connection (Allen-Bradley DF1) (Page 930)

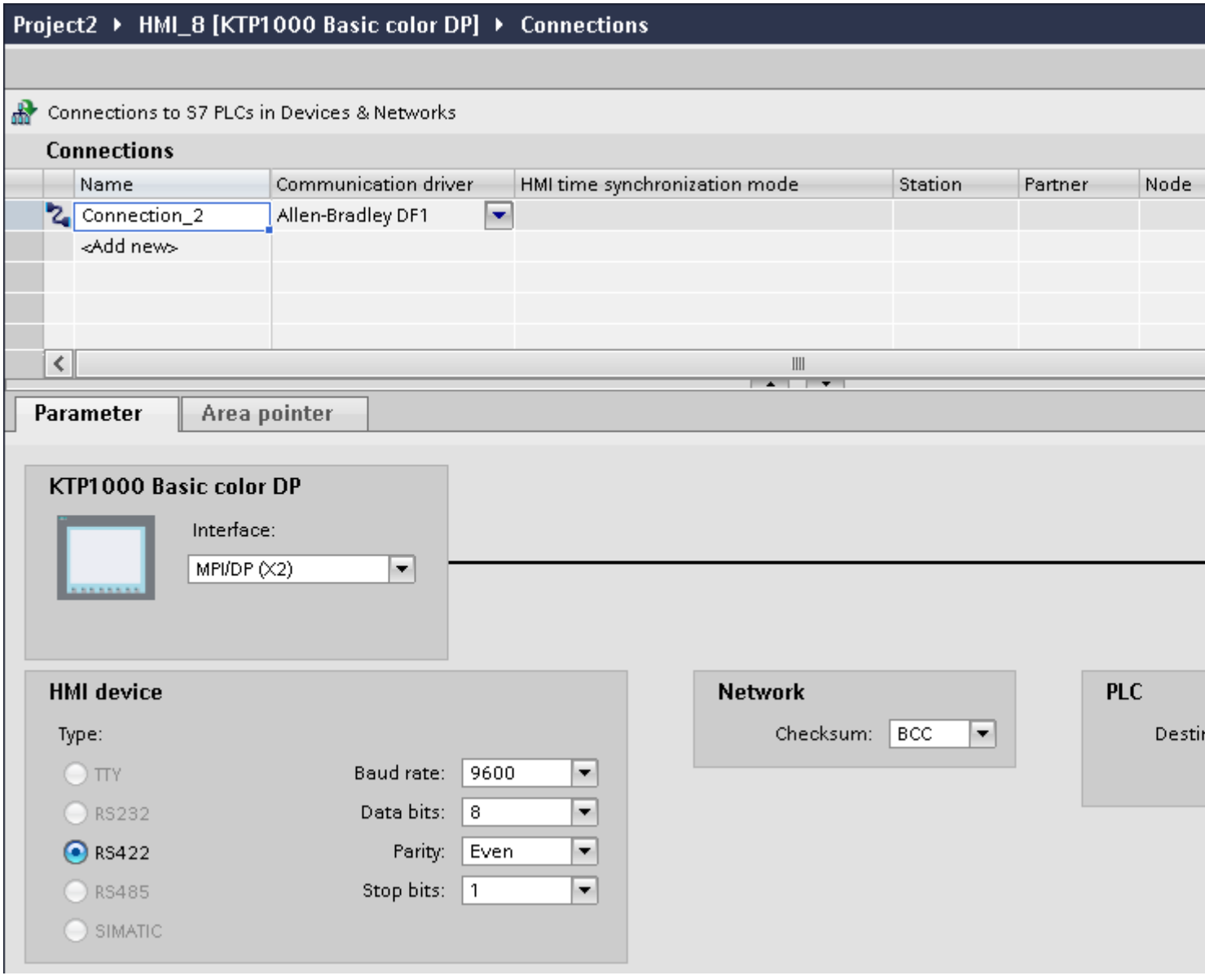
Optimizing the configuration (Page 948)

Parameters for the connection (Allen-Bradley DF1)

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 905)

Configuring a connection via Allen-Bradley DF1 (Page 928)

Connecting HMI device to PLC

Connections via Allen-Bradley DF1

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

Allen-Bradley communication drivers (Page 905)

Communication types (Page 932)

DF1 protocol with multi-point connection (Page 934)

DF1 protocol with multi-point connection via KF2 module (Page 935)

DF1 protocol with multi-point connection via KF3 module (Page 936)

Communication types

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 and 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

Allen-Bradley communication drivers (Page 905)

Connections via Allen-Bradley DF1 (Page 932)

DF1 protocol with multi-point connection (Page 934)

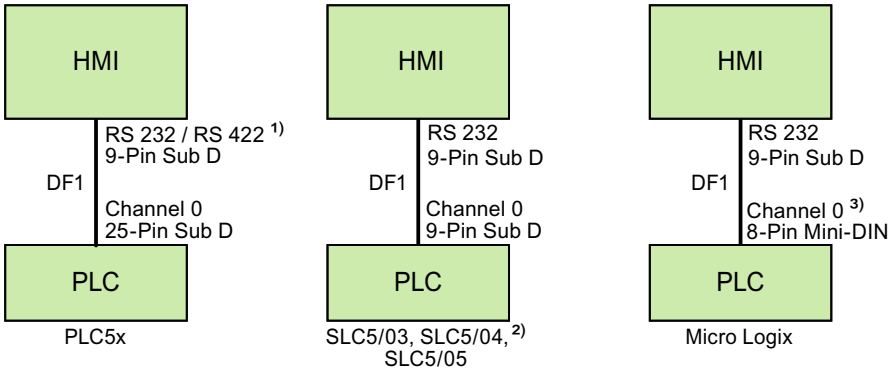
DF1 protocol with multi-point connection via KF2 module (Page 935)

DF1 protocol with multi-point connection via KF3 module (Page 936)

DF1 protocol with multi-point connection

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

Allen-Bradley communication drivers (Page 905)

Connections via Allen-Bradley DF1 (Page 932)

Communication types (Page 932)

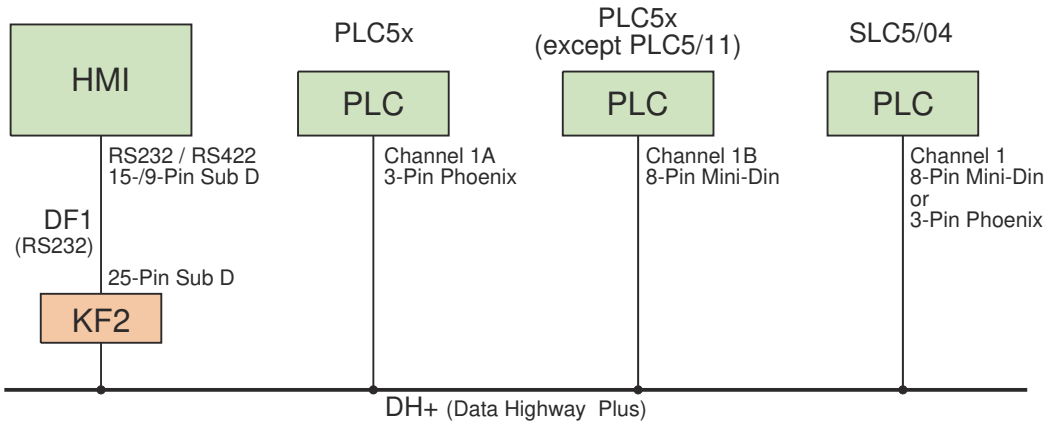
DF1 protocol with multi-point connection via KF2 module (Page 935)

DF1 protocol with multi-point connection via KF3 module (Page 936)

DF1 protocol with multi-point connection via KF2 module

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

Allen-Bradley communication drivers (Page 905)

Connections via Allen-Bradley DF1 (Page 932)

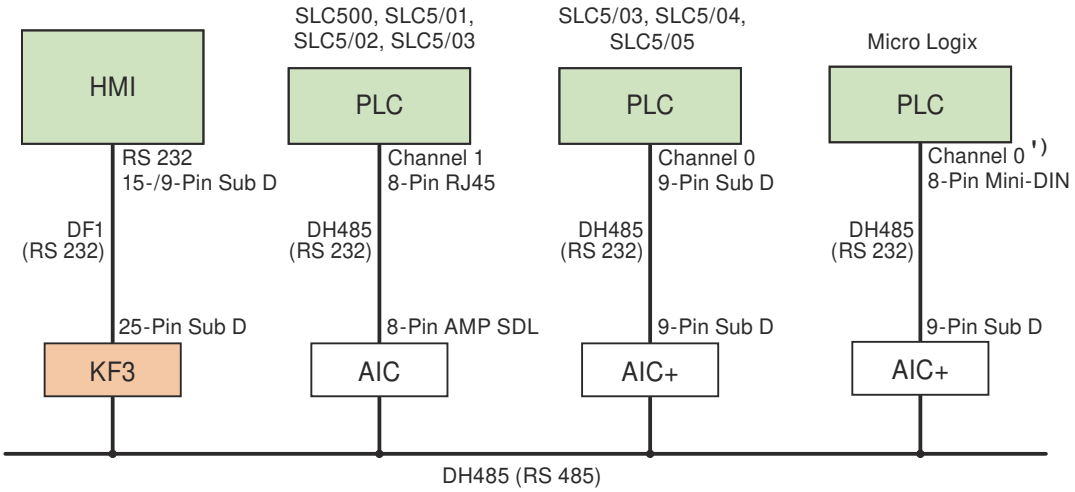
Communication types (Page 932)

DF1 protocol with multi-point connection (Page 934)

DF1 protocol with multi-point connection via KF3 module (Page 936)

DF1 protocol with multi-point connection via KF3 module

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

Allen-Bradley communication drivers (Page 905)

Connections via Allen-Bradley DF1 (Page 932)

Communication types (Page 932)

DF1 protocol with multi-point connection (Page 934)

DF1 protocol with multi-point connection via KF2 module (Page 935)

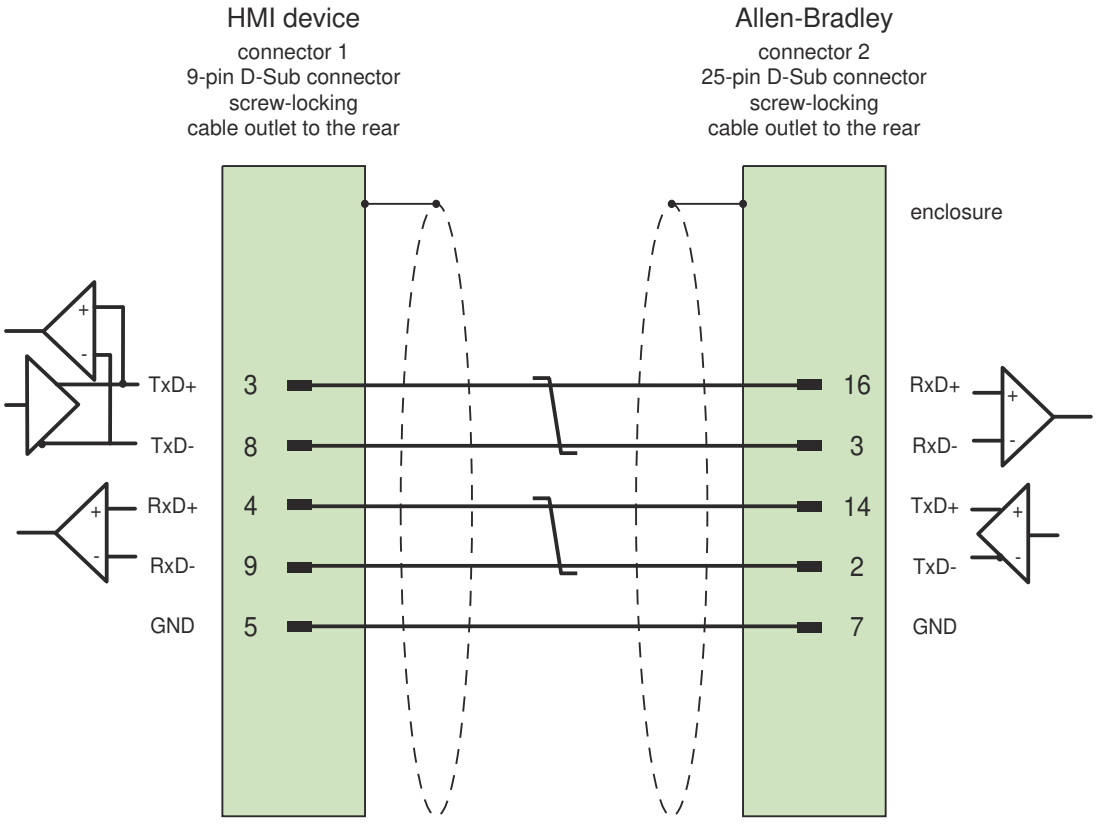
Connecting cables for Allen-Bradley DF1

Connecting cable 9-pin Sub D RS 422 for Allen-Bradley

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

Allen-Bradley communication drivers (Page 905)

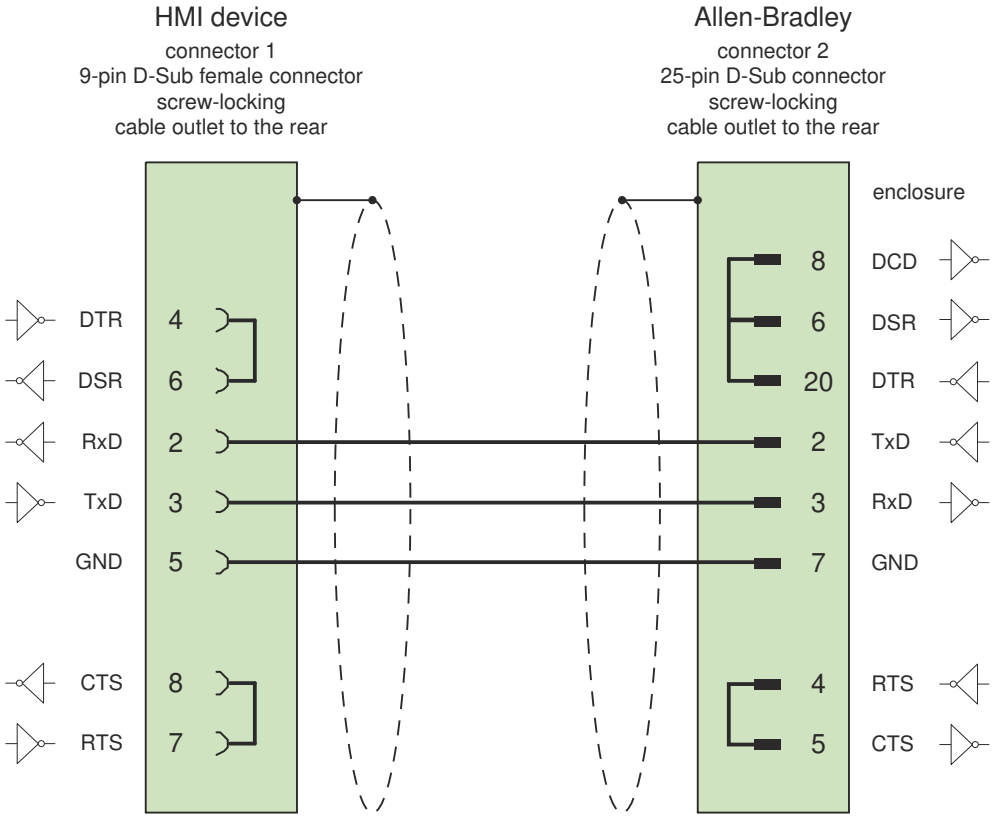
Connections via Allen-Bradley DF1 (Page 932)

Connecting cable 1784-CP10, RS 232, for Allen-Bradley

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

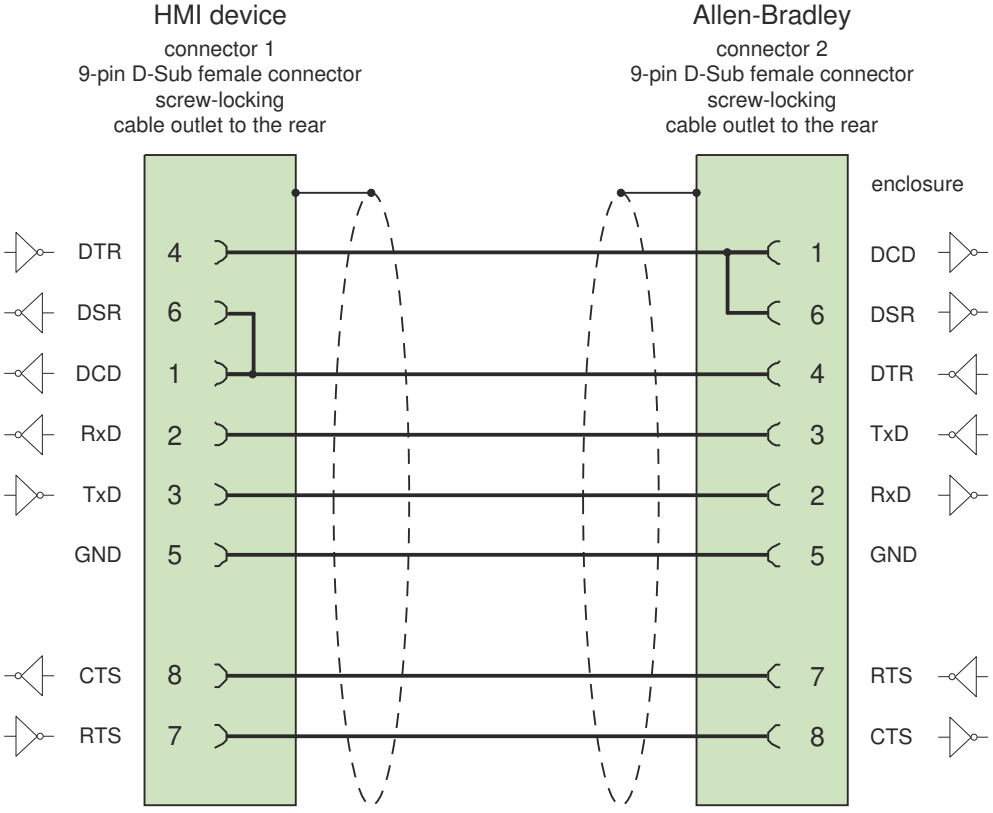
Allen-Bradley communication drivers (Page 905)

Connections via Allen-Bradley DF1 (Page 932)

Connecting cable 1747-CP3, RS-232, for Allen-Bradley

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

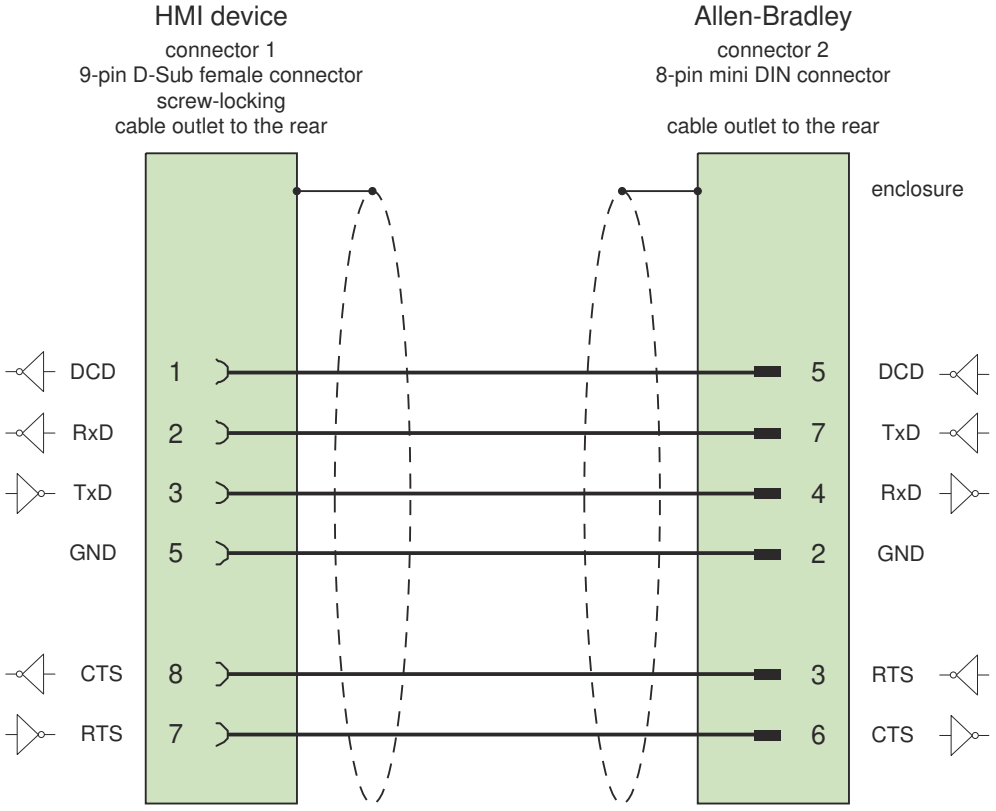
Allen-Bradley communication drivers (Page 905)

Connections via Allen-Bradley DF1 (Page 932)

Connecting cable 1761-CBL-PM02, RS-232, for Allen-Bradley

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

Allen-Bradley communication drivers (Page 905)

Connections via Allen-Bradley DF1 (Page 932)

Performance features of communication

Permitted data types for Allen-Bradley DF1

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
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 905)

Supported CPU types for Allen-Bradley DF1 (Page 942)

Addressing (Page 943)

Address areas for Allen-Bradley DF1 (Page 944)

Supported CPU types for Allen-Bradley DF1

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 905)

Permitted data types for Allen-Bradley DF1 (Page 941)

Addressing (Page 943)

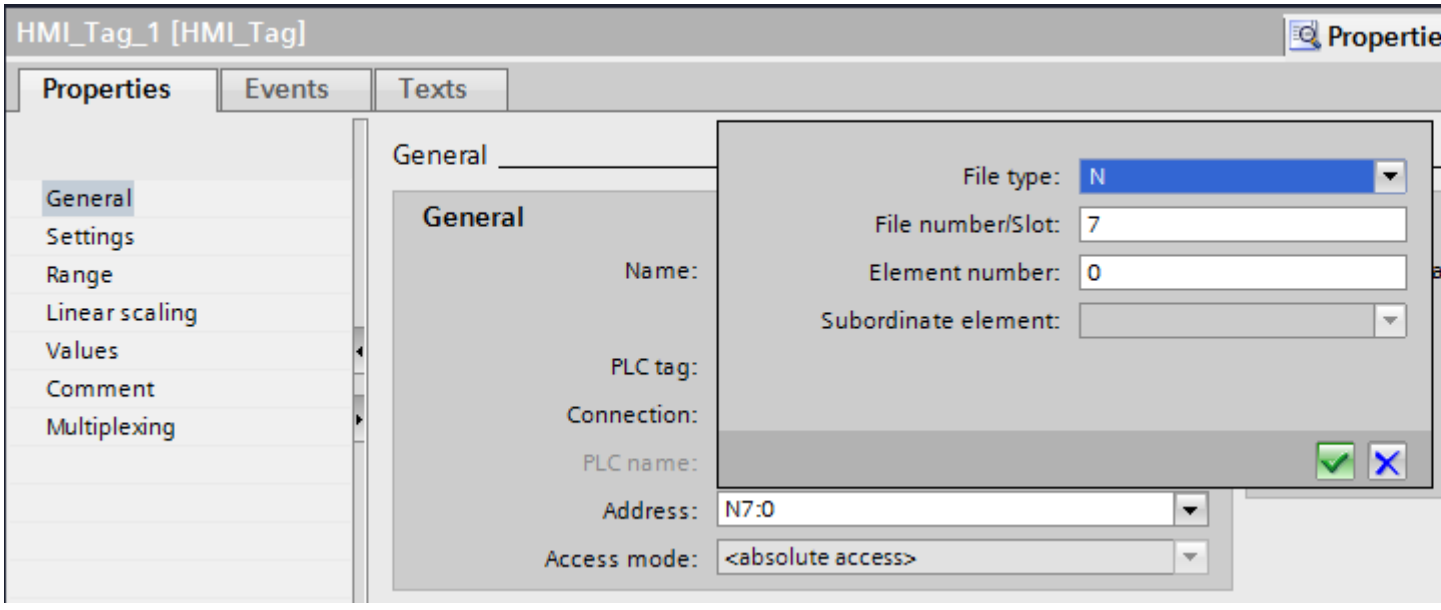
Address areas for Allen-Bradley DF1 (Page 944)

Addressing

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 905)

Permitted data types for Allen-Bradley DF1 (Page 941)

Supported CPU types for Allen-Bradley DF1 (Page 942)

Address areas for Allen-Bradley DF1 (Page 944)

Address areas for Allen-Bradley DF1

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	--

Address areas	Data types					
	Bool	**Int**	**UInt**	**DInt**	**UDInt**	**Real**
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 - S2:65/15	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	--	--	--

Address areas	Data types					
	Bool	**Int**	**UInt**	**DInt**	**UDInt**	**Real**
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
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 905)

Permitted data types for Allen-Bradley DF1 (Page 941)

Supported CPU types for Allen-Bradley DF1 (Page 942)
Addressing (Page 943)

Commissioning components

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

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.

See also

Configuring a connection via Allen-Bradley DF1 (Page 928)

Data exchange

Area pointers for Allen-Bradley

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".

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

Area pointer	Allen-Bradley EtherNet/IP	Allen-Bradley DF1
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

Allen-Bradley communication drivers (Page 905)

Trends

General information on trends

Trends

A trend is the graphical representation of one or more values from the PLC. The value is read out either time- or bit-triggered, depending on the configuration.

For additional information see:

AUTOHOTSPOT

Note

The value is read out time-triggered for Basic Panels.

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.

Bit-triggered trends

Through a trigger bit set in the trend request tag, the HMI device either reads in a trend value or an entire trend buffer. This setting is defined in the configuration. Bit-triggered trends are normally used to represent fast 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 "HMI tags" editor and connected to trend areas during configuration. The HMI device and PLC then communicate with each other via 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)

See also

Area pointers for Allen-Bradley (Page 949)

Trend request and trend transfer (Page 951)

Restrictions to the trend control (Page 952)

Trend request and trend transfer

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 closing the screen, the HMI device resets the relevant bits in the trend request area.

