# SIEMENS

## SIMATIC HMI

## WinCC flexible 2005
## Communication Part 1

User's Manual

This user's manual is part of the documentation package with the order number
6AV6691-1CA01-0AB0

**Edition 06/2005**

A5E00280703-02

## Safety Guidelines

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 to property damage only 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

with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

### Caution

without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

### Notice

indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

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 device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel**. Within the context of the safety notes in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

## Prescribed Usage

Note the following:

### Warning

This device may only be used for the applications described in the catalog or the technical description and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens. Correct, reliable operation of the product requires proper transport, storage, positioning and assembly as well as careful operation and maintenance.

## Trademarks

All names identified by ® are registered trademarks of the 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.

# Preface

## Preface

## Purpose of this manual

This user manual is part of the WinCC flexible documentation..

The purpose of the "WinCC flexible Communication" user manual is to explain:

- which communications protocols can be used for communication between a SIEMENS HMI device and a controller,
- which SIEMENS HMI devices can be used for communication,
- which controllers can be connected to a selected SIEMENS HMI device,
- which settings are required in the controller program for the connection, and
- which user data areas must be set up for communication.

Separate sections therefore explain the size, structure, and function of the user data areas and the assigned area pointers.

The manual is intended for newcomers, operators and configuration engineers involved in configuration, commissioning, installation and service with WinCC flexible.

The help integrated in WinCC flexible, the WinCC flexible Information System, contains detailed information. The information system contains instructions, examples and reference information in electronic form.

## Basic Knowledge Requirements

General knowledge in the field of automation engineering is required to understand this manual.

You should also have experience of using PCs running under the Windows 2000 or Windows XP operating systems. A knowledge of VBA or VBS is required for advanced configuration by using scripts.

## Scope of the manual

This manual is valid for the WinCC flexible 2005 software package.

## Position in the information scheme

This manual is part of the SIMATIC HMI documentation. The information below presents an overview of the information landscape of SIMATIC HMI.

**User manual**

- WinCC flexible Micro
  - describes the engineering basics based on the WinCC flexible Micro engineering system (ES)
- WinCC flexible Compact/ Standard/ Advanced
  - describes the engineering basics based on the WinCC flexible Compact, WinCC flexible Standard and WinCC flexible Advanced engineering systems (ES)
- WinCC flexible Runtime:
  - Describes how to commission and operate your Runtime project on a PC.
- WinCC flexible Migration:
  - Describes how to convert an existing ProTool project to WinCC flexible.
  - Describes how to convert an existing WinCC project to WinCC flexible.
  - Describes how to migrate ProTool projects with an HMI migration from OP7 to OP 77B or OP 73micro.
  - Describes how to migrate ProTool projects with an HMI migration from OP7 to OP 77B or OP 77A.
  - Describes how to migrate ProTool projects with an HMI migration from OP17 to OP 177B.
  - Describes how to migrate ProTool projects with HMI migration from graphic devices to Windows CE devices.
- Communication:
  - Communication Part 1 describes the connection of the HMI device to SIMATIC PLCs.
  - Communication Part 2 describes the connection of the HMI device to third-party PLCs.

**Operating Instructions**

- Operating instructions for SIMATIC operating units:
  - OP 73, OP 77A, OP 77B
  - TP 170micro, TP 170A, TP 170B, OP 170B
  - OP 73micro, TP 177micro
  - TP 177A, TP 177B, OP 177B
  - TP 270, OP 270
  - MP 270B
  - MP 370
- Operating instructions for mobile SIMATIC operating units:
  - Mobile Panel 170
- Operating instructions (compact) for SIMATIC operating units:
  - OP 77B
  - Mobile Panel 170

### Getting Started

- WinCC flexible for first time users:
    - Based on a sample project, this is a step-by-step introduction to the basics of configuring screens, alarms, and recipes, and screen navigation.
- WinCC flexible for advanced users:
    - Based on a sample project, this is a step-by-step introduction to the basics of configuring logs, project reports, scripts, user management, and multilingual projects, and integration into STEP 7.
- WinCC flexible options:
    - Based on a sample project, this is a step-by-step introduction to the basics of configuring the WinCC flexible Audit, Sm@rtServices, Sm@rtAccess and OPC Server options.

### Online availability

The following links provide direct access to technical documentation on SIMATIC products and systems in English, German, French, Italian, and Spanish.

- SIMATIC Guide Technische Dokumentation in Deutsch:
  "http://www.ad.siemens.de/simatic/portal/html_00/techdoku.htm"
- SIMATIC Guide for Technical Documentation in English:
  "http://www.ad.siemens.de/simatic/portal/html_76/techdoku.htm"

## Guide

The user manual consists of Parts 1 and 2. Part 1 is organized as follows:

- Communication basics
- Connecting to the SIMATIC S7
- Connecting via the SIMATIC HMI HTTP protocol
- Connecting via OPC
- Connecting to the SIMOTION PLC
- Connecting to the SIMATIC S5
- Connecting to the SIMATIC 500/505

## Conventions

A distinction is made in the naming conventions for the configuration and runtime software:

- "WinCC flexible 2005" refers to the configuration software.
- "Runtime" designates the runtime software running on the HMI devices.
- "WinCC flexible Runtime" designates the visualization product for use on standard PCs or panel PCs.

The term "WinCC flexible" is used in the general context. A version name such as "WinCC flexible 2005" is used whenever it is necessary to distinguish it from other versions.

The following formatting is used to facilitate reading of the manual:

| Notation | Scope |
|---|---|
| "Add screen" | • Terminology that occurs in the user interface, e.g., dialog names, tabs, buttons, menu commands.<br>• Inputs required, e.g., limit values, tag values<br>• Path information |
| "File > Edit" | Operational sequences, e.g., menu commands/shortcut menu commands. |
| <F1>, <Alt>+<P> | Keyboard inputs |

Please observe notes labeled as follows:

**Note**

Notes containing important information about the product and its use or a specific section of the documentation to which you should pay particular attention.

## Trademarks

| |
|---|
| HMI® |
| SIMATIC® |
| SIMATIC HMI® |
| SIMATIC ProTool® |
| SIMATIC WinCC® |
| SIMATIC WinCC flexible® |

Third parties using for their own purposes any other names in this documentation which refer to trademarks might infringe upon the rights of the trademark owners.

## Further Support

If you have any technical questions, please get in touch with your Siemens representative or agent responsible.

You will find your contact person at:

"http://www.siemens.com/automation/partner"

You will find a guide to the technical documentation offered for the individual SIMATIC Products and Systems here at:

"http://www.siemens.com/simatic-tech-doku-portal"

The online catalog and order system is found under:

"http://mall.automation.siemens.com/"

## Training Centers

Siemens offers a number of training courses to familiarize you with the SIMATIC S7 automation system. Please contact your regional training center or our central training center in D 90327 Nuremberg, Germany for details:

Telephone:   +49 (911) 895-3200.

Internet::     http://www.sitrain.com"

## Technical Support

You can reach the Technical Support for all A&D products
Via the Web formula for the Support Request

"http://www.siemens.com/automation/support-request"

Phone:        + 49 180 5050 222

Fax:           + 49 180 5050 223

Additional information about our Technical Support can be found on the Internet pages "http://www.siemens.com/automation/service"

## Service & Support on the Internet

In addition to our documentation, we offer our Know-how online on the internet at:

"http://www.siemens.com/automation/service&support"

where you will find the following:

- The newsletter, which constantly provides you with up-to-date information on your products.

- The right documents via our Search function in Service & Support.

- A forum, where users and experts from all over the world exchange their experiences.

- Your local representative for Automation & Drives.

- Information on field service, repairs, spare parts and more under "Services".

# Table of contents

# Working with connections

<div style="text-align: right; font-size: 3em;">1</div>

## 1.1 Basics

### 1.1.1 Communication basics

**Introduction**

The term communication refers to the data exchange between two communication partners. The communication partners can be interconnected via direct cable connection or network.

**Communication partners**

A communication partner can be any node which is capable of communicating and exchanging data with other nodes on the network. In the WinCC flexible environment, those nodes may be CPUs and communication modules of the automation system, HMI devices and CPs in the PC.

Data transferred between the communication partners may serve different purposes. In WinCC flexible, those are:

- process control
- process data acquisition
- reporting states in a process
- process data logging

## 1.1.2 Principles of communication

### Introduction

WinCC flexible controls communication between the HMI and the PLC by means of tags and area pointers.

### Communication using tags

In WinCC flexible, tags are centrally managed in the "Tag" editor. There are external and internal tags. External tags are used for communication, and represent the image of defined memory locations on the PLC. The HMI and the PLC both have read and write access to this memory location. Those read and write operations may cyclic or event-triggered.

In your configuration, create tags that point to specific PLC addresses. The HMI reads the value from the defined address, and then displays it. The operator may also enter values on the HMI which will be written to the relevant PLC address.

### Communication using area pointers

Area pointers are used to exchange data of specific user data areas. Area pointers are parameter fields WinCC flexible uses in Runtime to obtain information about the location and size of data areas of the PLC. During communication, the PLC and the HMI alternately access those data areas for read and write operations. Based on the results of the evaluation of data stored in those areas, the controller and HMI trigger defined actions.

Area pointers used in WinCC flexible:

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

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

### Communication between WinCC flexible and automation systems

Industrial communication using WinCC flexible means that data are exchanged using tags and area pointers. To acquire the data, the HMI sends request frames to the automation system using a communication driver. The automation system (AS) returns the requested data to the HMI in a response frame.

## Communication drivers

A communication driver is a software component that establishes a connection between an automation system and an HMI, and thus allows the transfer of process values to the WinCC flexible tags. WinCC flexible supports the interconnection of different automation systems with various communication drivers.

Users can select the interface, the profile and the transmission speed for each specific communication partner.

## Communication between HMIs

The SIMATIC HMI HTTP Protocol is available for the communication between HMIs. This protocol is a component of the "Sm@rtAccess" option. The Protocol can be used on PCs with WinCC flexible 2004 Runtime and on Panels as of the 270 series. For detailed information, refer to the SIMATIC HMI HTTP Protocol documentation.

## Communication via uniform and manufacturer-independent interface

WinCC flexible provides a uniform and manufacturer-independent software interface using OPC (OLE for Process Control). This interface allows a standardized data exchange between applications for industry, office, and production. For detailed information, refer to the OPC documentation.

# 1.2 Elements and basic settings

## 1.2.1 Connections Editor

### Introduction

In the Connections editor, you create and configure connections.

### Open

Select "Connections" from the project view, and then open the shortcut menu. Select "New connection" from in this shortcut menu. The new connection will be created and opened in the work area.

## Structure
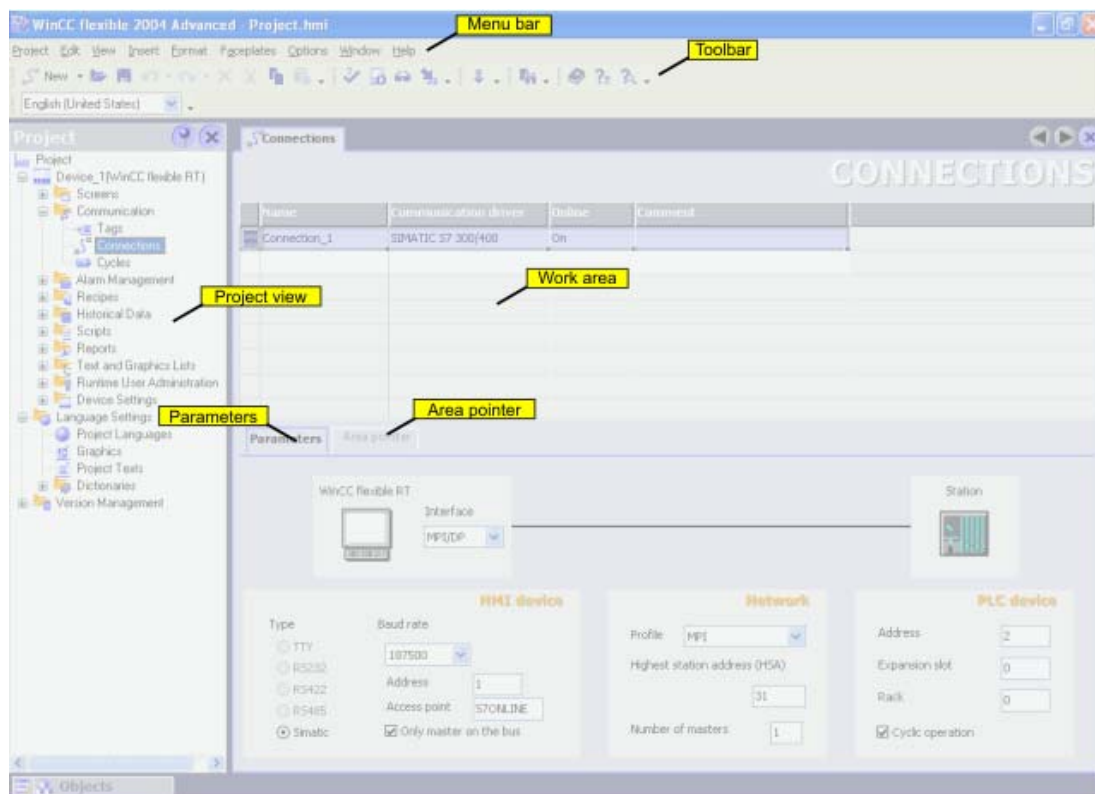


## Menu bar

The menu bar contains all commands required for operating WinCC flexible. Any available keystroke is indicated next to the menu command.

## Toolbars

The toolbars contain the most frequently used buttons.

Select "View > Toolbars" to show or hide the specific toolbars. The ⬝ button of a toolbar can be used to show or hide specific buttons of this toolbar.

## Work area

All connections are visualized in the work area in tabular format. You select the communication drivers from the table cells, and edit the relevant connection properties. To sort the table by its column entries, simply click the column header.

## "Parameters" tab

Using the "Parameters" tab in the work area, you configure the parameters of the communication driver you selected from the table. Select the settings for the HMI, the network and for the controller.

**"Area pointer" tab**

You configure the area pointers of the connections on the "Area pointer" tab in the work area.

## 1.2.2 Parameters for connections

**Introduction**

Select the "Parameters" tab of the "Connections" editor to configure the properties of a connection between the HMI and the communication partner.

**Structure**

The communication partners are visualized schematically on the "Parameters" tab. This tab provides the "HMI device", "Network" and "Controller" areas where you can declare the parameters of the relevant interface used.
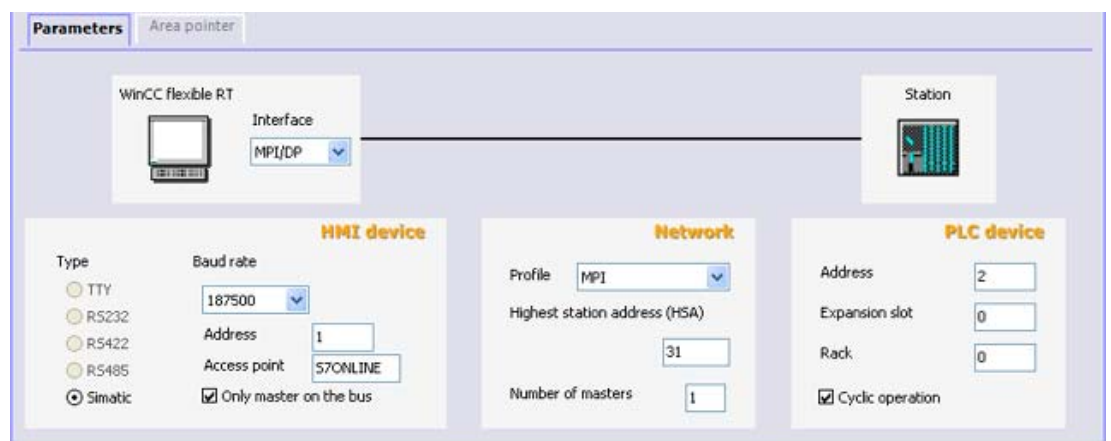


The system sets default parameters. Always ensure consistency on the network whenever you edit parameters. For detailed information on configurable parameters, refer to the description of the supported protocols

## 1.2.3 Area pointers for connections

### Introduction

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

### Structure

The "Area pointer" tab contains two tables of area pointers. The "For all connections" table contains those area pointers which are created only once in the project and can be used for only one connection.

The "For each connection" table contains the area pointers you can set separately for each available connection.

| | Connection | Name | Address | Length | Trigger mode | Acquisition cycle | Comment |
|---|---|---|---|---|---|---|---|
| | Connection_1 | Date/time PLC | DB 1 DBW 0 | 6 | Cyclic continuous | 1 min | |
| | <undefined> | Project ID | | 1 | Cyclic continuous | <undefined> | |
| | <undefined> | Screen number | | 5 | Cyclic continuous | <undefined> | |

For each connection

| | Active | Name | Address | Length | Trigger mode | Acquisition cycle | Comment |
|---|---|---|---|---|---|---|---|
| | On | Coordination | DB 1 DBW 12 | 2 | Cyclic continuous | <undefined> | |
| | Off | Data mailbox | | 5 | Cyclic continuous | <undefined> | |
| | Off | Date/time | | 6 | Cyclic continuous | <undefined> | |
| | Off | Job mailbox | | 4 | Cyclic continuous | <undefined> | |

The availability of the various area pointers is determined by the HMI used. For detailed information on area pointers and their configuration, refer to the description of the supported protocols

## 1.3 Configuring the connection

### Introduction

You create a new connection using the Connections editor.

### Requirements

A project is open.

## Procedure

1. In the project view, open the "Communication" group.

2. Select "New connection" from the "Connections" shortcut menu.

   The "Connections" editor opens and shows a new connection.

3. Rename the connection in the "Name" column as required.

4. From the "Communication driver" column, select a communication driver that is suitable for the PLC being used.

| Name | Communication driver |
|---|---|
| Connection_1 | SIMATIC S7 300/400 ▼ |
| | Allen Bradley DF1 |
| | Allen Bradley DH485 |
| | GE Fanuc SNP |
| | LG GLOFA-GM |
| | Mitsubishi FX |
| | Mitsubishi Protocol 4 |
| | Modicon MODBUS |
| | Omron Hostlink / Multilink |
| | OPC |
| | SIMATIC 500/505 seriell |
| | SIMATIC 500/505 DP |
| | SIMATIC S5 AS511 |
| | SIMATIC S5 DP |
| | SIMATIC S7 200 |
| | SIMATIC S7 300/400 |
| | SIMATIC HMI HTTP Protocol |
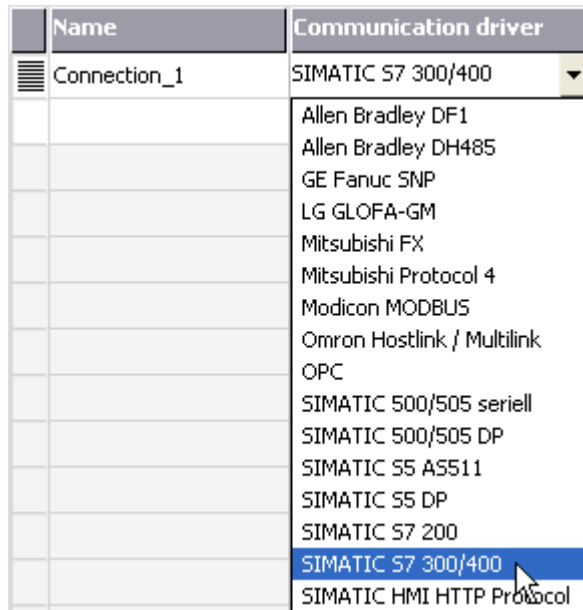
   Only those drivers that are supported by the selected HMI device will be displayed.

5. The system automatically sets suitable values for the communication partner in the "Parameters" tab.

6. Check the parameters, and edit these as required.

7. Save the project.

## Alternative procedure

Select "Insert > New item > Connection" from the menu bar. The "Connections" editor opens and shows a new connection. Edit the connection as described in steps 2 through 7.

## Result

A new connection was created. The connection parameters are configured.

# 1.4 Connections and protocols

## HMI functionality

The HMI is used to read, save and record alarms and tags. In addition, the HMI can be used to intervene in the process.

## Data exchange

Prerequisite for using the HMI's control and monitoring functionality is its connection to a PLC. Data exchange between the HMI and the PLC is controlled by a connection-specific protocol. Each connection requires a separate protocol.

## Criteria for selecting the connection

Criteria for selecting the connection between the HMI and the PLC are, for example:

- PLC type
- CPU in the PLC
- HMI type
- Number of HMIs per PLC
- Structure and and any bus systems of an existing plant
- Amount of components additionally required

## Protocols

Protocols are available for the following PLCs:

| PLC | Protocol |
|-----|----------|
| SIMATIC S7 | • PPI<br>• MPI [1]<br>• PROFIBUS<br>• TCP/IP (Ethernet) |
| SIMATIC S5 | • AS 511<br>• PROFIBUS |
| SIMATIC 500/505 | • NITP<br>• PROFIBUS |
| SIMATIC HMI HTTP Protocol | • HTTP/HTTPS (Ethernet) |
| SIMOTION | • PROFIBUS |
| OPC | • DCOM |
| Allen-Bradley | PLC series SLC500, SLC501, SLC502, SLC503, SLC504, SLC505, MicroLogix and PLC5/11, PLC5/20, PLC5/30, PLC5/40, PLC5/60, PLC5/80<br>• DF1 [2]<br>• DH+ via DF1 [3]<br>• DH485 + via DF1 [4]<br>• DH485 [4] |

| PLC | Protocol |
|---|---|
| GE Fanuc Automation | PLC series 90–, 9090<br>• SNP |
| LG Industrial Systems (Lucky Goldstar) / IMO | PLC series GLOFA GM (GM4, GM6 and GM7) / Series G4, G6 and G7<br>• Dedicated communication |
| Mitsubishi Electric | PLC series MELSEC FX and MELSEC FX0<br>• FX |
| Mitsubishi Electric | PLC series MELSEC FX0, FX1n, FX2n, AnA, AnN, AnS, AnU, QnA and QnAS<br>• Protocol 4 |
| OMRON | PLC series SYSMAC C, SYSMAC CV, SYSMAC CS1, SYSMAC alpha and CP<br>• Hostlink/Multilink (SYSMAC Way) |
| Schneider Automation (Modicon) | PLC series Modicon 984, TSX Quantum and TSX Compact<br>• MODBUS RTU |

[1]    Not possible when connected to S7-212.

[2]    Applies to controllers SLC503, SLC504, SLC505, PLC5, MicroLogix

[3]    Applies to controllers SLC504, PLC5 over DF1

[4]    Applies to controllers SLC500 to SLC 505 and MicroLogix

# 1.5     Device-dependency of the protocols

## Availability of the communication protocols

Communication between the HMI and the PLC is controlled using a network-specific protocol. The tables below shows the communication protocols supported by the HMIs.

## Overview

## Micro Panels

|  | OP 73micro | TP 170micro | TP 177micro |
|---|---|---|---|
| SIMATIC S7 - PPI [1] | No | yes | No |
| SIMATIC S7 - MPI | Yes [1] | Yes [1] | Yes [1] |
| SIMATIC S7 - PROFIBUS | Yes [1] | Yes [1] | Yes [1] |
| SIMATIC S7 - Ethernet | No | No | No |
| SIMATIC S5 - AS511 | No | No | No |
| SIMATIC S5 - PROFIBUS DP | No | No | No |
| SIMATIC 500/505 - NITP | No | No | No |
| SIMATIC 500/505 - PROFIBUS DP | No | No | No |

|  | OP 73micro | TP 170micro | TP 177micro |
|---|---|---|---|
| SIMATIC HMI HTTP Protocol | No | No | No |
| OPC | No | No | No |
| Allen-Bradley DF1 | No | No | No |
| Allen-Bradley DH 485 | No | No | No |
| GE Fanuc | No | No | No |
| LG GLOVA-GM | No | No | No |
| Mitsubishi MELSEC | No | No | No |
| Modicon MODBUS | No | No | No |

## Mobile Panels

|  | Mobile Panel 170 | Mobile Panel 177 DP | Mobile Panel 177 PN |
|---|---|---|---|
| SIMATIC S7 - PPI [1] | yes | Yes | no |
| SIMATIC S7 - MPI | yes | Yes | no |
| SIMATIC S7 - PROFIBUS | yes | Yes | no |
| SIMATIC S7 - Ethernet | No | no | Yes |
| SIMATIC S5 - AS511 | yes | yes [3] | no |
| SIMATIC S5 - PROFIBUS DP | yes | Yes | no |
| SIMATIC 500/505 - NITP | yes | Yes | no |
| SIMATIC 500/505 - PROFIBUS DP | yes | Yes | no |
| SIMATIC HMI HTTP Protocol | No | no | Yes |
| OPC | No | no | no |
| Allen-Bradley DF1 | yes | Yes | no [4] |
| Allen-Bradley DH 485 | yes | Yes | no [4] |
| GE Fanuc | yes | Yes | no [4] |
| LG GLOVA-GM | yes | Yes | no [4] |
| Mitsubishi MELSEC | yes | Yes | no [4] |
| Modicon MODBUS | yes | Yes | no [4] |

## Panels

|  | OP 73 | OP 77A | OP 77B | TP 170A | TP 170B OP 170B | TP 177A | TP 177B OP 177B | 270 series |
|---|---|---|---|---|---|---|---|---|
| SIMATIC S7 - PPI [1] | No | No | yes | yes | yes | No | Yes | yes |
| SIMATIC S7 - MPI | yes | yes | yes | yes | yes | yes | Yes | yes |
| SIMATIC S7 - PROFIBUS | yes | yes | yes | yes | yes | yes | Yes | yes |
| SIMATIC S7 - Ethernet | No | No | No | No | yes | No | Yes | yes |
| SIMATIC S5 - AS511 | No | No | yes | yes | yes | No | yes [3] | yes |
| SIMATIC S5 - PROFIBUS DP | No | No | yes | No | yes | No | Yes | yes |

| | OP 73 | OP 77A | OP 77B | TP 170A | TP 170B OP 170B | TP 177A | TP 177B OP 177B | 270 series |
|---|---|---|---|---|---|---|---|---|
| SIMATIC 500/505 - NITP | No | No | yes | yes | yes | No | Yes | yes |
| SIMATIC 500/505 - PROFIBUS DP | No | No | yes | No | yes | No | Yes | yes |
| SIMATIC HMI HTTP Protocol | No | No | No | No | No | No | no | yes |
| OPC | No | No | No | No | No | No | no | No |
| Allen-Bradley DF1 | No | No | yes | yes | yes | No | Yes | yes |
| Allen-Bradley DH 485 | No | No | yes | yes | yes | No | Yes | yes |
| GE Fanuc | No | No | yes | yes | yes | No | Yes | yes |
| LG GLOVA-GM | No | No | yes | yes | yes | No | Yes | yes |
| Mitsubishi MELSEC | No | No | yes | yes | yes | No | Yes | yes |
| Modicon MODBUS | No | No | yes | yes | yes | No | yes [5] | yes |

## Multi Panels

| | 270 series [2] | 370 series |
|---|---|---|
| SIMATIC S7 - PPI [1] | yes | yes |
| SIMATIC S7 - MPI | yes | yes |
| SIMATIC S7 - PROFIBUS | yes | yes |
| SIMATIC S7 - Ethernet | yes | yes |
| SIMATIC S5 - AS511 | yes | yes |
| SIMATIC S5 - PROFIBUS DP | yes | yes |
| SIMATIC 500/505 - NITP | yes | yes |
| SIMATIC 500/505 - PROFIBUS DP | yes | yes |
| SIMATIC HMI HTTP Protocol | yes | yes |
| OPC | yes | yes |
| Allen-Bradley DF1 | yes | yes |
| Allen-Bradley DH 485 | yes | yes |
| GE Fanuc | yes | yes |
| LG GLOVA-GM | yes | yes |
| Mitsubishi MELSEC | yes | yes |
| Modicon MODBUS | yes | yes |

## WinCC flexible Runtime

| | WinCC flexible Runtime |
|---|---|
| SIMATIC S7 - PPI [1] | yes |
| SIMATIC S7 - MPI | yes |
| SIMATIC S7 - PROFIBUS | yes |
| SIMATIC S7 - Ethernet | yes |
| SIMATIC S5 - AS511 | yes |
| SIMATIC S5 - PROFIBUS DP | yes |
| SIMATIC 500/505 - NITP | yes |
| SIMATIC 500/505 - PROFIBUS DP | yes |
| SIMATIC HMI HTTP Protocol | yes |
| OPC | yes |
| Allen-Bradley DF1 | yes |
| Allen-Bradley DH 485 | yes |
| GE Fanuc | yes |
| LG GLOVA-GM | yes |
| Mitsubishi MELSEC | yes |
| Modicon MODBUS | yes |

[1]     Only with SIMATIC S7-200

[2]     Only MP 270B

[3]     via PC/TTY adapter (option)

[4]     via http protocol, if supported by the controller.

[5]     Not released for TP 177A, TP 177B DP, TP 177B PN/DP, OP 177B DP, OP 177B PN/DP.

# 1.6     Conversion when changing controllers

## Changing the controller

With controllers and drivers for which it is pointless to attempt to adopt the addresses from other controllers, for example SIMOTION, SIMATIC HMI HTTP protocol or OPC, the address is deleted. With all other controllers, there is an attempt to adopt the data type. If this is successful, there is an attempt to adopt the address. If the adoption of the data type fails, a default data type and a default address are used. If the adoption of the address fails, a default address is used.

## Changing within a device family or changing the CPU type

As described above, there is an attempt to adopt the address and data type. If an address or a data type needs to be adapted before it can be accepted (for example because the new CPU/controller does not support the previous address format),

- a message is displayed

- the field is displayed on an orange background.
  Change the relevant value to a valid value.

# Communication with SIMATIC S7 controllers

# 2

## 2.1 Communication with SIMATIC S7

### 2.1.1 Communications partner (SIMATIC S7)

**Introduction**

This section describes communication between HMI devices and SIMATIC S7.

When working with SIMATIC S7, the HMI devices can be connected over various networks. The networks depend on the module being used. Networks supported:

| PLC | Modules | Network | Profile |
|---|---|---|---|
| SIMATIC S7-200 | CPU | PPI | PPI |
| | | MPI | MPI |
| | CPU with PROFIBUS DP interface or PROFIBUS CP | PROFIBUS | PROFIBUS DP Standard[)], Universal |
| | CP 243-1 | Ethernet | TCP/IP ISO[1)] |
| SIMATIC S7-300/400 | CPU FM with communication functions | MPI | MPI |
| | CPU with PROFIBUS DP interface or PROFIBUS CP | PROFIBUS | PROFIBUS- DP Standard , Universal |
| SIMATIC S7-300 | CP 343-1 | Ethernet | TCP/IP ISO[1)] |
| SIMATIC S7-400 | CP 443-1 | Ethernet | TCP/IP ISO[1)] |
| [1)] The current WinCC flexible versions do not support ISO. | | | |

**Special features of communication**

Applies to SINUMERIK modules

Windows CE devices ,except for TP 170A, also support the SINUMERIK modules. If you want to use the HMI device to communicate with the CPU and the SINUMERIK module, a connection must be created to each communication partner in WinCC flexible.

Applies to SIMODRIVE

The SIMODRIVE MCU 172A control system must be created in WinCC flexible as a PLC with an address.

## 2.1.2 Communication between the HMI and the PLC

**Communications principle**

The HMI and the SIMATIC S7 PLC communicate using

* Tags

* User data areas

* Networks

**Tags**

The PLC and the HMI exchange data using process values. In your configuration, create tags that point to specific PLC addresses. The HMI reads the value from the defined address, and then displays it. The operator may also enter values on the HMI which will be written to the relevant PLC address.

**User data areas**

User data areas are used to exchange special data, and are set up only when such data are being used.

User data areas are, for example, required for the following:

* Alarms

* Trends

* Recipes

* PLC jobs

* Sign-of-life monitoring

You set up the user data areas when you create the project in the project window of the HMI, by selecting "Communication ▸ Connections" from the "Area pointers" work area. They also need to be assigned to the corresponding address area on the PLC.

## Networks

WinCC flexible offers diverse networks for the communication between the HMI and SIMATIC S7 PLCs. Networks significant for the communication with WinCC flexible:

- PPI (Point-to-Point Interface)
- MPI (Multi-Point Interface)
- PROFIBUS (Process Field Bus)
- Ethernet

On the Ethernet network, the connection is established via communications processor (CP) or a CPU of the SIMATIC S7 PLCs. Other network connections can be established using the MPI/PROFIBUS interface of the CPU.

### Note

Allowances must be made for certain special features when routing connections between the HMI and the PLC. Whether a connection is made during startup of the HMI depends on the communications resources of the partner modules which support routing. For further information, refer to the SIMATIC NET documentation.

### Note

A SIMATIC S7 PLC supports parallel communication via MPI/PROFIBUS interface and Ethernet. When communication via MPI interface goes down, communication via Ethernet is also restricted. The interruption could be caused, for example, by the removal of the MPI connector.

## 2.1.3    Networks

### Introduction

HMIs and SIMATIC S7 PLC can communicate using the following networks:

- PPI (Point-to-Point Interface)
- MPI (Multi-Point Interface)
- PROFIBUS (Process Field Bus)
- Ethernet

### PPI network

PPI communication is only supported by the SIMATIC S7-200 PLC. A PPI connection is a PtP connection, whereby the HMI operates in master mode, and the SIMATIC S7-200 in slave mode.

Only one SIMATIC S7-200 can be connected to an HMI. The HMI is connected using the serial connector of the CPU. Several HMIs can be connected to a SIMATIC S7-200. A SIMATIC S7-200 supports only one connection at any given time.

**Note**

The PPI network supports up to six further masters in addition to the HMI. To avoid loss of performance, however, you should not configure more than four nodes as masters on the PPI network.

## MPI Network

The HMI is connected to the MPI interface of the SIMATIC S7 PLC. Several HMIs can be connected to the same SIMATIC S7 PLC, and several SIMATIC S7 PLCs can be connected to the same HMI. The maximum number of communications partners supported by the HMI depends on the HMI type being used. For further information, refer to the documentation of the relevant HMI.

The SIMATIC S7-200 PLC must be configured as a passive network node. It is connected via PPI port.

## PROFIBUS Network

On the PROFIBUS network, an HMI can by connected to S7 modules which are equipped with an integrated PROFIBUS or PROFIBUS DP interface. Several HMIs can be connected to the same SIMATIC S7 PLC, and several SIMATIC S7 PLCs can be connected to the same HMI. The maximum number of communications partners supported by the HMI depends on the HMI type being used. For further information, refer to the documentation of the relevant HMI.

The SIMATIC S7-200 PLC must be configured as a passive network node. The SIMATIC S7-200 is networked using a DP connector or PROFIBUS CP.

## Ethernet network

WinCC flexible supports connections to all S7 modules which are networked using an integrated Ethernet interface or CP. Several HMIs can be connected to the same SIMATIC S7 PLC, and several SIMATIC S7 PLCs can be connected to the same HMI. The maximum number of communications partners supported by the HMI depends on the HMI type being used. For further information, refer to the documentation of the relevant HMI.

# 2.2 Configuring the SIMATIC S7 communication driver

## 2.2.1 Installing the communication driver

### 2.2.1.1 Installing the communication driver

#### Drivers for the HMI

The communication drivers for SIMATIC S7-200 and SIMATIC S7-300/400 PLCs are supplied with WinCC flexible, and are installed automatically.

The PLC does not require any special blocks for the connection.

## 2.2.2 PPI communication

### 2.2.2.1 Requirements for PPI communication

#### Hardware requirements

- PPI communication is only supported by the SIMATIC S7-200 PLC.
- To connect a Panel PC or standard PC to a SIMATIC S7-200, you need a PC/PPI cable or a CP, such as CP 5611.

  WinCC flexible setup automatically installs the relevant driver.

#### Cables

---

**Note**

Set compatibility mode on the new PC/PPI cable (6ES7 901-3CB30-0XA0) as described in the cable data sheet in order to enable its use in WinCC flexible Runtime. It does not support the functionality for 187.5 kbps in this mode.

The new PC/PPI USB cable (6ES7 901-3DB30-0XA0) is currently not supported.

---

#### Software Requirements

Software components required for PPI communication:

- WinCC flexible engineering tool
- STEP 7 Micro Win software for the SIMATIC S7-200 PLC

## 2.2.2.2    Configuring protocol parameters using PPI

### Parameters to set

To edit the parameters, double-click "Communication > Connections" in the project view of the HMI. "SIMATIC S7-200" is selected from the "Communication drivers" column on the work area. You can now edit the protocol parameters in the properties view:

### Device-specific parameters

Select "HMI device" to set the network parameters for the HMI. You only need to do this once. The modification applies to all communication partners.

- "Interface"

  There, select the HMI interface at which it is connected to the PPI network.

| HMI | Interface |
|---|---|
| OP 73micro, TP 170micro, TP 177micro<br>Mobile Panel 170, Mobile Panel 177<br>OP 73, OP 77A, OP 77B, TP 170A, TP 170B,<br>OP 170B, TP 177A, TP 177B, OP 177B, TP 270,<br>OP 270,<br>MP 270B, MP 370 | IF1B |
| Panel PC, PC | MPI/PROFIBUS DP |

- "Transmission rate"

  There, set the data transmission rate for the network. The data transmission rate is determined by the slowest HMI on the network. The setting must be consistent on the entire network.

- "Address"

  There, set the PPI address of the HMI. The PPI address must be unique on the PPI network.

- "Access point"

  There, set the access point over which the communications partner can be reached.

- "Only master on the bus"

  This disables an additional safety function that prevents bus disturbances when the HMI connects to the network. A passive station (slave) can only send data on request by an active station (master.) If you only have slaves connected to the HMI, disable this safety function by setting the "Only master on the bus" check box.

  When operating the HMI on an S7-200 PLC, set the HMI to master mode.

## Network parameters

Select "Network" to set the network parameters for the HMI.

- "Profile"

  There, select the relevant network profile. In this case, set "PPI." The setting must be consistent on the entire network.

- "Highest station address"

  There, set the highest station address. The highest station address must be equal to or higher than the highest MPI address. The setting must be consistent on the entire network.

- "Number of masters"

  Set the number of masters on the network to "1."

## Communication partners

Select "Controller" to set the address of the S7 module with which the HMI will exchange data. Assign a connection name for each communications partner.

- "Address"

  There, set the PPI address of the S7 module (CPU) to which the HMI is connected.

- "Slot"

  This parameter is not necessary for PPI communication.

- "Rack"

  This parameter is not necessary for PPI communication.

- "Cyclic operation"

  This parameter is not necessary for PPI communication.

## 2.2.3    MPI communication

### 2.2.3.1    Requirements for MPI communication

## Hardware requirements

Hardware components required for MPI networking:

- To connect a Panel PC or standard PC to the MPI interface of the S7 PLC, you need either a communications processor (for example, CP 5512, CP 5611, CP 5613), or a PC/MPI adapter (for example, PC adapter). The PC adapter converts RS-232 (serial) to RS-485 (MPI).

---

### Note

The PC/MPI adapter must be supplied with power via the MPI connector. It cannot be connected to HMIs, but only to the MPI interface of a CPU. The PC adapter may only be used for PtP connections.

---

---

**Notice**

The PC adapter is not approved for MPI transfer.

---

- Use an MPI cable to connect the HMI to the SIMATIC S7 PLC.

---

**Note**

At a SIMATIC S7-200 of the first generation (CPU 214, 215, 216), always use port 1 for MPI/PROFIBUS DP communication. When using a CPU 214, set a transmission rate of 9.6 kbps.

---

- Use a serial data cable to connect the HMI to the PC.
- To set up an MPI network, you need a PROFIBUS bus cable with PROFIBUS connector.

---

**Note**

A cable must be terminated with its characteristic wave impedance. Enable the terminating resistor on the first and last node of the PROFIBUS/MPI network.

Both nodes must be supplied with power. Of the two end nodes (first and last) of a PROFIBUS/MPI network, only one can be an OP!

The PROFIBUS bus terminal RS-485 and PROFIBUS connector are equipped with a switch which is used to enable the terminating resistor. Meaning of the ON and OFF switch settings:

OFF: The terminating resistor is disabled, the bus cable is not terminated.

ON: The terminating resistor is enabled, the bus cable is terminated.

If a problem occurs on the connection between an S7-400 (version 1 only) and an OP, a new connection can only be established again after turning off the power on the OP.

---

## Software Requirements

Software components required for MPI communication:

- WinCC flexible engineering tool
- STEP 7 software for the SIMATIC S7 PLC
- SIMATIC NET software, for example, for CP5613

### 2.2.3.2 Configuring protocol parameters over MPI

## Parameters to set

To edit the parameters, double-click "Communication > Connections" in the project view of the HMI. On the work area, "SIMATIC S7-200" or "SIMATIC S7-300/400" is selected from the "Communication drivers" column. You can now edit the protocol parameters in the properties view:

## Device-specific parameters

Select "HMI device" to set the network parameters for the HMI. You only need to make the setting once. The modification applies to all communication partners.

- "Interface"

  There, select the HMI interface at which it is connected to the MPI network.

  | HMI | Interface |
  | --- | --- |
  | OP 73micro, TP 170micro, TP 177micro<br>Mobile Panel 170, Mobile Panel 177<br>OP 73, OP 77A, OP 77B, TP 170A, TP 170B, OP 170B,<br>TP 177A, TP 177B, OP 177B, TP 270, OP 270,<br>MP 270B, MP 370 | IF1B |
  | Panel PC, PC | MPI/PROFIBUS DP |

- "Transmission rate"

  There, set the data transmission rate for the network. The data transmission rate is determined by the slowest HMI on the network. The setting must be consistent on the entire network.

- "Address"

  There, set the MPI address of the HMI. The MPI address must be unique in the MPI network.

- "Only master on the bus"

  This disables an additional safety function that prevents bus disturbances when the HMI connects to the network. A passive station (slave) can only send data on request by an active station (master.) If you only have slaves connected to the HMI, disable this safety function by setting the "Only master on the bus" check box.

  When operating the HMI on an S7-200 PLC, set the HMI to master mode.

## Network parameters

Select "Network" to set the parameters of the MPI network to which the HMI is connected. Those network parameters will be applied if you have integrated WinCC flexible in STEP 7, and the HMI is connected to the network

- "Profile"

  There, select the relevant network profile. In this case, set "MPI." The setting must be consistent on the entire network.

- "Highest station address"

  There, set the highest station address. The highest station address must be equal to or higher than the highest MPI address. The setting must be consistent on the entire network.

- "Number of masters"

  This information is not required for MPI.

## Communication partners

Select "Controller" to set the address of the S7 module with which the HMI will exchange data. Assign a connection name for each communication partner.

- "Address"

  There, set the MPI address of the S7 module (CPU) to which the HMI is connected.

- "Slot"

  There, set the number of the slot in which the S7 module resides. This setting is not required for SIMATIC S7-200 PLCs.

- "Rack"

  There, set the number of the rack in which the S7 module resides. This setting is not required for SIMATIC S7-200 PLCs.

- "Cyclic operation"

  When cyclic operation is enabled, the PLC optimizes its data exchange with the HMI. This increases system performance. Disable cyclic mode if you are operating several HMIs in parallel. This setting is not required for SIMATIC S7-200 PLCs.

### 2.2.3.3 Assigning the MPI address to the SIMATIC S7

### 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.

The MPI address is created in STEP 7.

### MPI address of the communication partner of a SIMATIC S7-300

When assigning addresses, a distinction must be made 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 should communicate.

### MPI address of the FM

The HMI can only communicate with FM modules which are assigned an MPI address. This applies to all FMs which are connected to the communications bus (K-bus).

FMs without MPI address are connected to the P-bus. These includes, for example, the FM 350.

## 2.2.4 PROFIBUS communication

### 2.2.4.1 Requirements of PROFIBUS communication

### Hardware requirements

Hardware components required for PROFIBUS DP networking:

- To connect a Panel PC or standard PC to the PROFIBUS DP interface of the S7 PLC, you require a communications processor (e.g. CP 5512, CP 5611) or an adapter (PC adapter, for example). PC adapters are available for the RS232 (serial) and USB (USB 1.1) ports.
- Use a PROFIBUS cable to connect the HMI device to the SIMATIC S7 PLC.

---

**Note**

At a SIMATIC S7-200 of the first generation (CPU 214, 215, 216), always use port 1 for MPI/PROFIBUS DP communication. When using a CPU 214, set a transmission rate of 9.6 kbps.

---

- Use a serial data cable to connect the HMI to the PC.
- To set up a PROFIBUS DP network, you need a PROFIBUS bus cable with PROFIBUS connector.

---

**Note**

A cable must be terminated with its characteristic wave impedance. Enable the terminating resistor on the first and last node of the PROFIBUS/MPI network.

Both nodes must be supplied with power. Of the two end nodes (first and last) of a PROFIBUS/MPI network, only one can be an OP!

The PROFIBUS bus terminal RS-485 and PROFIBUS connector are equipped with a switch which is used to enable the terminating resistor. Meaning of the ON and OFF switch settings:

OFF: The terminating resistor is disabled, the bus cable is not terminated.

ON: The terminating resistor is enabled, the bus cable is terminated.

If a problem occurs on the connection between an S7-400 (version 1 only) and an OP, a new connection can only be established again after turning off the power on the OP.

---

## Software Requirements

The following software components are required for communication over PROFIBUS-DP:

- WinCC flexible engineering tool
- STEP 7 software for the SIMATIC S7 controller

### 2.2.4.2 Configuring protocol parameters using PROFIBUS DP

## Parameters to set

To edit the parameters, double-click "Communication > Connections" in the project view of the HMI. On the work area, "SIMATIC S7-200" or "SIMATIC S7-300/400" is selected from the "Communication drivers" column. You can now edit the protocol parameters in the properties view:

## Device-specific parameters

Select "HMI device" to set the network parameters for the HMI. You only need to do this once. The modification applies to all communication partners.

- "Interface"

  There, select the HMI interface at which it is connected to the PROFIBUS network.

| HMI | Interface |
|---|---|
| OP 73micro, TP 170micro, TP 177micro<br>Mobile Panel 170, Mobile Panel 177<br>OP 73, OP 77A, OP 77B, TP 170A, TP 170B, OP 170B,<br>TP 177A, TP 177B, OP 177B, TP 270, OP 270,<br>MP 270B, MP 370 | IF1B |
| Panel PC, PC | MPI/PROFIBUS DP |

- "Transmission rate"

  There, set the data transmission rate for the network. The data transmission rate is determined by the slowest HMI on the network. The setting must be consistent on the entire network.

- "Address"

  There, set the PROFIBUS DP address of the HMI. The PROFIBUS DP address must be unique on the PROFIBUS network.

---

### Note

You should assign intermittent addresses, with at least one free address, to the active PROFIBUS nodes when using certain modules. In this case, STEP 7 V5.1 automatically assigns addresses which takes this rule into account.

However, this only affects modules with ASIC Version ASPC2 to STEP D. Extract the current version from the GSD file or the manual.

---

- "Only master on the bus"

  This disables an additional safety function that prevents bus disturbances when the HMI connects to the network. A passive station (slave) can only send data on request by an active station (master.) If you only have slaves connected to the HMI, disable this safety function by setting the "Only master on the bus" check box.

  When operating the HMI on an S7-200 PLC, set the HMI to master mode.

## Network parameters

Select "Network" to set the PROFIBUS parameters of the network to which the HMI is connected. Those network parameters will be applied if you have integrated WinCC flexible in STEP 7, and the HMI is connected to the network

- "Profile"

  There, select the relevant network profile. Set "DP", "Universal" or "Standard." The setting must be consistent on the entire network.

- "Highest station address"

  Set the highest station address. The highest station address must be equal to or higher than the highest PROFIBUS address. The setting must be consistent on the entire network.

### Note

If you set a transmission rate of 1.5 Mbps at the TP 170A, the highest station address must be less than or equal to 63.

- "Number of masters"

  There, set the number of masters on the PROFIBUS network. This information is necessary to ensure the correct calculation of bus parameters.

## Communication partners

Select "Controller" to set the address of the S7 module with which the HMI will exchange data. Assign a connection name for each communication partner.

- "Address"

  There, set the PROFIBUS address of the S7 module (CPU) to which the HMI is connected.

- "Slot"

  There, set the number of the slot in which the S7 module resides. This setting is not required for SIMATIC S7-200 PLCs.

- "Rack"

  There, set the number of the rack in which the S7 module resides. This setting is not required for SIMATIC S7-200 PLCs.

- "Cyclic operation"

  When cyclic operation is enabled, the PLC optimizes its data exchange with the HMI. This increases system performance. Disable cyclic mode if you are operating several HMIs in parallel. This setting is not required for SIMATIC S7-200 PLCs.

## 2.2.5 Ethernet communication

### 2.2.5.1 Requirements of Ethernet communication

#### Hardware requirements

Hardware components required for Ethernet networking:

- To connect a Panel PC or standard PC to the S7 controller over Ethernet, a communications processor (for example CP 1613, CP 1612, or CP 1512) or Ethernet NIC (for example SIMATIC Panel PC On-Board Ethernet adapter, standard Ethernet module) is required.

- Ethernet connections between a Panel PC or standard PC to the S7-200 controller require a communication processor, such as CP243-1.

---

**Note**

To connect an HMI via CP243-1 communication processor, the CP243-1 must be installed in the slot next to the CPU's right side (module no. 0)

---

#### Software Requirements

- SIMATIC S7-300/400

  Software components required for Ethernet communication:

  – WinCC flexible engineering tool

  – STEP 7 software for the SIMATIC S7 PLC

  – IE SOFTNET-S7 LEAN software for PC and Panel PC with Ethernet ON Board interface, or alternatively with CP1512, CP1612

  – S7-1613 software for PC and Panel PC with CP1613

- SIMATIC S7-200

  Software components required for Ethernet communication:

  – WinCC flexible engineering tool

  – MicroWin

### 2.2.5.2 Configuring protocol parameters on Ethernet

#### Parameters to set

To edit the parameters, double-click "Communication > Connections" in the project view of the HMI. On the work area, "SIMATIC S7-200" or "SIMATIC S7-300/400" is selected from the "Communication drivers" column. You can now edit the protocol parameters in the properties view:

## Device-specific parameters

Select "HMI device" to set the network parameters for the HMI. The changes are not transferred automatically to the HMI. Edit the settings in the Control Panel of the HMI.

- "Interface"

  There, select the HMI interface at which the HMI is connected to the MPI network. In this case, set "Ethernet."

| HMI | Interface |
|---|---|
| Mobile Panel 177 PN<br>TP 170B, OP 170B, TP 177B PN/DP, OP 177B PN/DP,<br>TP 270, OP 270<br>MP 270B, MP 370<br>Panel PC, PC | Ethernet |

- "Type"

  The protocol type "IP" is set permanently.

### Note

The "ISO" protocol is not released for the current version of WinCC flexible.

- "Address"

  There, set the IP address of your PC system.

- "Subnet mask"

  There, set the subnet mask for your PC system. The subnet mask is used in combination with the IP address to identify the subnet in which the PC system resides. You can obtain the value from your network administrator.

### Note

On Ethernet, the parameters of the configuration computer will not be transferred to the destination device. The IP address and subnet mask parameters are not assigned on the HMI.

You thus need to configure the IP address and the subnet mask manually at the HMI.

## Communication partners

Select "Controller" to set the address of the S7 module with which the HMI will exchange data. Assign a connection name for each communication partner.