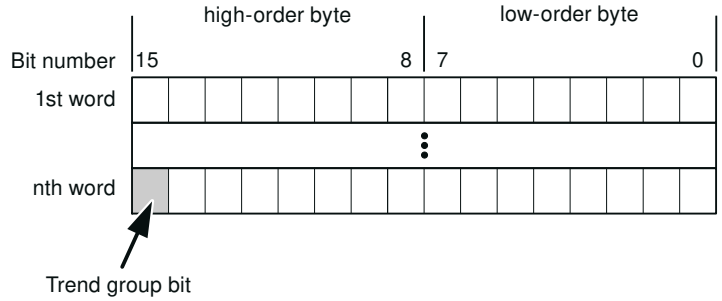
The trend request area can be used for evaluation purposes in the PLC to determine which trend is currently being displayed on the HMI device. Trends can also be triggered without evaluation of the trend request area.

Trend transfer area 1

This area is used to trigger trends. You must set the bit assigned to the trend in the trend transfer area and set the trend group bit in your control program. 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.

The PLC writes to Buffer 2 while the HMI device reads values from Buffer 1, and writes to Buffer 1 when the HMI device is reading Buffer 2. This prevents the PLC from overwriting trend values while the trend is being read by the HMI device.

See also

Area pointers for Allen-Bradley (Page 949)

General information on trends (Page 950)

Restrictions to the trend control (Page 952)

Restrictions to the trend control

For Allen-Bradley DF1 communication drivers

Tags of the following operand types are permitted:

- "N"
- "O"
- "I"
- "S"
- "B"

Permitted data types:

- "UInt"
- "Int"

Note

The "Array" data type is only available for the trend type "Bit-triggered buffer". The data type "Array" cannot be selected for the "Bit-triggered real time", "Cyclic real time" or "Data log" trend types.

Individual array elements cannot be selected as data type.

You assign a bit to a trend during configuration. This sets a defined bit assignment for all trend areas.

For Allen-Bradley EtherNet/IP communication drivers

Tags of data type "Int" or an array tag of data type "Int" are permitted. You assign a bit to a trend during configuration. This sets a defined bit assignment for all trend areas.

ControlLogix and CompactLogix

The following data types are possible for tags for the ControlLogix and CompactLogix CPU types:

- "UInt"
- "Int"

SLC and MicroLogix

The following operand types are permitted for tags of the SLC and MicroLogix CPU types:

- "N"
- "O"
- "I"
- "S"
- "B"

Permitted data types:

- "UInt"
- "Int"

Note

The "Array" data type is only available for the trend type "Bit-triggered buffer". The data type "Array" cannot be selected for the "Bit-triggered real time", "Cyclic real time" or "Data log" trend types.

Individual array elements cannot be selected as data type.

See also

Area pointers for Allen-Bradley (Page 949)

General information on trends (Page 950)

Trend request and trend transfer (Page 951)

Alarms

Configuring alarms

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

See also

Area pointers for Allen-Bradley (Page 949)

Acknowledgment of alarms (Page 955)

Acknowledgment of alarms

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

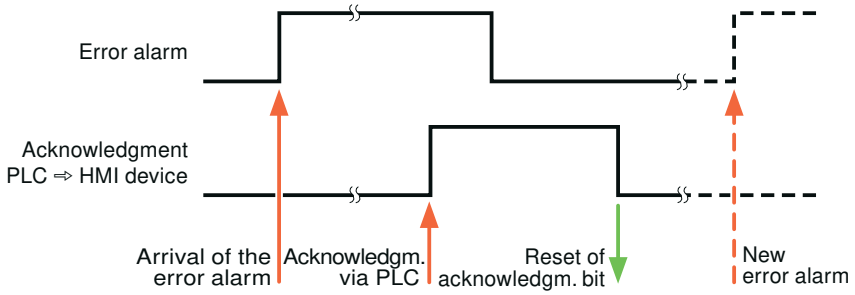
Acknowledgment by the PLC

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.

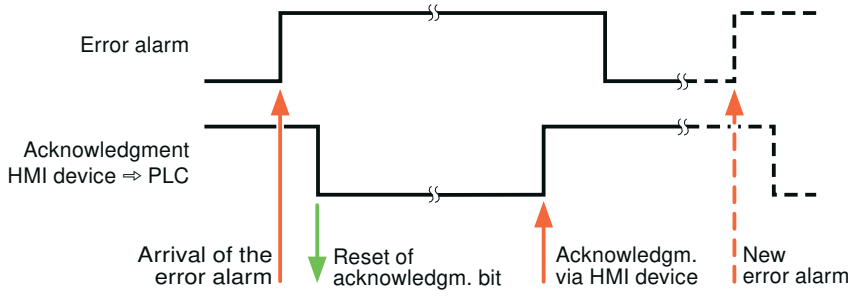
If a discrete alarm subject to acknowledgment is acknowledged by the HMI device, then the corresponding bit in the assigned acknowledgment tag will be set. The entire acknowledgment tag is then written to the PLC by the HMI device. This allows the PLC to recognize that a certain alarm message has been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

Area pointers for Allen-Bradley (Page 949)

Configuring alarms (Page 954)

LED mapping

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 Comfort Panels	Panel PCs
0	0	Off	Off
0	1	Rapid flashing	Flashing
1	0	Slow flashing	Flashing
1	1	On permanently	On permanently

2.19.4.2 Mitsubishi

Mitsubishi communication drivers

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

Configuring a connection via Mitsubishi MC TCP/IP

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

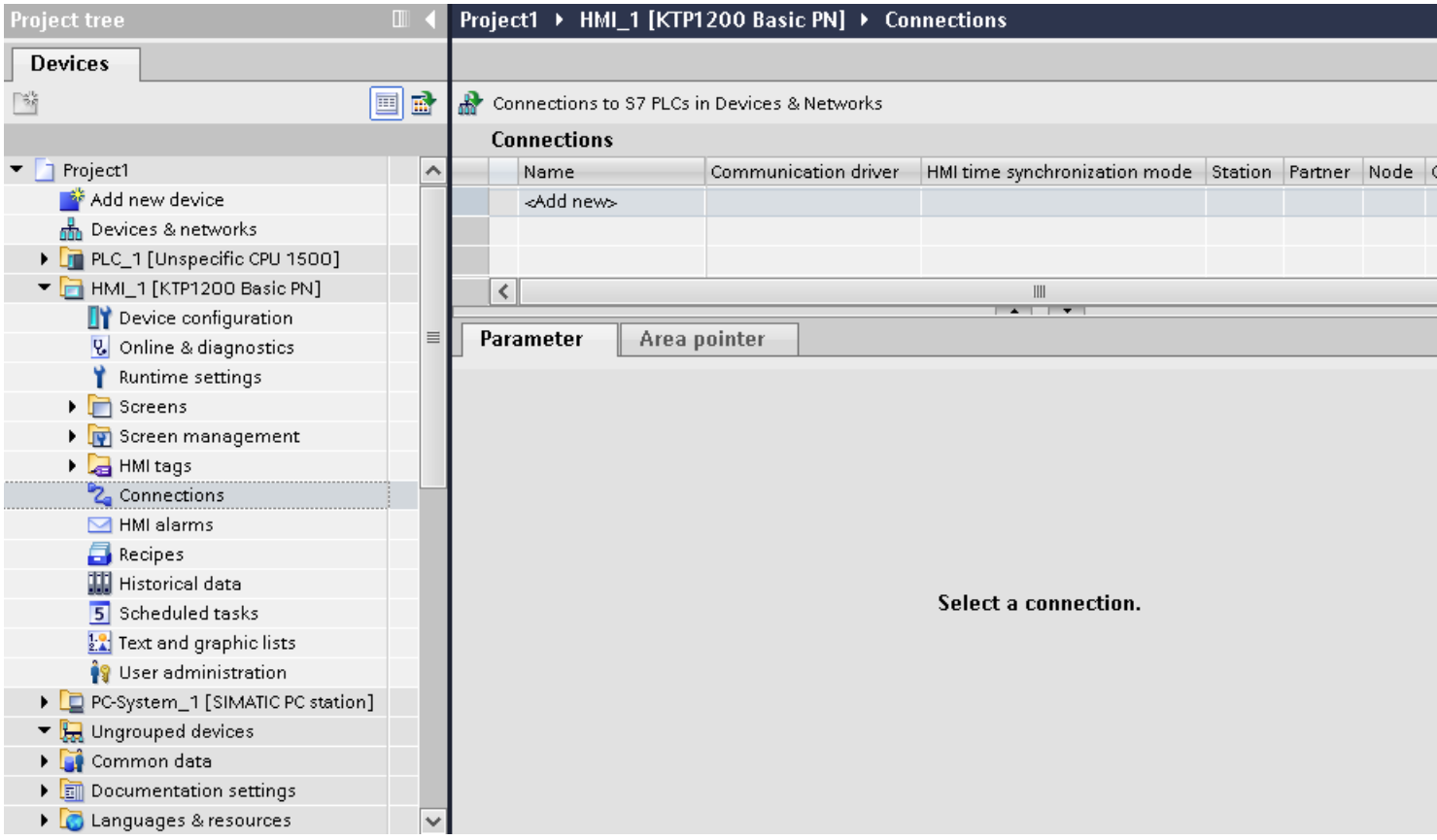
Requirements

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

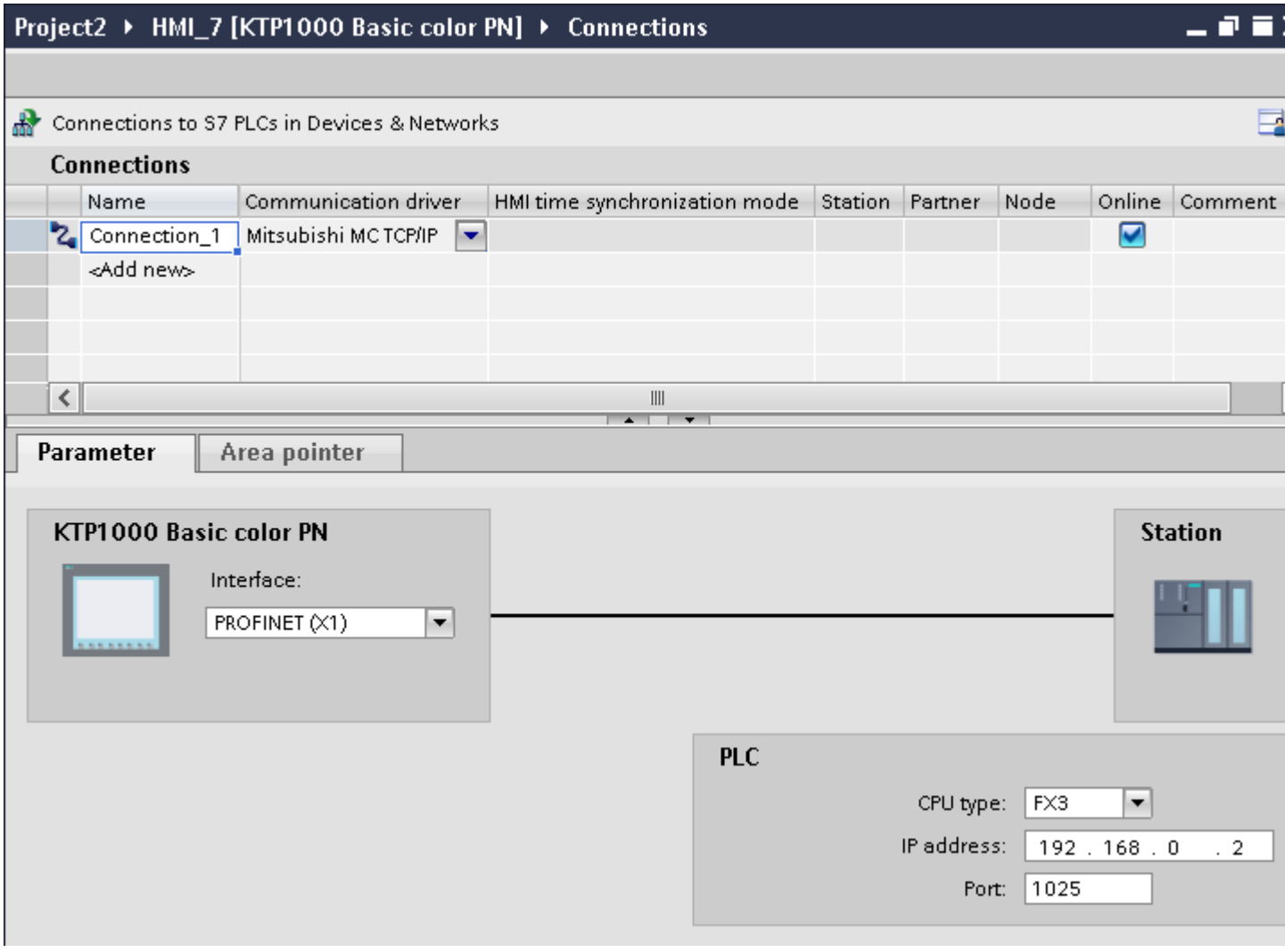
Procedure

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 958)

Parameters for the connection (Mitsubishi MC TCP/IP) (Page 962)

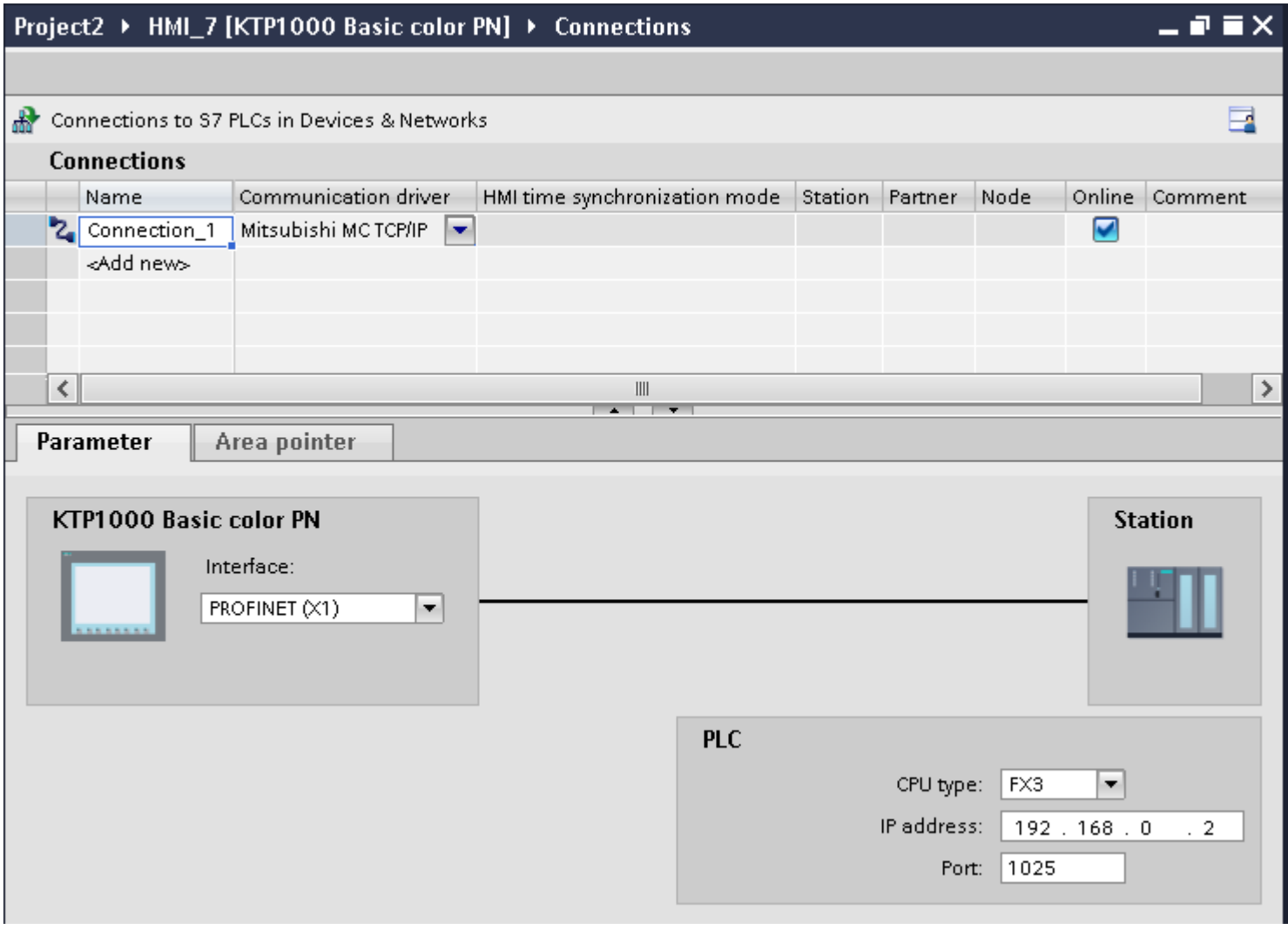
Optimizing the configuration (Page 978)

Parameters for the connection (Mitsubishi MC TCP/IP)

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

Configuring a connection via Mitsubishi MC TCP/IP (Page 959)

Connecting HMI device to PLC

Connections via Mitsubishi MC TCP/IP

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

Mitsubishi communication drivers (Page 958)

Configuring a connection via Mitsubishi MC TCP/IP (Page 959)

Communication types (Page 965)

Communication types

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 959)

Mitsubishi communication drivers (Page 958)

Connections via Mitsubishi MC TCP/IP (Page 964)

Parameterization of the communications modules

FX3 PLCs

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

Q PLCs (Page 967)

Internal Ethernet port of the Q0xUDEH CPU (Page 969)

Q PLCs

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 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 hexadecimal values.

See also

FX3 PLCs (Page 966)

Internal Ethernet port of the Q0xUDEH CPU (Page 969)

Internal Ethernet port of the Q0xUDEH CPU

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

FX3 PLCs (Page 966)

Q PLCs (Page 967)

Performance features of communication

Permitted data types for Mitsubishi MC TCPI/IP

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

Data type	Operand type	Length
24-bit block	M, X, Y B, F	4 bytes
28-bit block	M, X, Y B, F	4 bytes
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

Supported CPU types for Mitsubishi MC TCP/IP (Page 971)

Addresses for Mitsubishi MC TCP/IP (Page 971)

Address areas for Mitsubishi MC TCP/IP (Page 972)

Address areas for Mitsubishi MC TCP/IP (Page 974)

Supported CPU types for Mitsubishi MC TCP/IP

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

Permitted data types for Mitsubishi MC TCPI/IP (Page 969)

Addresses for Mitsubishi MC TCP/IP (Page 971)

Address areas for Mitsubishi MC TCP/IP (Page 972)

Address areas for Mitsubishi MC TCP/IP (Page 974)

Addresses for Mitsubishi MC TCP/IP

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

Name	Operand type	Max. address FX3	Max. address Q-Series
Link flag	B	--	Hex: B0 - FFF
Error flag	F	--	F0 - 2047

See also

Permitted data types for Mitsubishi MC TCPI/IP (Page 969)
Supported CPU types for Mitsubishi MC TCP/IP (Page 971)
Address areas for Mitsubishi MC TCP/IP (Page 972)
Address areas for Mitsubishi MC TCP/IP (Page 974)

Address areas for Mitsubishi MC TCP/IP

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	--	--	--	--	--	--	--	--
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 - YFFEC	Y0 - YFFE8	Y0 - YFFE4	Y0 - YFFE0

See also

Permitted data types for Mitsubishi MC TCPI/IP (Page 969)
Supported CPU types for Mitsubishi MC TCP/IP (Page 971)
Addresses for Mitsubishi MC TCP/IP (Page 971)
Address areas for Mitsubishi MC TCP/IP (Page 974)

Address areas for Mitsubishi MC TCP/IP

FX3

Address areas	Data types															
	Bool	**Int**	**Word**	**DInt**	**DWord**	**Real**	**String**	**Raw**	**4-bit block**	**8-bit block**	**12-bit block**	**16-bit block**	**20-bit block**	**24-bit block**	**28-bit block**	**32-bit block**
D	--	D0 - D65535	D0 - D65535	D0 - D65534	D0 - D65534	D0 - D65534	D0 - D65535	D0 - D65535	--	--	D0 - D65535	D0 - D65535	D0 - D65534	D0 - D65534	D0 - D65534	D0 - D65534
M	M0 - M65535	M0 - M65535	M0 - M65535	M0 - M65534	M0 - M65534	M0 - M65534	M0 - M65535	M0 - M65535	M0 - M65535	M0 - M65535	M0 - M65535	M0 - M65535	M0 - M65534	M0 - M65534	M0 - M65534	M0 - M65534
CN	CN0 - CN65535	CN0 - CN65535	CN0 - CN65535	CN0 - CN65534	CN0 - CN65534	CN0 - CN65534	CN0 - CN65535	CN0 - CN65535	CN0 - CN65535	CN0 - CN65535	CN0 - CN65535	CN0 - CN65535	CN0 - CN65534	CN0 - CN65534	CN0 - CN65534	CN0 - CN65534
TN	--	TN0 - TN65535	TN0 - TN65535	TN0 - TN65534	TN0 - TN65534	TN0 - TN65534	TN0 - TN65535	TN0 - TN65535	--	--	TN0 - TN65535	TN0 - TN65535	TN0 - TN65534	TN0 - TN65534	TN0 - TN65534	TN0 - TN65534
R	--	R0 - R65535	R0 - R65535	R0 - R65534	R0 - R65534	R0 - R65534	R0 - R65535	R0 - R65535	--	--	R0 - R65535	R0 - R65535	R0 - R65534	R0 - R65534	R0 - R65534	R0 - R65534
S	S0 - S65535	S0 - S65535	S0 - S65535	S0 - S65534	S0 - S65534	S0 - S65534	S0 - S65535	S0 - S65535	S0 - S65535	S0 - S65535	S0 - S65535	S0 - S65535	S0 - S65534	S0 - S65534	S0 - S65534	S0 - S65534
CS	CS0 - CS65535	CS0 - CS65535	CS0 - CS65535	CS0 - CS65534	CS0 - CS65534	CS0 - CS65534	CS0 - CS65535	CS0 - CS65535	CS0 - CS65535	CS0 - CS65535	CS0 - CS65535	CS0 - CS65535	CS0 - CS65534	CS0 - CS65534	CS0 - CS65534	CS0 - CS65534
TS	TS0 - TS65535	TS0 - TS65535	TS0 - TS65535	TS0 - TS65534	TS0 - TS65534	TS0 - TS65534	TS0 - TS65535	TS0 - TS65535	TS0 - TS65535	TS0 - TS65535	TS0 - TS65535	TS0 - TS65535	X0 - X776	X0 - X776	X0 - X776	X0 - X776
X	X0 - X777	X0 - X777	X0 - X777	X0 - X776	X0 - X776	X0 - X776	X0 - X777	X0 - X777	X0 - X777	X0 - X777	X0 - X777	X0 - X777	X0 - X776	X0 - X776	X0 - X776	X0 - X776
Y	Y0 - Y777	Y0 - Y777	Y0 - Y777	Y0 - Y766	Y0 - Y766	Y0 - Y766	Y0 - Y777	Y0 - Y777	Y0 - Y777	Y0 - Y777	Y0 - Y777	Y0 - Y777	Y0 - Y766	Y0 - Y766	Y0 - Y766	Y0 - Y766

Q

Address area	Data types															
	Bool	**Int**	**Word**	**DInt**	**DWord**	**Real**	**String**	**Raw**	**4-bit block**	**8-bit block**	**12-bit block**	**16-bit block**	**20-bit block**	**24-bit block**	**28-bit block**	**32-bit block**
B	B0 - BFFFF	B0 - BFFFF	B0 - BFFFF	B0 - BFFFE	B0 - BFFFE	B0 - BFFFE	B0 - BFFFF	B0 - BFFFF	B0 - BFFFF	B0 - BFFFF	B0 - BFFFF	B0 - BFFFF	B0 - BFFFE	B0 - BFFFE	B0 - BFFFE	B0 - BFFFE
SB	SB0 - SBFFFF	SB0 - SBFFFF	SB0 - SBFFFF	SB0 - SBFFFE	SB0 - SBFFFE	SB0 - SBFFFE	SB0 - SBFFFF	SB0 - SBFFFF	SB0 - SBFFFF	SB0 - SBFFFF	SB0 - SBFFFF	SB0 - SBFFFF	SB0 - SBFFFE	SB0 - SBFFFE	SB0 - SBFFFE	SB0 - SBFFFE
CC	CC0 - CC65535	CC0 - CC65535	CC0 - CC65535	CC0 - CC65534	CC0 - CC65534	CC0 - CC65534	CC0 - CC65535	CC0 - CC65535	CC0 - CC65535	CC0 - CC65535	CC0 - CC65535	CC0 - CC65535	CC0 - CC65534	CC0 - CC65534	CC0 - CC65534	CC0 - CC65534
SC	SC0 - SC65535	SC0 - SC65535	SC0 - SC65535	SC0 - SC65534	SC0 - SC65534	SC0 - SC65534	SC0 - SC65535	SC0 - SC65535	SC0 - SC65535	SC0 - SC65535	SC0 - SC65535	SC0 - SC65535	SC0 - SC65534	SC0 - SC65534	SC0 - SC65534	SC0 - SC65534
TC	TC0 - TC65535	TC0 - TC65535	TC0 - TC65535	TC0 - TC65534	TC0 - TC65534	TC0 - TC65534	TC0 - TC65535	TC0 - TC65535	TC0 - TC65535	TC0 - TC65535	TC0 - TC65535	TC0 - TC65535	TC0 - TC65534	TC0 - TC65534	TC0 - TC65534	TC0 - TC65534
D	--	D0 - D65535	D0 - D65535	D0 - D65534	D0 - D65534	D0 - D65534	D0 - D65535	D0 - D65535	--	--	D0 - D65535	D0 - D65535	D0 - D65534	D0 - D65534	D0 - D65534	D0 - D65534
SD	--	SD0 - SD65535	SD0 - SD65535	SD0 - SD65534	SD0 - SD65534	SD0 - SD65534	SD0 - SD65535	SD0 - SD65535	--	--	SD0 - SD65535	SD0 - SD65535	SD0 - SD65534	SD0 - SD65534	SD0 - SD65534	SD0 - SD65534
F	F0 - F65535	F0 - F65535	F0 - F65535	F0 - F65534	F0 - F65534	F0 - F65534	F0 - F65535	F0 - F65535	F0 - F65535	F0 - F65535	F0 - F65535	F0 - F65535	F0 - F65534	F0 - F65534	F0 - F65534	F0 - F65534
L	L0 - L65535	L0 - L65535	L0 - L65535	L0 - L65534	L0 - L65534	L0 - L65534	L0 - L65535	L0 - L65535	L0 - L65535	L0 - L65535	L0 - L65535	L0 - L65535	L0 - L65534	L0 - L65534	L0 - L65534	L0 - L65534
M	M0 - M65535	M0 - M65535	M0 - M65535	M0 - M65534	M0 - M65534	M0 - M65534	M0 - M65535	M0 - M65535	M0 - M65535	M0 - M65535	M0 - M65535	M0 - M65535	M0 - M65534	M0 - M65534	M0 - M65534	M0 - M65534
SM	SM0 - SM65535	SM0 - SM65535	SM0 - SM65535	SM0 - SM65534	SM0 - SM65534	SM0 - SM65534	SM0 - SM65535	SM0 - SM65535	SM0 - SM65535	SM0 - SM65535	SM0 - SM65535	SM0 - SM65535	SM0 - SM65534	SM0 - SM65534	SM0 - SM65534	SM0 - SM65534