- "Address"

  There, set the IP address of the S7 module to which the HMI is connected.

- "Slot"

  There, set the number of the slot in which the S7 module resides. This setting is not required for SIMATIC S7-200 PLCs.

- "Rack"

  There, set the number of the rack in which the S7 module resides. This setting is not required for SIMATIC S7-200 PLCs.

- "Cyclic operation"

  When cyclic operation is enabled, the PLC optimizes its data exchange with the HMI. This increases system performance.

  Disable cyclic mode if you are operating several HMIs in parallel.

## 2.2.6 Permitted data types

### Permitted data types

The data types listed in the tables below are available for your tag and area pointer configuration.

### SIMATIC S7-300/400

| Name | Object | Data type |
|---|---|---|
| Data block | DB | CHAR, BYTE, INT, WORD, DINT, DWORD, REAL, BOOL, STRING, STRINGCHAR, TIMER, COUNTER, DATE, TIME, DATE AND TIME, TIME OF DAY |
| Flag | M | CHAR, BYTE, INT, WORD, DINT, DWORD, REAL, BOOL, STRING, STRINGCHAR, TIMER, COUNTER, DATE, TIME, DATE AND TIME, TIME OF DAY |
| Input | I | CHAR, BYTE, INT, WORD, DINT, DWORD, REAL, BOOL, STRING, STRINGCHAR |
| Peripheral input | PI | CHAR, BYTE, INT, WORD, DINT, DWORD, REAL, BOOL, STRING, STRINGCHAR |
| Output | O | CHAR, BYTE, INT, WORD, DINT, DWORD, REAL, BOOL, STRING, STRINGCHAR |
| Peripheral output | PO | CHAR, BYTE, INT, WORD, DINT, DWORD, REAL, BOOL, STRING, STRINGCHAR |
| Timer | T | Timer |
| Counter | C | Counter |

### SIMATIC S7-200

| Name | Object | Data type |
|------|--------|-----------|
| Tag | V | CHAR, BYTE, INT, WORD, DINT, DWORD, REAL, BOOL, STRINGCHAR |
| Input | I | CHAR, BYTE, INT, WORD, DINT, DWORD, REAL, BOOL, STRINGCHAR |
| Output | O | CHAR, BYTE, INT, WORD, DINT, DWORD, REAL, BOOL, STRINGCHAR |
| Flag | M | CHAR, BYTE, INT, WORD, DINT, DWORD, REAL, BOOL, STRINGCHAR |
| Timer | T | Timer |
| Counter | C | INT |

### Time and Timer data type

You can configure a start value for numerical tags. Set the start value in ms units when you configure tags of the data type Time or Timer, for example, the value 1000 is equivalent to 1 s.

## 2.2.7 Optimizing the configuration

## 2.2.7.1 Optimizing the configuration

### Acquisition cycle and update time

The acquisition cycles for the "Area pointers" and tags defined in the engineering software are decisive factors for the update times which can actually be achieved.

The update time is equivalent to the acquisition cycle + transmission time + processing time.

To achieve optimum update times, remember the following points in your configuration:

- Set up the minimum and maximum size of the various data areas to suit requirements.
- Define associated data areas. The actual update time improves if you set up one large area, rather than several smaller ones.
- If the acquisition cycles you select are too short, this is detrimental to the overall performance. Set the acquisition cycle to suit the rate of change of the process values. The temperature profile of a furnace, for example, is rather sluggish compared to the speed profile of a drive. Guideline value of the acquisition cycle: approx. 1 s.
- To improve update times, you could do without the cyclic transfer of large numbers of tags. Enter "On request" as the acquisition cycle. Instead, use a deliberate, spontaneous, event-controlled transfer of tags (PLC job.)
- Write the tags of an alarm or a screen without gaps to a single data area.
- To allow the reliable detection of changes in the PLC, these must be available at least during the actual acquisition cycle.
- Set the highest possible transmission rate.

### Screens

The actually feasible update rate of screens depends on the data type and volume to be visualized.

To achieve shorter update times, you should always configure short acquisition cycles only for those objects which actually need to be updated quickly.

### Trends

When using bit-triggered trends, and the group bit is set in the "Trend transfer area", the HMI always refreshes the trends according to the set bits. It then resets the bits.

The group bit in the S7 program can only be set again after all bits have been reset by the HMI.

### PLC jobs

A large transfer volume of PLC jobs transferred rapidly and in succession, may cause an overload in the communication between the HMI and the PLC.

The HMI acknowledges receipt of a PLC job by writing a 0 value to the first data word of the PLC job. The HMI now processes the job, for which it requires further time. When the HMI immediately writes a new PLC job to the jobs mailbox, it may take some time to process the next PLC job. The next PLC job will not be accepted unless the system provides sufficient computing capacity.

### Multiplexing OP connections using SIMATIC NET communications processors

The SIMATIC NET communications processors listed below support multiplexing of OP connections (can be configured in STEP 7 V5.1 SP3 or higher):

- CP342-5 PROFIBUS (V5.0 and higher)

- CP343-1 ETHERNET

Each CP supports up to 16 OP, while requiring only one CPU connection resource. In multiplex mode, do not the rack/slot assignment of the CPU to address OP connections, but rather the rack/slot assignment of the CP. In integrated mode, activate the CP as the partner station, rather than the CPU. Enable multiplexing of OP connections (HMI) in HW Config (STEP 7), on the "Options" properties page of the communications processor. This does not require any communications blocks in the user program of the S7-300 CPU.

---

#### Note

Multiplexed OP connections are only suitable for low data volumes.

---

### Transmission rates of PPI connections (S7-200)

Allowances to be made as a result of the low transmission rates of PPI connections:

- The transfer of data records and the indirect transfer of tags is speeded up significantly if the release of the data mailbox and the positive acknowledgment happen at the same time instead of one after the other.

- When selecting a screen with a bit-triggered trend, wait until the trend values have been updated on the OP before selecting a different screen with a new trend because the area pointers (trend request) are updated less often.

### 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. If no tags are requested, for example no output tags in the current screen, there is no certainty that a breakdown of the connection will be detected.

Configure area pointer coordination for each controller. This ensures that even in the situation described above, the breakdown of the connection will be detected after approximately two minutes.

## 2.3 Direct keys

### 2.3.1 Configuration of DP direct keys

### Application

In addition to their normal use, the F, K and S buttons of the operator panels can also be used as DP direct keys in the configuration. The "DirectKey" function is appended to the configured button for the touch panel. DP direct keys means that a bit is set in the I/O area of the CPU by pressing the key or the button.

The DP direct keys are normal DP entries for the S7 CPU and are therefore configured the same as e.g. an ET 200 station. The cycle time of the DP bus is calculated as the sum of all configured inputs and outputs. Thus, the reaction time of the DP direct keys can also be determined. For a typical DP configuration, the reaction time of the DP direct keys is < 100 ms.

### Requirement

The HMI device must be connected to a SIMATIC S7 PLC over PROFIBUS DP.

WinCC flexible must be installed as "STEP 7 integrated" and the HMI device must be incorporated in the PROFIBUS network. For more detailed information, refer to the "WinCC flexible - Configuring WinCC Windows-based Systems" user manual.

### HMI devices

DP direct keys can be used for the following HMI devices:

- MP 370
- MP 270B
- TP 270, OP 270
- TP 170B, OP 170B
- TP 177B, OP 177B
- Mobile Panel 177
- OP 77B

## Configuration for STEP 7

The HMI device should be configured as an active node for general communication (reading and writing of variables). The HMI device should also be configured as a slave in the PROFIBUS-DP network for the DP direct keys. The following figure shows the basic configuration based on an S7-400.
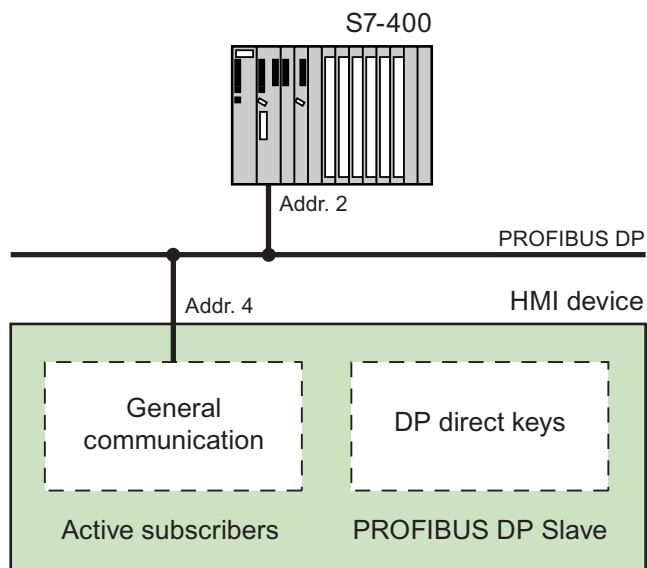


Configuration of the HMI device for DP direct keys

## Basic configuration procedure

Steps 1 to 3 below describe the basic procedure for configuring the HMI device (as a master) for general communication. Steps 4 to 6 describe how to configure the HMI device as a PROFIBUS DP slave in order to use the DP direct keys.

The same address is used to configure the HMI device as an active node and as a DP slave.

1. Create a new project in STEP 7 and configure the hardware with

   – a DP-capable CPU, CPU 414-2 DP

   – assign the PROFIBUS DP an address and connect the CPU to the network.

2. Insert a new "SIMATIC HMI Station" object into your project. Select the desired HMI device for this. You will see the inserted HMI device in your STEP 7 project:
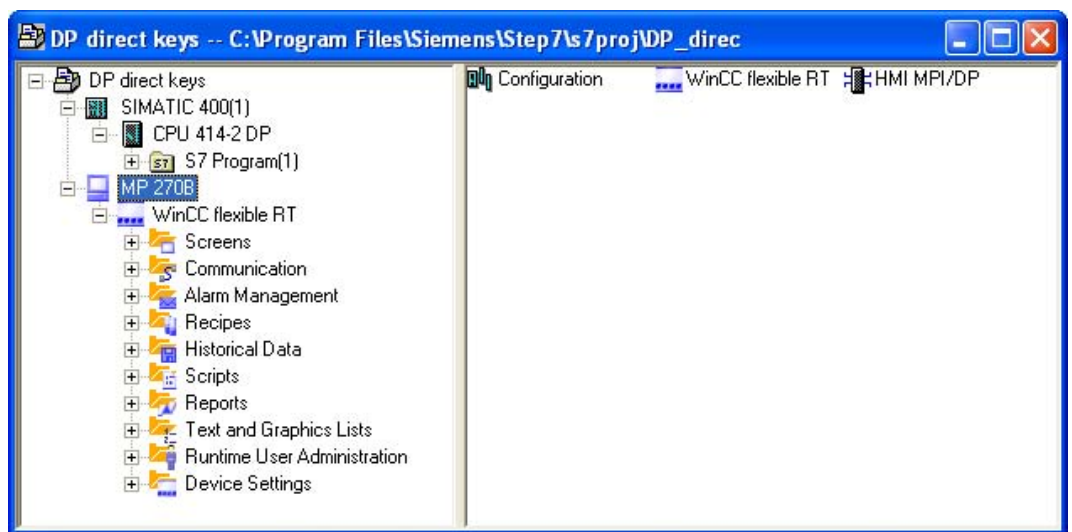


Configuring the HMI device in the STEP 7 project

3. Double-click on "Configuration" and connect the HMI device to the same PROFIBUS DP network as the one for the CPU.

4. Open the hardware configuration of the CPU and drag the configured HMI device type from the hardware catalog under "Configured stations > SIMATIC HMI Station" to the displayed PROFIBUS DP:
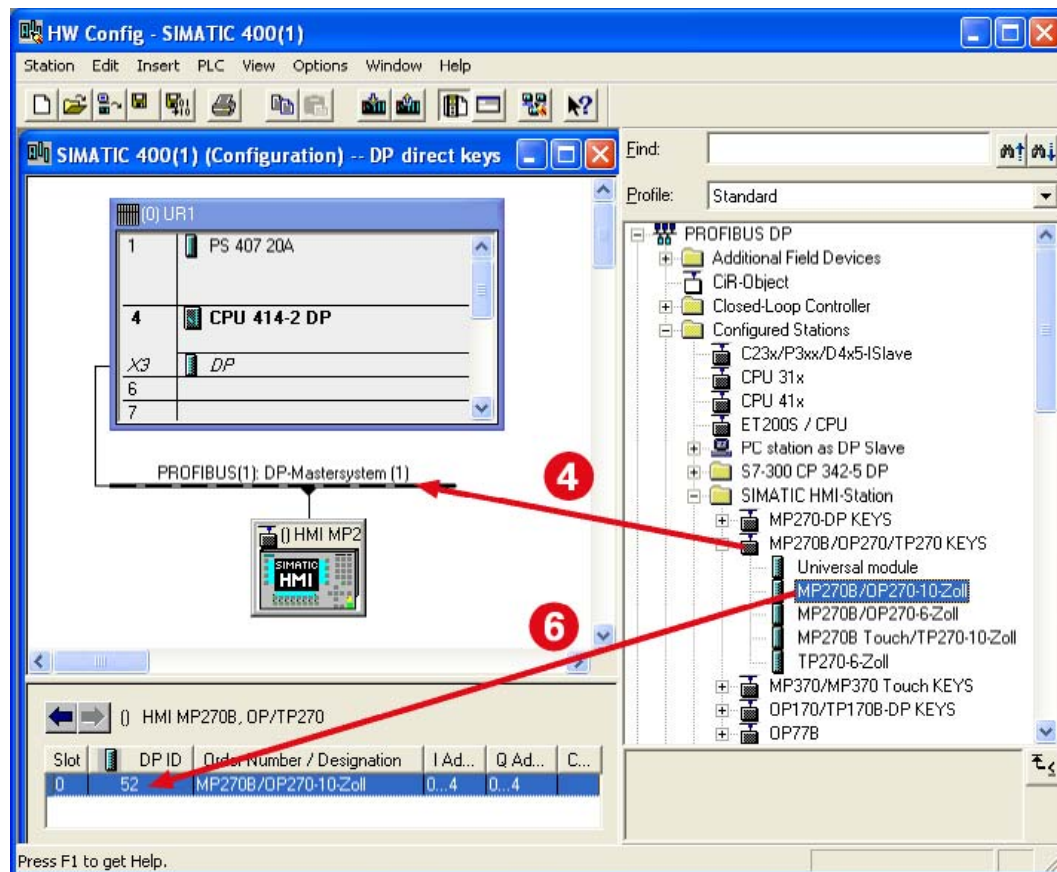


Configuration in the STEP 7 project for DP direct keys (example)

5. Once you have inserted the configured HMI device into the station window, the "Properties DP Slave" dialog opens. Select the DP slave and connect it to the PLC by clicking the "Connect" button.
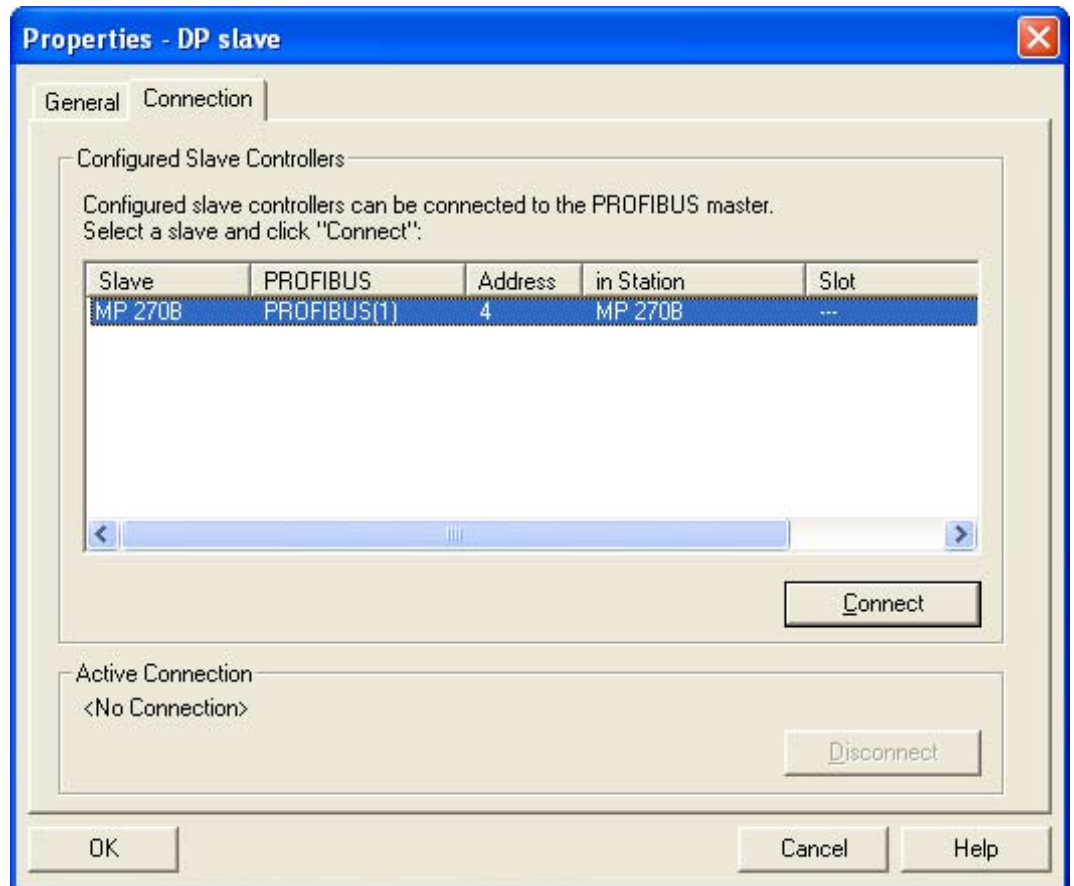


Connecting the HMI device as a slave

6. Specify the type of HMI device used by selecting the HMI device in the hardware catalog and dragging it into the detailed view of the station window. The type and address of the HMI device are displayed in the detailed view.

## Assigning the inputs/outputs

The keys or buttons of the HMI device are assigned to bytes in the DP input area while the LEDs are assigned to bytes in the DP output area. The following table shows the number of bytes used by the various HMI devices and the exact assignments are shown in the succeeding diagrams.

Touch panels do not have any permanently assigned keys. They only have user-configurable buttons. You can assign a button a bit in the DP input area using the Direct Keys function. The direction in which the bits are counted in the DP input area 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.

### Assignment for the DP inputs/outputs

| HMI device | Inputs | Outputs |
|---|---|---|
| MP 370 key device, MP 270B, OP 270-10" | 5 bytes | 5 bytes |
| MP 370 touch device, TP 270-10" | 5 bytes | – |
| OP 270-6", OP 170 | 4 bytes | 4 bytes |
| TP 270-6", TP 170 | 4 bytes | – |
| OP 177B | 9 bytes | 4 bytes |
| TP 177B | 4 bytes | – |
| Mobile Panel 177 | 9 bytes | 4 bytes |
| OP 77B | 4 bytes | 4 bytes |

### Input/output assignment for MP 370

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | S8 | S7 | S6 | S5 | S4 | S3 | S2 | S1 | n+0 | S8 | S7 | S6 | S5 | S4 | S3 | S2 | S1 |
| | S16 | S15 | S14 | S13 | S12 | S11 | S10 | S9 | n+1 | S16 | S15 | S14 | S13 | S12 | S11 | S10 | S9 |
| Keys | F8 | F7 | F6 | F5 | F4 | F3 | F2 | F1 | n+2 | F8 | F7 | F6 | F5 | F4 | F3 | F2 | F1 |
| | F16 | F15 | F14 | F13 | F12 | F11 | F10 | F9 | n+3 | F16 | F15 | F14 | F13 | F12 | F11 | F10 | F9 |
| | ACK | ALT | CTRL | SHIFT | F20 | F19 | F18 | F17 | n+4 | ACK | A-Z left | A-Z right | INFO | F20 | F19 | F18 | F17 |

Direct keys assignment — LED

Assignment of the keys /LED in the input and output areas for MP 370

### Input/output assignment for MP 270B and OP 270-10"

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | F8 | F7 | F6 | F5 | F4 | F3 | F2 | F1 | n+0 | F8 | F7 | F6 | F5 | F4 | F3 | F2 | F1 |
| | F16 | F15 | F14 | F13 | F12 | F11 | F10 | F9 | n+1 | | | | F12 | F11 | F10 | F9 |
| Keys | K4 | K3 | K2 | K1 | F20 | F19 | F18 | F17 | n+2 | K4 | K3 | K2 | K1 | | | | |
| | K12 | K11 | K10 | K9 | K8 | K7 | K6 | K5 | n+3 | K12 | K11 | K10 | K9 | K8 | K7 | K6 | K5 |
| | ACK | ALT | CTRL | SHIFT | K16 | K15 | K14 | K13 | n+4 | ACK | A-Z left | A-Z right | INFO | K16 | K15 | K14 | K13 |

Direct keys assignment — LED

Assignment of the keys /LED in the input and output areas for MP 270 and OP 270 10"

## Input/output assignment for OP 270-6" and OP 170

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Keys | F8 | F7 | F6 | F5 | F4 | F3 | F2 | F1 | n+0 | F8 | F7 | F6 | F5 | F4 | F3 | F2 | F1 |
| | K2 | K1 | F14 | F13 | F12 | F11 | F10 | F9 | n+1 | K2 | K1 | | | | | | |
| | K10 | K9 | K8 | K7 | K6 | K5 | K4 | K3 | n+2 | K10 | K9 | K8 | K7 | K6 | K5 | K4 | K3 |
| | ACK | ALT | CTRL | SHIFT | | | | | n+3 | ACK | A-Z left | A-Z right | INFO | | | | |

Direct keys assignment | LED

Assignment of the keys /LED in the input and output areas for MP 270 6" and OP 170

## Input/output assignment for OP 177B

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Membrane keys | F8 | F7 | F6 | F5 | F4 | F3 | F2 | F1 | n+0 | F8 | F7 | F6 | F5 | F4 | F3 | F2 | F1 |
| | K2 | K1 | F14 | F13 | F12 | F11 | F10 | F9 | n+1 | K2 | K1 | | | | | | |
| | K10 | K9 | K8 | K7 | K6 | K5 | K4 | K3 | n+2 | K10 | K9 | K8 | K7 | K6 | K5 | K4 | K3 |
| | K18 | K17 | K16 | K15 | K14 | K13 | K12 | K11 | n+3 | K18 | K17 | K16 | K15 | K14 | K13 | K12 | K11 |
| | | | | | | | | | n+4 | | | | LED | | | | |
| Touch buttons | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | n+5 | | | | | | | | |
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | n+6 | | | | | | | | |
| | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | n+7 | | | | | | | | |
| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | n+8 | | | | | | | | |

Direct keys assignment

Key/touch button/LED assignment in the input and output area for OP 177B

## Input/output assignment for TP 177B

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Byte |
|---|---|---|---|---|---|---|---|---|
| 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 |

Touch buttons

Direct keys assignment       No output area       LED

Touch button assignment in the input area for TP 177B

## Input/output assignment for Mobile Panel 177

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F8 | F7 | F6 | F5 | F4 | F3 | F2 | F1 | n+0 | F8 | F7 | F6 | F5 | F4 | F3 | F2 | F1 |
|  |  | F14 | F13 | F12 | F11 | F10 | F9 | n+1 |  |  |  |  |  | T1 |  | S0 |
|  |  |  |  |  | T0 | S1 | S0 | n+2 | K10 | K9 | K8 | K7 | K6 | K5 | K4 | K3 |
| I7 | I6 | I5 | I4 | I3 | I2 | I1 | I0 | n+3 | K18 | K17 | K16 | K15 | K14 | K13 | K12 | K11 |
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | n+4 |  |  |  |  | LED |  |  |  |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | n+5 |  |  |  |  |  |  |  |  |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | n+6 |  |  |  |  |  |  |  |  |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | n+7 |  |  |  |  |  |  |  |  |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | n+8 |  |  |  |  |  |  |  |  |

Membrane keys / Touch buttons

Direct keys assignment

Key/touch button/LED assignment in the input and output area for Mobile Panel 177

### Input/output assignment for OP 77B

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Membrane keys | | | | | K4 | K3 | K2 | K1 | n+0 | | | | | K4 | K3 | K2 | K1 |
| | | | | | F4 | F3 | F2 | F1 | n+1 | | | | | | | | |
| | | | | | | | | | n+2 | | | | | | | | |
| | | | | | | | | | n+3 | | | | | | | | |
| | | | | Direct keys assignment | | | | | | | | | LED | | | | |

Key/LED assignment in the input and output area for OP 77B

### PROFIBUS screen number (touch devices only)

If PROFIBUS direct keys use the same bits for different functions on different screens, the S7 needs to distinguish the various functions by means of the screen number. In order to avoid the delayed update of the screen number in the controller after a screen change, the PROFIBUS screen number screen function is available.

The "PROFIBUSSreenNumber" function allows you to set any bit in the DP input area to identify the screen and transfer it to the PLC at the same time as the direct key bits. This ensures unambiguous allocation of a control bit to screen number at all times..

## 2.4     User data areas

### 2.4.1     Area pointer

#### 2.4.1.1     General information about area pointers (SIMATIC S7)

#### Introduction

Area pointers are parameter fields from which WinCC flexible Runtime obtains information about the location and size of data areas in the controller. During communication, the controller and the HMI device read and write information for each other in these data areas. By evaluating the data entered in these areas by the partner device, the controller and HMI device trigger defined actions.

Physically, the area pointers reside in the memory of the controller. Their addresses are set up during configuration in the "Connections" editor in "Coordination".

WinCC flexible uses the following area pointers:

* Control request
* Project ID
* Screen number

- Data set
- Date/time
- Date/time controller
- Coordination

## Application

Before an area pointer can be used, it must be set up and activated by selecting "Communication ▸ Connections."



Activating an area pointer based on a SIMATIC S7 controller

- Active

  Activates the area pointer.
- Name

  Default name of the area pointer set by WinCC flexible.
- Address

  Here, enter the tag address of the area pointer In the PLC.
- Length

  The default length of the area pointer is set by WinCC flexible.
- Acquisition cycle

  An acquisition cycle must be specified here, so that an area pointer is read periodically by Runtime. Make allowances for the fact that a shorter acquisition time may have a negative influence on HMI performance.
- Comment

  Here, you can enter a comment, for example to describe the use of the area pointer.

## Functional scope

Which area pointers are actually available depends on the HMI device you are using. The following table provides you with an overview of the range of functions of the various HMI devices:

| Area pointer | PC Panel PC | MP 270B MP 370 | TP 270 MP 270 OP 270 | Mobile Panel 170 Mobile Panel 177 OP 77B TP 170B OP 170B TP 177B OP 177B | OP73 OP 77A OP 177A | OP 73micro TP 170micro TP 177micro TP 170A |
|---|---|---|---|---|---|---|
| Screen number | x | x | x | x | x | – |
| Data set | x | x | x | x | – | – |
| Date/time | x | x | x | x | x | – |
| Date/time controller | x | x | x | x | x | x |
| Coordination | x | x | x | x | x | – |
| Project ID | x | x | x | x | – | – |
| Control request | x | x | x | x | x | – |
| [1]OP 73micro , TP 170micro and TP 170micro devices only support communication with an S7-200 PLC. | | | | | | |

The following table shows how the controller and HMI device access the various data areas, reading (R) or writing (W).

| Data area | Required for | HMI device | Controller |
|---|---|---|---|
| Screen number | Evaluation by the controller to determine which screen is currently open | W | R |
| Data set | Transfer of data mailboxes with synchronization | R / W | R / W |
| Date/time | Transfer of date and time from the HMI device to the controller | W | R |
| Date/time controller | Transfer of date and time from the controller to the HMI device | R | W |
| Coordination | Querying the status of the HMI device in the controller program | W | R |
| Project ID | Runtime checks whether or not the WinCC flexible user version and the project on the controller are consistent | R | W |
| Control request | Triggering functions on the HMI device by the controller program | R / W | R / W |

The following sections explain the area pointers and the assigned job mailboxes.

## 2.4.1.2 "Screen number" area pointer

### Function

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

This allows information on the current screen content of the HMI device to be transferred to the controller and to trigger certain reactions from the controller, for example calling up a different screen.

### Application

Before the "Screen number" area pointer can be used, it must be set up and activated by selecting "Communication ▸ Connections." It can be created only in **one** controller and only **once** on this controller.

The screen number is transferred spontaneously to the controller; in other words, it is transferred whenever a new screen is selected on the HMI device. It is therefore unnecessary to configure an acquisition cycle.

### Layout

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

|  | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1. Word | Current screen type |||||||||||||||
| 2. Word | Current screen number |||||||||||||||
| 3. Word | Reserved |||||||||||||||
| 4th word | Current field number |||||||||||||||
| 5. Word | Reserved |||||||||||||||

- Current screen type

  "1" for root screen or
  "4" for permanent window

- Current screen number

  1 to 65535

- Current field number

  1 to 65535

## 2.4.1.3 "Date/time" area pointer

### Function

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

The controller writes job mailbox "41" to the jobs mailbox.

When it evaluates the job mailbox, 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 data area has the following structure:

| Data word | Most significant byte | | Least significant byte | | |
|:---:|:---:|:---:|:---:|:---:|:---:|
| | 7 | 0 | 7 | 0 | |
| n+0 | Reserved | | Hour (0-23) | | Time |
| n+1 | Minute (0-59) | | Second (0-59) | | |
| n+2 | Reserved | | Reserved | | |
| n+3 | Reserved | | Weekday (1-7, 1=Sunday) | | Date |
| n+4 | Day (1-31) | | Month (1-12) | | |
| n+5 | Year (80-99/0-29) | | Reserved | | |

---

**Note**

Note that when you enter the year, values 80-99 result in years 1980 through 1999 and the values 0-29 result in the years 2000 through 2029.

---

## 2.4.1.4 "Date/time controller" area pointer

### Function

This area pointer is used to transfer the date and time from the controller to the HMI device. This is useful when the controller is time master.

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

The HMI device reads the data cyclically at the configured acquisition interval and synchronizes itself.

---

**Note**

Do not configure an acquisition cycle for the date/time area pointer that is too short, since this can be detrimental to the performance of the HMI device.
Recommendation: Acquisition cycle of 1 minute, if the process allows this.

---

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 (80-99/0-29) | | | Month (1-12) | | |
| n+1 | Day (1-31) | | | Hour (0-23) | | |
| n+2 | Minute (0-59) | | | Second (0-59) | | |
| n+3 | Reserved | | | Reserved | Weekday (1-7, 1=Sunday) | |
| n+4 [1] | Reserved | | | Reserved | | |
| n+5 [1] | Reserved | | | Reserved | | |

> [1] The two data words must be present in the data area in order to ensure a correlation of the data format with WinCC flexible and to avoid the reading of the wrong information.

### Note

Note that when you enter the year, values 80-99 result in years 1980 through 1999 and the values 0-29 result in the years 2000 through 2029.

## 2.4.1.5 "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

Its length is two words.

### Application

### Note

Each time the area pointer is updated by the HMI device, the entire coordination area is always written.
For this reason, the PLC program must not make any changes in 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. After startup, the bit is set permanently to "1."

## Operating mode

As soon as the HMI device is switched offline by the user, the operating mode bit is set to 1. In normal operation of the HMI device, the state of the operating mode bit is "0." You can find out the current operating mode of the HMI device by querying this bit.

## Life bit

The life bit is inverted by the HMI device at intervals of approximately one second. By querying this bit in the controller program, you can check whether or not the connection to the HMI device still exists.

### 2.4.1.6 "User version" area pointer

## Function

When runtime starts, it is possible to check whether the HMI device is connected to the correct controller. This is important when several HMI devices are being used.

The HMI device compares a value stored on the controller with the value specified in the configuration data. This ensures the compatibility of the configuration data and the controller program. If there is a discrepancy, a system alarm is displayed on the HMI device and runtime is stopped.

## Application

If you want to use this area pointer, set up the following during configuration.

- Specify the version of the configuration data. Possible values between 1 and 65535.

  Enter the version in the "Device settings ▸ Device settings" editor in "Project ID."

- Data address of the value of the version stored on the controller:

  Enter the data address in the "Communication ▸ Connections" editor in "Address."

## 2.4.1.7 "Job mailbox" area pointer

### Function

Using the jobs mailbox, job mailboxes can be given to the HMI device so that actions are triggered 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 ||

If the first word of the job mailbox is a value other than 0, the HMI device evaluates the job mailbox. 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

Below, you can see all the job mailboxes and their parameters available on the various HMI devices. The "No." column contains the job number of the job mailbox. Job mailboxes can only be triggered by the controller when the HMI device is online.

### Note

There are no job mailboxes for the TP 170A HMI device.

| No. | Function | | PC Panel PC | Panel [1) |
|-----|----------|--|-------------|-----------|
| 14 | Set time (BCD-coded) | | x | x |
| | Parameter 1 | Left byte:  - <br> Right byte:  hours (0-23) | | |
| | Parameter 2 | Left byte:  minutes (0-59) <br> Right byte:  seconds (0-59) | | |
| | Parameter 3 | - | | |

| No. | Function | | PC<br>Panel PC | Panel [1) |
|---|---|---|---|---|
| 15 | Set date (BCD-coded) | | x | x |
| | Parameter 1 | Left byte: -<br>Right byte: weekday<br>          (1-7: Sunday-Saturday) | | |
| | Parameter 2 | Left byte: day (1-31)<br>Right byte: month (1-12) | | |
| | Parameter 3 | Left byte: year | | |
| 23 | User login | | x | x |
| | Logs the user on with the name "PLC user" at the HMI device with the group number transferred in parameter 1.<br>The logon is possible only when the transferred group number exists in the project. | | | |
| | Parameter 1 | Group number 1 - 255 | | |
| | Parameter 2, 3 | - | | |
| 24 | User logoff | | x | x |
| | Logs off the currently logged on user.<br>The function corresponds to the "logoff" system function) | | | |
| | Parameter 1, 2, 3 | - | | |
| 40 | Transfer date/time to controller | | x | x |
| | (in the S7 DATE_AND_TIME format)<br>There should be at least 5 seconds between two jobs, otherwise the HMI device will be overloaded. | | | |
| | Parameter 1, 2, 3 | - | | |
| 41 | Transfer date/time to controller | | x | x |
| | (in the OP/MP format)<br>There should be at least 5 seconds between two jobs, otherwise the HMI device will be overloaded. | | | |
| | Parameter 1, 2, 3 | - | | |
| 46 | Updating tags | | x | x |
| | Causes the HMI device to read the current value of the tags from the controller whose update ID matches the value transferred in parameter 1.<br>(Function corresponds to the "UpdateTag" system function.) | | | |
| | Parameter 1 | 1 - 100 | | |
| 49 | Clear event buffer | | x | x |
| | Parameter 1, 2, 3 | - | | |
| 50 | Clear error alarm buffer | | x | x |
| | Parameter 1, 2, 3 | - | | |
| 51 | Display selection | | x | x |
| | Parameter 1 | Screen number | | |
| | Parameter 2 | - | | |
| | Parameter 3 | Field number | | |

| No. | Function | | PC<br>Panel PC | Panel [1] |
|-----|----------|---|----|-------|
| 69 | Read data mailbox from controller | | x | x |
| | Parameter 1 | Recipe number (1-999) | | |
| | Parameter 2 | Data mailbox number (1-65535) | | |
| | Parameter 3 | 0: Do not overwrite existing data record<br>1: Overwrite existing data mailbox | | |
| 70 | Write data mailbox to controller | | x | x |
| | Parameter 1 | Recipe number (1-999) | | |
| | Parameter 2 | Data mailbox number (1-65535) | | |
| | Parameter 3 | - | | |

[1] These include the HMI devices MP 370, MP 270B, TP 270, MP 270, OP 270, TP 170B, OP 170B, Mobile Panel 170 and OP 77B

## 2.4.1.8 "Data mailbox" area pointer

### "Data mailbox" area pointer

### Function

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

### Data transfer types

There are two ways of transferring data mailboxes between the HMI device and controller:

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

Data records are always transferred directly, i.e. tag values are directly read from the address or written to the address that was configured for the tag, without any detour through temporary storage.

### 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 the data mailboxes can also be triggered by the controller.

- Triggering by configured functions

If the transfer of data mailboxes is triggered by a configured function or by a job mailbox, the recipe display on the HMI device remains operable since the data mailboxes 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.

## Transfer without synchronization

If you select asynchronous transfer of data mailboxes between the HMI device and controller, 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 partner.

- The controller does not require information about the recipe number and data mailbox number.

- The transfer of data records is triggered by the operator on the HMI device.

## Reading values

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

- Triggering by the operator in the recipe view:

  The values are loaded on the HMI device. There you can then edit them, for example, modify values, save etc.

- Triggering by a function or job mailbox:

  The values are saved to the data medium immediately.

## Writing values

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

- Triggering by the operator in the recipe view:

  The current values are written to the controller.

- Triggering by a function or job mailbox:

  The current values are written to the controller from the data medium.

## Transfer with synchronization (SIMATIC S7)

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 mailbox transfer can be a useful alternative, for example when

- the controller is the "active partner" in the transfer of data mailboxes,

- Information about the recipe number and data mailbox number needs to be evaluated on the controller.

- The transfer of data mailboxes is triggered by a job mailbox.

## Requirement

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

- An area pointer has been set up: "Communication ▸ Connections" editor in "Area pointer"

- The controller with which the HMI device synchronizes transfer of data mailboxes is specified in the recipe. "Recipes" editor, properties window of the recipe, "Properties" group in "Transfer".

## Structure of the data area

The data area has a fixed length of 5 words and is structured as follows:

|         | 15 |                                     | 0 |
|---------|----|-------------------------------------|---|
| 1. Word | Current recipe number (1 - 999)               |||
| 2. Word | Current data mailbox 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 | Transfer active                    |
| 4       | 0000 0100 | Transfer completed without error   |
| 12      | 0000 1100 | Transfer completed with error      |

## Sequence of a transfer started by the operator in the recipe display

### Reading from the controller started by the operator in the recipe display

| Step | Action | |
|------|--------|--|
| 1 | Check: Status word = 0? | |
| | yes | No |
| 2 | The HMI device enters the recipe number to be read and the status "Transfer active" in the data mailbox and sets the data mailbox number to 0. | Abort with system alarm. |
| 3 | The HMI device reads the values from the controller and displays them in the recipe display.<br>If the recipes have synchronized tags, the values from the controller are also written to the tags. | |
| 4 | The HMI device sets the status "Transfer completed." | |
| 5 | To allow further transfers, the controller program must set the status word to 0 again. | |

### Writing to the controller started by the operator in the recipe display

| Step | Action | |
|------|--------|--|
| | Check: Status word = 0? | |
| 1 | yes | No |
| | The HMI device enters the recipe and data mailbox number to be written and the status "Transfer active" in the data mailbox. | Abort with system alarm. |
| 2 | The HMI device writes the current values to the controller.<br>If the recipes have synchronized tags, the changed values are synchronized between the recipe display and tags and then written to the controller. | |
| 3 | The HMI device sets the status "Transfer completed." | |
| 4 | If required, the controller program can now evaluate the transferred data. | |
| 5 | To allow further transfers, the controller program must set the status word to 0 again. | |

## Sequence of the transfer triggered by a job mailbox SIMATIC S7

Transfer of data mailboxes between the HMI device and controller can be initiated as follows:

- By the HMI device or
- By the controller

The two controller jobs No. 69 and No. 70 are available for this type of transfer.

## No. 69: Read data mailbox from controller ("PLC → DAT")

Job mailbox no. 69 transfers data mailboxes from the controller 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 mailbox number (1-65,535) | |
| Word 4 | Do not overwrite existing data mailbox: 0 Overwrite existing data mailbox: 1 | |

## No. 70: Write data mailbox to controller ("DAT → PLC")

Job mailbox no. 70 transfers data mailboxes from the HMI device to the controller. 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 mailbox number (1-65,535) | |
| Word 4 | — | |

## Sequence when reading from the controller 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 mailbox number specified in the job and the status "Transfer active" in the data mailbox. | Abort without return message. |
| 3 | The HMI device reads the values and stores the values in the data mailbox specified in the job mailbox. | |
| 4 | • If "Overwrite" was selected in the job, an existing data mailbox 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 mailbox already exists, the HMI device aborts the job and enters 0000 1100 in the status word of the data mailbox. | |
| 5 | To allow further transfers, the controller program must set the status word to 0 again. | |

### Sequence writing to the controller 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 mailbox number specified in the job and the status "Transfer active" in the data mailbox. | Abort without return message. |
| 3 | The HMI device fetches the values of the data mailbox specified in the job from the data medium and writes the values to the controller. | |
| 4 | The HMI device sets the status "Transfer completed." | |
| 5 | The controller program can now evaluate the transferred data. To allow further transfers, the controller program must set the status word to 0 again. | |

### Sequence of the transfer when triggered by a configured function

### Reading from the controller 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 "Transfer active" in the data mailbox. | Abort with system alarm. |
| 3 | The HMI device reads the values from the controller and stores them in the data mailbox specified in the function. | |
| 4 | • If "Yes" was selected for the "Overwrite" function, an existing data mailbox is overwritten without any prompt for confirmation.<br><br>The HMI device sets the status "Transfer completed."<br><br>• If "No" was selected for the "Overwrite" function and the data mailbox already exists, the HMI device aborts the job and enters 0000 1100 in the status word of the data mailbox. | |
| 5 | To allow further transfers, the controller program must set the status word to 0 again. | |

## Writing to the controller with 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 "Transfer active" 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 writes the values to the controller. | |
| 4 | The HMI device sets the status "Transfer completed." | |
| 5 | The controller program can now evaluate the transferred data. To allow further transfers, the controller program must set the status word to 0 again. | |

## Possible causes of error when transferring data records

## Possible causes of error

If the transfer of data records is terminated with error, this may be the result of one of the following causes:

- Tag address not set up on the controller
- Overwriting data records not possible
- Recipe number does not exist
- Data record number does not exist

### Note

To ensure data consistency, the evaluation of the recipe and data mailbox number on the controller must only be started after the status in the data mailbox is set to "Transfer completed" or "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 display

  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 nevertheless evaluate the status of the transfer by querying the status word in the data mailbox.

## 2.4.2 Events, alarms, and acknowledgments

### 2.4.2.1 General information on events, alarms, and acknowledgments

**Function**

Alarms provide the user with information on controller or HMI device operating states or problems by displaying them on the HMI device. The alarm texts consist of configurable texts and/or tags with actual values.

Alarms and events must be distinguished. The programmer defines what is an event and what is an alarm.

**Events**

An event indicates a status, for example

- Motor on
- Controller in manual mode

**Alarm**

Alarms indicate operational errors, for example

- Valve does not open
- Excess motor temperature

Alarms indicate exceptional operational states, and must therefore be acknowledged.

**Acknowledgment**

Alarms can be acknowledged

- By operator input at the HMI device
- By the controller setting an acknowledgment bit.

**Triggering alarms**

An alarm is triggered on the controller

- By setting a tag bit
- By a measured value violating a limit.

The location of tags, or of the tag array, is defined in WinCC flexible ES. The tag or array must be set up on the controller.

**Number of alarms**

The table below shows the number of alarms and words available for the various HMI devices.

| HMI device | Events and alarms | |
|---|---|---|
| | Words in total | Alarms in total |
| Standard PC | 250 | 4000 |
| MP 370 | 250 | 4000 |
| TP 270, MP 270, OP 270, MP 270B | 250 | 4000 |
| TP 177B, OP 177B Mobile Panel 177 | | |
| TP 177micro | | 500 |
| TP 170B, OP 170B Mobile Panel 170 | 125 | 2000 |
| TP 170A [1] | 63 | 1000 |
| TP 170micro | | 500 |
| OP 77B | 63 | 1000 |

[1]Only events possible.

## 2.4.2.2    Step 1: Creating tags or an array

**Procedure**

You create tags or arrays in the "Tags" editor. The dialog box is shown below.



- Specifying the names of the tags or the name of the array
- Select the connection to the controller.

    The connection must already be configured in the "Connections" editor.

- Select the data type

    The available data types depend on the controller being used. If you select an illegal data type, the tag will not be available in the "Discrete alarms" and "Analog alarms" editors.

The following data types are supported for SIMATIC S5 controllers:

| Controller | Permitted data types | |
|---|---|---|
| | Discrete alarms | Analog alarms |
| AS 300/400 | WORD, INT | CHAR, BYTE, INT, WORD, DINT, DWORD, REAL, COUNTER, TIME |

- Specify the address

  The tag addressed here contains the bit that triggers the alarm

  As soon as the bit of the tag is set on the controller and is transferred to the HMI device in the configured acquisition cycle, the HMI device recognizes the alarm as "incoming."

  After the same bit is reset on the controller, the HMI device recognizes the alarm as "outgoing."

- Specify array elements

  If the number of array elements is increased, you can select more bit numbers in the "Bit messages" editor. An array with 3 words, for example, makes 48 alarm bits available.

### 2.4.2.3    Step 2: Configuring an alarm

#### Procedure

Alarms fall into the following categories:

- Discrete alarms
- Analog alarms

You create alarms in the "Discrete alarms" and "Analog alarms" editors.

#### Discrete alarms

The editor is shown in the figure below.



- Edit text

  Here, you enter the text that will be shown in runtime. It can be formatted in fonts and contain fields for the visualization of tags.

  The text appears, for example, when an alarm view was configured in the "Screens" editor.

- Specify number

  Every alarm has a number that must only occur once within the project. It is used to uniquely identify the alarm and is indicated with the alarm in runtime.

  The permitted range of values is 1 to 32767.

  The WinCC flexible engineering system assigns consecutive numbers. You can edit these numbers, for example, if you want to organize the numbers in groups.

- Specify the alarm class

  Available alarm classes:

  – Error alarms

    This class must be acknowledged.

  – Process events

    This class signals events with incoming and outgoing alarms.

  – User-defined alarm classes

- Assign trigger tag

  In the "Trigger tag" column, you link the configured alarm with the tag created in step 1. All tags with permitted data types are shown in the selection list.

- Specify the bit number

  In the "bit number" column, specify the relevant bit position in the created tag.

  Remember that the way the bit positions are counted depends on the particular controller. With SIMATIC S5 controllers, the bit positions are counted as follows:

## Counting method in memory bit area and data block

| How the bit positions are counted | Byte 0 | | | | | | | | Byte 1 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Most significant byte | | | | | | | | Least significant byte | | | | | | | |
| in SIMATIC S5 controllers | 7 | | | | | | | 0 | 7 | | | | | | | 0 |
| In the WinCC flexible, you configure the following: | 15 | | | | | | | 8 | 7 | | | | | | | 0 |

## Analog alarms

The only difference between discrete alarms and analog alarms is that you configure a limit value, rather than a bit number. If the limit value is violated, the alarm is triggered. The outgoing alarm is triggered when the low limit is violated, and where applicable, taking into account any configured hysteresis.

## 2.4.2.4    Step 3: Configure the acknowledgment

### Procedure

Create suitable tags on the controller to acknowledge an alarm. You assign these tags to an alarm in the "Bit messages" editor. You make the assignment in "Properties ▸ Acknowledgment",

The following figure shows the dialog for configuring an acknowledgment.



The following distinctions are made for acknowledgments:

- Acknowledgment on the HMI device
- Acknowledgment by the PLC

## Acknowledgment by the PLC

In "Ack write tag", you configure the tag or the array tag and the bit number based on which the HMI device can recognize an acknowledgment by the controller.

A bit set in the tag acknowledges the assigned alarm bit on the HMI device and, therefore, has the same function as an acknowledgment from the HMI device, for example, by pressing the "ACK" key.

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

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



## Acknowledgment on the HMI device

In "Ack read tag", you configure the tag or the array tag and the bit number that is written to the controller after acknowledgment from the HMI device.

To ensure that a signal change always occurs when the acknowledgment bit is set, the HMI device first resets the acknowledgment bit assigned to an alarm as soon as the alarm bit is set. There is a certain time-based offset between these two operations, due to the processing time of the HMI device.

If the alarm is acknowledged on the HMI device, the bit is then set in the assigned acknowledgment tag on the controller. This allows the controller to recognize that the alarm has been acknowledged.

The figure below shows the pulse diagram.

## Function

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

## Time-triggered trends

The HMI device reads in the trend values cyclically at an interval specified in the configuration. Time-triggered trends are suitable for continuous processes, for example, the operating temperature of a motor.

## Bit-triggered trends

By setting a trigger bit in the tag trend transfer, the HMI device either reads in a trend value or an entire trend buffer. This is specified in the configuration. Bit-triggered trends are normally used to visualize rapidly changing values. One example might be the injection pressure in the production of plastic parts.

To trigger bit-triggered trends, create suitable external tags in the "Tags" editor of WinCC flexible and interconnect those to trend areas. The HMI device and controller then communicate with each other over these trend areas.

The following areas are available for trends:

- Trend request area

- Trend transfer area 1

- Trend transfer area 2 (required only with switch buffers)

Assign a bit to each trend. Valid are tags and array tags of the data type "Word" or "Int."

## Trend request area

If a screen is opened on the HMI device with one or more trends, the HMI device sets the corresponding bits in the trend request area. After deselecting the screen, the HMI device resets the relevant bits in the trend request area.

Using the trend request area, the controller can recognize which trend is currently displayed on the HMI device. Trends can also be triggered without evaluating the curve request area.

## Trend transfer area 1

This area is used to trigger trends. In your controller program, you set the bit assigned to the trend in the trend transfer area and set the trend group bit. The trend group bit is the last bit in the trend transfer area.

The HMI device recognizes the trigger and either reads a value or the entire buffer from the controller. It then resets the trend bit and the trend group bit.

The following picture shows the structure of a trend transfer area.



Trend group bit

The trend transfer area must not be modified by the controller 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. It is structured exactly the same as trend transfer area 1.

## Switch buffers

The switch buffer is a second buffer for the same trend that can be set up during configuration.

While the HMI device reads the values from buffer 1, the controller writes to buffer 2. If the HMI device is reading buffer 2, the controller writes to buffer 1. This prevents the trend values being overwritten by the controller while the trend is being read out by the HMI device.

## 2.4.3    LED mapping

### Function

The HMI devices Operator Panel (OP), Multi Panel (MP) and Panel PC have LEDs in their function keys. These LEDs can be activated by the controller. In a particular situation, this allows the controller to activate an LED, for example, to tell the operator which key to press.

### Requirement

To be able to activate an LED, an LED tag or an array tag must be set up on the controller and specified as an LED tag during configuration.

### LED assignment

The assignment of individual LEDs to the bits in the LED tags is specified when you configure the function keys. You specify the "LED tag" and the assigned "bit" in the properties window in the "General" group.

The bit number "bit" identifies the first of two consecutive bits that control the following LED states:

| Bit n+ 1 | Bit n | LED function |
|----------|-------|--------------|
| 0 | 0 | Off |
| 0 | 1 | Flashing |
| 1 | 0 | Flashing |
| 1 | 1 | On permanently |

## 2.5 Commissioning components

### 2.5.1 Commissioning components

### Transferring the PLC program to the controller

1. Connect the PC and CPU with the appropriate cable.

2. Download the program files to the CPU.

3. Then set the CPU to RUN.

### Transferring the project to the HMI device

1. To transfer the project, the HMI device must be in transfer mode.

   A distinction must be made between the following situations:

   – Initial startup

   When commissioning for the first time, there is no configuration on the HMI device. The project data and runtime software required for operation must be transferred from the configuration computer to the device: The HMI device automatically changes to transfer mode.
   The following message appears on the HMI device: "Transfer mode"

   – Recommissioning

   When you recommission, you replace an existing configuration on the HMI device with another.