Address area	Data types															
	Bool	**Int**	**Word**	**DInt**	**DWord**	**Real**	**String**	**Raw**	**4-bit block**	**8-bit block**	**12-bit block**	**16-bit block**	**20-bit block**	**24-bit block**	**28-bit block**	**32-bit block**
CN	--	CN0 - CN65535	CN0 - CN65535	CN0 - CN65534	CN0 - CN65534	CN0 - CN65534	CN0 - CN65535	CN0 - CN65535	--	--	CN0 - CN65535	CN0 - CN65535	CN0 - CN65534	CN0 - CN65534	CN0 - CN65534	CN0 - CN65534
SN	--	SN0 - SN65535	SN0 - SN65535	SN0 - SN65534	SN0 - SN65534	SN0 - SN65534	SN0 - SN65535	SN0 - SN65535	--	--	SN0 - SN65535	SN0 - SN65535	SN0 - SN65534	SN0 - SN65534	SN0 - SN65534	SN0 - SN65534
TN	--	TN0 - TN65535	TN0 - TN65535	TN0 - TN65534	TN0 - TN65534	TN0 - TN65534	TN0 - TN65535	TN0 - TN65535	--	--	TN0 - TN65535	TN0 - TN65535	TN0 - TN65534	TN0 - TN65534	TN0 - TN65534	TN0 - TN65534
R	--	R0 - R65535	R0 - R65535	R0 - R65534	R0 - R65534	R0 - R65534	R0 - R65535	R0 - R65535	--	--	R0 - R65535	R0 - R65535	R0 - R65534	R0 - R65534	R0 - R65534	R0 - R65534
S	S0 - SFFFF	S0 - SFFFF	S0 - SFFFF	S0 - SFFFE	S0 - SFFFE	S0 - SFFFE	S0 - SFFFF	S0 - SFFFF	S0 - SFFFF	S0 - SFFFF	S0 - SFFFF	S0 - SFFFF	S0 - SFFFE	S0 - SFFFE	S0 - SFFFE	S0 - SFFFE
CS	CS0 - CS65535	CS0 - CS65535	CS0 - CS65535	S0 - SFFFE	S0 - SFFFE	S0 - SFFFE	S0 - SFFFF	S0 - SFFFF	S0 - SFFFF	S0 - SFFFF	S0 - SFFFF	S0 - SFFFF	S0 - SFFFE	S0 - SFFFE	S0 - SFFFE	S0 - SFFFE
SS	SS0 - SS65535	SS0 - SS65535	SS0 - SS65535	SS0 - SS65534	SS0 - SS65534	SS0 - SS65534	SS0 - SS65535	SS0 - SS65535	SS0 - SS65535	SS0 - SS65535	SS0 - SS65535	SS0 - SS65535	SS0 - SS65534	SS0 - SS65534	SS0 - SS65534	SS0 - SS65534
TS	TS0 - TS65535	TS0 - TS65535	TS0 - TS65535	TS0 - TS65534	TS0 - TS65535	TS0 - TS65535	TS0 - TS65535	TS0 - TS65535	TS0 - TS65535	TS0 - TS65535	TS0 - TS65535	TS0 - TS65535	TS0 - TS65534	TS0 - TS65534	TS0 - TS65534	TS0 - TS65534
V	V0 - V65535	V0 - V65535	V0 - V65535	V0 - V65534	V0 - V65534	V0 - V65534	V0 - V65534	V0 - V65534	V0 - V65534	V0 - V65534	V0 - V65534	V0 - V65534	V0 - V65534	V0 - V65534	V0 - V65534	V0 - V65534
W	--	W0 - WFFFF	W0 - WFFFF	W0 - WFFFE	W0 - WFFFE	W0 - WFFFE	W0 - WFFFF	W0 - WFFFF	--	--	W0 - WFFFF	W0 - WFFFF	W0 - WFFFE	W0 - WFFFE	W0 - WFFFE	W0 - WFFFE
SW	--	SW0 - SWFFFF	SW0 - SWFFFF	SW0 - SWFFFE	SW0 - SWFFFE	SW0 - SWFFFE	SW0 - SWFFFF	SW0 - SWFFFF	--	--	SW0 - SWFFFF	SW0 - SWFFFF	SW0 - SWFFFE	SW0 - SWFFFE	SW0 - SWFFFE	SW0 - SWFFFE
X	X0 - XFFFF	X0 - XFFFF	X0 - XFFFF	X0 - XFFFE	X0 - XFFFE	X0 - XFFFE	X0 - XFFFF	X0 - XFFFF	X0 - XFFFF	X0 - XFFFF	X0 - XFFFF	X0 - XFFFF	X0 - XFFFE	X0 - XFFFE	X0 - XFFFE	X0 - XFFFE
DX	DX0 - DXFFFF	DX0 - DXFFFF	DX0 - DXFFFF	DX0 - DXFFFE	DX0 - DXFFFE	DX0 - DXFFFF	DX0 - DXFFFF	DX0 - DXFFFF	DX0 - DXFFFF	DX0 - DXFFFF	DX0 - DXFFFF	DX0 - DXFFFF	DX0 - DXFFFE	DX0 - DXFFFE	DX0 - DXFFFE	DX0 - DXFFFE

Address area	Data types															
	Bool	**Int**	**Word**	**DInt**	**DWord**	**Real**	**String**	**Raw**	**4-bit block**	**8-bit block**	**12-bit block**	**16-bit block**	**20-bit block**	**24-bit block**	**28-bit block**	**32-bit block**
Y	Y0 - YFFF F	Y0 - YFFF F	Y0 - YFFF F	Y0 - YFFF E	Y0 - YFFF E	Y0 - YFFF E	Y0 - YFFF F	Y0 - YFFF F	Y0 - YFFF F	Y0 - YFFF F	Y0 - YFFF F	Y0 - YFFF F	Y0 - YFFF E	Y0 - YFFF E	Y0 - YFFF E	Y0 - YFFF E
DY	DY0 - DYFF FF	DY0 - DYFF FF	DY0 - DYFF FF	DY0 - DYFF FE	DY0 - DYFF FE	DY0 - DYFF FE	DY0 - DYFF FF	DY0 - DYFF FF	DY0 - DYFF FF	DY0 - DYFF FF	DY0 - DYFF FF	DY0 - DYFF FF	DY0 - DYFF FE	DY0 - DYFF FE	DY0 - DYFF FE	DY0 - DYFF FE
Z	--	Z0 - Z655 35	Z0 - Z655 35	Z0 - Z655 34	Z0 - Z655 34	Z0 - Z655 34	Z0 - Z655 35	Z0 - Z655 35	--	--	Z0 - Z655 35	Z0 - Z655 35	Z0 - Z655 34	Z0 - Z655 34	Z0 - Z655 34	Z0 - Z655 34

See also

Permitted data types for Mitsubishi MC TCPI/IP (Page 969)

Supported CPU types for Mitsubishi MC TCP/IP (Page 971)

Addresses for Mitsubishi MC TCP/IP (Page 971)

Address areas for Mitsubishi MC TCP/IP (Page 972)

Commissioning components

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

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.

See also

Configuring a connection via Mitsubishi MC TCP/IP (Page 959)

Mitsubishi FX

Configuring a connection via Mitsubishi FX

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.

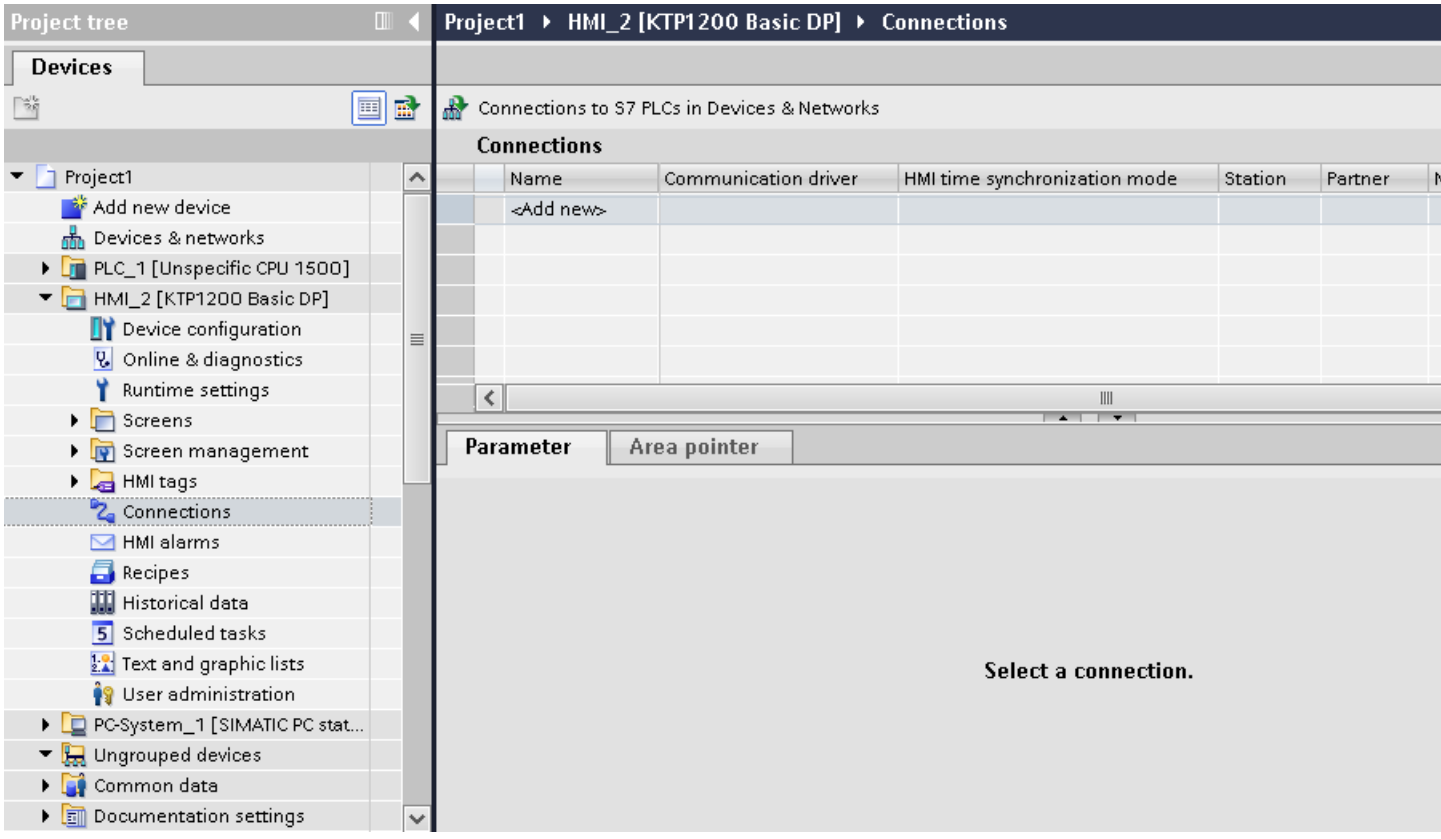
The interfaces are named differently depending on the HMI device.

Requirements

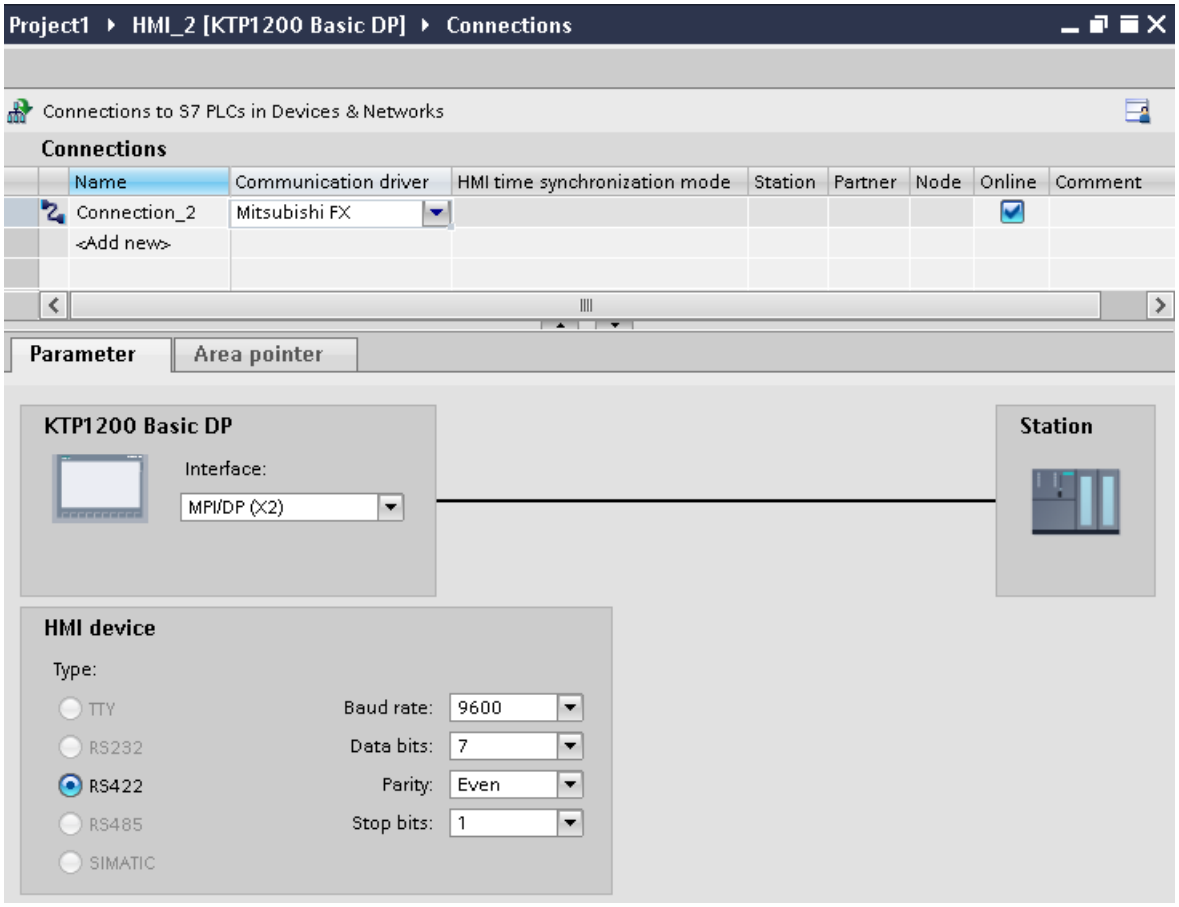
- A project is open.
- An HMI device has been created.

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".

See also

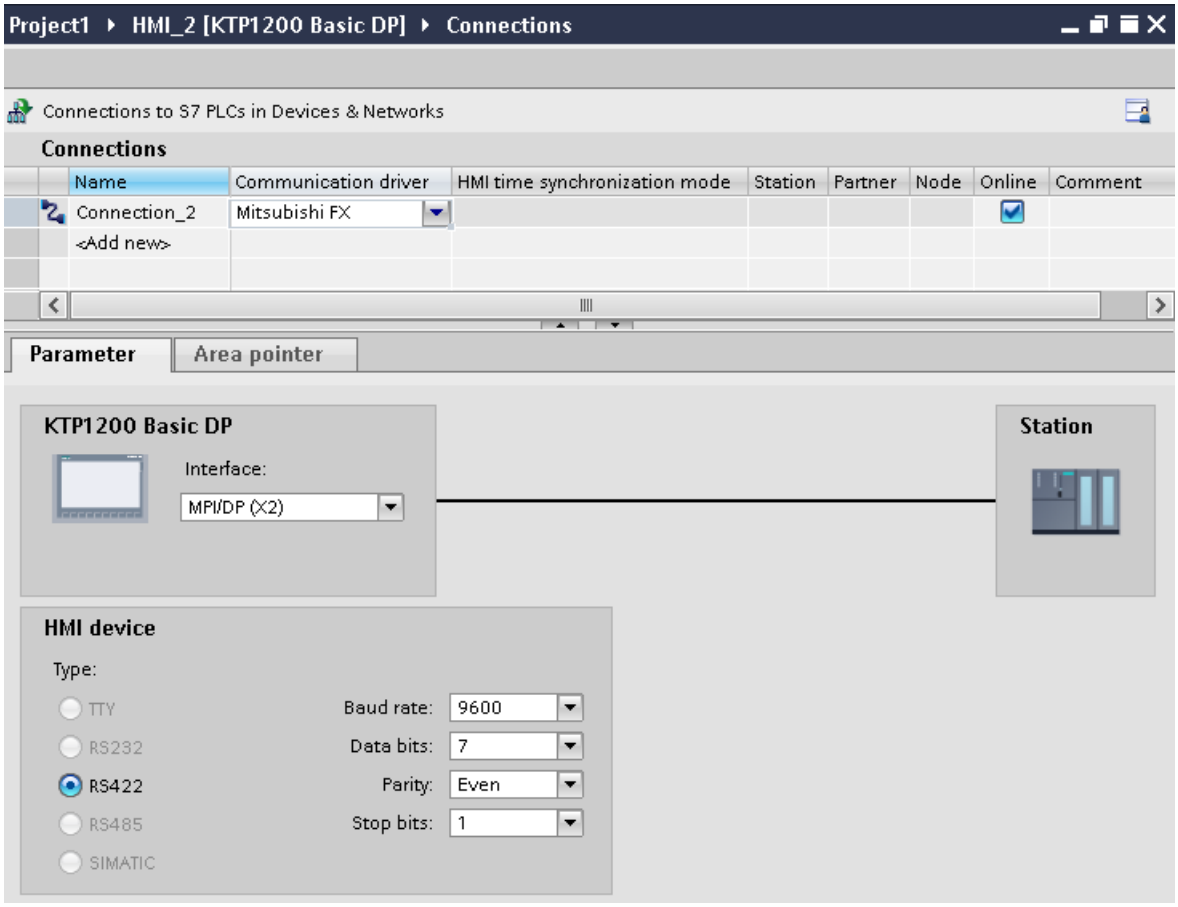
Parameters for the connection (Mitsubishi FX) (Page 980)

Parameters for the connection (Mitsubishi FX)

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

  When using the IF1B interface, you must switch over the RS422 receive data and the RTS signal additionally using 4 DIP switches on the rear of the HMI device. The newer versions of Comfort Panels use electronic circuits instead of DIP switches. The newer versions of Comfort Panels use electronic circuits instead of DIP switches.

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".

See also

Configuring a connection via Mitsubishi FX (Page 979)

Connecting HMI device to PLC

Connections via Mitsubishi FX

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

Applies only to RS 232:

Cable length is restricted to 0.32 m.

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 Mitsubishi FX".

See also

Configuring a connection via Mitsubishi FX (Page 979)

Communication types

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:

PLC	Mitsubishi FX (PG protocol)	Mitsubishi MC TCP/IP
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

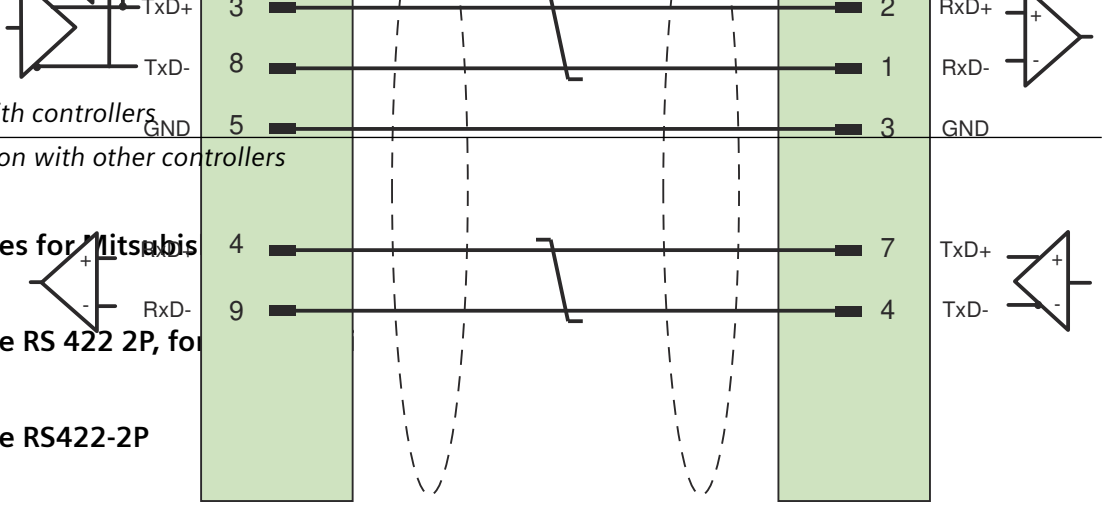
See also

Configuring a connection via Mitsubishi FX (Page 979)

Connecting cables for Mitsubis

Connecting cable RS 422 2P, fo

Connecting cable RS422-2P



Shield with large-area contact to housing at both ends
Cable: 3 x 2 x 0.14 mm^2, shielded,
max. length 500 m

See also

Connections via Mitsubishi FX (Page 982)

Configuring a connection via Mitsubishi FX (Page 979)

Performance features of communication

Permitted data types for Mitsubishi FX

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.

See also

Configuring a connection via Mitsubishi FX (Page 979)

Supported CPU types for Mitsubishi FX

CPU types

The following CPU types are supported for configuring the Mitsubishi FX communication driver.

- FX1 series
  - FX1n
- FX2 series
  - FX2n

See also

Configuring a connection via Mitsubishi FX (Page 979)

Address areas for Mitsubishi FX

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								

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**
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

See also

Configuring a connection via Mitsubishi FX (Page 979)

Commissioning components

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

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

Area pointers for Mitsubishi

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".

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

Mitsubishi communication drivers (Page 958)

Trends

General information on trends

Trends

A trend is the graphical representation of one or more values from the PLC. The value is read out either time- or bit-triggered, depending on the configuration.

For additional information see:

AUTOHOTSPOT

Note

The value is read out time-triggered for Basic Panels.

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.

Bit-triggered trends

Through a trigger bit set in the trend request tag, the HMI device either reads in a trend value or an entire trend buffer. This setting is defined in the configuration. Bit-triggered trends are normally used to represent fast 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 "HMI tags" editor and connected to trend areas during configuration. The HMI device and PLC then communicate with each other via 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)

See also

Area pointers for Mitsubishi (Page 988)

Trend request and trend transfer (Page 989)

Restrictions to the trend control (Page 990)

Trend request and trend transfer

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 closing the screen, the HMI device resets the relevant bits in the trend request area.

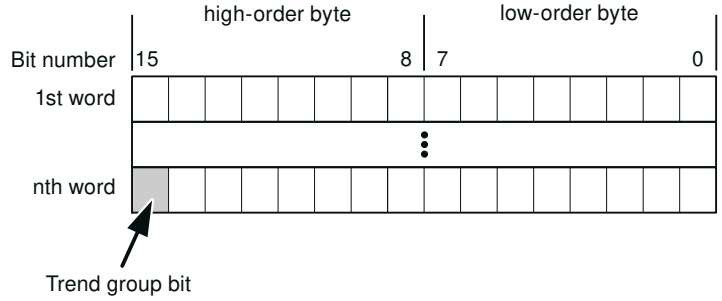
The trend request area can be used for evaluation purposes in the PLC to determine which trend is currently being displayed on the HMI device. Trends can also be triggered without evaluation of the trend request area.

Trend transfer area 1

This area is used to trigger trends. You must set the bit assigned to the trend in the trend transfer area and set the trend group bit in your control program. 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.

The PLC writes to Buffer 2 while the HMI device reads values from Buffer 1, and writes to Buffer 1 when the HMI device is reading Buffer 2. This prevents the PLC from overwriting trend values while the trend is being read by the HMI device.

See also

Area pointers for Mitsubishi (Page 988)

General information on trends (Page 988)

Restrictions to the trend control (Page 990)

Restrictions to the trend control

Data types for Mitsubishi MC TCPI/IP

For trend control you can create tags and array tags of the operand type "D" with the data type "Word" or "Int".

Data types for Mitsubishi FX

For trend control you can create tags and array tags of the operand type "D" with the data type "Word".

See also

Area pointers for Mitsubishi (Page 988)

General information on trends (Page 988)

Trend request and trend transfer (Page 989)

Alarms

Configuring alarms

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."

See also

Acknowledgment of alarms (Page 992)

Area pointers for Mitsubishi (Page 988)

Acknowledgment of alarms

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

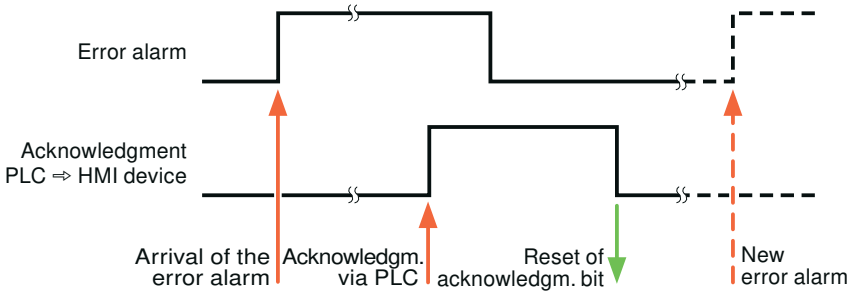
Acknowledgment by the PLC

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

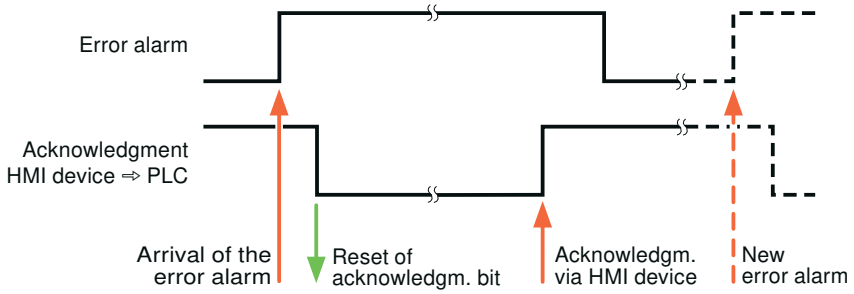
acknowledgment tag is then written to the PLC by the HMI device. This allows the PLC to recognize that a certain alarm message has been acknowledged at the HMI device.

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 991)

Area pointers for Mitsubishi (Page 988)

LED mapping

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 Comfort Panels	Panel PCs
0	0	Off	Off
0	1	Rapid flashing	Flashing
1	0	Slow flashing	Flashing
1	1	On permanently	On permanently

2.19.4.3 Modicon Modbus

Modicon Modbus communication drivers

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

Configuring a connection via Modicon Modbus TCP/IP

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

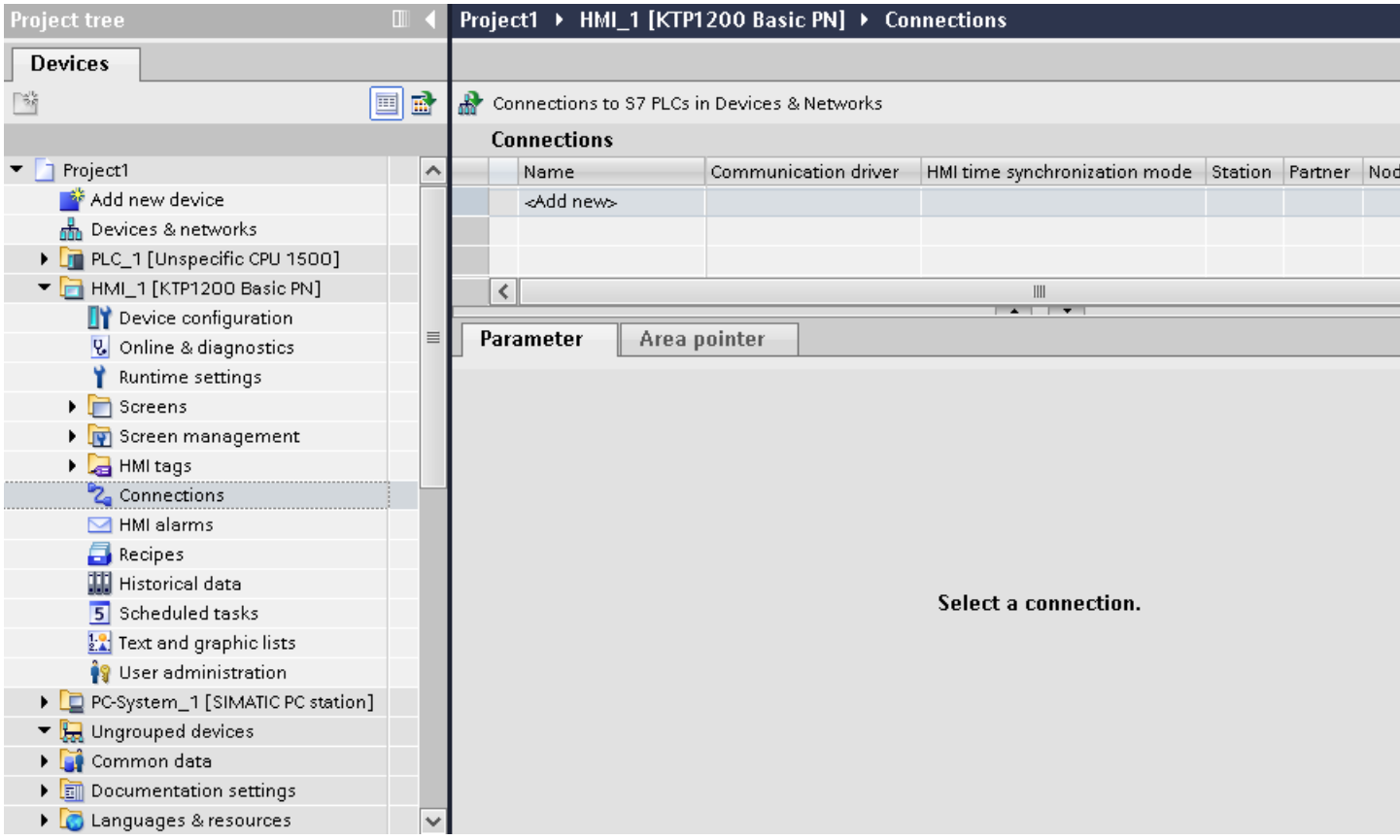
Requirements

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

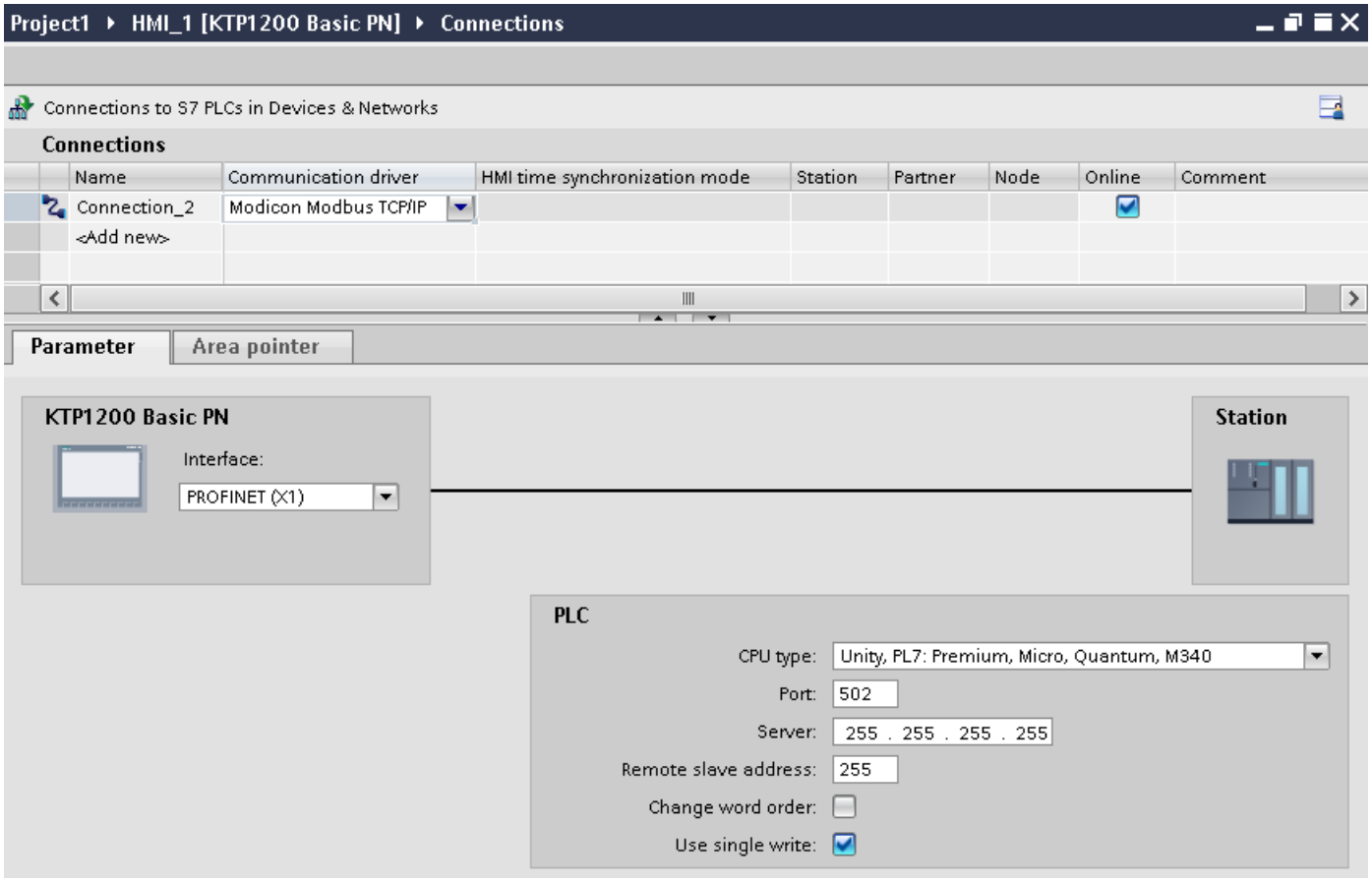
Procedure

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 995)

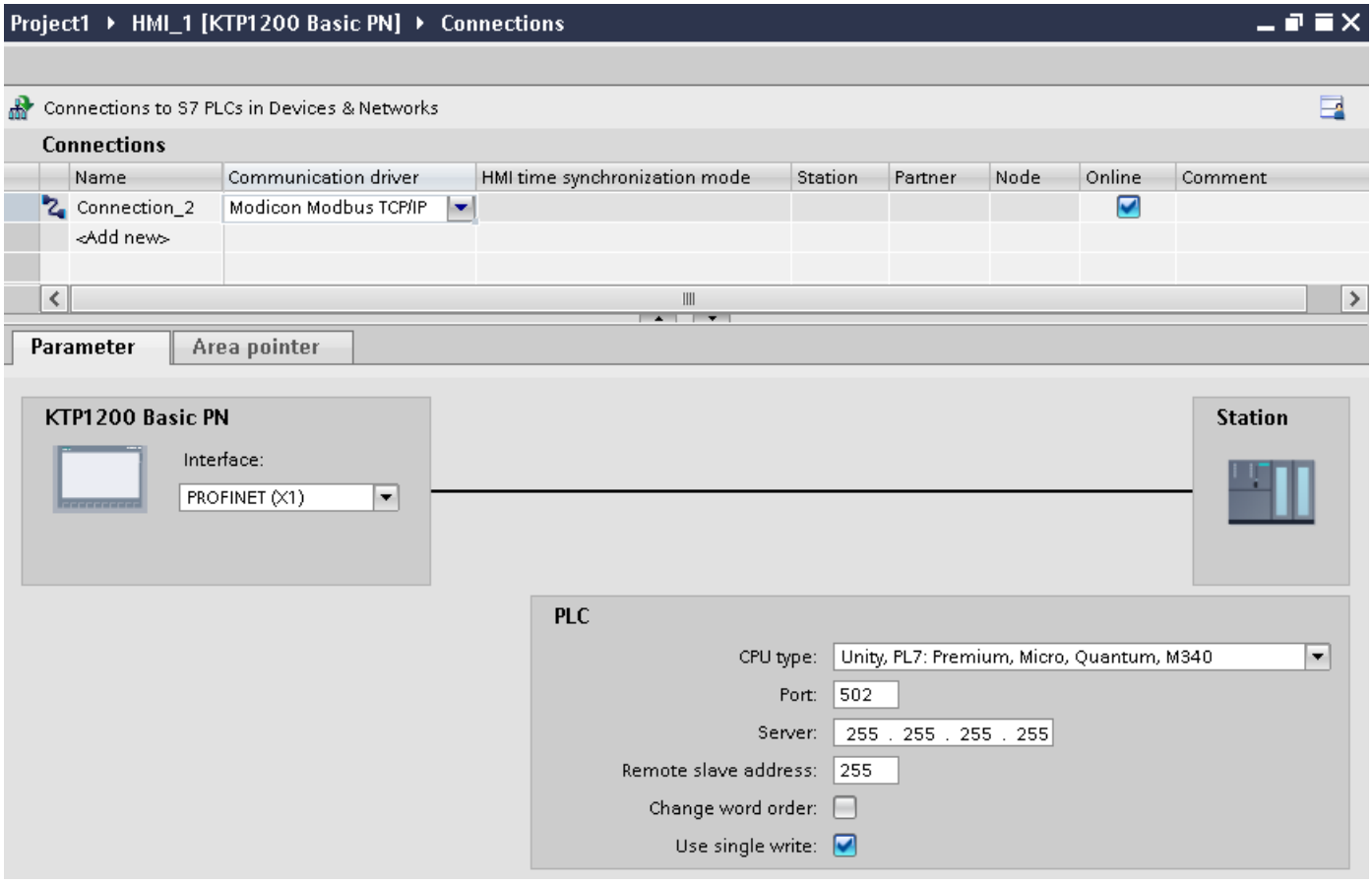
Parameters for the connection (Modicon Modbus TCP/IP) (Page 998)

Parameters for the connection (Modicon Modbus TCP/IP)

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 995)

Configuring a connection via Modicon Modbus TCP/IP (Page 996)

Connecting HMI device to PLC

Connections via Modicon Modbus TCP/IP

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 996)

Modicon Modbus communication drivers (Page 995)

Communication types

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 996)

Modicon Modbus communication drivers (Page 995)

Performance features of communication

Permissible data types for Modicon Modbus TCP/IP

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, PL7: Premium, Micro, Quantum M340"

Data type	Operand type	Length
+/- Double	%MW	4 bytes
+/- Int	%MW, %IW	2 bytes
16-bit group	%M, %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

Configuring a connection via Modicon Modbus TCP/IP (Page 996)

Modicon Modbus communication drivers (Page 995)

Supported CPU types for Modicon Modbus TCP/IP (Page 1005)

Address areas for Modicon Modbus TCP/IP (Page 1006)

Address areas for Modicon Modbus TCP/IP (Page 1007)

Supported CPU types for Modicon Modbus TCP/IP

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 995)
Configuring a connection via Modicon Modbus TCP/IP (Page 996)
Permissible data types for Modicon Modbus TCP/IP (Page 1004)
Address areas for Modicon Modbus TCP/IP (Page 1006)
Address areas for Modicon Modbus TCP/IP (Page 1007)

Address areas for Modicon Modbus TCP/IP

Unity, PI7

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

Concept, ProWORX

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	--	--	--	--

Address areas	Data types							
	Bool	**16 Bit Group**	**Int**	**+/- Int**	**DInt**	**+/- DInt**	**Float**	**ASCII**
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 PLC, an offset of 1 is configured for each address.

The offset is due to the different address formats of the PLC and Unity Pro. The valid address area for MODBUS Quantum PLCs starts with "0x400001", "0x300001", etc., while the address area for Unity Pro starts with "%MW0". During mapping, an offset of 1 is added for each read or write access by the MODBUS Quantum PLCs.

See also

Modicon Modbus communication drivers (Page 995)

Configuring a connection via Modicon Modbus TCP/IP (Page 996)

Permissible data types for Modicon Modbus TCP/IP (Page 1004)

Supported CPU types for Modicon Modbus TCP/IP (Page 1005)

Address areas for Modicon Modbus TCP/IP (Page 1007)

Address areas for Modicon Modbus TCP/IP

UnityPl7

Address areas	Data types							
	Bool	**16 Bit Group**	**Int**	**+/- Int**	**DInt**	**+/- DInt**	**Float**	**ASCII**
%1	%I0 - %I9999999	%I0 - %I9999999	--	--	--	--	--	--
%M	%M0 - %M9999999	%M0 - %M9999999	--	--	--	--	--	--

Address areas	Data types							
	Bool	**16 Bit Group**	**Int**	**+/- Int**	**DInt**	**+/- DInt**	**Float**	**ASCII**
%IW	%IW0.0 - %IW9999999.15	--	%IW0 - %IW9999999	%IW0 - %IW9999999	%IW0 - %IW9999998	%IW0 - %IW9999998	%IW0 - %IW9999998	%IW0 - %IW9999999
%MW	%MW0.0 - %MW9999999.15	--	%MW0 - %MW9999999	%MW0 - %MW9999999	%MW0 - %MW9999998	%MW0 - %MW9999998	%MW0 - %MW9999998	%MW0 - %MW9999999
6x	6x0.0:1 - 6x9999999.15:10	--	6x0:1 - 6x9999999:10	6x0:1 - 6x9999999:10	6x0:1 - 6x9999998:10	6x0:1 - 6x9999998:10	6x0:1 - 6x9999998:10	6x0:1 - 6x9999999:10

ConceptProWORX

Address areas	Data types							
	Bool	**16 Bit Group**	**Int**	**+/- Int**	**DInt**	**+/- DInt**	**Float**	**ASCII**
0x	0x0 - 0x9999999	0x0 - 0x9999999	--	--	--	--	--	--
1x	1x0 - 1x9999999	1x0 - 1x9999999	--	--	--	--	--	--
3x	3x0.1 - 3x9999999.16	--	3x0 - 3x9999999	3x0 - 3x9999999	3x0 - 3x9999999	3x0 - 3x9999999	3x0 - 3x9999999	3x0 - 3x9999999
4x	4x0.1 - 4x9999999.16	--	4x0 - 4x9999999	4x0 - 4x9999999	4x0 - 4x9999999	4x0 - 4x9999999	4x0 - 4x9999999	4x0 - 4x9999999
6x	6x0.1:1 - 6x9999999.16:10	--	6x0:1 - 6x9999999:10	6x0:1 - 6x9999999:10	6x0:1 - 6x9999999:10	6x0:1 - 6x9999999:10	6x0:1 - 6x9999999:10	6x0:1 - 6x9999999:10

Note

If you select the address format for Unity Pro when configuring the connection between HMI device and a MODBUS quantum PLC, an offset of 1 is configured for each address.

This is due to the different address formats of the PLC and Unity Pro. Whereas the valid address area for MODBUS Quantum PLCs starts with "0x400001", "0x300001" etc., the address area for Unity Pro starts with "%MW0". During mapping, an offset of 1 is added for each reading or writing access by MODBUS Quantum PLCs.

See also

Modicon Modbus communication drivers (Page 995)

Configuring a connection via Modicon Modbus TCP/IP (Page 996)

Permissible data types for Modicon Modbus TCP/IP (Page 1004)

Supported CPU types for Modicon Modbus TCP/IP (Page 1005)

Address areas for Modicon Modbus TCP/IP (Page 1006)

Commissioning components

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.

See also

Modicon Modbus communication drivers (Page 995)

Optimizing the configuration

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.

See also

Modicon Modbus communication drivers (Page 995)

Modicon Modbus RTU

Configuring a connection via Modicon Modbus RTU

Introduction

You configure a connection to a PLC with a Modicon Modbus RTU communication driver in the "Connections" editor of the HMI device.

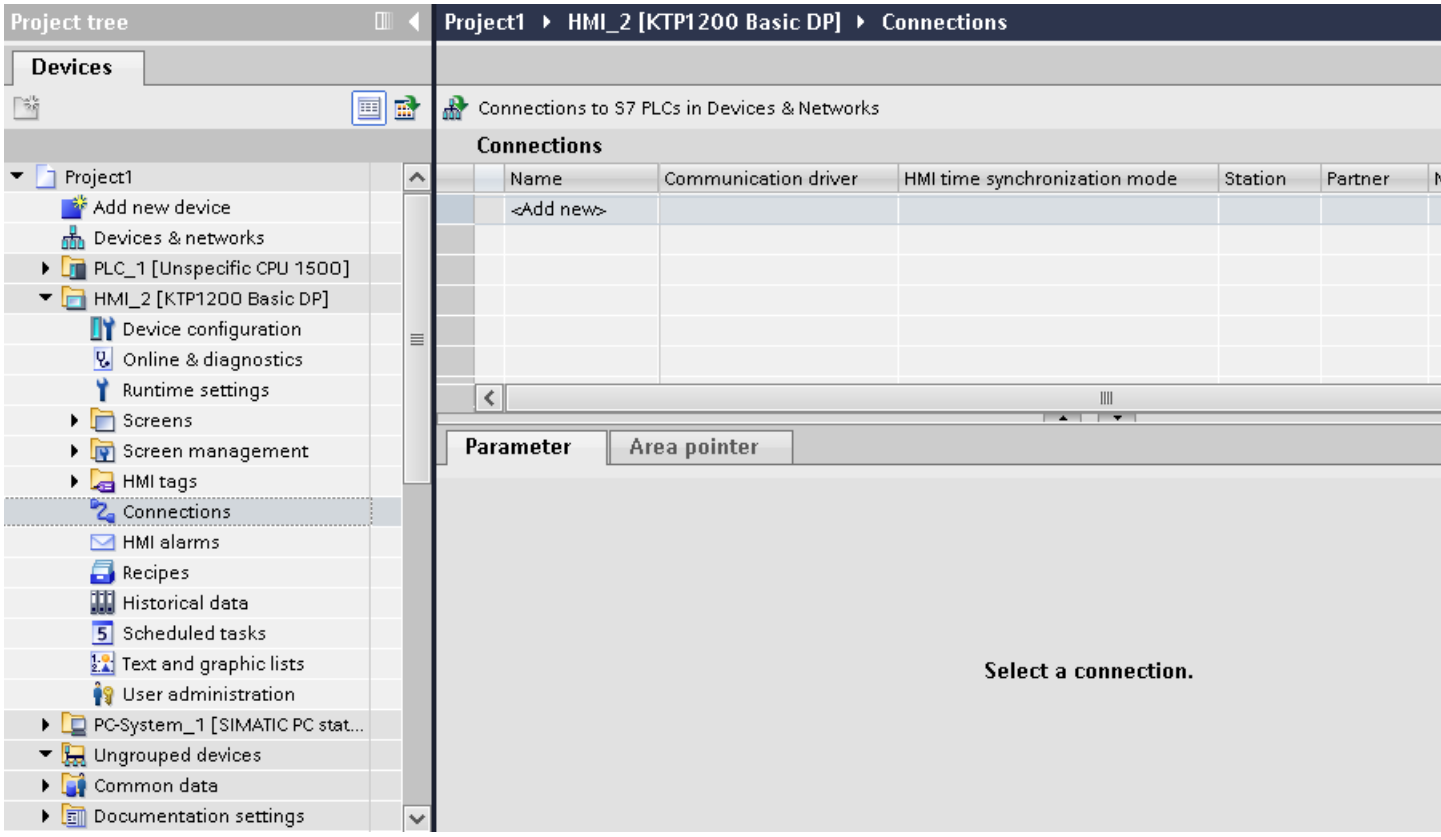
The interfaces are named differently depending on the HMI device.

Requirements

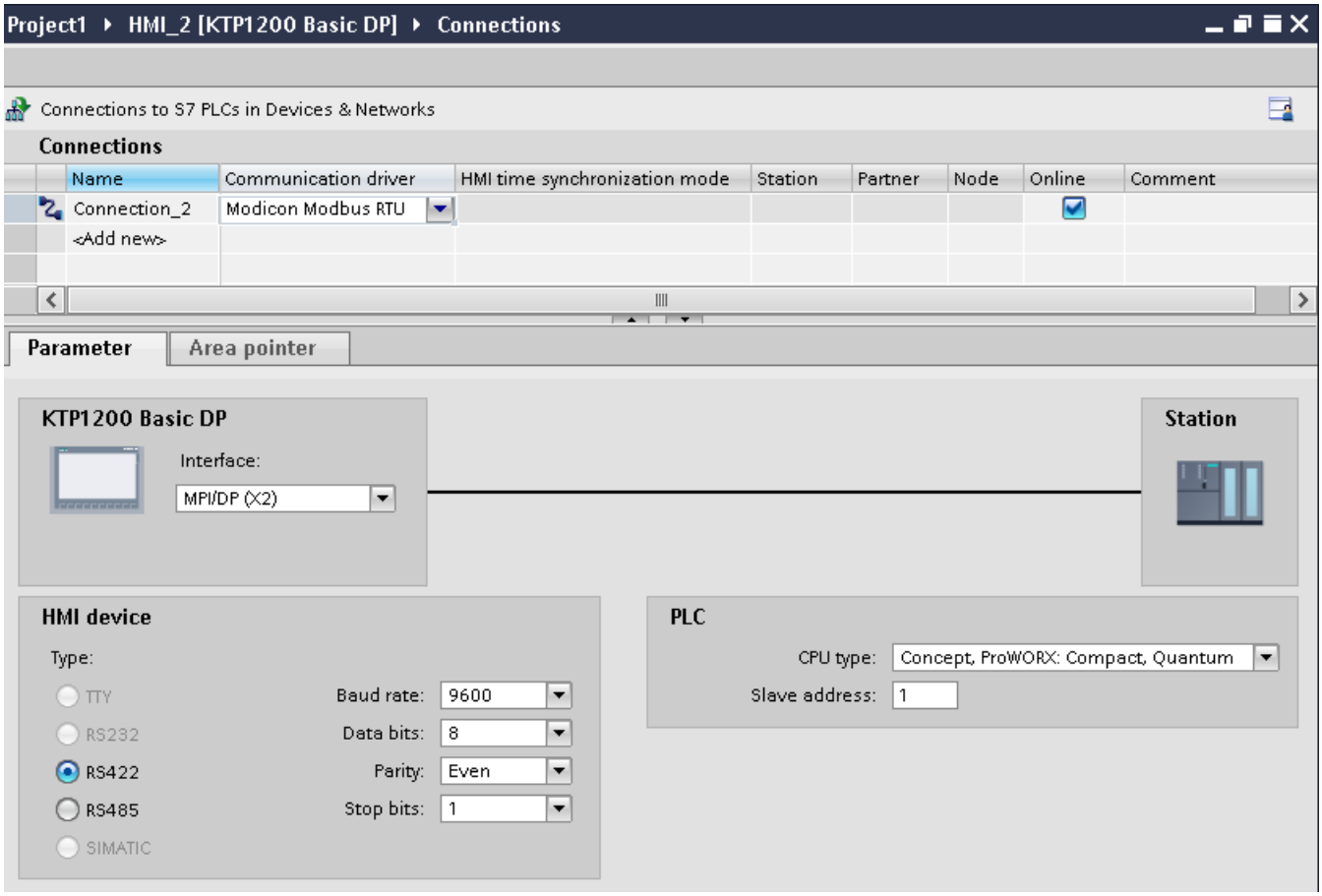
- A project is open.
- An HMI device has been created.