   For detailed information on how to do this, refer to the manual supplied with your HMI device.

2. In your WinCC flexible project, check whether the alarm settings meet your requirements.

3. Before you transfer the project to the HMI device, set the transfer parameters using the "Project" menu "Transfer" ▸ "Transfer settings":

4. To transfer the project to the HMI device, click on the "transfer" button.

    – The project is compiled automatically.

    – Each step of the compilation and transfer is listed in a window.

    If the transfer was successful, the following message is output to the configuration computer: "Transfer completed."

    The start screen appears on the HMI device.

## Connecting the controller and the HMI device

1. Connect the controller and the HMI device using a suitable cable.

2. The connection message appears on the HMI device. Please remember that the system alarm texts can be edited by the user in WinCC flexible.

---

**Notice**

During commissioning make sure you are aware of and keep to the safety-related notices in the HMI device manual.

Remember that high-frequency emissions, for example from mobile phones, can cause unwanted operating states.

---

## 2.5.2    Commissioning the interface

### Introduction

To connect a Panel PC or PC to the interface of the S7 controller, you require a communications processor or a PC/MPI adapter. To establish communication with the S7 controller, the PG/PC interface must be set.

### How to commission the interface

1. Go to the Start menu of the operating system and select the command "Settings ▶ Control Panel".

2. Double-click on "Set PG/PC interface" The "Set PG/PC interface" dialog opens.

3. Select the access point of the application "S7ONLINE."

4. Select the interface parameter assignment to suit your CP or adapter.

5. Close the dialog with the "OK" button.

### Alternative procedure

1. Select the "Set PG/PC interface" command in the "Tools" menu of the SIMATIC Manager.

2. Select the access point of the application "S7ONLINE."

3. Select the interface parameter assignment to suit your CP or adapter.

4. Close the dialog with the "OK" button.

# Communication via SIMATIC HMI HTTP protocol

<div style="text-align: right; font-size: 3em;">3</div>

## 3.1 Principles on the protocol

### 3.1.1 Principles on the protocol

**Introduction**

The SIMATIC HMI HTTP Protocol is a component part of the Sm@rtAccess option and is used to exchange data between the devices.

Its use is possible when

- PCs with WinCC flexible Runtime and Panels from the 270 Series should exchange tags with each other via existing LANs (Sm@rtAccess).

- transfer should take place using a firewall server.

The SIMATIC HMI HTTP protocol is not suitable for exchanging mass data.

Data exchange occurs according to the Request-Response method. The HTTP client sends his request to the HTTP server which processes it and returns his 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.
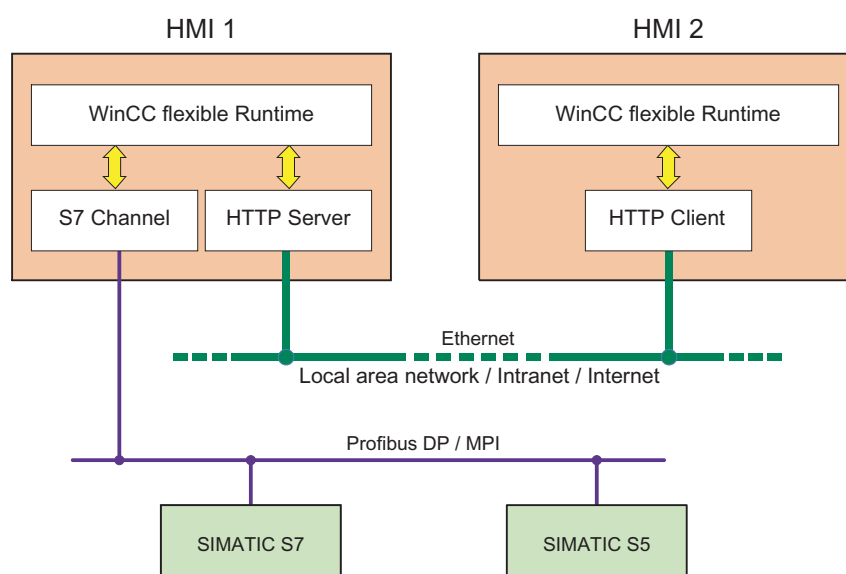
## HTTP / HTTPS

The SIMATIC HMI HTTP Protocol provides two standards:

- HTTP

  Is implemented in local networks for a fast, uncoded transfer of uncritical data.

- HTTPS

  Enables a reliable HTTP connection between devices. Firstly, there is an exchange of Secret Keys so that, in a later stage, the respective Public Keys in the form of digital certificates can be exchanged reliably and used for encrypting user data to ensure confidential 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.

---

## Connectable HMI devices

The following devices can be connected with each other over the SIMATIC HMI HTTP Protocol:

| Operating system | HMI device |
|---|---|
| Windows CE | TP 270, OP 270, MP 270B, MP 370 OP 177B, TP 177B Mobile Panel 177 PN |
| Windows 2000, Windows XP | PC, Panel PC (WinCC flexible Runtime must be installed on the units) |

# 3.2 Configuring the communication driver

## 3.2.1 Installing the communication driver

### 3.2.1.1 Installing the communication driver

## Installing HTTP components

The following HTTP components are supplied with WinCC flexible and installed when you transfer the configuration to the HMI devices:

- HTTP server
- HTTP client

For a standard PC or Panel PC, the following must also be installed:

- WinCC flexible Runtime

No special blocks are required on the HMI device for communication.

## 3.2.2 Configure HTTP server

### 3.2.2.1 Configuring HTTP server functions

#### Procedure

In addition to the HTTP communication channels (in the control panel), it must also be ensured that the HTTP server is supported by WinCC flexible Runtime . These settings are defined in the project from WinCC flexible ES.

1. Double-click "Device settings" in the project window, and open the "Device settings" editor.



2. Select the "Sm@rtAccess: SIMATIC HMI HTTP Server" check box in the "Services in Runtime" area.

Before the HMI device can be used as an HTTP server, the project must be transferred to the HMI device.

### 3.2.2.2 Configuring tags in HTTP servers

### 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:

1. The data type of the server tags must match the data type in the client,

2. The tag name configured in the HTTP server must be identical to the name of the address tag of the HTTP client.

## 3.2.3 Configuring HTTP clients

### 3.2.3.1 Configuring HTTP connections in the client

### Procedure

To be able to access the tags on the HTTP server, a SIMATIC HMI HTTP protocol communication connection must be created.

1. Create a new connection using "Communication > Connections and assign it to the "SIMATIC HMI HTTP Protocol" communication driver.

The "Connections" editor is shown in the following diagram.
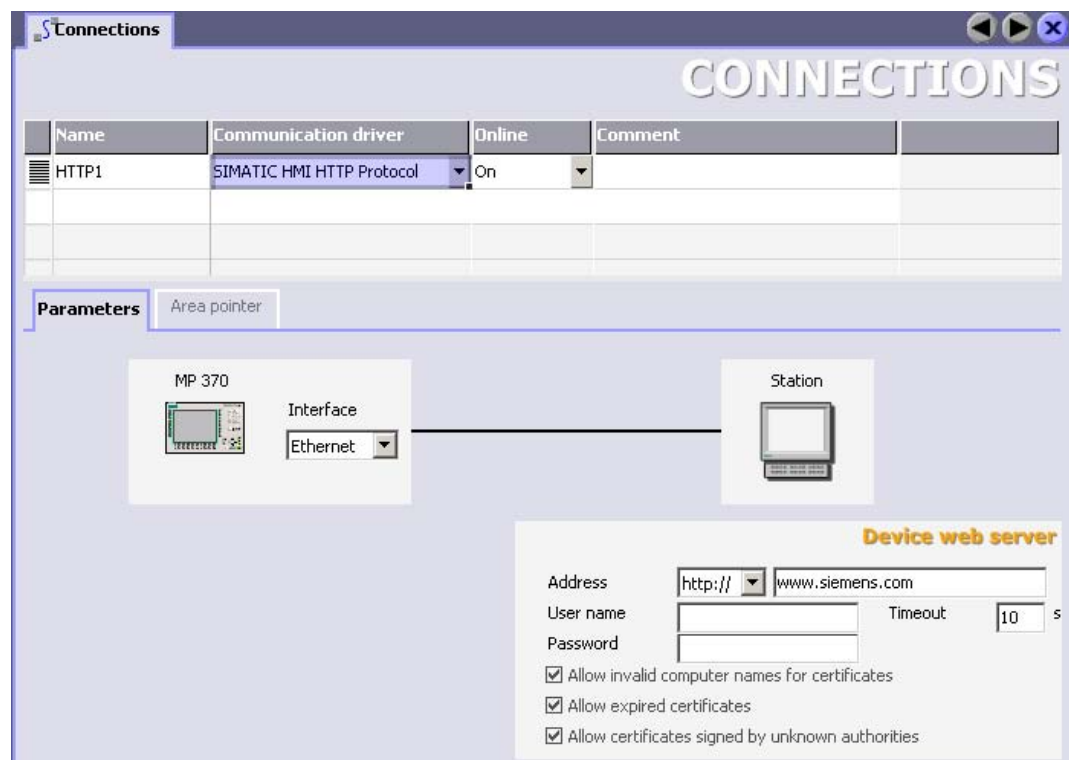
2. Assign a name to this connection that indicates its function.

3. Define the parameters in the Properties view:

| | |
|---|---|
| Interface | Select "Ethernet." |
| Address | Select the protocol http:// or https:// and enter the name of the HTTP server or its address to which communication is required. |
| | Ask your network administrator for the specific name or parameters of your network. |
| | If the server has already been commissioned, you can read out the IP address on the server as well: |
| | • Panel<br>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>. |
| | • PC/Panel PC<br> Click "Start > Run", enter "Cmd", and press <Ret>: The command interpreter is displayed. Enter the "ipconfig" command. The IP address is displayed after pressing <Ret>. |
| User name and password | If the "Authentication required" check box was selected in the HTTP server in the "Control Panel > WinCC Internet Settings > Web Server" dialog, a user name and password must be entered here in the client. |
| Timeout | The period after which a connection break is recognized. |

4. If you have selected the HTTPS protocol, you can use the settings "Allow invalid computer names for certificates", "Allow expired certificates", and "Allow certificates signed by unknown authorities" to specify how the HTTPS client checks the properties of the server certificate and how it reacts to errors.

### 3.2.3.2 Configuring tags in HTTP clients

**Procedure**

In order to be able to access tags on the HTTP server, they must be configured in the client as tag addresses.

1. Create tags in the client project under "Communication > Tags" for all server tags you wish to access.

The following figure illustrates the "Tags" editor with the browser open.
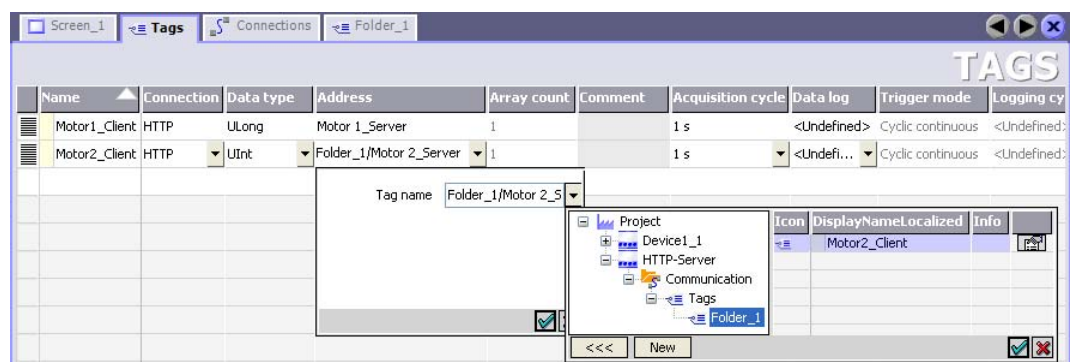
2. Define the parameters in the working area:

| | |
|---|---|
| Name | Enter the tag names on the HTTP client |
| Connection | Select the HTTP connection |
| Data type | Select the data type of the tags: |
| | Note |
| | 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. |
| | Note |
| | Array tags are not permissible. |
| Address | Enter the name of the tag to be used to communicate on the HTTP server exactly as it is written. If the tag to be addressed is located in a subfolder, the full path including tag name must be specified as the address, e.g. [folder name]\[tag name]. |
| | This is a particular advantage when the devices for server and client are in the same WinCC flexible Project. In this case, the names of the server tags can be selected in the editor browser and thus accepted. |

## 3.2.4 Permitted data types

### Permitted data types

When configuring tags, the data types listed below can be used.

| Data types in the HTTP Protocol | Length | Signs | Range of values |
|---|---|---|---|
| Bool | 0 | No | true (-1) or false (0) |
| Char | 1 bytes | yes | -128 to 127 |
| Byte | 1 bytes | No | 0 to 255 |
| Int | 2 bytes | yes | -32768 to 32767 |
| UInt | 2 bytes | No | 0 to 65535 |
| Long | 4 bytes | yes | -2,147,483,648 to 2,147,483,647 |
| ULong | 4 bytes | No | 0 to 4,294,967,295 |
| Float | 4 bytes | yes | -3.402823E38 to -1.401298E-45 for negative values and 1.401298E-45 to 3.402823E38 for positive values |
| Double | 8 bytes | yes | -1.79769313486231E308 to -4.94065645841247E-324 for negative values and 4.94065645841247E-324 to 1.79769313486232E308 for positive values |
| String | 1 to 255 byte | —— | |
| DateTime | 8 bytes | —— | 1.1.1970 00:00:00 up to 31.12.2037 23:59:59 |

Please note that data types can be defined in external controllers which have different names in WinCC flexible. To ensure correct assignment, please observe the tag definition in the external controllers.

---

**Note**

It is not possible to access array tags from an HTTP client.

---

# 3.3 Initiating an HTTP connection

## 3.3.1 Setting the Internet Settings

**Introduction**

Initiating an HTTP/HTTPS connection requires settings to be made in WinCC Internet Settings on the Control Panel of both the server and client, in addition to the configuration in WinCC flexible.

The number of tabs and the names of the tabs in the "WinCC flexible Internet Settings" dialog depend on the software you installed. Only the required tabs for the MP 270B HMI device are shown in the example below.

**WinCC flexible Internet Settings, Proxy tab**

The valid network settings are specified by the network administrator.

The following settings must be made:

- Internet settings using Windows 2000 or XP based units

  To do so, call the dialog by selecting "Internet Options > Connections > 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 flexible 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 flexible Internet Settings if the Web Server is installed on the HMI.
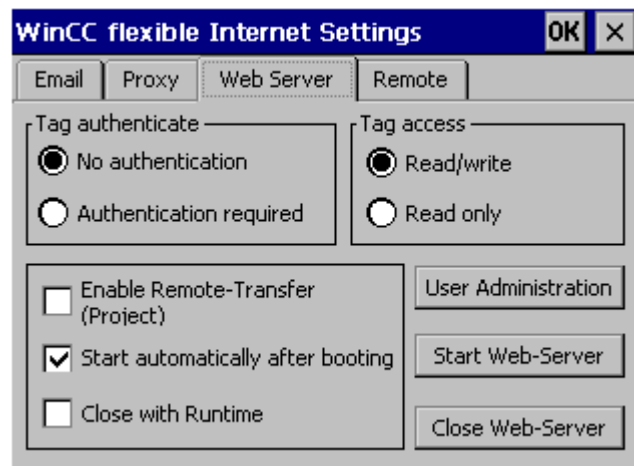


- Tag authenticate

  Controls authentication for tag access:

  – "No authentication": Authentication (user name password) is not required for access.

  – "Authentication required": Authentication (user name 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)

  Activating this check box enables an HTTP transfer from the configuration computer to the HMI device, e.g., to load a WinCC flexible Runtime project.

  – 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.
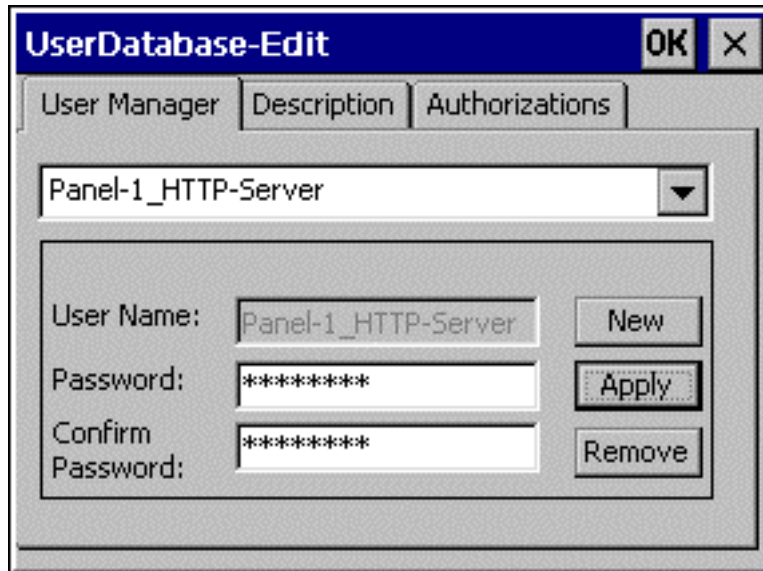


  In the "User Manager" tab of this dialog box, you can select, delete, or create a new user. The "Description" and "Authorizations" tabs always apply to the user currently selected.

  (This example dialog box shows a user with the name 'Panel-1_HTTP-Server'.)

- User Manager

  "Administrator" is the default user name, and "100" is the default password setting. If a user name and password have been entered for the HTTP server in "Communication > Connections" in your WinCC flexible project, the same user name and password must be specified here.

- 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.
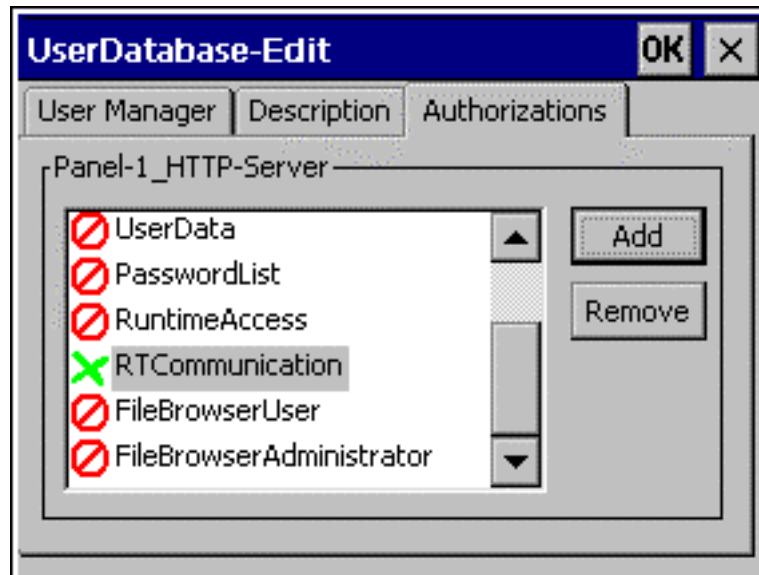


## 3.3.2 Setting up an HTTPS connection

### Introduction

To establish an HTTP connection, you must perform the following actions:

- In the "Connections" editor of WinCC flexible ES, 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. They ensure that the server to be connected to is the one it states it is.

### 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. This 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

The first time an HTTPS server is started, it creates the certificate itself and saves it in the "Cert.cer" file. This file is located

- On a PC/Panel PC (with Windows 2000/ XP): in the directory containing WinCC flexible Runtime

- On Windows CE-based devices, in the "\SystemRoot\SSL\cacert.pem" folder.

The "cacert.pem" file must be available on the HTTPS client on a medium from which you can start it by double-clicking the file. You can select from the following transfer options:

| Server | Client | Possible file transfer |
|---|---|---|
| With Windows 2000 / XP (PC, Panel PC) | With Windows 2000 / XP (PC, Panel PC) | • Diskette <br> • USB stick <br> • LAN (Ethernet) <br> • Internet Explorer (via TCP/IP if service is already running) |
| With Windows CE (xP 270, MP 370) | With Windows 2000 / XP (PC, Panel PC) | • Memory card <br> • ActiveSync (serial) |
| With Windows 2000 / XP (PC, Panel PC) | With Windows CE (xP 270, MP 370, xP 177B, Mobile Panel 177 PN) | |
| With Windows CE (xP 270, MP 370, xP 177B, Mobile Panel 177 PN) | With Windows CE (xP 270, MP 370, xP 177B, Mobile Panel 177 PN) | • Memory card |

## Installing a certificate on a client with Windows 2000 or XP

Insert the storage medium containing the saved "cacert.pem" file in the HTTPS client, or open the folder in which the file resides. 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 recognizes if a certificate has not yet been imported and prompts the user to install it. Any faults in the certificate are displayed.

## Installing a certificate on a client with Windows CE

Insert the storage medium containing the saved "cacert.pem" file in the HTTPS client. WinCC flexible provides the "InstallCert.exe" tool for importing certificates using Windows CE.

Installation can take place

- In Explorer:
  Double-click the "cacert.pem" file to install the certificate.

- In the command prompt:
  Enter "InstallCert /[command parameter] [filename]".

  – command parameters:

    Parameter /r must be specified because the certificate used in WinCC flexible 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

    Specify the certificate file complete with path (e.g. "\Storage Card\cacert.pem".)

A status alarm is output when you completed the installation.

# Communication via OPC

<div style="text-align: right; font-size: 3em;">4</div>

## 4.1 OPC - OLE for process control

### 4.1.1 Basic principles

#### 4.1.1.1 OPC

**Introduction**

OPC (OLE for Process Control) is a uniform, multi-vendor software interface. OPC Data Access (OPC DA) is based on the Windows technology COM (Component Object Model) and DCOM (Distributed Component Object Model). OPC XML, on the other hand, is based on the Internet standards XML, SOAP, and HTTP.

**DCOM**

DCOM expanded COM by adding the ability to access objects beyond the limits of one computer.

This basis allows a standardized data exchange between applications from industry, office, and manufacturing.

Previously, applications that access process data were restricted to the access mechanisms of the communications network. In OPC, devices and applications of different manufacturers can be uniformly combined.

The OPC client is an application that accesses process data of an OPC server. The OPC server is a program that offers a standardized software interface to applications of different vendors. The OPC server is an intermediate layer between these applications to process the process data, the various network protocols, and the interfaces for access to the data.

When exchanging data with OPC, you can use only HMI devices with operating systems based on the Windows technology of COM and DCOM. Windows 2000 and Windows XP currently have this software interface.

**XML**

Communication with DCOM is restricted to local area networks. Data exchange using XML allows communication with the platform-independent protocol SOAP over the Web or an intranet.

## 4.1.1.2 OPC over DCOM

### Communications concept

An OPC configuration must consist of at least one OPC server and one OPC client. The OPC server is a DCOM application that transfers data to an OPC client for further processing. In the opposite direction, a client can also supply a server with data. The data is exchanged in the form of OPC items. These are addressed using their symbolic names that you assign during configuration of the OPC server.

### Access mechanism

The open interface standard OPC uses the RPC (Remote Procedure Call) access mechanism. RPC is used to forward messages with which a distributed application can call up services on several computers in the network.

The OPC client is an application that requests process data from the OPC server over the OPC software interface.

The OPC server is a program that offers a standardized software interface to applications of different vendors. The OPC server is an intermediate layer between these applications to process the process data, the various network protocols, and the interfaces for access to the data.

### Protocol profile

OPC can use all standard protocols available to DCOM (Distributed Component Object Model) on a computer to access data of an automation system via an OPC server. OPC is not restricted to any particular standard protocol. The preferred standard protocol for communication is the datagram TCP/IP.

### Transport protocols

The following network protocols are available to DCOM for communication:

- TCP/IP tunnel
- Datagram UDP/IP
- Datagram IPX
- Connection-oriented TCP/IP
- Connection-oriented SPX
- Connection-oriented NetBEUI
- Connection-oriented Netbios over IPX

OPC attempts to establish a communication connection over one of the protocols in the order shown in the above list. If the attempt was successful, this protocol is used for communication. Otherwise, OPC continues the procedure until a connection is established using one of the listed protocols.

**Note**

The protocols in the list are shown in descending priority. To allow the communications connection to be established more quickly over OPC, either set the protocol to the highest priority or remove all unnecessary protocols.

## 4.1.1.3    OPC via XML

### Communications concept

An OPC configuration must consist of at least one OPC XML server and one OPC client.

The OPC XML server supports the OPC XML data access specification. Data access over XML has a range of functions based on OPC data access.

The OPC XML server is a Web service that passes data to an OPC client for further processing and receives data from an OPC client. The OPC XML server is not visible in WinCC flexible. It is started automatically by the Web server when an OPC client requests data.

Currently, OPC clients for access to OPC DA servers are most commonly found. The OPC client accesses the data of the OPC XML server over an OPC gateway. The OPC gateway handles communication between DCOM and XML.



DCOM    OPC Gateway    XML

OPC gateway
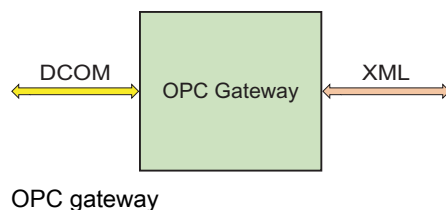
The data is exchanged in the form of OPC items. This data is addressed using symbolic names. You specify the symbolic name in the OPC server WinCC flexible project. Before the OPC client can access the data, runtime must be activated on the OPC Server.

### Access mechanism

Data exchange over XML uses the SOAP (Simple Object Access Protocol). SOAP is a platform-independent XML-based protocol. Using SOAP, applications on the Internet or in heterogeneous computer networks can communicate with each other using HTTP (HyperText Transfer Protocol)

### Transport protocols

HTTP is used to transfer the data.

## 4.1.2    Using the HMI device

### Possible configuration

To communicate using the OPC communication driver, WinCC flexible has a suitable COM interface over which data can be exchanged between automation systems.

An HMI device can be used either as an OPC server or OPC client. As an OPC client, the HMI device can connect to a maximum of 8 OPC servers.

| HMI device | Data exchange over | Operating system | OPC server | OPC client |
|---|---|---|---|---|
| PC, Panel PC | DCOM | Windows 2000 Windows XP | OPC DA server | OPC DA client |
| MP 270B, MP 370 | XML | Windows CE | OPC-XML Server | - |

### HMI device as OPC client

When using an HMI device as an OPC client, the "OPC" communication driver must be included in the WinCC flexible project.

To allow data exchange, a connection is created in the WinCC flexible project of the OPC client over which access to the tags of the OPC server is handled. An HMI device can access several OPC servers. A separate connection is created for each OPC server you want to access. The HMI device can be used as a central operator control station and monitoring station.
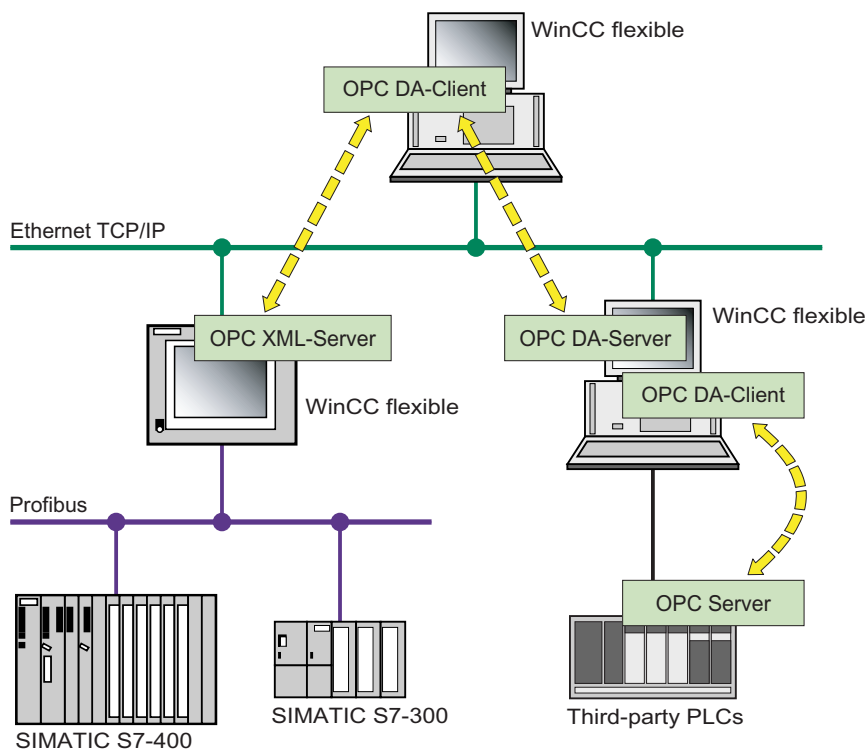
## HMI device as OPC server

The OPC server makes the data available to other applications. The applications can run on the same computer or on computers in the connected network environment. This means that process values can, for example, be displayed in Microsoft Excel.
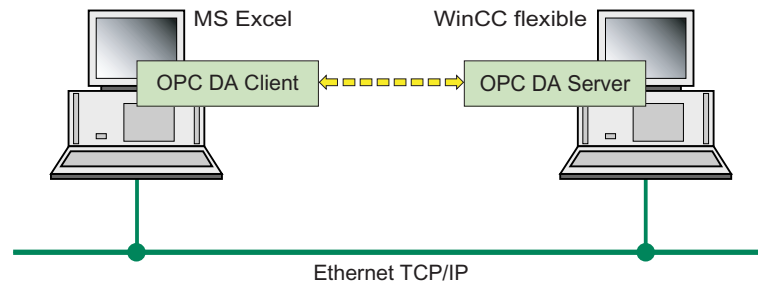


# 4.2 Configuring OPC

## 4.2.1 Installing OPC

### Installing OPC components

The following OPC components are supplied with the WinCC flexible engineering system and installed:

- OPC DA server
- Communication driver OPC (OPC client)
- OPC-XML Server
- OPC server browser
- OPC item browser

For a standard PC or Panel PC, the following must also be installed:

- OPC server
- WinCC flexible Runtime
- OPC XML Manager Included in the Setup of OPC XML Wrapper.

## 4.2.2 Configuring OPC server mode for the HMI

### 4.2.2.1 Configuring OPC server mode for the HMI

**Introduction**

The OPC server used depends on the HMI device. On HMI devices with Windows 2000 and Windows XP, the OPC DA server is used, whereas an HMI device with Windows CE uses the OPC XML server.
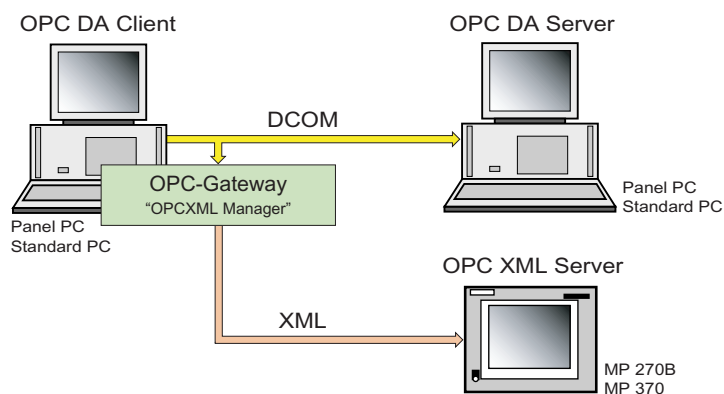


**Procedure**

1. Double-click on "Device settings\Device settings" in the project window.

2. Activate "Act as OPC server" in the working area.

3. Save the project.

4. To transfer the project to the HMI device, select the "Transfer" ▸ "Transfer settings" command in the "Project" menu.

5. Click on the "Transfer" button.

6. Start runtime on the HMI device.

## 4.2.3　Configure the HMI device as OPC client

### 4.2.3.1　Configure the HMI device as OPC client

#### Introduction

HMI devices with the Windows 2000 or Windows XP operating systems are used as the OPC client of WinCC flexible. Data exchange between the OPC DA server and OPC client uses WinCC flexible tags. In order to simplify the configuration of tags, OPC-Item-Browser is added by using the WinCC flexible setup program.

To this purpose you create connections in the project of the OPC client. The OPC server is accessed over this connection.

To enable access of the OPC client to one or several OPC-XML servers, configure "OPC XML Manager".

These settings allow communication between DCOM and XML.
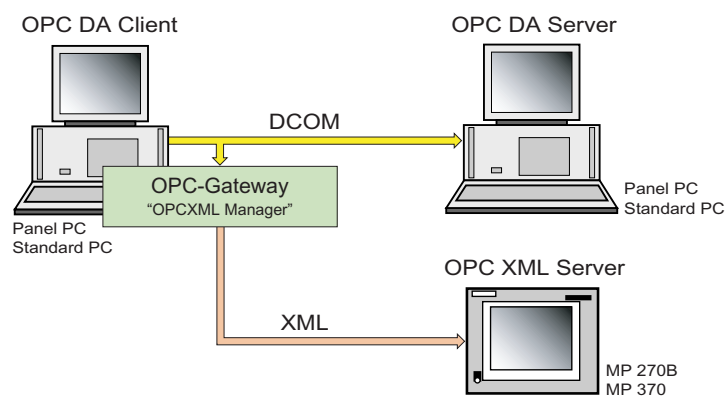


#### Requirement

The HMI device with the OPC server can be reached over its IP address.

WinCC flexible runtime is activated on the HMI device with the OPC server.

The launch and access permissions of DCOM of the OPC DA server must be set appropriately.

At least one tag is configured in the project of the OPC server.

## Procedure

1.  Double-click in the project window on "Communication ▸ Connections" Create a new connection.

2.  Select the OPC communication driver in the "Communication driver" column of the working area.

3.  Using the OPC server browser, select the name of the computer you want to use as the OPC server.

4.  Select the name of the OPC server from the displayed list. Example: for WinCC flexible OPC-DA-Server "OPC.SimaticHMI.HmiRTm", and for OPC-     XML-Server "OPC.Siemens.XML".

5.  Double-click in the project window on "Communication ▸ Tags". Create a new tag. As "Connection", select the connection you just created.

6.  Click on "Addressing" in the "Properties group" in the properties view.

7.  Select "Symbol" and select the "Item" in the tag management of the OPC server in the OPC item browser.

8.  Save the project.

9.  To transfer the project to the HMI device, select the "Transfer" ▸ "Transfer settings" command in the "Project" menu.

10. Click on the "Transfer" button.

11. Start runtime on the HMI device.

### 4.2.3.2    Managing access to OPC XML Server

## Introduction

An OPC client may access one or several OPC-XML servers. A separate connection has to be created for each OPC server you want to access.

Before the OPC client can access an OPC XML server, you must also configure the "OPC XML Manager" dialog. These settings allow communication between DCOM and XML. You can add, edit or delete the configuration of OPC XML servers in "OPC XML Manager".

## Installing "OPC XML Manager"

1.  The Setup for the installation of "OPC XML Manager" is available on WinCC flexible CD2, under "WinCCflexible\setup\OPCXMLWrapper\Disk1". Double-click "setup.exe."

2.  Select the user interface language of the Setup program under "Language."

3.  Select "Installation" to run the installation.

4.  Follow the setup instructions on the screen.

## Opening "OPC XML Manager"

1.  In Windows Explorer, go to "…\Program Files\Common Files\Siemens\OPCXML."

2.  Double-click "XMLMuxConifg.exe". "OPC XML Manager" opens.

## Adding the configuration of OPC XML servers

1. In "OPC XML Manager", click "Add…"". This opens the "Add/Edit Webservice" dialog box.
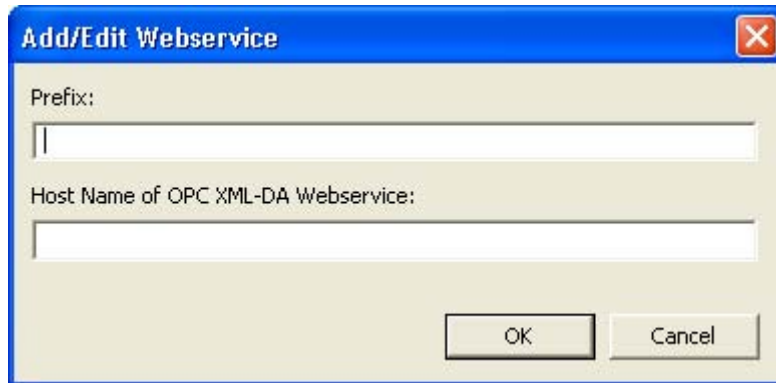
2. In the "Add/Edit Webservice" dialog box, enter a prefix at "Prefix". You can view the prefix in OPC-Item-Browser.

3. Enter the computer name, for example station 2, or the IP address of the HMI device that will be used as the OPC XML server. Click "OK" to close the dialog box.
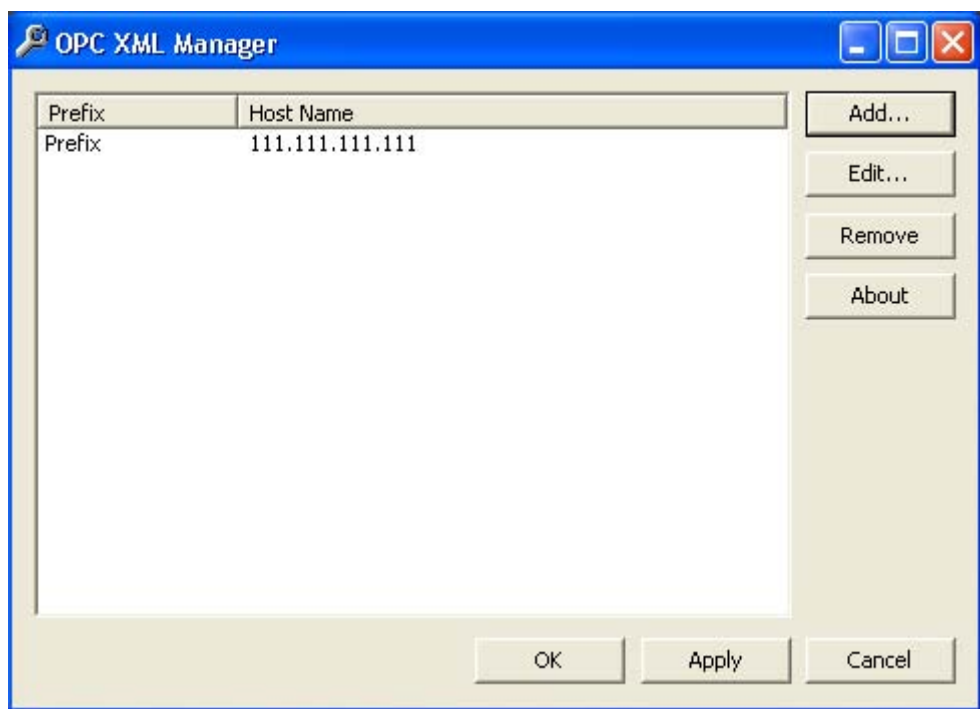
4. Click "Apply"

5. Close "OPC XML Manager" by clicking "OK."

## Editing the configuration of an OPC XML server.

1. In "OPC XML Manager", select the entry you want to edit.

2. In "OPC XML Manager", click "Edit…"".

3. On the "Add/Edit Webservice" dialog box rename the computer, for example "Station-2" or the IP address of the HMI. Edit the prefix as required.

4. Click "OK" to close the dialog box.

5. Close "OPC XML Manager" by clicking "OK."

## Deleting the configuration of an OPC XML server.

1. In "OPC XML Manager", select the entry you want to delete.

2. In "OPC XML Manager", click "Remove…"".

3. Close "OPC XML Manager" by clicking "OK."

### 4.2.3.3    Configuring protocol parameters

## Parameters to be set

To set the parameters, double-click on "Communication\Connections" in the project window. "OPC" is selected in the "Communication drivers" column in the working area. You can now enter or modify the protocol parameters in the properties window:

## Communication partners

With "OPC server", you address the OPC server with which the OPC client exchanges data.

- "Name of the OPC server"

  Define the name of the OPC server (ProgID). To make configuration easier, use the "OPC server" browser.

| OPC server | OPC server name |
|---|---|
| OPC DA server | OPC.SimaticHMI.HmiRTm |
| OPC-XML Server | OPC.Siemens.XML |

- "Name of the remote computer"

  Define the name of the computer on which the OPC server runs.

### 4.2.3.4    Configuring tag parameters

## Parameters to be set

To set the parameters, double-click on "Communication ▸ Tags" in the Project view. To communicate over OPC, set the "Name" and "Type of acquisition" for the tags of the OPC client in the properties window of the "General" group and the following parameters.

## Connection parameters

- "Connection"

  Define the "Communication connection".

- "Data type"

  Define the data type of the tag to be read from the OPC server or written to the OPC server.
  The data type defined on the OPC client must be identical to the data type used on the OPC server.

---

### Note

The "SHORT" data type is used by default in the OPC communication connection. The OPC item browser automatically sets the data type, name and address parameters of the tag. If access is made to a tag without using the OPC item browser, the data type must be adapted manually. The data type on the OPC client must be the same as the data type on the OPC server.

---

## Address parameters

Click on "Addressing" in the Properties view to define the address parameters.

- "Icon"

  Select "Symbol" and select the "Item" in the tag management of the OPC server in the OPC item browser.

  If numerous tags are created in the project, they can be collated in folders. The "item" then has the following syntax in the OPC item browser: "Foldername(1-n)\Tagname".

| Address space | Item | Access Rights |
|---|---|---|
| OPC.SimaticHMI.HmiRTm | Folder_1\Folder_1_1\Tag_in_Folder_1_1 | Read/Writeable |
| Variables | Folder_1\Folder_1_2\Tag_in_Folder_1_2 | Read/Writeable |
| | Folder_1\Tag_in_Folder_1 | Read/Writeable |
| | Folder_2\Folder_2_1\Tag_in_Folder_2_1 | Read/Writeable |
| | Folder_2\Folder_2_2\Tag_in_Folder_2_2 | Read/Writeable |
| | Folder_2\Tag_in_Folder_2 | Read/Writeable |
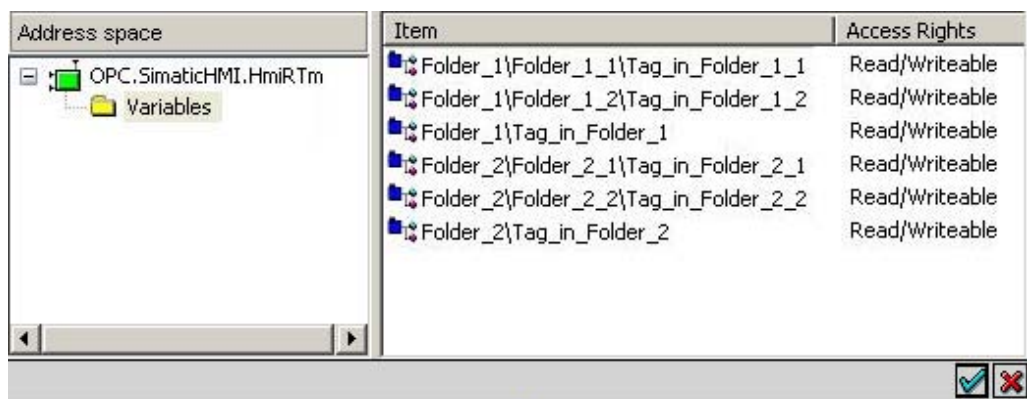
- "Item name"

  Enter the name of the tag of the OPC server in the "Item Name" field. Please take the tag name from the documentation of the OPC server.

  On OPC XML Server, the item name is <prefix>:WinCC Flexible RT<@>Tag_1. The <prefix> defines the OPC XML Server to which the tag is assigned. In OPC XML Manager, define the <prefix>.

  In WinCC flexible, the tag name is the symbolic name of the tag.
  In SIMATIC NET, the tag name might be as follows:
  "S7:[CPU416-2DP|S7-OPC-Server|CP_L2_1:]DB100,REAL0,1"
  or
  "S7:[CPU416-2DP|S7-OPC-Server|CP_L2_1:]MB0"

---

**Note**

The tag name may not contain either contain special characters, nor mutated vowels. This also applies to " , " and " * ".

---

- "Access path"

  Enter the path for access to the OPC server. When connected to the OPC server of WinCC flexible, no access path needs to be specified.

  In the case of communication to an external OPC server, an access path may be necessary. Further information is provided by the respective OPC server manufacturer.

## Permitted data types

When OPC servers are connected to WinCC flexible, only certain data types are supported.

| OPC data type | WinCC flexible 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 data types have the following value ranges:

| 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_UI4 | -2147483648 to 2147483647 |
| VT_I4 | 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.

# 4.3 Commissioning OPC

## 4.3.1 DCOM

### 4.3.1.1 DCOM settings

### DCOM settings

Data is exchanged between an OPC DA server and OPC client over the DCOM interface. Before communication is possible, the launch and access permissions of DCOM must be set correctly.

The DCOM settings depend, for example, on the network configuration and security aspects.

### Notice

The description below deals with the full enabling of the OPC DA server without considering safety aspects. With these settings, it is possible to communicate over OPC. There is, however, no guarantee that the functionality of other modules will not be impaired.

We recommend that you ask your network administrator to make these settings. To configure DCOM, you require fundamental knowledge of the network technology of Windows 2000 and Windows XP.

For more detailed information on DCOM, refer to the documentation of Windows 2000 and Windows XP.

You configure DCOM using the program "dcomcnfg.exe."

### 4.3.1.2 Starting the "dcomcnfg" program

### Introduction

You modify DCOM settings using the "dcomcnfg" program.

### Starting dcomcnfg in Windows 2000

1. Click on "Run" in the start menu of the operating system. Type in "dcomcnfg.exe." The "Properties of DCOM configuration" dialog opens.

2. Click on the "Applications" tab. Select "OPC.SimaticHMI.HmiRTm" as "Applications."

3. Click "Properties". The "OPC.SimaticHMI.HmiRTm properties" dialog opens. Set the properties of the application.

## Starting dcomcnfg in Windows XP

Windows XP basically provides the same configuration options for DCOM as the other versions of Windows. The difference is that the dialog layouts have changed.

1. Go to the Start menu of the operating system and select the command "Settings ▸ Control Panel".

2. Double-click on "Administrative tools ▸ Component services". The "Component services" dialog opens.

3. In the structure tree, expand "Console Root\Component\Services\Computers\My Computer\DCOM Configuration"

4. Select "OPC.SimaticHMI.HmiRT" as "Applications." Open the context-sensitive menu of "OPC.SimaticHMI.HmiRT" and select "Properties." The "OPC.SimaticHMI.HmiRTm properties" dialog opens. Set the properties of the application.

### 4.3.1.3    Configuring DCOM on the OPC DA server

### Introduction

Before the OPC client can launch the OPC DA server and establish the process communications connection successfully, the launch and access permissions of the OPC server must be set correctly.

### Requirements

- You have started the "dcomncfg.exe" program.

- The "OPC.SimaticHMI.HmiRTm properties" dialog is open.

### Procedure

1. Click on the "General" tab.

2. As the "Authentication level", select "None."

3. Click on the "Security" tab.

4. Click "Use custom access permissions."

5. Click the "Edit" button. The "Registry value permissions" dialog opens.

6. Add the users "Administrators", "Interactive", "Everyone", "Network", and "System" and select "Allow Access" as the "Type of Access." Click "OK" to close the dialog box.

7. Click "Use custom launch permissions."

8. Click the "Edit" button. The "Registry value permissions" dialog opens.

9. Add the users "Everyone" and "Network" and select "Allow Launch" as the "Type of Access." Click "OK" to close the dialog box.

10. Click on the "Location" tab. Click "Run application on this computer."

11. Click "OK" to close all open dialogs.

# Communication with SIMOTION controllers

<div align="right">

# 5

</div>

## 5.1 Requirements for communication with SIMOTION

### Introduction

WinCC flexible enables communication with a SIMOTION controller (Siemens Motion Control System) over PROFIBUS DP, Ethernet and MPI networks. The protocol SIMOTION is used for this.

### Requirements

The following products are required for creating SIMOTION controller projects using WinCC flexible 2005:

- Configuration software SIMATIC STEP 7 V5.3 SP2 or higher
- Configuration software SIMOTION SCOUT V3.2 SP1
- Configuration software WinCC flexible 2005

### Connectable HMI devices

The following HMIs can be connected to a SIMOTION:

|  | HMI device | Operating system |
|---|---|---|
| Panel PC | Panel PC IL 70<br>Panel PC IL 77<br>Panel PC 670<br>Panel PC 677<br>Panel PC 870<br>Panel PC 877 | Windows 2000<br>Windows XP |
| Standard | - | Windows 2000<br>Windows XP |
| Multi Panels | MP 370<br>MP 270B | Windows CE |
| Panel | Mobile Panel 170<br>Mobile Panel 177<br>TP 170B<br>OP 170B<br>TP 177B<br>OP 177B<br>TP 270<br>OP 270 | Windows CE |

|  | HMI device | Operating system |
|---|---|---|
| SIMOTION Panel | P012K<br>P012T<br>P015K<br>P015T<br>PC-R | Windows XP |

## Configuring an HMI station with SIMOTION

Requirement:

- To be able to configure an HIMI station for a SIMOTION controller, WinCC flexible has to be integrated in Step 7 for operation.

- There must be a project with SIMOTION controller.

Procedure:

1. Two different methods are available for performing the configuration:

    – The HMI project is generated integrated in SIMOTION. The WinCC flexible editors in SIMOTION Scout are available for this.

    – The HMI project is generated separate from SIMOTION. To do this, in WinCC flexible use the project wizard to select your SIMOTION project in the selection field "Integrate S7 project". Your HMI project will be integrated in this project.

2. Open the HW configuration of the HMI station created and connect the HMI station to SIMOTION controller network.

    You can select this as a network:

    – MPI

    – PROFIBUS DP

    – Ethernet

    The symbolic addresses used in SIMOTION are then available in the HMI project.

## Further information on communication

For further information on communication with SIMOTION, please refer to the chapter "Communication with SIMATIC S7".

# Communication with WinAC controllers

<div style="text-align: right; font-size: 3em;">6</div>

## 6.1 Requirements for communication with WinAC

### Introduction

Communication with a SIMATIC WinAC (Windows Automation Center) controller is possible with WinCC flexible. The SIMATIC S7 - 300/400 protocol is used for this.

### Requirements

You need the following products for configuring communication between WinCC flexible and a WinAC controller:

- A WinAC product
    - WinAC Basic or
    - WinAC RTX or
    - WinAC Slot
- WinCC flexible Advanced
- Configuration software SIMATIC STEP 7 V5.3 or later
- Configuration software SIMATIC NET CD 11/2003 or later (optional)

When selecting these products, take into consideration their release for the various operating systems.

WinCC flexible must also be integrated in STEP 7. An integrated configuration means that WinCC flexible can independently read the parameters for the controller from STEP 7.

### Note

External communication requires the installation of the configuration software SIMATIC NET CD 11/2003 or later. Otherwise, WinCC flexible Runtime can only communicate internally with the WinAC controller (i.e., within one HMI device).