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 995)

Parameters for the connection (Modicon Modbus RTU) (Page 1012)

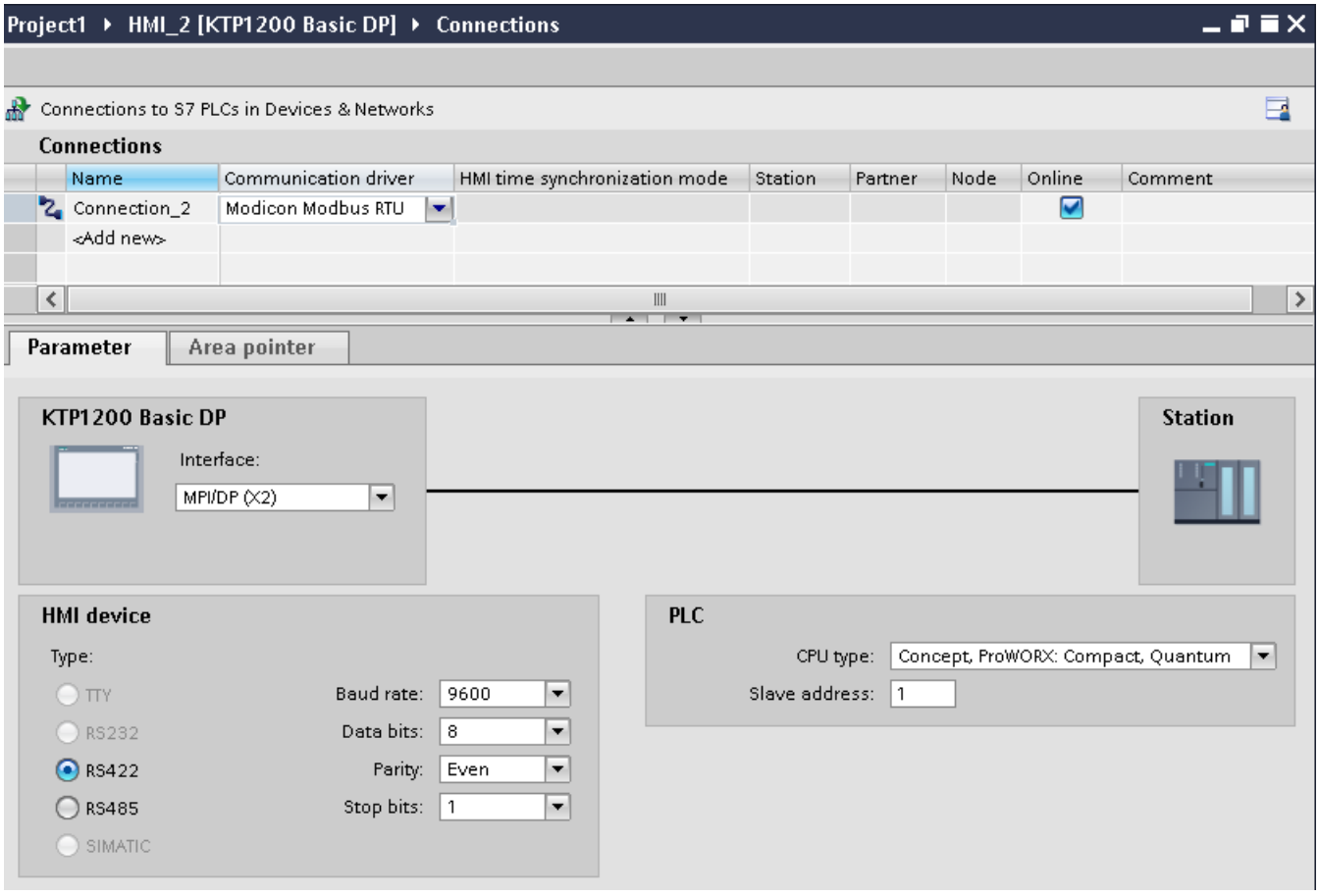
Optimizing the configuration (Page 1023)

Parameters for the connection (Modicon Modbus RTU)

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 995)

Configuring a connection via Modicon Modbus RTU (Page 1010)

Connecting HMI device to PLC

Connections via Modicon Modbus RTU

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

Modicon Modbus communication drivers (Page 995)

Configuring a connection via Modicon Modbus RTU (Page 1010)

Communication types (Page 1015)

Communication types

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

Reading function codes		Address range	
03	ReadHoldingRegisters	4x	USERDATA
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

Connections via Modicon Modbus RTU (Page 1014)

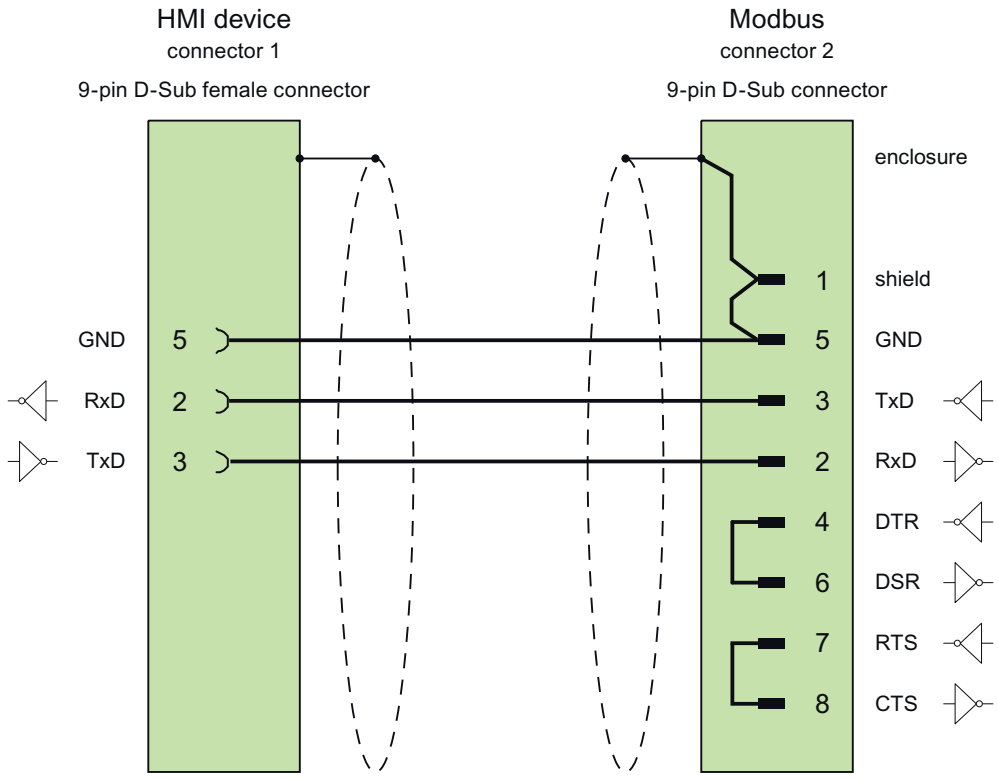
Modicon Modbus communication drivers (Page 995)

Configuring a connection via Modicon Modbus RTU (Page 1010)

Connecting cables for Modicon Modbus RTU

Connecting cable PP1, RS-232, for Modicon

Point-to-point cable 1: PLC > PC ...



Cables: 3 x 0.14 mm^2, shielded,
max. length 15 m

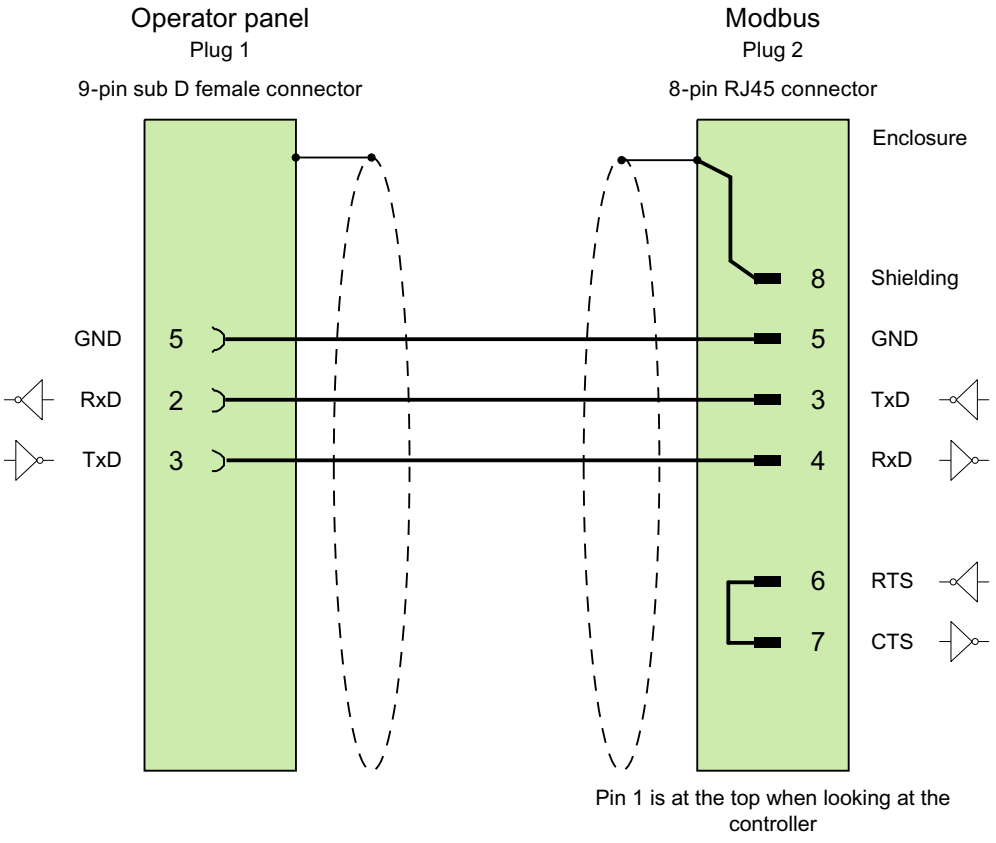
See also

Connections via Modicon Modbus RTU (Page 1014)

Modicon Modbus communication drivers (Page 995)

Connecting cable PP2, RS-232, for Modicon

Point-to-point cable 2: PLC (TSX Compact) > PC...



Cables: 3 x 0.14 mm², shielded,
max. length 15 m

See also

Connections via Modicon Modbus RTU (Page 1014)

Modicon Modbus communication drivers (Page 995)

Performance features of communication

Permitted data types for Modicon Modbus RTU

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

Configuring a connection via Modicon Modbus RTU (Page 1010)

Supported CPU types for Modicon Modbus RTU (Page 1020)

Address areas for Modicon Modbus RTU (Page 1020)

Address areas for Modicon Modbus RTU (Page 1021)

Supported CPU types for Modicon Modbus RTU

CPU types

The following CPU types are supported in the configuration of the Modicon Modbus RTU communication driver.

- Compact
- Momentum
- Quantum

See also

Configuring a connection via Modicon Modbus RTU (Page 1010)

Permitted data types for Modicon Modbus RTU (Page 1019)

Address areas for Modicon Modbus RTU (Page 1020)

Address areas for Modicon Modbus RTU (Page 1021)

Address areas for Modicon Modbus RTU

Concept, ProWORX

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 PLC, an offset of 1 is configured for each address.

The offset is due to the different address formats of the PLC and Unity Pro. The valid address area for MODBUS Quantum PLCs starts with "0x400001", "0x300001", etc., while the address area for Unity Pro starts with "%MW0". During mapping, an offset of 1 is added for each read or write access by the MODBUS Quantum PLCs.

See also

Configuring a connection via Modicon Modbus RTU (Page 1010)

Permitted data types for Modicon Modbus RTU (Page 1019)

Supported CPU types for Modicon Modbus RTU (Page 1020)

Address areas for Modicon Modbus RTU (Page 1021)

Address areas for Modicon Modbus RTU

UnityPl7

Address areas	Data types							
	Bool	**16 Bit Group**	**Int**	**+/- Int**	**DInt**	**+/- DInt**	**Float**	**ASCII**
%1	%I0 - %I9999999	%I0 - %I9999999	--	--	--	--	--	--
%M	%M0 - %M9999999	%M0 - %M9999999	--	--	--	--	--	--
%IW	%IW0.0 - %IW9999999.15	--	%IW0 - %IW9999999	%IW0 - %IW9999999	%IW0 - %IW9999998	%IW0 - %IW9999998	%IW0 - %IW9999998	%IW0 - %IW9999999
%MW	%MW0.0 - %MW9999999.15	--	%MW0 - %MW9999999	%MW0 - %MW9999999	%MW0 - %MW9999998	%MW0 - %MW9999998	%MW0 - %MW9999998	%MW0 - %MW9999999
6x	6x0.0:1 - 6x9999999.15:10	--	6x0:1 - 6x9999999:10	6x0:1 - 6x9999999:10	6x0:1 - 6x9999998:10	6x0:1 - 6x9999998:10	6x0:1 - 6x9999998:10	6x0:1 - 6x9999999:10

ConceptProWORX

Address areas	Data types							
	Bool	**16 Bit Group**	**Int**	**+/- Int**	**DInt**	**+/- DInt**	**Float**	**ASCII**
0x	0x0 - 0x9999999	0x0 - 0x9999999	--	--	--	--	--	--
1x	1x0 - 1x9999999	1x0 - 1x9999999	--	--	--	--	--	--
3x	3x0.1 - 3x9999999 .16	--	3x0 - 3x9999999	3x0 - 3x9999999	3x0 - 3x9999999	3x0 - 3x9999999	3x0 - 3x9999999	3x0 - 3x9999999
4x	4x0.1 - 4x9999999 .16	--	4x0 - 4x9999999	4x0 - 4x9999999	4x0 - 4x9999999	4x0 - 4x9999999	4x0 - 4x9999999	4x0 - 4x9999999
6x	6x0.1:1 - 6x9999999 .16:10	--	6x0:1 - 6x9999999 :10	6x0:1 - 6x9999999 :10	6x0:1 - 6x9999999 :10	6x0:1 - 6x9999999 :10	6x0:1 - 6x9999999 :10	6x0:1 - 6x9999999 :10

Note

If you select the address format for Unity Pro when configuring the connection between HMI device and a MODBUS quantum PLC, an offset of 1 is configured for each address.

This is due to the different address formats of the PLC and Unity Pro. Whereas the valid address area for MODBUS Quantum PLCs starts with "0x400001", "0x300001" etc., the address area for Unity Pro starts with "%MW0". During mapping, an offset of 1 is added for each reading or writing access by MODBUS Quantum PLCs.

See also

Configuring a connection via Modicon Modbus RTU (Page 1010)

Permitted data types for Modicon Modbus RTU (Page 1019)

Supported CPU types for Modicon Modbus RTU (Page 1020)

Address areas for Modicon Modbus RTU (Page 1020)

Commissioning components

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

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.

See also

Configuring a connection via Modicon Modbus RTU (Page 1010)

Data exchange

Area pointers for Modicon Modbus

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".

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

Modicon Modbus communication drivers (Page 995)

Trends

General information on trends

Trends

A trend is the graphical representation of one or more values from the PLC. The value is read out either time- or bit-triggered, depending on the configuration.

For additional information see:

AUTOHOTSPOT

Note

The value is read out time-triggered for Basic Panels.

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.

Bit-triggered trends

Through a trigger bit set in the trend request tag, the HMI device either reads in a trend value or an entire trend buffer. This setting is defined in the configuration. Bit-triggered trends are normally used to represent fast 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 "HMI tags" editor and connected to trend areas during configuration. The HMI device and PLC then communicate with each other via 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)

See also

Area pointers for Modicon Modbus (Page 1024)

Trend request and trend transfer (Page 1026)

Permitted data types for trend control (Page 1027)

Trend request and trend transfer

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 closing the screen, the HMI device resets the relevant bits in the trend request area.

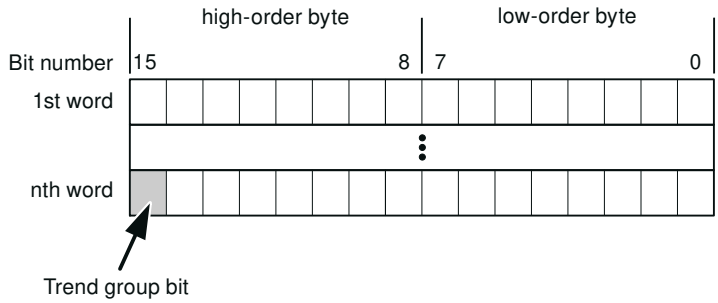
The trend request area can be used for evaluation purposes in the PLC to determine which trend is currently being displayed on the HMI device. Trends can also be triggered without evaluation of the trend request area.

Trend transfer area 1

This area is used to trigger trends. You must set the bit assigned to the trend in the trend transfer area and set the trend group bit in your control program. 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.

The PLC writes to Buffer 2 while the HMI device reads values from Buffer 1, and writes to Buffer 1 when the HMI device is reading Buffer 2. This prevents the PLC from overwriting trend values while the trend is being read by the HMI device.

See also

Area pointers for Modicon Modbus (Page 1024)

General information on trends (Page 1025)

Permitted data types for trend control (Page 1027)

Permitted data types for trend control

Modicon Modbus TCP/IP

Tags of the following operand types are permitted for CPU type "Concept, ProWORX: Compact, Quantum, Momentum".

- "4x"
- "6x"

Tags of the following operand types are permitted for CPU type "Unity, PL7: Premium, Micro, Quantum, M340".

- "%MW"

The tags for trend control must have data type "Int" or "+/- Int" or be an array tag with data type "Int" or "+/-Int".

You assign a bit to a trend during configuration. This sets a defined bit assignment for all areas.

Modicon Modbus RTU

Tags of the following operand types are permitted for CPU type "Concept, ProWORX: Compact, Quantum".

- "4x"
- "6x"

The tags for trend control must have data type "Int" or "+/- Int" or be an array tag with data type "Int" or "+/-Int".

You assign a bit to a trend during configuration. This sets a defined bit assignment for all areas.

See also

Area pointers for Modicon Modbus (Page 1024)

General information on trends (Page 1025)

Trend request and trend transfer (Page 1026)

Alarms

Configuring alarms

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

Area pointers for Modicon Modbus (Page 1024)

Acknowledgment of alarms (Page 1029)

Acknowledgment of alarms

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

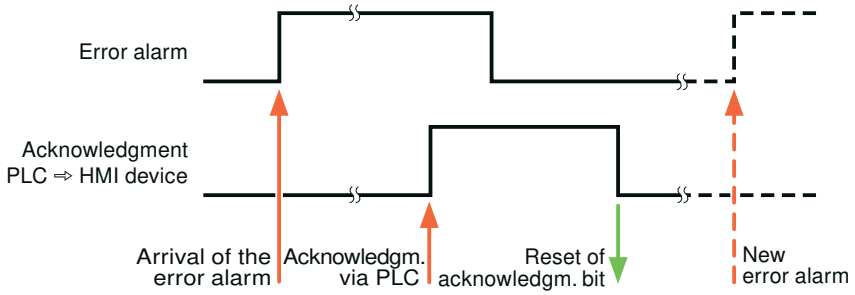
Acknowledgment by the PLC

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.

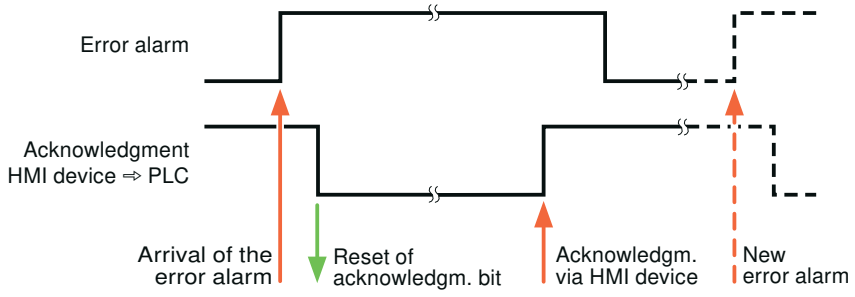
If a discrete alarm subject to acknowledgment is acknowledged by the HMI device, then the corresponding bit in the assigned acknowledgment tag will be set. The entire acknowledgment tag is then written to the PLC by the HMI device. This allows the PLC to recognize that a certain alarm message has been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

Area pointers for Modicon Modbus (Page 1024)

Configuring alarms (Page 1028)

LED mapping

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 Comfort Panels	Panel PCs
0	0	Off	Off
0	1	Rapid flashing	Flashing
1	0	Slow flashing	Flashing
1	1	On permanently	On permanently

2.19.4.4 Omron

Omron communication drivers

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
- Omron Ethernet IP

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

Configuring a connection via Omron Host Link

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.

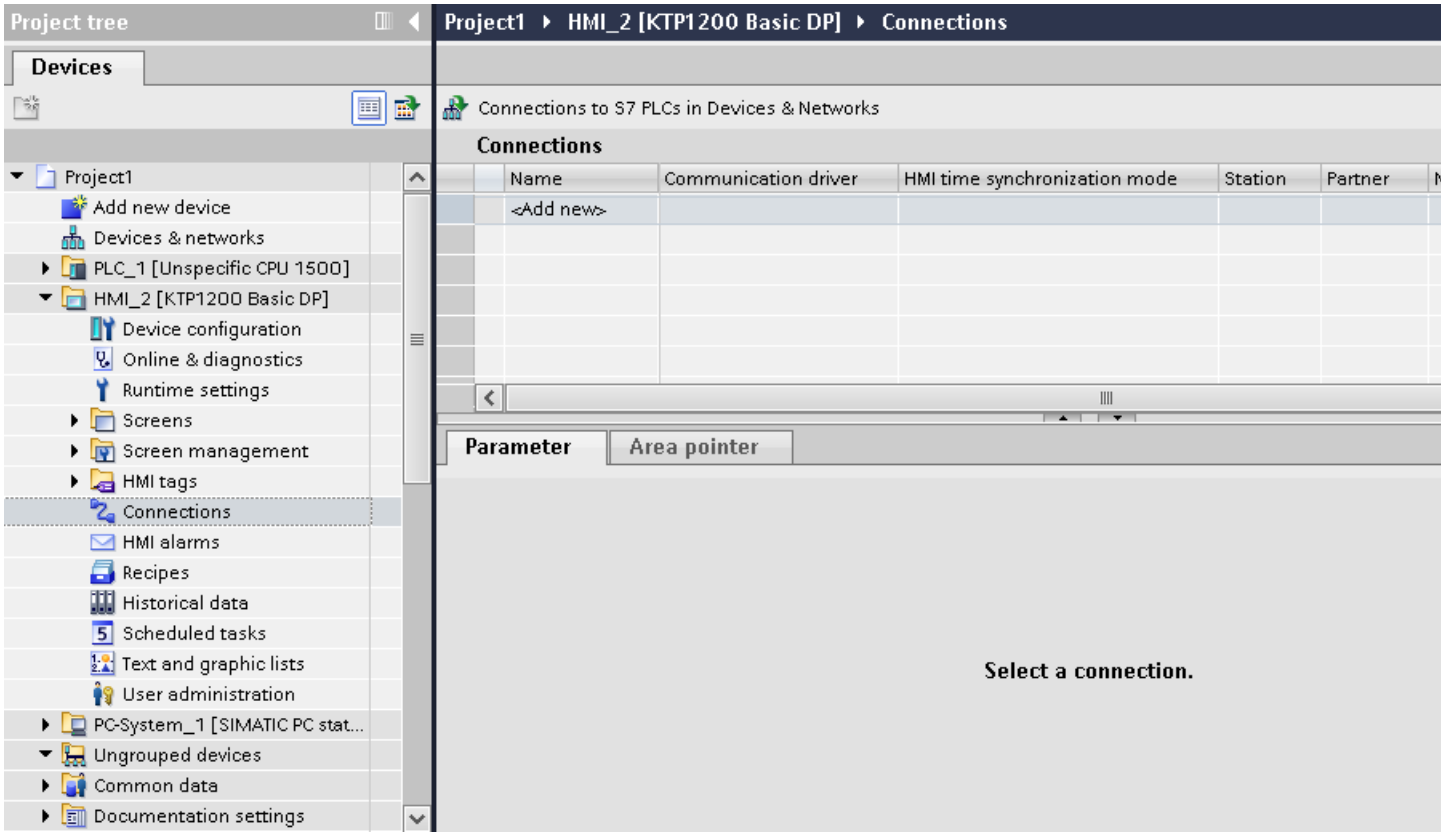
The interfaces are named differently depending on the HMI device.

Requirements

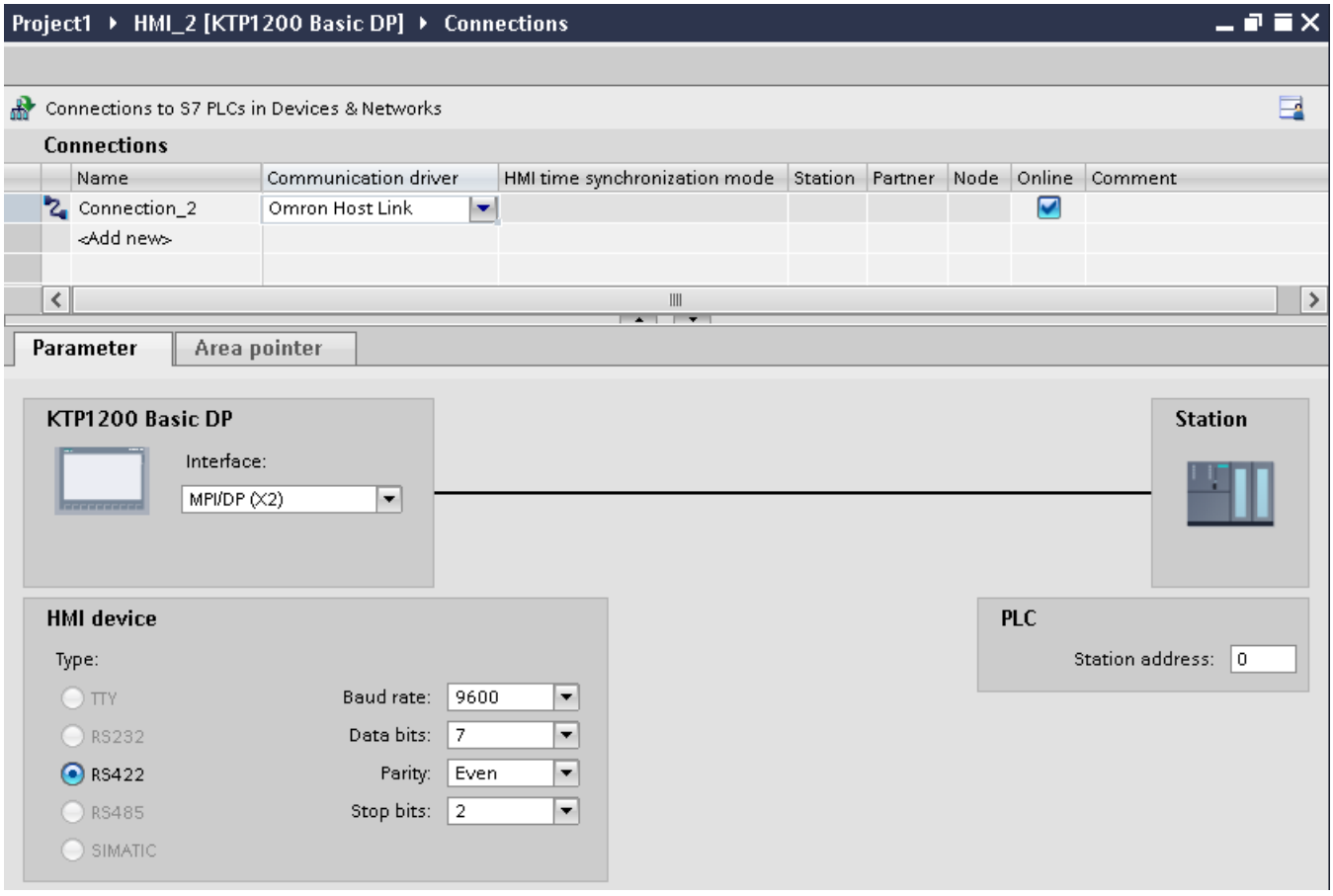
- A project is open.
- An HMI device has been created.

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 1032)

Parameters for the connection (Omron Host Link) (Page 1034)

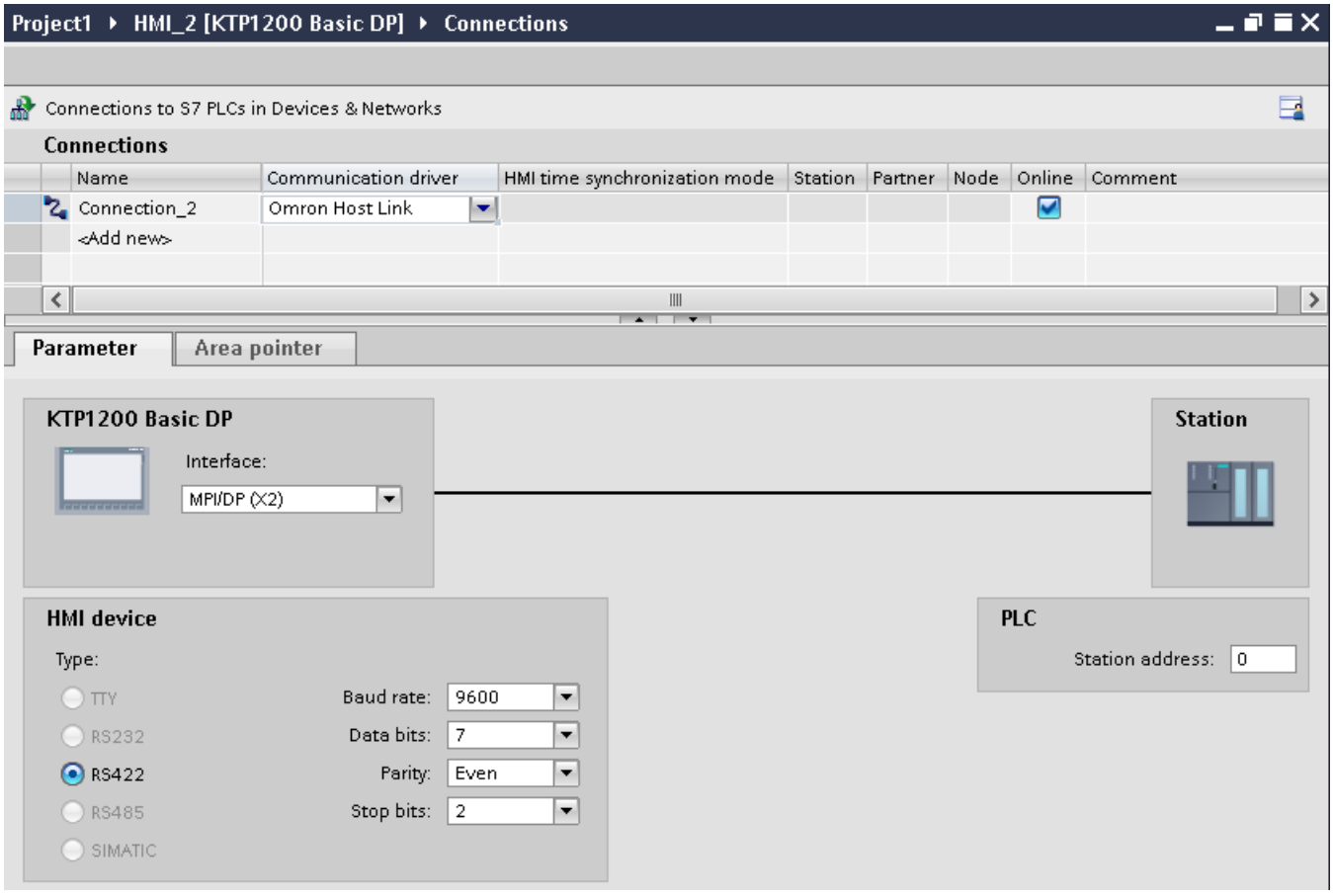
Optimizing the configuration (Page 1045)

Parameters for the connection (Omron Host Link)

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 1032)

Configuring a connection via Omron Host Link (Page 1032)

Connecting HMI device to PLC

Connections via Omron Host Link

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 1032)

Omron communication drivers (Page 1032)

Communication types (Page 1036)

Communication types

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:

- 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

Parameters for the connection (Omron Host Link) (Page 1034)

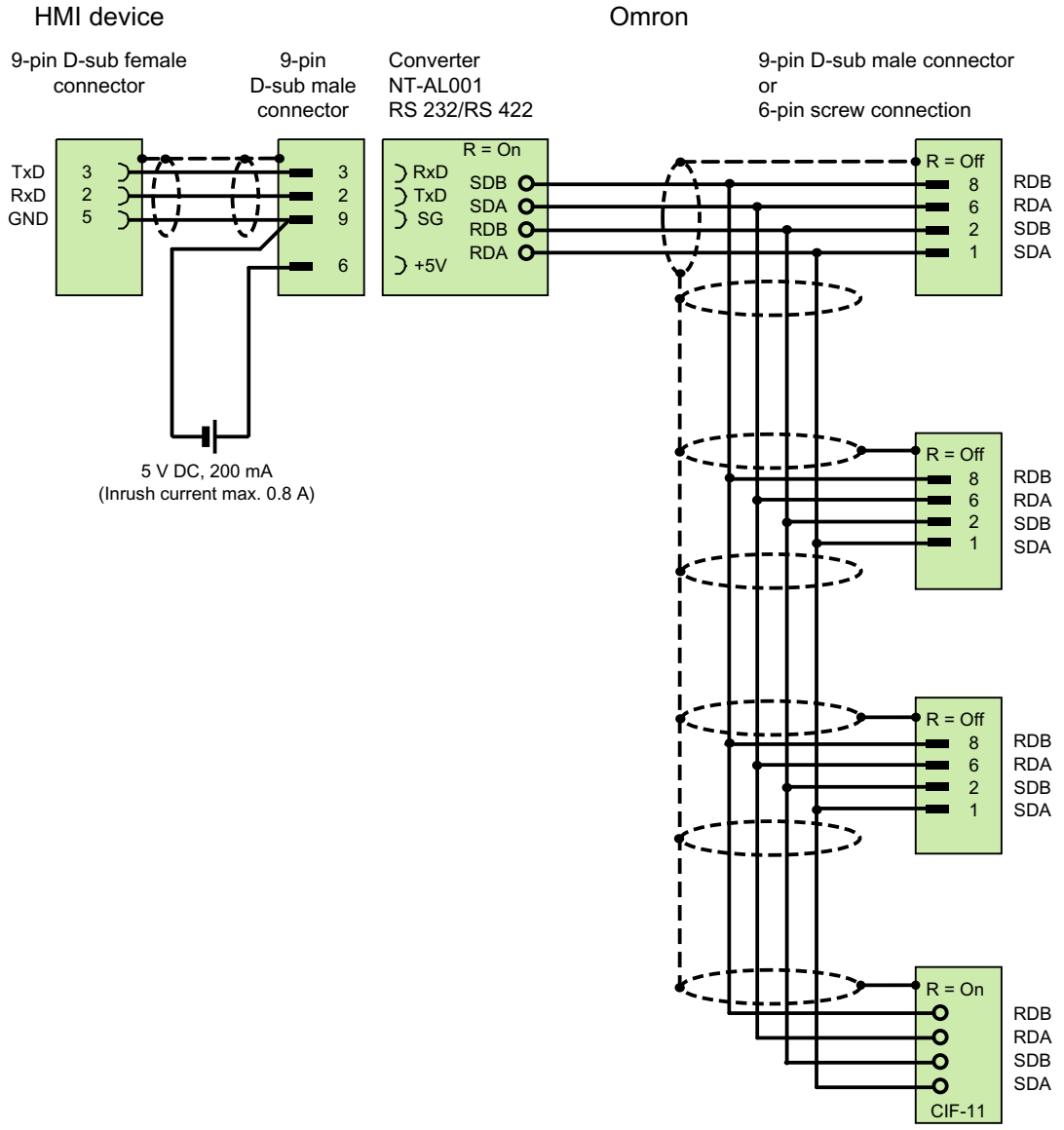
Omron communication drivers (Page 1032)

Connections via Omron Host Link (Page 1036)

Connection cable for Omron Host Link

Connecting cable MP1, RS-232, over converter, for Omron

Multipoint cable 1: MP/TP/PC > PLC



[1] Inrush current max. 0.8 A

shielded, max. length 500 m

See also

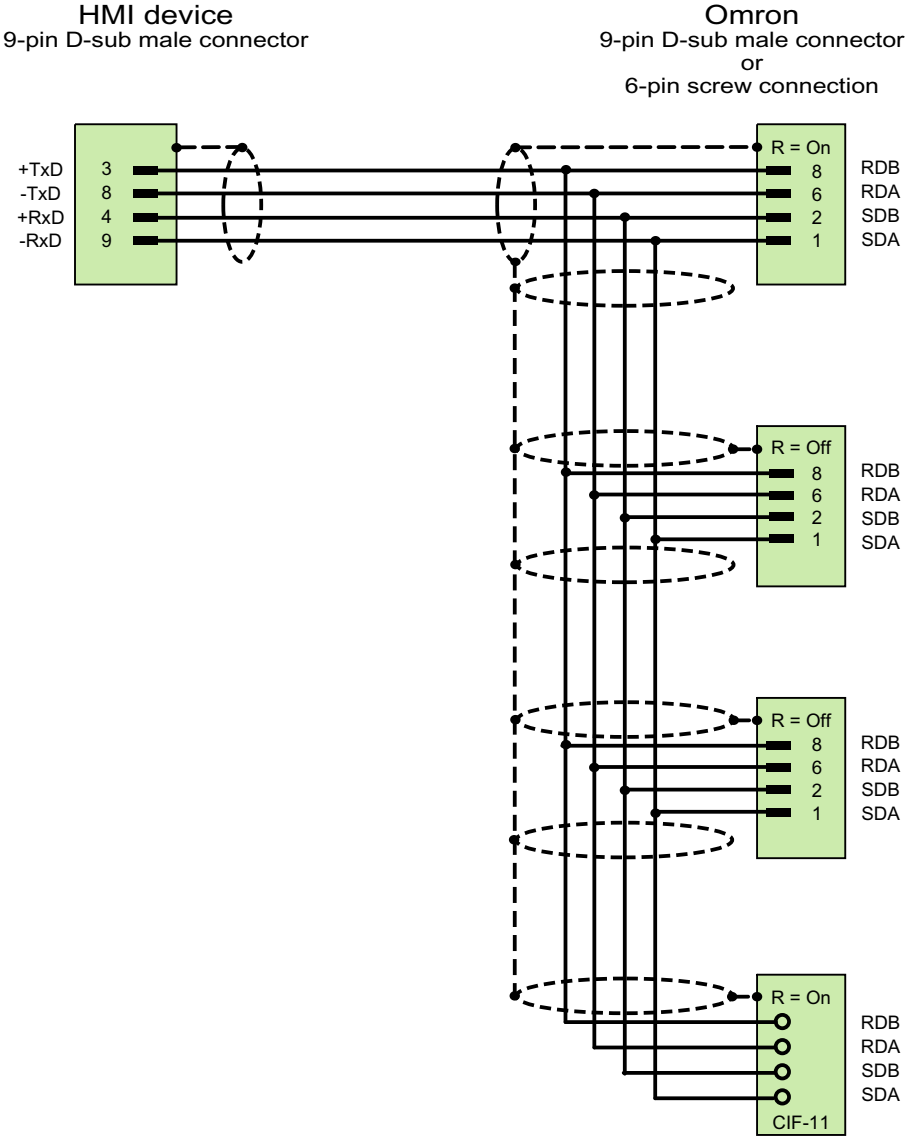
Connections via Omron Host Link (Page 1036)

Configuring a connection via Omron Host Link (Page 1032)

Omron communication drivers (Page 1032)

Connecting cable MP2, RS-422, for Omron

Multipoint cable 2: RS422, MP/TP/PC > SPS_



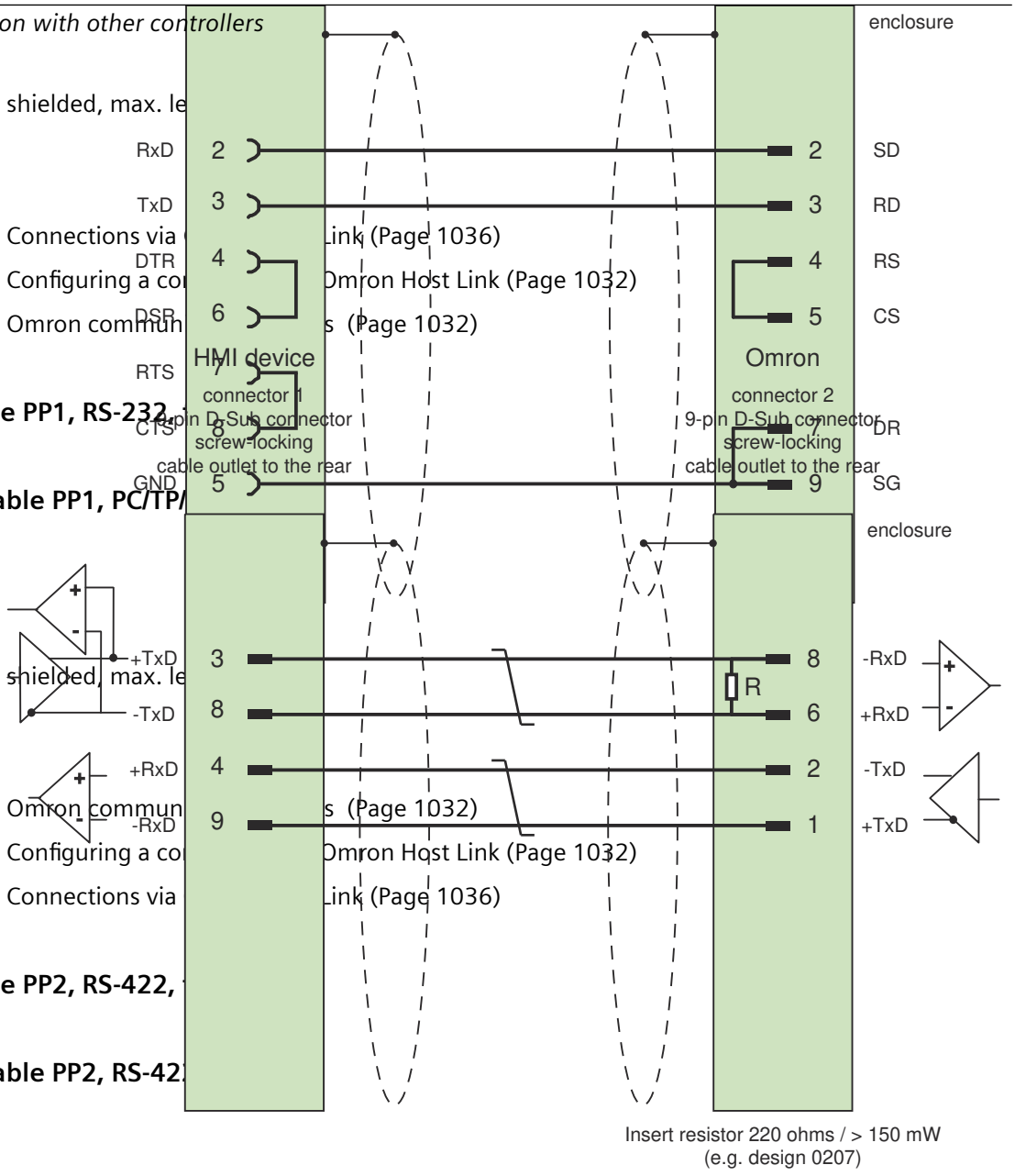
shielded, max. le

See also

Connections via Link (Page 1036)
Configuring a co Omron Host Link (Page 1032)
Omron commun s (Page 1032)

Connecting cable PP1, RS-232

Point-to-point cable PP1, PC/TP/



shielded, max. le

See also

Omron commun s (Page 1032)
Configuring a co Omron Host Link (Page 1032)
Connections via Link (Page 1036)

Connecting cable PP2, RS-422,

Point-to-point cable PP2, RS-42

shielded, max. length 500 m

See also

Connections via Omron Host Link (Page 1036)
Configuring a connection via Omron Host Link (Page 1032)
Omron communication drivers (Page 1032)

Performance features of communication

Permissible data types for Omron Host Link

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
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

Operand type old PLC	Operand type CS and CJ PLC
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

Configuring a connection via Omron Host Link (Page 1032)

Omron communication drivers (Page 1032)

Supported CPU types for Omron Host Link (Page 1042)

Addressing in Omron Host Link (Page 1043)

Address areas for Omron Host Link (Page 1044)

Supported CPU types for Omron Host Link

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 1032)

Configuring a connection via Omron Host Link (Page 1032)

Permissible data types for Omron Host Link (Page 1041)

Addressing in Omron Host Link (Page 1043)

Address areas for Omron Host Link (Page 1044)

Addressing in Omron Host Link

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 1032)

Configuring a connection via Omron Host Link (Page 1032)

Permissible data types for Omron Host Link (Page 1041)

Supported CPU types for Omron Host Link (Page 1042)

Address areas for Omron Host Link (Page 1044)

Address areas for Omron Host Link

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 1032)

Configuring a connection via Omron Host Link (Page 1032)

Permissible data types for Omron Host Link (Page 1041)

Supported CPU types for Omron Host Link (Page 1042)

Addressing in Omron Host Link (Page 1043)

Commissioning components

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.

See also

Omron communication drivers (Page 1032)

Optimizing the configuration (Page 1045)

Optimizing the configuration

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.

See also

Omron communication drivers (Page 1032)

Commissioning components (Page 1045)

Configuring a connection via Omron Host Link (Page 1032)

Data exchange

Area pointers for Omron

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".

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

Omron communication drivers (Page 1032)

Trends

General information on trends

Trends

A trend is the graphical representation of one or more values from the PLC. The value is read out either time- or bit-triggered, depending on the configuration.

For additional information see:

AUTOHOTSPOT

Note

The value is read out time-triggered for Basic Panels.

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.

Bit-triggered trends

Through a trigger bit set in the trend request tag, the HMI device either reads in a trend value or an entire trend buffer. This setting is defined in the configuration. Bit-triggered trends are normally used to represent fast 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 "HMI tags" editor and connected to trend areas during configuration. The HMI device and PLC then communicate with each other via 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)

See also

Trend request and trend transfer (Page 1048)

Restrictions to the trend control (Page 1049)

Trend request and trend transfer

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 closing the screen, the HMI device resets the relevant bits in the trend request area.

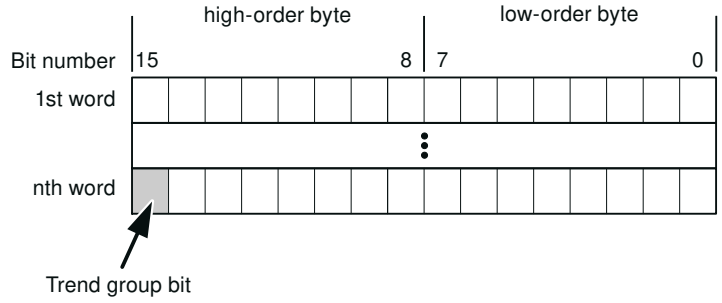
The trend request area can be used for evaluation purposes in the PLC to determine which trend is currently being displayed on the HMI device. Trends can also be triggered without evaluation of the trend request area.

Trend transfer area 1

This area is used to trigger trends. You must set the bit assigned to the trend in the trend transfer area and set the trend group bit in your control program. 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.

The PLC writes to Buffer 2 while the HMI device reads values from Buffer 1, and writes to Buffer 1 when the HMI device is reading Buffer 2. This prevents the PLC from overwriting trend values while the trend is being read by the HMI device.

See also

General information on trends (Page 1047)

Restrictions to the trend control (Page 1049)

Restrictions to the trend control

For Omron Host Link communication drivers

Tags of "Operand type" "DM", "I/O", "HR", "AR" or "LR" are permitted.

They must be data type "UInt", "Int" or an array tag of data type "UInt", "Int". You assign a bit to a trend during configuration. This sets a defined bit assignment for all areas.

See also

General information on trends (Page 1047)

Trend request and trend transfer (Page 1048)

Alarms

Configuring alarms

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

Omron communication drivers (Page 1032)

Acknowledgment of alarms (Page 1050)

Acknowledgment of alarms

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

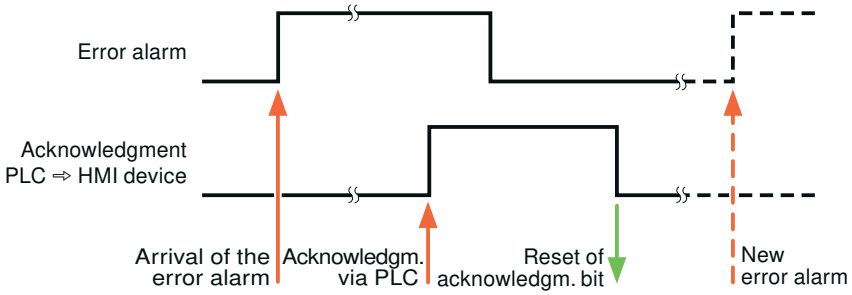
Acknowledgment by the PLC

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.

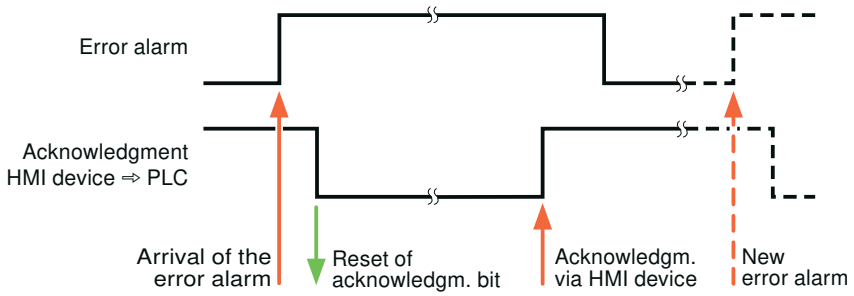
If a discrete alarm subject to acknowledgment is acknowledged by the HMI device, then the corresponding bit in the assigned acknowledgment tag will be set. The entire acknowledgment tag is then written to the PLC by the HMI device. This allows the PLC to recognize that a certain alarm message has been 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 1049)

LED mapping

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 Comfort Panels	Panel PCs
0	0	Off	Off
0	1	Rapid flashing	Flashing

Bit n+ 1	Bit n	LED function	
		all Mobile Panels, all Comfort Panels	**Panel PCs**
1	0	Slow flashing	Flashing
1	1	On permanently	On permanently

2.19.5 Connection diagnostics

2.19.5.1 Operating principle

Introduction

By means of WinCC "Channel Diagnosis", you can quickly get an overview in Runtime of the status of the active connections. "Channel Diagnosis" outputs status and diagnostic information and at the same time acts as operator interface for configuring the diagnostic output:

- Output of status/statistics information of the communication, for example, in the process picture
- Text output to a logbook file for fault analysis, and troubleshooting by the Service
- Text output to a trace file to help the Hotline in pinpointing the cause of communication problems

You can start the diagnostics module as separate application via the Windows Start menu or insert it as an object in a process picture. Only the status information of channels that support the diagnosis is displayed.

Logbook file

A logbook file with the name <Name.log> is created by "Channel Diagnosis" for each configured connection. Important information and errors are displayed in this file. The contents depend on the communication driver.

The creation of the file and the displayed texts are not configurable.

The logbook file contains, for example, start messages and end messages, version information, information on communication errors.

Each entry in the file consists of a date stamp and time stamp, the flag name, and a description. The file is always saved immediately after an entry in order that, for example, all information is available after a voltage drop.

Trace file

A trace file, in which additional information and errors are displayed, with the name <Name.trc> can also be created for each configured connection. The use of the trace file is selectable in Runtime. When the function is activated a message appears that the runtime of the communication is affected by this.

Each entry in the trace file contains a time stamp with appended flag name and description.

When the trace function is activated all logbook outputs are also written to the trace file.

The text output in the trace file is used to support the hotline in pinpointing communication problems.

Note

The outputs in trace and logbook files are exclusively in English.

Both files are stored in the "Diagnosis" directory in the WinCC directory structure.

The current counter values are not output in these files.

Performance tags

Alternatively, you have the option of using performance tags to diagnose connections.

See also

Checking connections with performance tags (Page 1060)

Diagnostics with "WinCC Channel Diagnosis" (Page 1054)

Diagnostics with the Channel diagnostics view (Page 1056)

Description of the log file (Page 1057)

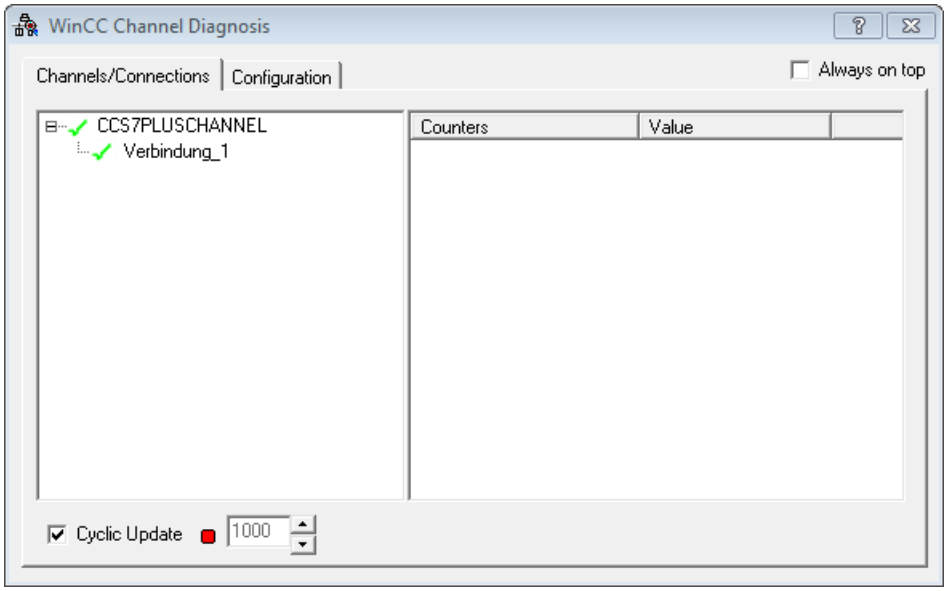
Configuring the trace function (Page 1058)

2.19.5.2 Diagnostics with "WinCC Channel Diagnosis"

Introduction

The channel diagnostics "Chanel Diagnosis" can also be started separately from WinCC as application via the Windows Start menu.