## Configuration example

The following communication example shows the access from WinCC flexible Runtime via the Softbus and WinAC to an external SIMATIC S7 300/400 controller. The communication can only be performed even with an installed SIMATIC NET CD.
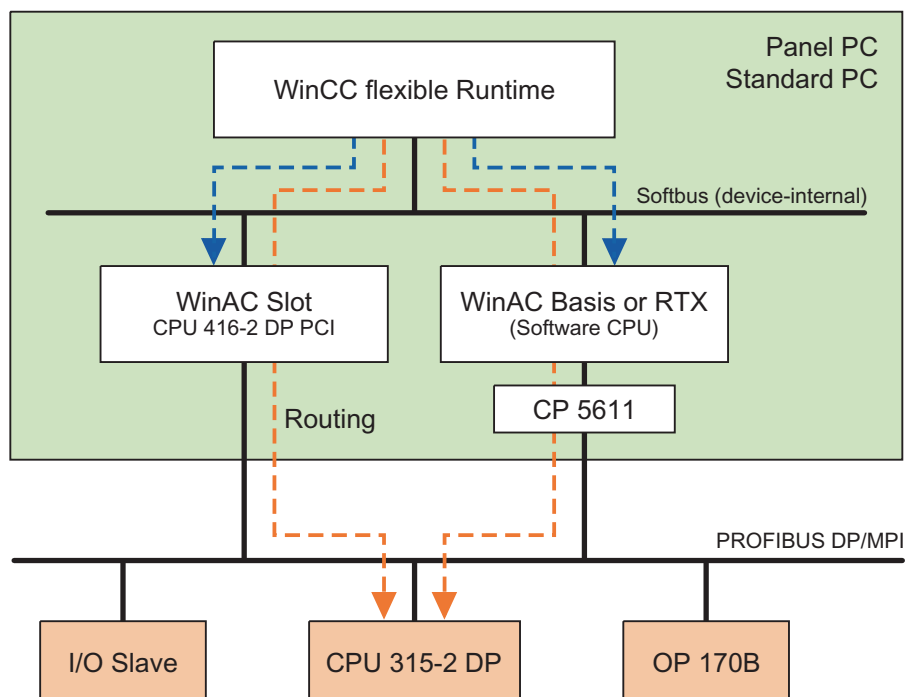


Note: Internal communication (blue arrow) is also possible without the installation of the SIMATIC NET CD.

## Configuring communication

The following is also true for SIMATIC HMI and SIMATIC PC stations. For purposes of clarity, only the SIMATIC HMI Station is mentioned here.

Please note the documentation on STEP 7 and other information on the SIMATIC NET CD.

The configuration of the internal communication is divided into three steps:

1. Configuring in STEP 7

   – In the SIMATIC Manager, insert a SIMATIC HMI Station in your STEP 7 project

   – Select the HMI device you have configured in the SIMATIC Manager, and select "Properties" with a right mouse click. The following window opens:
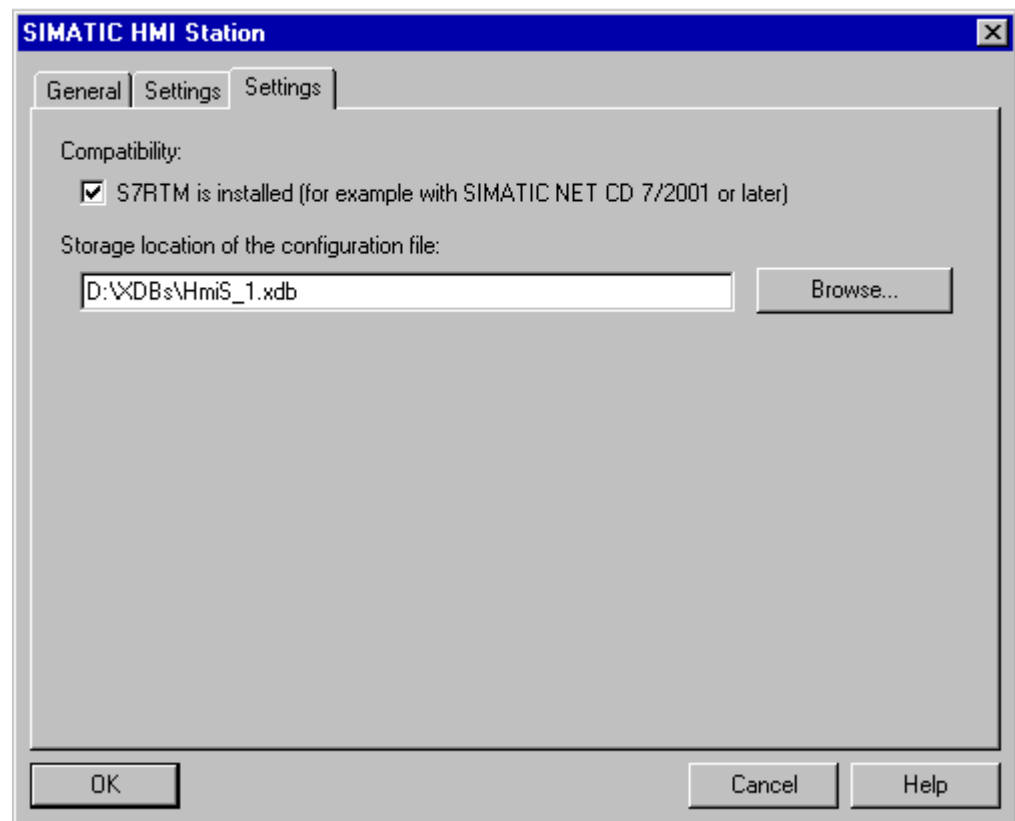


   – Set the "Compatibility" check box:

   When you set the check box "Compatibility", WinCC assumes that the SIMATIC NET CD 11/2003 is installed and offers you external as well as internal communications partners in the control dialog.

   – Switch to HW Config

   – You can find the entry "SIMATIC WinCC flexible Runtime" in the catalog under "SIMATIC HMI-Station → HMI". Place this at the desired index.

   – Complete the configuration of the HMI device (depending on the equipment e.g. WinAC, CP...).

   – In the SIMATIC Manager you can find the WinCC flexible configuration under the SIMATIC HMI Station

2. Configuring the communication settings in WinCC flexible

   – Open the configuration you made in step 1 in WinCC flexible under "Open Existing Project"

   – Select your HMI device

   – Double click in the project window in the area "Communication" on "Connections" and activate the connection

   – Edit the project

3. Configuring the PC with WinCC flexible Runtime

   – Please note the documentation on the SIMATIC NET CD

   – Open the component configurator

   – Insert SIMATIC WinCC flexible Runtime at the same index as in HW Config

   – Complete the configuration of the HMI device (see the documentation from the SIMATIC NET CD or other products).

---

**Note**

The ActiveX Control SIMATIC Panel OCX for WinAC can be integrated in WinCC flexible Runtime.

---

## Communication and user data areas

Refer to the section "Communication with SIMATIC S7" to find all necessary information about communication, the valid data types and the user data areas in WinAC.

# Communication with SIMATIC S5 controllers

<div style="text-align: right; font-size: 2em;">7</div>

## 7.1 Communication with SIMATIC S5

### 7.1.1 Communications partner (SIMATIC S5)

**Introduction**

This section describes communication between an HMI device and the SIMATIC S5 controller.

With these controllers, the connections use protocols that are specific to the controllers.

- AS511
- PROFIBUS DP

**Connectable controllers**

A connection using AS511 and PROFIBUS DP is supported by the following PLCs:

| Controller | AS511 | PROFIBUS DP |
|---|---|---|
| S5-90U | x | – |
| S5-95U | x | – |
| PLC 95U DP master | x | x |
| S5-100U (CPU 100, CPU 102, CPU 103) | x | – |
| S5-115U<br>(CPU 941, CPU 942, CPU 943, CPU 944, CPU 945) | x | x |
| S5-135U<br>(CPU 922, CPU 928A, CPU 928B) | x | x |
| S5-155U<br>(CPU 946/947, CPU 948) | x | x |

## 7.1.2 Communication between the HMI and the PLC

### Communications principle

The HMI device and the controller communicate using

- Tags
- User data areas

### Tags

The controller and the HMI device exchange data using process values. In your configuration, create tags that point to an address on the controller. The HMI device reads the value from the specified address and displays it. The operator can also make an entry on the HMI device that is then written to the address on the controller.

### User data areas

User data areas are intended for the exchange of special data and are set up only when such data is used.

User data areas are, for example, required for the following:

- Job mailboxes
- Transferring data mailboxes
- Date/time synchronization
- Sign-of-life monitoring

During configuration in WinCC flexible, create user data areas and assign the corresponding addresses on the controller.

## 7.2 Configuring the communication driver for SIMATIC S5

### 7.2.1 Communication over AS511

#### 7.2.1.1 Requirements of communication

### Connections

Connecting the HMI device to a SIMATIC S5 controller effectively involves setting the interface parameters. No special blocks are required on the controller for the connection.

The HMI device is connected directly to the CPU.

Ideally, the SI1 CPU interface should be used with TTY (current loop) electrical characteristics. If available, you can also use the SI2 CPU interface with TTY electrical characteristics. On the SI2 interface, however, remember that performance is restricted.

## Cables

To connect the HMI device to the SIMATIC S5 controller, you can use the following connecting cables:

| HMI device | Connecting cable  [1] |
|---|---|
| OP 77B<br>TP 170A, TP 170B,<br>OP 170B, TP 177B,<br>OP 177B, TP 270,<br>OP 270, MP 270B | 6ES5 734-1BD20 (includes V.24/TTY converter)<br>An adapter 9-pin female > 25-pin male is required. |
| Mobile Panel 170<br>Mobile Panel 177 | 6ES5 734-1BD20 (includes V.24/TTY converter)<br>An adapter 25 pin male > screw terminals is required. |
| MP 370 | 6XV1 440-2A _ _ _  [2] |
| Panel PC 670, 677, 870, 877<br>(COM 1, COM 2) | COM 1: 6ES5 734-2BD20  [3]<br>COM 2: 6ES5 734-1BD20 (includes V.24/TTY converter)<br>An adapter 9-pin female > 25-pin male is required. |
| Panel PC IL (COM 1)  [4] | 6ES5 734-1BD20 (includes V.24/TTY converter)<br>An adapter 9-pin female > 25-pin male is required. |
| Standard PC<br>(COM 1, COM 2) | 6ES5 734-1BD20 (includes V.24/TTY converter)<br>For a 9-pin interface, an adapter 9-pin female > 25-pin male is also required. |
| '_'  Enter the length key<br>[1]  The connecting cables fit the PG interface of all CPUs of the SIMATIC S5.<br>[2]  Also all CPUs that have an SI2 interface.<br>[3]  You can obtain cable lengths not included in the standard cables using order number 6ES5 734-2 _ _ _ 0<br>[4]  COM 2 port is used internally by the TFT display. | |

### 7.2.1.2 Installing the communication driver

## Driver for the HMI device

The communication driver for SIMATIC S5 controllers for which WinCC flexible supports a connection utilizing the AS511 protocol is supplied with WinCC flexible and installed automatically.

No special blocks are required on the controller for communication.

### 7.2.1.3 Configuring the controller type and protocol

## Select the controller

For communication with a SIMATIC S5 controller over the AS511 protocol, double-click "Communication ▶ Connections" in the project window of the HMI device and select the protocol in the "Communication driver" column:

- SIMATIC S5 AS511

The properties window displays the parameters of the selected protocol.

The settings for the HMI device are made by WinCC flexible and must be transferred to the HMI device.

---

**Note**

Please remember that remote transfer over the serial port must be disabled.

You do this on the HMI device in the "Loader ▸ Control panel ▸ Transfer" menu by selecting the "Transfer settings" command and deactivating "Remote control."

---

To edit the parameters, double-click "Communication ▸ Connections" in the project view on the HMI device, select the connection, and then edit the parameters in the properties window.

### 7.2.1.4    Configuring protocol parameters

#### Parameters to be set

To edit the parameters, double-click "Communication > Connections" in the project window of the HMI device. "SIMATIC S5 AS511" is selected in the "Communication drivers" column in the project window. Change the protocol parameters of the connection in the properties window.

#### Device-specific parameters

- Interface

   Here, you select the interface of the HMI device to which the SIMATIC S5 controller is connected.

| HMI device | Interface |
|---|---|
| OP 77B, TP 170A, | IF1A |
| Mobile Panel 170, Mobile Panel 177 DP | IF1A (connection box) |
| OP 170B, TP 170B, <br> TP 270, OP 270, MP 270B, <br> MP 370 | IF1A or IF2 |
| Panel PC, PC | COM1 through COM4 |

For more detailed information, refer to the HMI device manual.

- Type

   Here, the physical TTY interface (current loop) is set as default and cannot be changed.

- Baud rate

   Here, the transmission rate of 9600 bauds is set as default and cannot be changed.

- Data bits

   Here, "8 bits" is the default and cannot be changed.

- Parity

   Here, "Even" is the default and cannot be changed.

- Stop bits

  Here, "1" is the default value.

  If this value is changed in the S5 PLC, it has to be adapted here in the WinCC flexible project.

## Controller-dependent parameters

- CPU type

  Here, you set the CPU type you are using.

- Reading the DB address list cyclically

  If you select "Read DB address list cyclically", each time the HMI device writes or reads, the address list is read again in the controller.

  This is important when blocks are set up, modified, or deleted on the controller.

---

### Note

The HMI device can only access a DB if the address list was read previously. Since this selection has a significant effect on the performance of the HMI device, this should only be set during commissioning.

---

## Permitted data types

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

| Name | Range | Data type |
|------|-------|-----------|
| Data block word | DB DW | KF, KH, KM, KY, KS, KT, KC, bit |
| Data block double word | DB DD | DF, DH, KS, KG, bit *) |
| Extended data block word *) | DX DW | KF, KH, KM, KY, KS, KT, KC, bit |
| Extended data block double word *) | DX DD | DF, DH, KS, KG, bit *) |
| Input word | IW | KF, KH, KM, KY, KS, KT, KC, bit *) |
| Input double word | ID | DF, DH, KS, KG, bit *) |
| Output word | QW | KF, KH, KM, KY, KS, KT, KC, bit |
| Output double word | QD | DF, DH, KS, KG, bit *) |
| Flag word | FW | KF, KH, KM, KY, KS, KT, KC, bit *) |
| Flag double word | FD | DF, DH, KS, KG, bit *) |
| Timer | T | KT, KH, KM |
| Counter | C | KT, KH, KM |
| S flag word *) | SW | KF, KH, KM, KY, KS, KT, KC, bit |
| S flag double word *) | SD | DF, DH, KS, KG, bit |
| *) This data type is not supported by all CPUs. | | |

The data types have the following value ranges:

| Data type | Range of values |
|---|---|
| KF | -32768 to 32767 |
| KH | 0000 through FFFF |
| KM | 0000000000000000 through 1111111111111111 |
| KY | 0.0 to 255.255 |
| KG | 3.402823466 e-38 through 3.402823466 e+38 |
| DF | -2147483648 to 2147483647 |
| DH | 00000000 through FFFFFFFF |
| KS | 1 through 128 characters for rack devices<br>1 through 80 characters for all other devices. |
| KC | 0 to 999 |
| KT | 10 ms to 9990 sec. |
| Bit | 0 or 1 |

**Note**

The representation of the tags on the HMI device is selected in the properties window of the particular screen object, "General" group, in the "Format pattern" field.

## 7.2.2 Communication over PROFIBUS DP

### 7.2.2.1 Requirements of communication

**Hardware requirements**

To include the HMI devices in an existing PROFIBUS DP network, you require the following hardware components:

- An HMI device released for communication with a SIMATIC S5 controller. When using a standard PC, this must have a CP 5611 or CP 5511 communications processor installed.

- The controller must have a communications module (CP 5431 or IM308C).

- Every device (HMI device or controller) requires a PROFIBUS DP bus connector or other approved component (except for an FSK bus terminal, refer to the configurator in the SIMATIC HMI catalog ST80.1).

**Software Requirements**

For the PROFIBUS DP connection, you require the following software components:

- COM PROFIBUS 3.x or higher

- WinCC flexible Engineering System version 1.0 or higher

## System limits

When networking over PROFIBUS DP, a maximum of 122 HMI devices can be attached as slaves.

This value is a theoretical limit value. The actual limit is decided by the memory capacity and performance of the controller.

### Note

For more detailed information on the individual components, refer to the communication manual for Windows-based systems.

### 7.2.2.2 Communication driver for HMI device

## Installing the communication driver

## Driver for the HMI device

For communication on PROFIBUS DP, you require the function blocks "FB158" and "FB159."

"FB159" must only be loaded on the controller and must not be called in the user program.

The required parameter settings are transferred with the "FB158" call.

These function blocks are examples and support linear P addressing. You can expand these function blocks at any time to adapt to your requirements.

The function blocks you require depend on the PLC.

| PLC | Folder SimaticS5\... |
|---|---|
| PLC 95U DP master | ..\PLC95UDP |
| PLC 115U | ..\PLC115UDP\CPU941_4 for CPU 941 through 944<br>..\PLC115UDP\CPU945 for CPU 945 |
| PLC 135U | ..\PLC135U |
| PLC 155U | ..\PLC135U |

## Configuring function block FB 158

## Addressing the function block

The function blocks provided with WinCC flexible are examples and support linear P addressing. You can expand these function blocks at any time to adapt to your requirements.

The following table explains the changes required for the respective types of addressing when the two supplied function blocks, FB 158 and FB 159, are used.

| Type of addressing | Changes |
|---|---|
| Linear P | Not necessary |
| Linear Q | The P addresses and Q addresses must be changed in FB 158 and FB 159. |

| Type of addressing | Changes |
|---|---|
| P | The page frame number must be entered into the I/O byte 255 before FB 158 is called. |
| | Example:<br>L  KB x<br>T  PY 255 |
| Q | The page frame number must be entered into the I/O byte 255 before FB 158 is called. |
| | The P addresses and Q addresses must be changed in FB 158 and FB 159. |
| | Example:<br>L  KB x<br>T  QB 255 |

The following table shows the allowed address range in relation to the type of addressing for all PLCs except S5 95U.

| Type of addressing | Allowed address range |
|---|---|
| Linear P range | 128 to 255 |
| Linear Q range [1] | 0 to 255 |
| P | 192 to 254 |
| Q [1] | 0 to 254 |
| [1]  Only possible for S5 115U with CPU 945, S5 135U and S5 155U. | |

For S5 95U, only the area between 64 and 127 is allowed. The FB 158 creates an error message for addresses larger than 127.

## Calling FB 158

FB 158 must be called in the cyclic program, e.g. OB 1, with the following parameters:

- PERA:

  I/O start address It must match with the setting in COM PROFIBUS.

- BLEN:

  Block length (not for PLC 95U since only tiny is possible.)

  0:  tiny

  1:  small

  2:  middle

  3:  big

  The block length must match with the setting in WinCC flexible ("Communication > Connections > Communication Driver > SIMATIC S5 DP").

- CADB:

  An available DB that is used by FB 158 as working memory. FB 158 uses the first 11 words of the DB to buffer data. The user can use this DB after the 11th data word.

After FB 158 has been called, AKKU 1 contains the number of any error that has occurred. This error needs to be evaluated in the STEP 5 program since the error number is reset the next time the FB is called.

| Example of a call for FB 158: | |
|---|---|
| | :JU FB 158 |
| NAME | :DPHMI |
| PERA: | :KF 128 |
| BLEN: | :KF 0 |
| CADB: | :DB 58 |

### Note

FB158 is responsible for recording the process, FB159 reads the data in and out of the memory. FB 159 should only be available in the PLC and may not be called by the user.

### Multiple calling of FB158

When using several units, the FB must be called once for each unit.

### Note

The performance is increased considerably when the FB158 is called from an Alarm OB and the PLC polling time is greater than the alarm time.

### 7.2.2.3    Configuring the controller type and protocol

### Select the controller

To connect to a SIMATIC S5 controller using the PROFIBUS DP protocol, double-click on "Communication ▶ Connections" in the project view and select the following protocol in the work area in the "Communications driver" column:

- SIMATIC S5 DP

The properties window displays the parameters of the selected protocol.

To edit the parameters, double-click "Communication > Connections" in the project view on the HMI device, select the connection, and then edit the parameters in the properties window.

### Note

The settings on the HMI device and on the controller must match.

### 7.2.2.4    Configuring protocol parameters

### Parameters to be set

To edit the parameters, double-click "Communication > Connections" in the project window of the HMI device. "SIMATIC S5 DP" is selected in the "Communication drivers" column in the project window. Change the protocol parameters of the connection in the properties window.

### Device-specific parameters

- Interface

  Here, you select the interface of the HMI device to which the SIMATIC S5 controller is connected.

| HMI device | Interface |
|---|---|
| Mobile Panel 170, Mobile Panel 177 DP | IF1B (connection box) |
| OP 77B, TP 170A, OP 170B, TP 170B, TP 177B, OP 177B, TP 270, OP 270 MP 270B, MP 370 | IF1B |
| Panel PC | MPI/DP |
| PC/ | PROFIBUS DP adapter (for example CP 5611) |

For more detailed information, refer to the HMI device manual.

- Type

  Here, the physical interface "Simatic" is set as default and cannot be changed.

- Baud rate

  Here, you select the transmission rate.

### Controller-dependent parameters

- HMI address

  Here, you select the PROFIBUS DP address of the HMI device.
  Range of values: 3 to 125

- Block length

  Here, you specify the assignment of the I/O area used for communication between the HMI device and controller. The size of the I/O area influences performance.

  The desired configuration is implemented according to class B (basic DP slave complying with EN 50170). You can select four different desired configurations:

  – Class B tiny

  – Class B small

  – Class B middle

  – Class B big

  The Class B configuration is set by the system.

  The assignment of the I/O areas is fixed with the four various settings. The following table shows the assignment of the I/O areas:

| Class | Inputs (bytes) | Outputs (bytes) |
|---|---|---|
| Class B tiny | 32 | 22 |
| Class B small | 42 | 22 |
| Class B middle | 64 | 32 |
| Class B big | 128 | 64 |

The settings in WinCC flexible must match the configuration data of the interface module CP 5431 or IM308C.

---

**Note**

To transfer large amounts of data, it is advisable to set a large I/O area. This means that the displays on the HMI device are updated more quickly since the data is fetched in one cycle.

---

### Permitted data types

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

| Name | Range | Data type |
|---|---|---|
| Data block word | DB DW | KF, KH, KM, KY, KS, KT, KC, bit |
| Data block double word | DB DD | DF, DH, KS, KG, bit *) |
| Input word | IW | KF, KH, KM, KY, KS, KT, KC, bit *) |
| Input double word | ID | DF, DH, KS, KG, bit *) |
| Output word | QW | KF, KH, KM, KY, KS, KT, KC, bit |
| Output double word | QD | DF, DH, KS, KG, bit *) |
| Flag word | FW | KF, KH, KM, KY, KS, KT, KC, bit *) |
| Flag double word | FD | DF, DH, KS, KG, bit *) |
| Timer | T | KT, KH, KM |
| Counter | C | KT, KH, KM |
| *) This data type is not supported by all CPUs. | | |

The data types have the following value ranges:

| Data type | Range of values |
|---|---|
| KF | -32768 to 32767 |
| KH | 0000 through FFFF |
| KM | 0000000000000000 through 1111111111111111 |
| KY | 0.0 to 255.255 |
| KG | 3.402823466 e-38 through 3.402823466 e+38 |
| DF | -2147483648 to 2147483647 |
| DH | 00000000 through FFFFFFFF |
| KS | 1 through 128 characters for rack devices<br>1 through 80 characters for all other devices. |
| KC | 0 to 999 |
| KT | 10 ms to 9990 sec. |
| Bit | 0 or 1 |

---

**Note**

The representation of the tags on the HMI device is selected in the properties window of the particular screen object, "General" group, in the "Format pattern" field.

---

## 7.2.3 Notes on avoiding errors

### Changes during operation

To avoid errors, note the following points:

- Modifying data blocks during operation is not permitted.
- Compressing the internal program memory of the PLC ("Compress" PG function, integrated FB COMPR) is not permitted if an HMI device is connected.

---

**Note**

When memory is compressed, the absolute addresses of the blocks in the program memory change. Since the HMI device only reads the address list during startup, it does not recognize the address change and accesses the wrong memory areas.

If you cannot avoid compressing memory during operation, turn off the HMI device before running the compress function.

---

**Notice**

In hazardous areas, turn off the power to the HMI device when disconnecting the connectors.

---

### HMI device is connected to SI2 of the CPU

If communication using AS511 is active over both interfaces of the CPU, the second interface is serviced with low priority.

A possible configuration could be, for example: PG on SI1 and HMI device on SI2. In this case, error messages may occur on the HMI device indicating communication problems. This problem can be particularly extreme with the CPU 928B.

## 7.2.4    Optimizing the configuration

### Acquisition cycle and update time

The acquisition cycles for the "Area pointers" and the acquisition cycles of the tags specified in the configuration software are decisive factors for the actual update times that can be achieved.

The update time is the sum of the acquisition cycle + transmission time + processing time.

To achieve optimum update times, remember the following points in your configuration:

- Keep the individual data areas as small as possible and as large as necessary.

- Define data areas that belong together as belonging together. The actual update time improves if you set up a large area rather than several smaller areas.

- If the acquisition cycles you select are too short, this is detrimental to the overall performance. Set the acquisition cycle to suit the rate of change of the process values. The temperature curve of a furnace, for example, changes much more slowly than the speed curve of an electric drive. As a general guideline, the acquisition cycle should be approx. 1 second.

- Put the tags of an alarm or a screen in one data area without gaps.

- To allow changes in the controller to be recognized reliably, these must be available at least during the actual acquisition cycle.

- Set the transmission rate to the highest possible value.

### Screens

With screens, the update rate that can actually be achieved depends on the type and amount of data to be displayed.

To achieve shorter update times, during configuration you should make sure that you only configure short acquisition cycles for those objects that actually need to be updated quickly.

### Trends

When using bit-triggered trends, if the group bit is set in the "Trend transfer area", the HMI device always updates all the trends whose bit is set in this area. It then resets the bits again.

The group bit in the S5 program can only be set again after all bits have been reset by the HMI device.