"Channel Diagnostics" is therefore permanently available and not dependent on the selection of a process picture, as when you use "WinCC Channel Diagnostics Control". The status information is only displayed by "Channel Diagnosis" when WinCC is in Runtime.



Requirements

- A connection and a tag are configured.
- The WinCC project is in Runtime.

Procedure

1. In the Windows Start menu, select the entry "WinCC Channel Diagnostics" under "All Programs > Siemens Automation > Runtime Systems > WinCC Runtime Professional".
2. "WinCC Channel Diagnosis" opens.
3. The configured connections appear in the "Channels/Connection" tab. If there is no WinCC project in Runtime, the message "No connection to WinCC" appears.
4. Click a connection. The status information for this connection is displayed in the area on the right.
5. Switch to the "Configuration" tab. Select one of the displayed channels and configure which error displays are to be entered in the associated log file.
6. For activating the trace function, you can find additional information under "Configuration of the trace function".

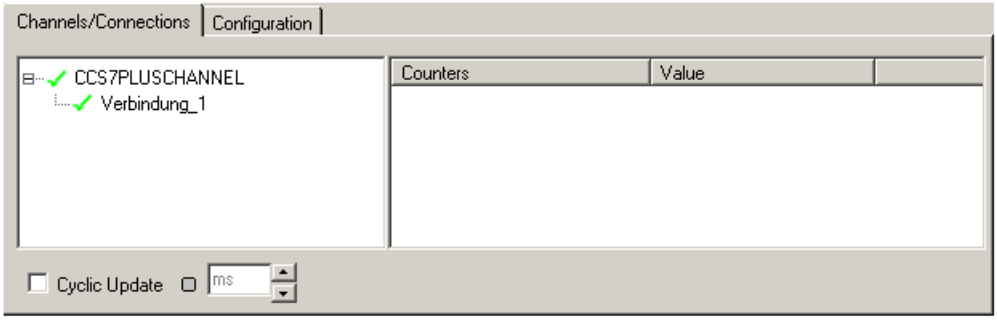
Note

"Channel Diagnosis" shows only status information on the channels that support the channel diagnostics.

2.19.5.3 Diagnostics with the Channel diagnostics view

Introduction

The status information of a connection can also be displayed with the object "Channel diagnostics view" in a process screen.



Requirements

- A connection and a tag are configured.

Procedure

1. Open a screen.
2. Insert the "Channel diagnostics view" from the "Controls" object pane into the screen.
3. Save the screen.
4. Start Runtime.
5. Select the screen in which you have configured the Channel diagnostics view.
6. The configured connections are displayed in the "Channels/Connection" tab.
7. Check the icons in front of the connection name. If the status of a channel, and connection are OK, a green check mark is displayed in front the associated entry.
8. If there is no green check mark in front of the name of channel and connection, click a connection. The status information for this connection is displayed in the area on the right.
9. Switch to the "Configuration" tab. Select one of the displayed channels and configure which error displays are to be entered in the associated log file.
10. Check the channel-specific log file. Open the log file from the "Siemens\Automation\SCADA_2007\WinCC\Diagnose" folder in a text editor.
11. Check the latest entries with the "ERROR" flag.
12. Activate the trace function.
    Contact Customer Support if you are unable to locate the error, even after checking the log file.

See also

Description of the log file (Page 1057)

Configuring the trace function (Page 1058)

2.19.5.4 Description of the log file

Introduction

State changes and errors are entered in the log file. These entries can be used to further analyze a communication problem.

Each entry in the file contains a date stamp and time stamp with information on the type and description.

Example of a logbook entry:

2018-11-26 11:44:16,110 INFO Log starting ...

2018-11-26 11:44:16,110 INFO | LogFileName : C:\Program Files (x86)\Siemens\Automation\SCADA-RT_V11\WinCC\Diagnose\CCS7PLUSCHANNEL_01.LOG

2018-11-26 11:44:16,110 INFO | LogFileCount : 3

2018-11-26 11:44:16,110 INFO | LogFileSize : 1400000

2018-11-26 11:44:16,110 INFO | TraceFlags : c4000000

2018-11-26 11:44:16,110 INFO | Process-ID : 0000090c

2018-11-26 11:44:16,110 INFO CR_UG5_CLP: No or uncompleted register of channel specific tags

2018-11-26 11:44:16,110 INFO FM_UG4_CLP: No or uncompleted register of channel specific tags

2018-11-26 11:44:16,298 INFO CR_UG5_CLP: connecting...

2018-11-26 11:44:16,298 INFO FM_UG4_CLP: connecting...

2018-11-26 11:44:25,939 INFO CR_UG5_CLP: Client received a 'logged_out' message on session: 'Session: 1'

2018-11-26 11:44:25,939 ERROR CR_UG5_CLP: Session: 1:Login() failed, errorCode = 0x800015210000008d

2018-11-26 11:44:25,939 ERROR CR_UG5_CLP: Open failed, errorCode = 0x6

2018-11-26 11:44:25,940 INFO FM_UG4_CLP: Client received a 'logged_out' message on session: 'Session: 1'

2018-11-26 11:44:25,940 ERROR FM_UG4_CLP: Session: 1:Login() failed, errorCode = 0x800015210000008d

2018-11-26 11:44:25,940 ERROR FM_UG4_CLP: Open failed, errorCode = 0x6

Explanations

Message text	Meaning
LogFileName : C:\Program Files (x86)\Siemens\Automation\SCADA-RT_V11\WinCC\Diagnose\CCS7PLUSCHANNEL_01.LOG	Name of the log file with path information
LogFileCount : "n"	Number of log files of the channel
LogFileSize : "x"	Size of the individual log files in bytes
TraceFlags : c4000000	Display of the flags used by the trace function, as hexadecimal number

See also

Diagnostics with the Channel diagnostics view (Page 1056)

2.19.5.5 Configuring the trace function

Introduction

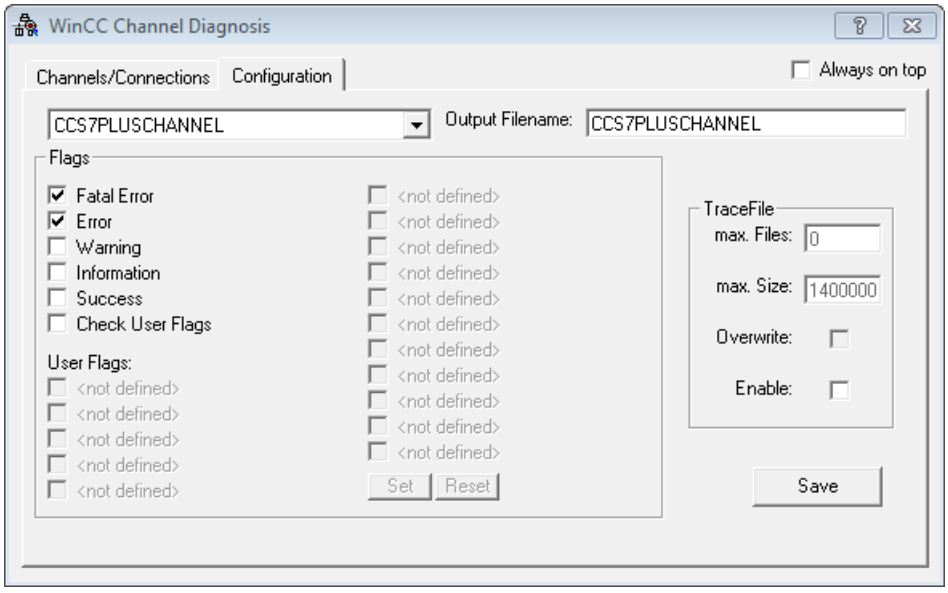
Additional status and error information on the communication is entered in the trace file.

Requirements

- A connection and a tag are configured.
- The WinCC project is in Runtime.

Procedure

1. Start "WinCC Channel Diagnostics" via the Start menu.
2. Select the required connection from the "Configuration" tab.



3. In the area "Flags", activate the status and error displays that are to be output in the trace file.
4. If the "User Flags" are to be output in the trace file, select the relevant check box. The number and meaning of the "User Flags" depends on the connection.
5. All "User Flags" can be set or reset using the "Set" and "Reset" buttons.
6. Select "Enable" in the "Trace File" area. Other fields in this area are hereby activated.
7. Enter the maximum number of trace files in the "max. Files" field.
8. Specify the size of the individual trace files in the "max. Size" field.
9. Activate the field "Overwrite", if existing trace files are to be overwritten in recycle mode when the maximum file number and size are reached.
10. Press the "Save" button to save the settings and activate the changes.

Overview of standard flags

Note

The text output in the trace file is used to support the hotline in pinpointing communication problems. Only a description of the general flags is therefore provided below.

Flag	Meaning
Fatal Error	Serious errors (such as, user access required)
Error	Errors (for example, frame errors)
Warning	Warning (for example, note on checksum error)

Flag	Meaning
Information	Information (for example, function call)
Success	Successful execution (for example, completion of a function call)
Check User Flags	Releases the check boxes of the "User Flags" for operation

See also

Diagnostics with the Channel diagnostics view (Page 1056)

2.19.5.6 Checking connections with performance tags

WinCC provides the following system tags to analyze a communication channel:

- @PRF_DMRT_CHNCON_... (Tags)
- @PRF_ALGRT_CHNCON_... (Alarm logging)

You can evaluate the time response of a connection with these performance tags.

Note

Connection-specific performance tags

If a connection is configured, connection-specific performance tags are created in the database.

These tags cannot be viewed in the "System tags" tab but can be used like other performance tags.

When you link the connection-specific performance tags to objects, the tag assignment will have a red background in the engineering system. Access in Runtime still works.

Use the @<...>@ConnectionStateEx system tags to determine the status of a connection.

Creating performance tags

As soon as you create a new connection, the associated performance tags are created.

If you rename the connection, the performance tags are also renamed automatically.

Types of performance tags

Performance tags can be assigned to the following data types:

Tags	Data type	Access	Description
Relative tags	Floating-point number 64-bit IEEE 754	Read	Values which apply relatively to the time of reading, e.g. currently pending values or values per second. The reset tag does not have an influence on these values. The tag name ends in: • ..._PENDING • ..._SECOND Update cycle: 1 second
Counter tags	Floating-point number 64-bit IEEE 754	Read	Absolute values since Runtime activation You can reset the value to "0" by using the reset tag. The tag name ends in: • ..._AVERAGE • ..._PEAK • ..._TOTAL Update cycle: 1 second
Reset tags	Unsigned 32-bit value	Read Write	You can set the value of the reset tags via scripts, for example: • 0: Disabled • 1: The value of all associated counter tags is reset to "0". The value of the reset tag itself is also reset to "0". The tag name ends in: • ...RESET

Overview of performance tags

System tag [1]	Description
Alarm logging	
@PRF_ALGRT_RESET	The reset tag resets the values of the following performance tags: • @PRF_ALGRT_CHNCON_..._AVERAGE • @PRF_ALGRT_CHNCON_..._PEAK The @PRF_ALGRT_... counter tags without relation to the communication driver are also reset.
@PRF_ALGRT_CHNCON_<...>_ALARMS_PER_SECOND [2]	Number of alarms generated per second that are transferred via the connection
@PRF_ALGRT_CHNCON_<...>_ALARMS_PER_SECOND_AVERAGE	Average number of alarms per second
@PRF_ALGRT_CHNCON_<...>_ALARMS_PER_SECOND_PEAK	Maximum number of alarms per second

System tag [1]	Description
Tags	
@PRF_DMRT_CHNCON_<...>_RESET	The reset tag resets the values of the following performance tags: • @PRF_DMRT_CHNCON_<...>..._TOTAL • @PRF_DMRT_CHNCON_<...>_RESET The reset applies to all counter tags that were created for the same connection.
@PRF_DMRT_CHNCON_<...>_TAG_READ_BYTES_PER_SECOND [2]	Bytes read/second Bits are rounded up to a byte. Metadata is not included, e.g. time stamp or SetValue Callback data.
@PRF_DMRT_CHNCON_<...>_TAG_READ_BYTES_TOTAL	Bytes read since activation of Runtime Bits are rounded up to a byte. Metadata is not included, e.g. time stamp or SetValue Callback data.
@PRF_DMRT_CHNCON_<...>_TAG_READS_PENDING	Started read requests that are not yet complete A constantly rising value indicates a system overload. Possible cause: • A data source or connection is not processing the read requests fast enough because it is overloaded or blocked.
@PRF_DMRT_CHNCON_<...>_TAG_READS_PER_SECOND [2]	Tags read/second
@PRF_DMRT_CHNCON_<...>_TAG_READS_TOTAL	Tags read since activation of Runtime
@PRF_DMRT_CHNCON_<...>_TAG_WRITES_PENDING	Started write requests that are not yet complete A constantly rising value indicates a system overload. Possible cause: • A data source or connection is not processing the write requests fast enough because it is overloaded or blocked.
@PRF_DMRT_CHNCON_<...>_TAG_WRITES_PER_SECOND 2[2]	Tags written/second
@PRF_DMRT_CHNCON_<...>_TAG_WRITES_TOTAL	Tags written since activation of Runtime
@PRF_DMRT_CHNCON_<...>_TAG_WRITTEN_BYTES_PER_SECOND [2]	Bytes written/second Bits are rounded up to a byte. Metadata is not included, e.g. time stamp or SetValue Callback data.
@PRF_DMRT_CHNCON_<...>_TAG_WRITTEN_BYTES_TOTAL	Bytes written since activation of Runtime Bits are rounded up to a byte. Metadata is not included, e.g. time stamp or SetValue Callback data.

1) <...> stands for the name of the communication connection, e.g. "@PRF_DMRT_CHNCON_S7-417_TAG_READS_PER_SECOND".

2) The information "PER_SECOND" relates to the last second before the tag update.

See also

Setting up switch on/switch off of a connection in Runtime (Page 138)

Index

:

:PROFIBUS DP
    Direct key, 843

A

A&E Server, 45
    Mapping of the WinCC message system, 45
Absolute Addressing
    of a tag, 145
Access level, 206, 233, 620
Accessibility
    WinCC OPC server, 32
Accessible devices, 214, 326, 426, 691
Acknowledgment theory, 47
Acquisition cycle
    Area pointer, 151
Active
    Area pointer, 151
Address
    Area pointer, 151
    Mitsubishi, 971
    Omron Host Link, 1043
Addressing
    Allen-Bradley, 919, 943
    Allen-Bradley Ethernet IP, 914
    Ethernet/IP, 919, 943
    MPI, 458
    SIMATIC S7 1500, 272
    Tags, 272
    Valid, 921
Alarm message
    Acknowledgment by the PLC, 261, 372, 481, 647, 727, 795, 815, 955, 993, 1029, 1051
    Acknowledgment on the HMI device, 261, 372, 481, 648, 727, 796, 816, 956, 993, 1030, 1051
    Configure acknowledgment, 260, 371, 480, 647, 726, 795, 815, 955, 992, 1029, 1050
Alarms
    Configuring, 259, 370, 479, 646, 725, 794, 814
    Data types, 259, 370, 479, 646, 725, 794, 814
    Modicon Modbus, 1028
    Non-integrated connection, 1028
    Restriction, 992
Allen-Bradley, 901, 933
    Allen-Bradley DF1 communication driver, 933
    Analog alarm, 955
    Area pointer, 949
    Basic Panel, 161
    Communication drivers, 905
    Data type, 953, 955
    DF1, 905
    Ethernet/IP, 921
    EtherNet/IP, 905
    Mitsubishi, 992
Allen-Bradley DF1
    Configuring a connection, 928
    Connection, 928, 932
    Connection parameters, 930
    CPU type, 942
    KF2 module, 933
    KF3 module, 933
    Valid data type, 941
Allen-Bradley Ethernet IP
    Address multiplexing, 918
    Addressing, 914
    Addressing type, 916
Allen-Bradley EtherNet/IP
    Configuring a connection, 905
    Connection, 905, 910
    Connection parameters, 907
    Data type, 911
Analog alarm
    Allen-Bradley, 955
    Mitsubishi, 992
    Omron, 1049
API error messages, 561
Area pointer, 149, 238, 349, 459, 624, 705, 774
    Acquisition cycle, 151
    Active, 151
    Address, 151
    Allen-Bradley, 949
    Basic Panel, 164
    Comment, 151
    Connection, 150, 156
    Connections editor, 151
    Data record, 248, 359, 469, 635, 715, 783
    Date/time, 349, 351, 707, 708
    Job mailbox, 246, 357, 466, 632, 712, 781
    Length, 151
    Mobile Panel, 179
    Modicon Modbus, 1024
    PLC tag, 151
    Pointer name, 151
    Project ID, 244, 355, 465, 631, 711, 779

Screen number, 239, 354, 460, 625, 706, 774
Tab, 151
Trends, 257, 368, 477, 643, 723, 791, 811, 950, 988, 1025, 1047
Area pointers
Configuring, 153
Coordination, 243, 352, 464, 630, 710, 778
Creating, 153
Date/time, 240, 461, 626, 775
Date/time PLC, 241, 462, 628, 777
Length, 154
Mitsubishi, 988
Omron, 1046
Asynchronous
Transferring data, 249, 360, 470, 636, 716, 789
Automation system, 92, 121
Setting up, 93
Autonegotiation
Disabled, 212, 324, 425, 689

B

Basic Panel
Area pointer, 164
Communication drivers, 161
ETHERNET, 163
IF1B, 163
Interface, 163
Mitsubishi, 161
Modicon Modbus, 161
Omron, 161
OPC, 161
Basic settings
OPC, 32
Basics
HMI HTTP protocol, 853
Bit-triggered trends, 257, 368, 477, 643, 723, 791, 811, 950, 988, 1025, 1047
Boundaries, 214, 326, 426, 691
Byte assignment
Direct key, 835

C

Cabling rules (PROFINET), 212, 324, 425, 689
Certificate, 82, 867
Importing on HTTP client, 867
Installing on devices, 867
Installing under Windows XP, 867
Change the CP type of a channel unit, 523
Change the logical device name of a channel unit, 522
Change word order
Modicon MODBUS TCP/IP, 1000
Changing the PROFIBUS connection
Operating mode, 826
Changing the PROFINET connection
Operating mode, 825
Channel diagnosis
Check channel, 1054
Channel diagnostics
Channel OPC, 87
Channel with, 1054
Diagnostics option, 87
Info flag of a logbook entry, 88
Log file, 88
Channel DLL SIMATIC S7-300/400, 515
Channel OPC, 87
Diagnostics option, 87
Checking the AS with STEP 7, 519
Checking the bus system, 518
CJ1, 1042
CJ2, 1042
Client
Configuring, 858
Parameters, 859
Tags, 863
Client certificate, 82
Comfort Panel
Area pointer, 174
Interface, 173
Comment
Area pointer, 151
Communication, 91
Basics, 92
between HMI devices, 855
Configuring, 881
Definition, 91
ET 200 CPU, 670
OPC, 870
Routing, 877, 879, 881
Runtime settings, 856, 871
S7 1200, 298
S7 1500, 185
S7 200, 763
S7 300, 401
S7 400, 401
SIMATIC LOGO!, 805
SIMATIC S7-1500 Software Controller, 582
Third-party drivers, 901
using area pointers, 97

Communication drivers, 98
Allen-Bradley, 905
Basic Panel, 161
Mitsubishi, 958
Modicon Modbus, 995
Omron, 1032
Communication in the security mode
Configuring, 17
Communication network, 121
PROFINET, 122
Communication partners
HMI device, 92
Networking, 126
PLC, 92
SIMATIC S7, 279, 390, 502, 743, 756, 769, 772
Communication types
Allen-Bradley, 910
Connectable PLCs, 1003, 1016
Enabled, 965, 983, 1002, 1015, 1036
EtherNet/IP, 910
Restriction, 1002, 1015
Compact, 1005, 1020
CompactLogix, 913
Compatibility, 14, 30
Components
Commissioning, 926, 948, 977, 986, 1009, 1023, 1045
Configuration file
File location, 75
Structure, 75, 76
Configuration PC
WinCC Runtime, 191, 220, 307, 335, 410, 435, 451, 590, 608, 676
Configure acknowledgment
Alarm message, 260, 371, 480, 647, 726, 795, 815, 955, 992, 1029, 1050
Configure alarms
Allen-Bradley, 954
Data types, 992
Non-integrated connection, 991
Special considerations, 991
Configure the correct AS network address, 518
Configuring
Alarms, 259, 479, 646, 725, 794, 814
Connections, 272, 275, 281, 383, 385, 391, 492, 495, 497, 505, 659, 662, 737, 746, 751, 763, 806
HMI connection, 191, 194, 305, 307, 407, 410, 590, 593, 676, 679
HMI device as OPC DA server, 19
HMI device as OPC DA Server, 20
OPC XML DA server in the OPC XML Manager, 25
Server, 856
WinCC OPC A&E server, 55
WinCC OPC Client, 23, 80
WinCC OPC UA Server, 78
Configuring a connection
Allen-Bradley DF1, 928
Allen-Bradley EtherNet/IP, 905
Mitsubishi FX, 979
Mitsubishi MC TCP/IP, 959
Modicon Modbus RTU, 1010
Modicon Modbus TCP, 996
Omron Host Link, 1032
OPC, 871
Configuring a tag
HTTP client, 861
HTTP server, 858
Configuring the network with Ethernet, 203, 315, 421
Creating private subnets, 204, 316, 422
Linking networks, 205, 317, 423
Relationship between IP address and subnet mask, 204, 316, 422
Setting the IP address, 203, 315, 421
Setting the subnet mask, 203, 315, 421
Connecting
PLC, 910, 964, 982, 1001, 1036
Connecting cable
6XV1440 - 2P for Mitsubishi PG protocol, 984
Allen-Bradley cable 1784-CP10, 938
Mitsubishi FX, 982
Connecting cable 9-pole
Sub D, RS 422, 937
Connecting cables
Point-to-point cable 1,
Point-to-point cable 2, 1018
Connection, 96
Allen-Bradley DF1, 928, 932
Allen-Bradley EtherNet/IP, 905
Area pointer, 150, 156
Configuring, 127, 131, 135, 413, 905, 928, 959, 979, 996, 1010, 1032
Creating, 127, 131, 135
Highlighting, 129
Integrated, 127
integrated connection, 145
Mitsubishi MC TCP/IP, 959
Modicon, 1014
Modicon Modbus RTU, 1010
Modicon Modbus TCP, 996
offline, 245, 356, 466, 632, 712, 780
Omron communication drivers, 1046
Omron Host Link, 1032

Parameters, 907, 930, 962, 998, 1012, 1034
Password, 210, 226, 237, 321, 340, 348, 602, 614, 624
Routed, 135
S7 200, 763, 806
Table, 126
Connection boxes
For IRT communication, 891
For media redundancy, 895
For Mobile Panels, 895
Connection cable
Allen-Bradley cable 1747-CP3, 939
Allen-Bradley cable 1761-CBL-PM02, 939
Modicon, 1014
Multipoint cable 1:MP/TP/PC, 1038
Multipoint cable 2:RS422, MP/TP/PC, 1039
Omron Host Link, 1036
Point-to-point cable PP2 for Omron, 1040
Connection information
Diagnostics, 109
Connection parameters
Allen-Bradley DF1, 930
Allen-Bradley EtherNet/IP, 907
Mitsubishi MC TCP/IP, 962
Modicon Modbus RTU, 1012
Modicon Modbus TCP, 998
Omron Host Link, 1034
Runtime Professional, 284
Connections
Allen-Bradley DF1, 933
Configuring, 272, 275, 281, 383, 385, 391, 492, 495, 497, 505, 659, 662, 737, 746, 751, 763, 806
Editor, 272, 275, 281, 383, 385, 391, 492, 495, 497, 505, 659, 662, 737, 746, 751
ControlLogix, 913
Coordination, 97
CP1, 1042
CPM, 1042
CPU type
Allen-Bradley DF1, 942
Allen-Bradley EtherNet/IP, 913
FX3 series, 971
Mitsubishi FX, 985
Mitsubishi MC TCP/IP, 971
Modicon Modbus RTU, 1020
Modicon Modbus TCP, 1005
Omron Host Link, 1042
Create
Connection, 300
Creating
Connection, 127, 131, 135, 187, 403, 586, 672
Connection to an OPC UA Server, 23, 81
Connection to OPC DA server, 22
OPC connection, 79
Creating a new project, 516
CS1, 1042
Cyclic operation, 504, 773, 810
Cyclical read services, 510

D

Data areas
Area pointer, 153
Reading, 153
Writing, 153
Data exchange, 91
Tags, 96
Trends, 257, 368, 477, 643, 723, 791, 811, 950, 988, 1025, 1047
using area pointers, 97
Data record, 97
Data type, 86
Allen-Bradley, 953, 955
Allen-Bradley EtherNet/IP, 911
Discrete alarm, 955, 992, 1049
Internal tag, 148
Mitsubishi, 992
Mitsubishi FX, 984
Mitsubishi MC TCP/IP, 969
Modicon Modbus RTU, 1019
Modicon Modbus TCP, 1004
Omron, 1049
Omron Host Link, 1041
OPC UA, 86
Trend, 953
Valid, 911, 969, 984, 1004, 1019, 1041
Data types
ET 200 CPU, 729
Mitsubishi FX, 991
Mitsubishi MC TCP/IP, 990
Modicon Modbus RTU, 1028
Modicon Modbus TCP/IP, 1027
OPC DA, 873, 875
OPC UA, 875
OPC XML DA, 874, 876
S7 1500, 270, 658
S7 200, 804
S7 300/400, 491
S7-1200 V1, 382
S7-300/400, 490
Trends, 259, 370, 479, 645, 725, 793, 813
Valid, 270, 382, 490, 491, 658, 729, 804, 873, 874, 875, 876

Date/time, 97
Area pointer, 349, 707
Date/time PLC, 97
Deleting
OPC XML DA server in the OPC XML Manager, 25
Device
Inserting, 94
Device dependency
S7 200, 797
S7 300/400, 483
SIMATIC LOGO!, 816
SIMATIC S7-1500 V1.0, 263, 649
Device information
Diagnostics, 109
Device view
Hardware and network editor, 111
Devices
Connecting, 96
Networking, 96, 126
Devices & Networks, 96, 126
Devices and networks, 96, 126
Connections, 127, 135
Devices & Networks, 332
HMI connections, 127, 135
Diagnostics
Connection, 109
Connection information, 109
Device, 109
Device information, 109
Diagnostics option, 87
Channel OPC, 87
Direct key, 825, 832, 847
Assigning the inputs/outputs, 832, 847
Bit coding, 836
Byte assignment, 835
configuring, 827
KP 1500 Comfort, 842, 852
KP1200 Comfort, 841, 851
KP700 Comfort, 838, 849
KP900 Comfort, 840, 850
KTP400 Comfort, 838, 848
KTP400F Mobile, 833
PROFIBUS DP, 843, 845, 846
PROFINET IO, 828, 830, 831
TP1200 Comfort, 842, 852
TP1500 Comfort, 842, 853
TP1900 Comfort, 842, 853
TP2200 Comfort, 842, 853
TP700 Comfort, 839, 849
TP900 Comfort, 840, 851
Discrete alarm
Allen-Bradley, 955
Mitsubishi, 992
Omron, 1049
DP, 231, 344, 442, 618

E

Editing
OPC XML DA server in the OPC XML Manager, 25
Editor
Connection, 131
Connections, 272, 275, 281, 383, 385, 391, 492, 495, 497, 505, 659, 662, 737, 746, 751
Devices and networks, 129
Enable autonegation, 211, 323, 424, 688
End of discovery of accessible devices, 214, 326, 426, 691
End of sync domain, 214, 326, 426, 691
End of topology discovery, 214, 326, 426, 691
Error codes during connection fault, 527
Error codes for connection fault, 515
Error xx occurred in 'function name' function!, 515, 563
ET 200 CPU
Communication, 670
Data types, 729
Ethernet
Parameters, 280, 282, 388, 393, 395, 500, 507, 744, 757, 767, 809
ETHERNET
Basic Panel, 163
Ethernet/IP
Allen-Bradley, 921
Example
Communication, 879
External tags
Data exchange, 96

F

FAQs, 120
Functions
Updating tag value, 145
FX1 series, 985
FX2 series, 985
FX3
PLC, 966
FX3 series, 971

G

Global data block
Configuring, 154
Creating, 154

H

Hardware and network editor
Device view, 111
Network view, 104
Topology view, 114
Hardware catalog
Task card, 118
Hardware editor
Components, 102
Function, 101
Hardware catalog, 118
Inspector window, 116
Highlighting
Connection, 129
HMI connection, 96
Configuring, 191, 194, 305, 307, 407, 410, 590, 593, 676, 679
Creating, 187, 300, 403, 586, 672
MPI, 444, 448, 451
MPI parameters, 453
Password, 210, 226, 237, 321, 340, 348, 602, 614, 624
PC, 191, 220, 307, 335, 410, 435, 451, 590, 608, 676, 698
PROFIBUS, 216, 220, 223, 332, 335, 428, 432, 435, 603, 608, 611, 696, 698
PROFIBUS parameters, 225, 339, 437, 613, 703
PROFINET, 186, 187, 189, 191, 194, 299, 300, 303, 305, 307, 402, 403, 405, 407, 410, 585, 586, 588, 590, 593, 671, 672, 674, 676, 679
PROFINET parameters, 197, 310, 416, 595, 681
S7 1200, 300
S7 1500, 187
S7 300/400, 403
SIMATIC ET 200 CPU, 672
SIMATIC PC, 194, 223, 305, 332, 407, 432, 448, 593, 611, 679, 696
SIMATIC S7-1500 Software Controller, 586
WinCC RT Advanced, 189, 223, 303, 332, 335, 405, 432, 448, 588, 611, 674, 696, 698
WinCC RT Professional, 189, 220, 223, 303, 332, 335, 405, 432, 435, 448, 451, 608, 611, 674
WinCC Runtime, 189, 303, 405, 588, 674
WinCC RT Advanced, 220, 435, 451, 608
HMI connections
Devices & Networks, 127, 135
HMI device
as OPC client, 15
as OPC server, 16
MPI parameters, 455
PROFIBUS parameters, 227, 341, 439, 615, 700
PROFINET parameters, 199, 311, 417, 597, 683
Server, 856
HMI device on a PLC
Commissioning, 926, 948, 977, 986, 1009, 1023, 1045
HMI HTTP protocol, 853
Basics, 853
HTTP, 853
HTTP client
Configuring a tag, 861
Importing certificates, 867
HTTP connection
Client, 858
HTTP protocol
Timeout, 860
HTTP server
Configuring a tag, 858
Setting up the WinCC Internet connections, 864
HTTPS, 853, 855

I

IF1B
Basic Panel, 163
iNA960 messages, 543
Industrial Ethernet, 121
INFO flag, 88
Inspector window
Hardware and network editor, 116
Installing
OPC XML DA server, 35
Instance certificate, 82
Integrated
Connection, 127, 131
Integrated connection, 145
Interface
Basic Panel, 163
Comfort Panel, 173
Mobile Panel, 178
Parameters, 511
PLC, 513
Serial communication, 511
WinCC RT Advanced, 183

WinCC RT Professional, 183
WinCC Runtime Advanced, 183
WinCC Runtime Professional, 183
Internal
Ethernet port, 969
Internal error codes and constants, 543
Internal Ethernet port
Open settings, 969
Internal tags
Data exchange, 96
IP address
Reading out, 861
IP protocol, 313
iQ series, 971
IRT (Isochronous Realtime Ethernet), 884
IRT communication, 891

J

Job mailbox, 97
Transferring data, 253, 364, 473, 640, 719, 785

L

LED
Assignment, 262, 373, 482, 648, 728, 957, 994, 1031, 1052
Function, 262, 373, 482, 648, 728, 957, 994, 1031, 1052
Image, 262, 373, 482, 648, 728, 957, 994, 1031, 1052
LED function key
Bit assignment, 837
Length
Area pointer, 151
Area pointers, 154
License
OPC server, 30
LLDP (Link Layer Discovery Protocol), 214, 326, 426, 691
Log file, 88
Logbook entry, 88

M

Manuals, 120
Mapping rules (OPC A&E)
"Mode 1", 50
"Mode 2", 51
"Mode 3", 51
Media redundancy (MRP), 892, 894, 895, 898
Micro, 1005
MicroLogix, 913, 942
Mitsubishi, 901
Address, 971
Analog alarm, 992
Basic Panel, 161
Communication drivers, 958
FX, 958
TCP/IP, 958
Mitsubishi FX
Configuring a connection, 979
Connection, 979, 982
CPU type, 985
Data type, 984
Mitsubishi MC TCP/IP, 965, 983
Mitsubishi MC TCP/IP
Configuring a connection, 959
Connection, 959, 964
Connection parameters, 962
CPU type, 971
Data type, 969
Mobile Panel
Area pointer, 179
Interface, 178
OPC, 14
Modicon
Approved communication with Modbus RTU, 1015
Connection, 1014
Connection cable, 1014
Restrictions with Modbus RTU, 1015
Modicon M340, 1005
Modicon Modbus, 901
Basic Panel, 161
Communication drivers, 995
RTU, 995
TCP, 995
Modicon Modbus connection
Data types, 1029
Modicon Modbus RTU, 1020
Configuring a connection, 1010
Connection, 1010
Connection parameters, 1012
Data type, 1019
Modicon Modbus TCP, 1005
Configuring a connection, 996
Connection, 996, 1001
Connection parameters, 998
Data type, 1004
Modicon MODBUS TCP/IP
Change word order, 1000
Momentum, 1005, 1020

Monitor, 211, 323, 424, 688
Moving the view
Overview navigation, 105, 112, 115
MPI, 122
Addressing, 458
Connection, 497
HMI connection, 444, 448, 451
Network, 124
Network architecture, 125
Parameters, 453, 455, 457
S7 200, 503, 770
S7 300/400, 443, 444, 446, 448, 451
WinCC RT Advanced, 446
WinCC RT Professional, 446
MPI address
S7 300, 459
S7 400, 459
MPI communication
S7 300/400, 443, 446
MPI parameters
S7 300/400, 453, 455, 457
MRP (Media Redundancy Protocol), 892, 894
MRP domain, 895, 898
Multiplexing
Address multiplexing Allen-Bradley Ethernet IP, 918
Multi-point connection
Allen-Bradley DF1, 935, 936

N

Named connections, 413
Configuring a connection, 413
Network, 91
Ethernet, 122
MPI, 124
PPI, 125
PROFIBUS, 123
PROFINET, 122
Network architecture
PPI, 126
Network editor
Components, 102
Function, 101
Hardware catalog, 118
Inspector window, 116
Network view, 126
Hardware and network editor, 104
Networking
Communication partners, 126
Devices, 126
Non-integrated
Connection, 131

O

Omron, 901
Analog alarm, 1049
Basic Panel, 161
Communication drivers, 1032
Data type, 1049
Host Link, 1032
Trend, 1049
Omron Host Link, 1042
Address, 1043
Configuring a connection, 1032
Connection, 1032, 1036
Connection cable, 1036
Connection parameters, 1034
Data type, 1041
OPC, 13, 29, 870
Accessing tags, 27
Basic Panel, 161
Basic settings, 32
Basics, 870
Channel diagnostics, 87
Communication, 870
Communication concept, 31
Compatibility, 14, 30
Configuring a connection, 871
DCOM user rights, 89
Functionality of the WinCC OPC A&E Server, 44
HMI device as OPC client, 15
HMI device as OPC server, 16
How the WinCC OPC DA Server works, 33
HTTP connection, 14
License, 30
Mobile Panel, 14
OPC XML DA server, 14
Permitted characters in tag names, 27
Permitted data type, 26
Principle of operation of the WinCC OPC HDA server, 36
Principle of operation of the WinCC OPC UA Server, 61
Principle of operation of the WinCC OPC XML DA server, 33
Specifications, 13, 29
OPC A&E
License, 30
OPC A&E Server, 44, 53
Access to logged events, 56
Mapping rules, 44

Principle of operation, 44
ProgID, 44
Quality codes, 49
supported specifications, 44
OPC A&Raw Data, 49
OPC connection
Creating, 79
OPC DA
Data types, 873, 875
License, 30
OPC DA server, 33
Configuring HMI device as ~, 19
Creating a connection, 22
Principle of operation, 33
ProgID, 15, 33
supported specifications, 33
OPC DA Server
Configuring HMI device as ~, 20
OPC Gateway
Proxy setting, 24
OPC HDA
License, 30
OPC HDA Server, 36
Assemblies, 41
Attributes, 40
Deactivate writing access, 37
Principle of operation, 36
ProgID, 36
Quality codes, 36, 49
Supported interfaces, 42
supported specifications, 36
Write Accesses, 37
OPC server
Accessibility, 15, 32
License, 30
OPC UA
Conformance units, 65
Data type, 86
Data types, 875
Display of the logging tags, 67
Display of the WinCC tags, 67
OPC UA Service Sets, 64
Profile, 65
OPC UA Data Access, 68
OPC UA Log Access, 68
OPC UA server
Configuration file, 75
Security concept, 16, 62
Security settings, 18, 63
User identification, 63
OPC UA Server
Communication profile, 61
Creating a connection, 23, 81
Principle of operation, 61
supported specifications, 61
URL, 61
OPC XML
"Date/Time" data type, 27
"STRING" data type, 27
Configuring server, 25
Permitted cycle time, 27
OPC XML DA
Data types, 874, 876
Deleting a server, 25
Editing a server, 25
License, 30
OPC XML DA client, 14
OPC XML DA server
Installing, 35
URL, 15
OPC XML DA Server, 33
Principle of operation, 33
supported specifications, 33
URL, 34
Operating mode, 825, 826
Operator input in the recipe view
Transferring data, 251, 363, 472, 638, 718, 784
Optimizing the configuration
Allen-Bradley DF1, 927, 948, 978, 987, 1009, 1023, 1045
Other PLCs
Addressing, 902
Data types, 902
Special features, 902
Overview navigation, 105, 112, 115

P

Parallel communication
Communication drivers, 184, 903
Parameter
SIMATIC LOGO!, 808
Parameters
Client, 859
Configurable, 513
Connection, 907, 930, 962, 998, 1012, 1034
Ethernet, 507
Interface, 511
MPI, 453, 455, 457
PLC, 513
PROFIBUS, 225, 227, 229, 284, 339, 341, 343, 437, 439, 441, 613, 615, 617, 700, 702, 703
S7 1200, 387
S7 1500, 277, 745

S7 200, 498, 503, 766, 770, 771
Tags, 863
Password
HMI connection, 210, 226, 237, 321, 340, 348, 602, 614, 624
Password protection, 206, 233, 318, 320, 346, 620
PC
HMI connection, 191, 220, 307, 335, 410, 435, 451, 590, 608, 676, 698
S7 1500, 191
S7 300/400, 410
SIMATIC ET 200 CPU, 676
SIMATIC S7-1500 Software Controller, 590
WinCC RT Advanced, 335, 590, 676
WinCC RT Professional, 191, 220, 307, 335, 410, 435, 451, 608, 676
WinCC RT Advanced, 191, 220, 307, 410, 435, 451, 608
Permitted data type
OPC, 26
Permitted data types
Trends, 259, 370, 479, 645, 725, 793, 813
Pin assignment
6XV1440 - 2P for Mitsubishi PG protocol, 984
Allen-Bradley, 937
Allen-Bradley cable 1747-CP3, 939
Allen-Bradley cable 1761-CBL-PM02, 939
Allen-Bradley cable 1784-CP10, 938
Multipoint cable 1:MP/TP/PC, 1038
Multipoint cable 2:RS422, MP/TP/PC, 1039
Point-to-point cable 1,
Point-to-point cable 2, 1018
Point-to-point cable PP1 for Omron, 1040
Point-to-point cable PP2 for Omron, 1040
PLC
Connecting, 910, 964, 982, 1001, 1036
MPI parameters, 457
PROFIBUS parameters, 229, 343, 441, 617, 702
PROFINET parameters, 201, 313, 419, 599, 685
PLC5, 942
Pointer name
Area pointer, 151
Point-to-point connection
Allen-Bradley DF1, 934
Port options, 212, 324, 425, 689
Enable autonegation, 211, 323, 424, 688
Monitor, 211, 323, 424, 688
Transmission rate / duplex, 211, 323, 423, 688
Possible cause of error
Transferring data, 256, 367, 476, 642, 722, 788
PPI, 122
Network, 125
Network architecture, 126
S7 200, 771
Premium, 1005
Product support, 120
PROFIBUS, 121, 231, 344, 395, 442, 618
HMI connection, 216, 220, 223, 328, 332, 335, 428, 432, 435, 603, 608, 611, 693, 696, 698
Network, 123
Parameters, 225, 227, 229, 278, 339, 341, 343, 389, 437, 439, 441, 501, 613, 615, 617, 700, 702, 703, 742, 755, 768
S7 1200, 327, 330, 335
S7 1500, 215, 218, 220, 223, 603, 606, 608, 611
S7 300/400, 216, 427, 428, 430, 432, 603
S7-1200, 328, 332, 692, 693, 698
S7-300/400, 435
SIMATIC ET 200 CPU, 694, 696
Standard, 231, 344, 442, 618
Universal, 231, 344, 442, 618
WinCC RT Advanced, 218, 330, 430, 606, 694
WinCC RT Professional, 218, 330, 430, 606
PROFIBUS communication
S7 1200, 327, 330
S7 1500, 215, 218, 603, 606
S7 300/400, 427, 430
SIMATIC ET 200 CPU, 692, 694
PROFIBUS DP, 123
Connection, 275, 385, 495
Direct key, 845
PROFIBUS parameters
S7 1200, 339, 341, 343, 700, 702, 703
S7 1500, 225, 227, 229, 613, 615, 617
S7 300/400, 437, 439, 441
PROFIBUS profile, 231, 344, 442, 618
Different profiles on the same subnet, 231, 344, 442, 618
Effect on the transmission rate, 231, 344, 442, 618
Meaning of profiles, 232, 345, 443, 619
PROFINET, 121, 884, 895, 898
Connection, 272, 383, 492, 659, 737
HMI connection, 186, 187, 189, 191, 194, 299, 300, 303, 305, 402, 403, 405, 407, 410, 585, 586, 588, 590, 593, 671, 672, 674, 676, 679
S7 1200, 299, 303
S7 1500, 186, 189
S7 300/400, 402, 405
SIMATIC ET 200 CPU, 671, 674
SIMATIC S7-1500 Software Controller, 585, 588
PROFINET IO, 883
Port options, 211, 323, 423, 688

PROFINET parameters
HMI connection, 197, 310, 416, 595, 681
HMI device, 199, 311, 417, 597, 683
PLC, 201, 313, 419, 599, 685
S7 1200, 310, 313
S7 1500, 197, 201, 595, 597, 599, 681, 685
S7 300/400, 416, 417, 419
S7-1500, 199, 683
PROFINET RT class, 886
ProgID
OPC A&E Server, 44
OPC DA server, 33
OPC HDA Server, 36
OPC XML DA Server, 34
Project
Multiple connections, 245, 356, 466, 632, 712, 780
Project ID, 97
Protection concept, 207, 234, 318, 347, 621
Protection level, 206, 233, 318, 320, 346, 620

Q

Q
PLC, 967
Q series, 971
Q0xUDEH CPU
PLC, 969
Quality codes, 36, 49
Quantum, 1005

R

Reading out
IP address, 861
Read-only, 320
Restriction
Modicon Modbus RTU, 1025
Modicon Modbus TCP/IP, 1025
Ring port, 895, 898
Routed
Connection, 135
Routing connection
Configuring, 877
RT class, 886
Runtime Professional
Connection parameters, 282
Runtime settings
Communication, 856, 871

S

S7 1200
Communication, 298
Connection parameters, 387
HMI connection, 300
Parameters, 387
PROFIBUS, 327, 330, 335
PROFIBUS parameters, 339, 341, 343, 700, 702, 703
PROFINET, 299, 303
S7 1500
Communication, 185
Connection parameters, 277, 745
Data types, 270, 658
HMI connection, 187
Parameters, 277, 745
PC, 191
PROFIBUS, 215, 216, 218, 220, 223, 603, 606, 608, 611
PROFIBUS parameters, 225, 227, 229, 613, 615, 617
PROFINET, 186, 189
PROFINET parameters, 197, 201, 595, 597, 599, 681, 685
S7 200
Communication, 763
Connection, 763, 806
Connection parameters, 766
Data types, 804
MPI, 503, 770
Parameters, 503, 766, 770, 771
PPI, 771
S7 300
Communication, 401
MPI address, 459
S7 300/400
Connection parameters, 498
Data types, 491
HMI connection, 403
MPI, 443, 444, 446, 448, 451
MPI parameters, 453, 455, 457
Parameters, 498
PC, 410
PROFIBUS, 427, 428, 430, 432
PROFIBUS parameters, 437, 439, 441
PROFINET, 402, 405
PROFINET parameters, 416, 417, 419
S7 400
Communication, 401
MPI address, 459

S7 channel troubleshooting of connection errors, 515
S7 connection status, 542
S7 log function, 519
s7_last_detailed_err_no, 534
S7-1200
PROFIBUS, 328, 332, 693, 698
S7-1200 V2
Data types, 382
S7-1500
PROFINET parameters, 199, 683
S7-300/400
Data types, 490
PROFIBUS, 435
s7chn.ini, 525
S7DOS Error codes, 547
S7DOS Function types, 546
S7DOS Trace function, 561
SCI messages, 544
Screen number, 97
Script
Updating tag value, 145
Security mode, 17
Selection of the correct channel unit, 517
Selection of the correct communication subsystem, 516
Serial communication
Interface, 511
Server
Configuring, 856
HMI device, 856
Server certificate, 82
Service & Support, 120
Setting the Internet Settings
HTTP server, 864
Setting up
DCOM user rights, 89
Setting up DCOM user rights, 89
Setup of an S7 logbook, 524
SIMATIC ET 200 CPU
HMI connection, 672
PC, 676
PROFIBUS, 694, 696
PROFINET, 671, 674
SIMATIC HMI HTTP protocol
Valid data type, 869
SIMATIC HMI HTTP Protocol
HTTP Protocol, 855
HTTPS Protocol, 855
SIMATIC LOGO!
Communication, 805
Connection parameters, 808
Parameter, 808
SIMATIC PC
HMI connection, 194, 223, 305, 332, 407, 432, 448, 593, 611, 679, 696
S7 1500, 194
S7 300/400, 407
SIMATIC ET 200 CPU, 679
SIMATIC S7-1500 Software Controller, 593
WinCC RT Advanced, 194, 332, 407, 593, 679
WinCC RT Professional, 194, 223, 305, 332, 407, 432, 448, 611, 679
WinCC RT Advanced, 223, 305, 432, 448, 611
SIMATIC S7
Communication partners, 279, 390, 502, 743, 756, 769, 772
SIMATIC S7-1500 Software Controller
Communication, 582
HMI connection, 586
PC, 590
PROFINET, 585, 588
SIMATIC S7-300/400, 515
SIMATIC S7 1500
Addressing, 272
SIMTIC ET 200 CPU
PROFIBUS, 692
SLC, 942
Special features
Omron Host Link, 1046
Special settings of the S7 channel, 522
Standard
PROFIBUS, 231, 344, 442, 618
Storage locations of the instance certificates, 17
Switch buffer, 258, 369, 478, 644, 724, 792, 812, 951, 989, 1026, 1048
Switchport, 212, 324, 425, 689
Symbolic addressing
of a tag, 146
Sync domain, 214, 326, 426, 691
Synchronize, 144
Synchronous
Transferring data, 250, 361, 470, 637, 716, 790
Syntax
Addressing, 915

T

Table
Connection, 126
Tag
absolute addressing, 145
External tag, 143
Internal tag, 148
symbolic addressing, 146

Tag table
Default, 141
for HMI devices, 141
user-defined, 141
Tags
Addressing, 272
Basics, 141
Data exchange, 96
Task card
Hardware catalog, 118
TCP/IP
Parameters, 282, 393
Third-party drivers
Communication, 901
Special features, 902
Time synchronization, 286, 396, 665, 758
Configuring, 288, 289, 667, 668, 760, 761
integrated connection, 288, 667, 760
Non-integrated connection, 289, 398, 668, 761
Restriction, 287, 397, 666, 759
System limits, 287, 397, 666, 759
Timeout
HTTP protocol, 860
Time-triggered trends, 257, 368, 477, 643, 723, 791, 811, 950, 988, 1025, 1047
Topology discovery, 214, 326, 426, 691
Topology view
Hardware and network editor, 114
Transferring data
Area pointer, 149, 238, 349, 459, 624, 705, 774
Area pointer screen number, 239, 354, 460, 625, 706, 774
Coordination area pointer, 243, 352, 464, 630, 710, 778
Data record area pointer, 248, 359, 469, 635, 715, 783
Date/time area pointer, 240, 349, 351, 461, 626, 707, 708, 775
Date/time PLC area pointer, 241, 462, 628, 777
Job mailbox, 253, 364, 473, 640, 719, 785
Job mailbox area pointer, 246, 357, 466, 632, 712, 781
Operator input in the recipe view, 251, 363, 472, 638, 718, 784
Possible cause of error, 256, 367, 476, 642, 722, 788
Project ID area pointer, 244, 355, 465, 631, 711, 779
Triggering by means of a configured function, 254, 366, 475, 639, 721, 787
With synchronization, 250, 361, 470, 637, 716, 790
Without synchronization, 249, 360, 470, 636, 716, 789
Transmission rate / duplex, 211, 323, 423, 688
Trend
Data type, 953, 1049
Trend control, 1027
Trend request
Trend transfer, 258, 369, 478, 644, 724, 792, 812, 951, 989, 1026, 1048
Trend request area, 258, 369, 478, 644, 724, 792, 812, 951, 989, 1026, 1048
Trend transfer area, 258, 369, 478, 644, 724, 792, 812, 951, 989, 1026, 1048
Trend transfer area 1
Trend transfer area 2, 258, 369, 478, 644, 724, 792, 812, 951, 989, 1026, 1048
Triggering by means of a configured function
Transferring data, 254, 366, 475, 639, 721, 787

U

Universal
PROFIBUS, 231, 344, 442, 618
URL
OPC UA Server, 61
OPC XML DA Server, 34
Using local ID
for communication, 139

V

Valid
Data type, 911, 969, 984, 1004, 1019, 1041
Data types, 270, 382, 490, 491, 658, 729, 804, 873, 874, 875, 876
Valid data type
Allen-Bradley DF1, 941
SIMATIC HMI HTTP protocol, 869

W

WinCC
Channel Diagnosis, 1053
WinCC message system
Attributes, 49
Display of the WinCC alarm classes and alarm methods, 53
on OPC A&OPC-A&WinCC message system, 45

WinCC alarm methods on OPC A&E server, 53
WinCC Message Classes on OPC A&E Server, 53
WinCC OPC A&E server
Condition Related Events, 46
Configuring, 55
Simple Events, 46
WinCC OPC Client
Configuring, 23, 80
WinCC OPC HDA Server
Configuring, 43
WinCC OPC server
Accessibility, 32
WinCC OPC UA Server, 61
Configuring, 43, 78
Discovery Server, 61
WinCC RT Advanced
HMI connection, 223, 332, 335, 432, 448, 611, 696, 698
Interface, 183
MPI, 446
PC, 335, 590, 676
PROFIBUS, 218, 330, 430, 606, 694
SIMATIC PC, 194, 332, 407, 593, 679
WinCC RT Professional
HMI connection, 220, 223, 332, 335, 432, 435, 448, 451, 608, 611
Interface, 183
MPI, 446
PC, 191, 220, 307, 335, 410, 435, 451, 608, 676
PROFIBUS, 218, 330, 430, 606
SIMATIC PC, 194, 223, 305, 332, 407, 432, 448, 611, 679
WinCC Runtime Advanced
Interface, 183
WinCC Runtime Professional
Interface, 183
WinCC RT Advanced
HMI connection, 220, 435, 451, 608
PC, 191, 220, 307, 410, 435, 451, 608
SIMATIC PC, 223, 305, 432, 448, 611

Z

Zoom
Adjusting the zoom setting, 105, 112, 115