### Job mailboxes

If large numbers of job mailboxes are sent in quick succession, this can lead to overload in the communication between the HMI device and controller.

If the HMI device enters the value 0 in the first data word of the job mailbox, the HMI device has received the job mailbox. The HMI device now processes the job for which it requires time. If a new job mailbox is entered again immediately in the jobs mailbox, it may take some time before the HMI device can process the next controller job. The next job mailbox will only be accepted when there is computing capacity available.

## Reading the DB address list cyclically (AS511 only)

Reading the DB address list each time the PLC is accessed is necessary only when, for example, the user data areas have not yet been set up during the commissioning phase. During subsequent operation, this should be disabled to improve performance.

# 7.3 User data areas

## 7.3.1 Area pointer

### 7.3.1.1 General information on area pointers (SIMATIC S5)

## Introduction

Area pointers are parameter fields from which WinCC flexible Runtime obtains information about the location and size of data areas in the controller. During communication, the controller and the HMI device read and write information for each other in these data areas. By evaluating the data entered in these areas by the partner device, the controller and HMI device trigger defined actions.

Physically, the area pointers reside in the memory of the controller. Their addresses are set up during configuration in the "Connections" editor in "Coordination".

WinCC flexible uses the following area pointers:

- Control request
- Project ID
- Screen number
- Data set
- Date/time
- Date/time controller
- Coordination

### Application

Before an area pointer can be used, it must be set up and activated by selecting "Communication ► Connections."



Activating an area pointer based on a SIMATIC S7 controller

- Active

  Activates the area pointer.

- Name

  Default name of the area pointer set by WinCC flexible.

- Address

  Here, enter the tag address of the area pointer In the PLC.

- Length

  The default length of the area pointer is set by WinCC flexible.

- Acquisition cycle

  An acquisition cycle must be specified here, so that an area pointer is read periodically by Runtime. Make allowances for the fact that a shorter acquisition time may have a negative influence on HMI performance.

- Comment

  Here, you can enter a comment, for example to describe the use of the area pointer.

## Functional scope

Which area pointers are actually available depends on the HMI device you are using. The following table provides you with an overview of the range of functions of the various HMI devices:

| Area pointer | PC<br>Panel PC | MP 270B<br>MP 370 | TP 270<br>MP 270<br>OP 270 | Mobile Panel 170<br>Mobile Panel 177<br>P 77B<br>TP 170B<br>OP 170B<br>TP 177B<br>OP 177B | OP73<br>OP 77A<br>OP 177A | OP 73micro<br>TP 170micro<br>TP 177micro<br>TP 170A |
|---|---|---|---|---|---|---|
| Screen number | x | x | x | x | x | – |
| Data set | x | x | x | x | – | – |
| Date/time | x | x | x | x | x | – |
| Date/time controller | x | x | x | x | x | x |
| Coordination | x | x | x | x | x | – |
| Project ID | x | x | x | x | – | – |
| Control request | x | x | x | x | x | – |
| [1]OP 73micro , TP 170micro and TP 170micro devices only support communication with an S7-200 PLC. | | | | | | |

The following table shows how the controller and HMI device access the various data areas, reading (R) or writing (W).

| Data area | Required for | HMI device | Controller |
|---|---|---|---|
| Screen number | Evaluation by the controller to determine which screen is currently open | W | R |
| Data set | Transfer of data mailboxes with synchronization | R / W | R / W |
| Date/time | Transfer of date and time from the HMI device to the controller | W | R |
| Date/time controller | Transfer of date and time from the controller to the HMI device | R | W |
| Coordination | Querying the status of the HMI device in the controller program | W | R |
| Project ID | Runtime checks whether or not the WinCC flexible user version and the project on the controller are consistent | R | W |
| Control request | Triggering functions on the HMI device by the controller program | R / W | R / W |

The following sections explain the area pointers and the assigned job mailboxes.

## 7.3.1.2    "Screen number" area pointer

## Function

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

This allows information on the current screen content of the HMI device to be transferred to the controller and to trigger certain reactions from the controller, for example calling up a different screen.

## Application

Before the "Screen number" area pointer can be used, it must be set up and activated by selecting "Communication ▸ Connections." It can be created only in **one** controller and only **once** on this controller.

The screen number is transferred spontaneously to the controller; in other words, it is transferred whenever a new screen is selected on the HMI device. It is therefore unnecessary to configure an acquisition cycle.

## Layout

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

|          | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 1. Word  | Current screen type |||||||||||||||
| 2. Word  | Current screen number |||||||||||||||
| 3. Word  | Reserved |||||||||||||||
| 4th word | Current field number |||||||||||||||
| 5. Word  | Reserved |||||||||||||||

- Current screen type

  "1" for root screen or
  "4" for permanent window

- Current screen number

  1 to 65535

- Current field number

  1 to 65535

### 7.3.1.3    "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

Its length is two words.

## Application

---

**Note**

Each time the area pointer is updated by the HMI device, the entire coordination area is always written.
For this reason, the PLC program must not make any changes in 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. After startup, the bit is set permanently to "1."

## Operating mode

As soon as the HMI device is switched offline by the user, the operating mode bit is set to 1. In normal operation of the HMI device, the state of the operating mode bit is "0." You can find out the current operating mode of the HMI device by querying this bit.

## Life bit

The life bit is inverted by the HMI device at intervals of approximately one second. By querying this bit in the controller program, you can check whether or not the connection to the HMI device still exists.

### 7.3.1.4 "Date/time controller" area pointer

## Function

This area pointer is used to transfer the date and time from the controller to the HMI device. This is useful when the controller is time master.

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

The HMI device reads the data cyclically at the configured acquisition interval and synchronizes itself.

### Note

Do not configure an acquisition cycle for the date/time area pointer that is too short, since this can be detrimental to the performance of the HMI device.
Recommendation: Acquisition cycle of 1 minute, if the process allows this.

The date/time data area has the following structure:

## DATE_AND_TIME format (in BCD code)

| Data word | Left byte | | | Right byte | | |
|---|---|---|---|---|---|---|
| | 15 | . . . . . . | 8 | 7 | . . . . . . | 0 |
| n+0 | Year (80-99/0-29) | | | Month (1-12) | | |
| n+1 | Day (1-31) | | | Hour (0-23) | | |
| n+2 | Minute (0-59) | | | Second (0-59) | | |
| n+3 | Reserved | | | Reserved | Weekday (1-7, 1=Sunday) | |
| n+4 [1] | Reserved | | | Reserved | | |
| n+5 [1] | Reserved | | | Reserved | | |

1) The two data words must be present in the data area in order to ensure a correlation of the data format with WinCC flexible and to avoid the reading of the wrong information.

### Note

Note that when you enter the year, values 80-99 result in years 1980 through 1999 and the values 0-29 result in the years 2000 through 2029.

## 7.3.1.5 "Date/time" area pointer

### Function

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

The controller writes job mailbox "41" to the jobs mailbox.

When it evaluates the job mailbox, 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 data area has the following structure:

| Data word | Left byte | | | Right byte | | | |
|---|---|---|---|---|---|---|---|
| | 15 | | 8 | 7 | | 0 | |
| n+0 | Reserved | | | Hour (0-23) | | | Time |
| n+1 | Minute (0-59) | | | Second (0-59) | | | |
| n+2 | Reserved | | | Reserved | | | |
| n+3 | Reserved | | | Weekday (1-7, 1=Sunday) | | | Date |
| n+4 | Day (1-31) | | | Month (1-12) | | | |
| n+5 | Year (80-99/0-29) | | | Reserved | | | |

### Note

Note that when you enter the year, values 80-99 result in years 1980 through 1999 and the values 0-29 result in the years 2000 through 2029.

### 7.3.1.6 "User version" area pointer

#### Function

When runtime starts, it is possible to check whether the HMI device is connected to the correct controller. This is important when several HMI devices are being used.

The HMI device compares a value stored on the controller with the value specified in the configuration data. This ensures the compatibility of the configuration data and the controller program. If there is a discrepancy, a system alarm is displayed on the HMI device and runtime is stopped.

#### Application

If you want to use this area pointer, set up the following during configuration.

- Specify the version of the configuration data. Possible values between 1 and 65535.

  Enter the version in the "Device settings ▶ Device settings" editor in "Project ID."

- Data address of the value of the version stored on the controller:

  Enter the data address in the "Communication ▶ Connections" editor in "Address."

### 7.3.1.7 "Job mailbox" area pointer

#### Function

Using the jobs mailbox, job mailboxes can be given to the HMI device so that actions are triggered on the HMI device. These functions include, for example:

- Display screen
- Set date and time

#### Data structure

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

| Word | Left byte | Right byte |
|------|-----------|------------|
| n+0 | 0 | Job number |
| n+1 | Parameter 1 | |
| n+2 | Parameter 2 | |
| n+3 | Parameter 3 | |

If the first word of the job mailbox is a value other than 0, the HMI device evaluates the job mailbox. 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

Below, you can see all the job mailboxes and their parameters available on the various HMI devices. The "No." column contains the job number of the job mailbox. Job mailboxes can only be triggered by the controller when the HMI device is online.

### Note

There are no job mailboxes for the TP 170A HMI device.

| No. | Function | | PC<br>Panel PC | Panel [1] |
|-----|----------|---|----------------|-----------|
| 14 | Setting time (in BCD code) | | x | x |
| | Parameter 1 | Left byte:  -<br>Right byte:  hours (0-23) | | |
| | Parameter 2 | Left byte:  minutes (0-59)<br>Right byte:  seconds (0-59) | | |
| | Parameter 3 | - | | |
| 15 | Setting date (in BCD code) | | x | x |
| | Parameter 1 | Left byte:  -<br>Right byte:  weekday<br>            (1-7: Sunday-Saturday) | | |
| | Parameter 2 | Left byte:  day (1-31)<br>Right byte:  month (1-12) | | |
| | Parameter 3 | Left byte:  year | | |
| 23 | User login | | x | x |
| | 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 - 255 | | |
| | Parameter 2, 3 | - | | |
| 24 | User logoff | | x | x |
| | Logs off the currently logged on user.<br>The function corresponds to the "logoff" system function) | | | |
| | Parameter 1, 2, 3 | - | | |
| 40 | Transfer date/time to controller | | x | x |
| | (in the S7 DATE_AND_TIME format)<br>There should be at least 5 seconds between two jobs, otherwise the HMI device will be overloaded. | | | |
| | Parameter 1, 2, 3 | - | | |
| 41 | Transfer date/time to controller | | x | x |
| | (in the OP/MP format)<br>There should be at least 5 seconds between two jobs, otherwise the HMI device will be overloaded. | | | |
| | Parameter 1, 2, 3 | - | | |

| No. | Function | | PC Panel PC | Panel [1] |
|-----|----------|--|-------------|-----------|
| 46 | Updating tags | | x | x |
| | Causes the HMI device to read the current value of the tags from the controller 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 | | x | x |
| | Parameter 1, 2, 3 | - | | |
| 50 | Clear error alarm buffer | | x | x |
| | Parameter 1, 2, 3 | - | | |
| 51 | Display selection | | x | x |
| | Parameter 1 | Screen number | | |
| | Parameter 2 | - | | |
| | Parameter 3 | Field number | | |
| 69 | Read data mailbox from controller | | x | x |
| | Parameter 1 | Recipe number (1-999) | | |
| | Parameter 2 | Data mailbox number (1-65535) | | |
| | Parameter 3 | 0: Do not overwrite existing data record 1: Overwrite existing data mailbox | | |
| 70 | Write data mailbox to controller | | x | x |
| | Parameter 1 | Recipe number (1-999) | | |
| | Parameter 2 | Data mailbox number (1-65535) | | |
| | Parameter 3 | - | | |

[1] These include the HMI devices MP 370, MP 270B, TP 270, MP 270, OP 270, TP 170B, OP 170B, Mobile Panel 170 and OP 77B

## 7.3.1.8 "Data mailbox" area pointer

### "Data mailbox" area pointer

### Function

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

## Data transfer types

There are two ways of transferring data mailboxes between the HMI device and controller:

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

Data records are always transferred directly, i.e. tag values are directly read from the address or written to the address that was configured for the tag, without any detour through temporary storage.

## 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 the data mailboxes can also be triggered by the controller.

- Triggering by configured functions

If the transfer of data mailboxes is triggered by a configured function or by a job mailbox, the recipe display on the HMI device remains operable since the data mailboxes 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.

## Transfer without synchronization

If you select asynchronous transfer of data mailboxes between the HMI device and controller, 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 partner.
- The controller does not require information about the recipe number and data mailbox number.
- The transfer of data records is triggered by the operator on the HMI device.

## Reading values

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

- Triggering by the operator in the recipe view:

  The values are loaded on the HMI device. There you can then edit them, for example, modify values, save etc.

- Triggering by a function or job mailbox:

  The values are saved to the data medium immediately.

## Writing values

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

- Triggering by the operator in the recipe view:

  The current values are written to the controller.

- Triggering by a function or job mailbox:

  The current values are written to the controller from the data medium.

## Transfer with synchronization (SIMATIC S5)

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 mailbox transfer can be a useful alternative, for example when

- the controller is the "active partner" in the transfer of data mailboxes,

- Information about the recipe number and data mailbox number needs to be evaluated on the controller.

- The transfer of data mailboxes is triggered by a job mailbox.

## Requirement

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

- An area pointer has been set up: "Communication ▶ Connections" editor in "Area pointer"

- The controller with which the HMI device synchronizes transfer of data mailboxes is specified in the recipe. "Recipes" editor, properties window of the recipe, "Properties" group in "Transfer".

## Structure of the data area

The data area has a fixed length of 5 words and is structured as follows:

| | 15 | 0 |
|---|---|---|
| 1. Word | Current recipe number (1 - 999) | |
| 2. Word | Current data mailbox 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 | Transfer active |
| 4 | 0000 0100 | Transfer completed without error |
| 12 | 0000 1100 | Transfer completed with error |

## Sequence of a transfer started by the operator in the recipe display

## Reading from the controller started by the operator in the recipe display

| Step | Action | |
|---|---|---|
| 1 | Check: Status word = 0? | |
| | yes | No |
| 2 | The HMI device enters the recipe number to be read and the status "Transfer active" in the data mailbox and sets the data mailbox number to 0. | Abort with system alarm. |
| 3 | The HMI device reads the values from the controller and displays them in the recipe display. If the recipes have synchronized tags, the values from the controller are also written to the tags. | |
| 4 | The HMI device sets the status "Transfer completed." | |
| 5 | To allow further transfers, the controller program must set the status word to 0 again. | |

## Writing to the controller started by the operator in the recipe display

| Step | Action | |
|---|---|---|
| | Check: Status word = 0? | |
| 1 | yes | No |
| | The HMI device enters the recipe and data mailbox number to be written and the status "Transfer active" in the data mailbox. | Abort with system alarm. |
| 2 | The HMI device writes the current values to the controller. If the recipes have synchronized tags, the changed values are synchronized between the recipe display and tags and then written to the controller. | |
| 3 | The HMI device sets the status "Transfer completed." | |
| 4 | If required, the controller program can now evaluate the transferred data. | |
| 5 | To allow further transfers, the controller program must set the status word to 0 again. | |

## Sequence of the transfer triggered by a job mailbox

Transfer of data mailboxes between the HMI device and controller can be initiated as follows:

- By the HMI device or
- By the controller

The two controller jobs No. 69 and No. 70 are available for this type of transfer.

## No. 69: Read data mailbox from controller ("PLC → DAT")

Job mailbox no. 69 transfers data mailboxes from the controller 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 mailbox number (1-65,535) | |
| Word 4 | Do not overwrite existing data mailbox: 0<br>Overwrite existing data mailbox: 1 | |

## No. 70: Write data mailbox to controller ("DAT → PLC")

Job mailbox no. 70 transfers data mailboxes from the HMI device to the controller. 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 mailbox number (1-65,535) | |
| Word 4 | — | |

## Sequence when reading from the controller 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 mailbox number specified in the job and the status "Transfer active" in the data mailbox. | Abort without return message. |
| 3 | The HMI device reads the values and stores the values in the data mailbox specified in the job mailbox. | |
| 4 | • If "Overwrite" was selected in the job, an existing data mailbox is overwritten without any prompt for confirmation.<br><br>   The HMI device sets the status "Transfer completed."<br><br>• If "Do not overwrite" was selected in the job, and the data mailbox already exists, the HMI device aborts the job and enters 0000 1100 in the status word of the data mailbox. | |

| Step | Action | |
|------|--------|---|
| 5 | To allow further transfers, the controller program must set the status word to 0 again. | |

## Sequence of writing to the PLC using PLC job "DAT → PLC" (no. 70)

| Step | Action | |
|------|--------|---|
| 1 | Check: Status word = 0? | |
| | yes | No |
| 2 | The HMI device enters the recipe and data mailbox number specified in the job and the status "Transfer active" in the data mailbox. | Abort without return message. |
| 3 | The HMI device fetches the values of the data mailbox specified in the job from the data medium and writes the values to the controller. | |
| 4 | The HMI device sets the status "Transfer completed." | |
| 5 | The controller program can now evaluate the transferred data. To allow further transfers, the controller program must set the status word to 0 again. | |

## Sequence of the transfer when triggered by a configured function

## Reading from the controller 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 "Transfer active" in the data mailbox. | Abort with system alarm. |
| 3 | The HMI device reads the values from the controller and stores them in the data mailbox specified in the function. | |
| 4 | • If "Yes" was selected for the "Overwrite" function, an existing data mailbox is overwritten without any prompt for confirmation.<br><br>  The HMI device sets the status "Transfer completed."<br>• If "No" was selected for the "Overwrite" function and the data mailbox already exists, the HMI device aborts the job and enters 0000 1100 in the status word of the data mailbox. | |
| 5 | To allow further transfers, the controller program must set the status word to 0 again. | |

## Writing to the controller with 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 "Transfer active" 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 writes the values to the controller. | |
| 4 | The HMI device sets the status "Transfer completed." | |
| 5 | The controller program can now evaluate the transferred data. To allow further transfers, the controller program must set the status word to 0 again. | |

## Possible causes of error when transferring data records

## Possible causes of error

If the transfer of data records is terminated with error, this may be the result of one of the following causes:

- Tag address not set up on the controller
- Overwriting data records not possible
- Recipe number does not exist
- Data record number does not exist

### Note

To ensure data consistency, the evaluation of the recipe and data mailbox number on the controller must only be started after the status in the data mailbox is set to "Transfer completed" or "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 display

  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 nevertheless evaluate the status of the transfer by querying the status word in the data mailbox.

## 7.3.2     Events, alarms, and acknowledgments

### 7.3.2.1     General information on events, alarms, and acknowledgments

#### Function

Alarms provide the user with information on controller or HMI device operating states or problems by displaying them on the HMI device. The alarm texts consist of configurable texts and/or tags with actual values.

Alarms and events must be distinguished. The programmer defines what is an event and what is an alarm.

#### Events

An event indicates a status, for example

- Motor on

- Controller in manual mode

#### Alarm

Alarms indicate operational errors, for example

- Valve does not open

- Excess motor temperature

Alarms indicate exceptional operational states, and must therefore be acknowledged.

#### Acknowledgment

Alarms can be acknowledged

- By operator input at the HMI device

- By the controller setting an acknowledgment bit.

#### Triggering alarms

An alarm is triggered on the controller

- By setting a tag bit

- By a measured value violating a limit.

The location of tags, or of the tag array, is defined in WinCC flexible ES. The tag or array must be set up on the controller.

**Number of alarms**

The table below shows the number of alarms and words available for the various HMI devices.
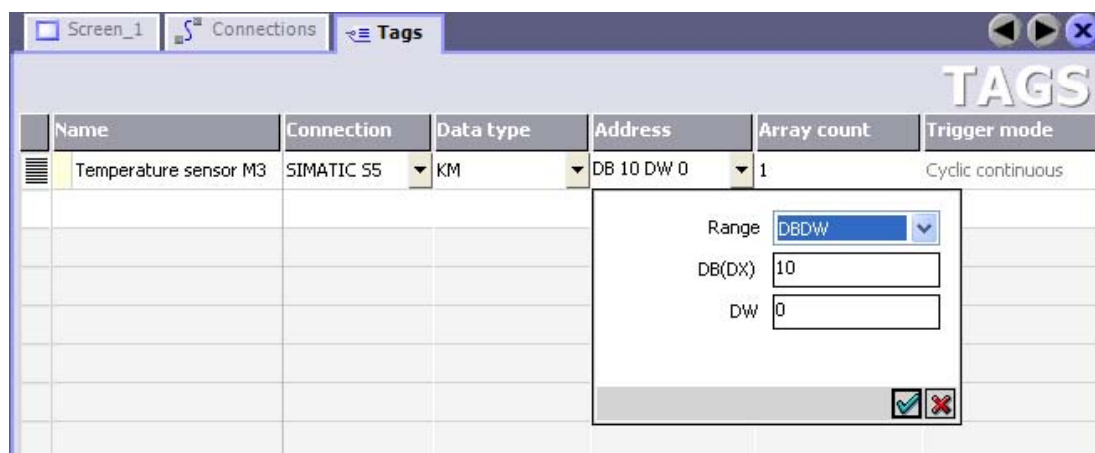
| HMI device | Events and alarms | |
|---|---|---|
| | Words in total | Alarms in total |
| Standard PC | 250 | 4000 |
| MP 370 | 250 | 4000 |
| TP 270, MP 270, OP 270, MP 270B | 250 | 4000 |
| TP 177B, OP 177B Mobile Panel 177 | | |
| TP 177micro | | 500 |
| TP 170B, OP 170B Mobile Panel 170 | 125 | 2000 |
| TP 170A [1] | 63 | 1000 |
| TP 170micro | | 500 |
| OP 77B | 63 | 1000 |

[1]Only events possible.

## 7.3.2.2 Step 1: Creating tags or an array

**Procedure**

You create tags or arrays in the "Tags" editor. The dialog box is shown below.



- Specifying the names of the tags or the name of the array

- Select the connection to the controller.

  The connection must already be configured in the "Connections" editor.

- Select the data type

  The available data types depend on the controller being used. If you select an illegal data type, the tag will not be available in the "Discrete alarms" and "Analog alarms" editors.

The following data types are supported for SIMATIC S5 controllers:

| Controller | Permitted data types | |
|---|---|---|
| | Discrete alarms | Analog alarms |
| S5-90U, S5-95U, S5-95 DP master, S5-100U, S5-115U, S5-135U, S5-155U | KF | KF, KH, KM, KY, KS, KC |

- Specify the address

  The tag addressed here contains the bit that triggers the alarm

  As soon as the bit of the tag is set on the controller and is transferred to the HMI device in the configured acquisition cycle, the HMI device recognizes the alarm as "incoming."

  After the same bit is reset on the controller, the HMI device recognizes the alarm as "outgoing."

- Specify array elements

  If the number of array elements is increased, you can select more bit numbers in the "Bit messages" editor. An array with 3 words, for example, makes 48 alarm bits available.

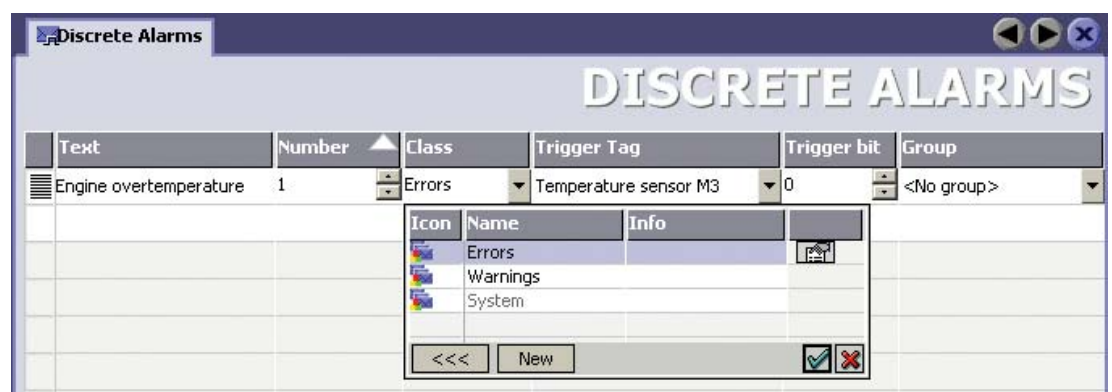### 7.3.2.3 Step 2: Configuring an alarm

### Procedure

Alarms fall into the following categories:

- Discrete alarms
- Analog alarms

You create alarms in the "Discrete alarms" and "Analog alarms" editors.

### Discrete alarms

The editor is shown in the figure below.



- Edit text

  Here, you enter the text that will be shown in runtime. It can be formatted in fonts and contain fields for the visualization of tags.

  The text appears, for example, when an alarm view was configured in the "Screens" editor.

- Specify number

    Every alarm has a number that must only occur once within the project. It is used to uniquely identify the alarm and is indicated with the alarm in runtime.

    The permitted range of values is 1 to 100,000.

    The WinCC flexible engineering system assigns consecutive numbers. You can edit these numbers, for example, if you want to organize the numbers in groups.

- Specify the alarm class

    Available alarm classes:

    – Error alarms

        This class must be acknowledged.

    – Process events

        This class signals events with incoming and outgoing alarms.

    – User-defined alarm classes

- Assign trigger tag

    In the "Trigger tag" column, you link the configured alarm with the tag created in step 1. All tags with permitted data types are shown in the selection list.

- Specify the bit number

    In the "bit number" column, specify the relevant bit position in the created tag.

    Remember that the way the bit positions are counted depends on the particular controller. With SIMATIC S5 controllers, the bit positions are counted as follows:

## Counting mode in the data block

| How the bit positions are counted | Data word | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| in SIMATIC S5 controllers | 15 | | | | | | | 8 | 7 | | | | | | | 0 |
| In WinCC flexible, you configure: | 15 | | | | | | | 8 | 7 | | | | | | | 0 |

## Counting mode in the flag area

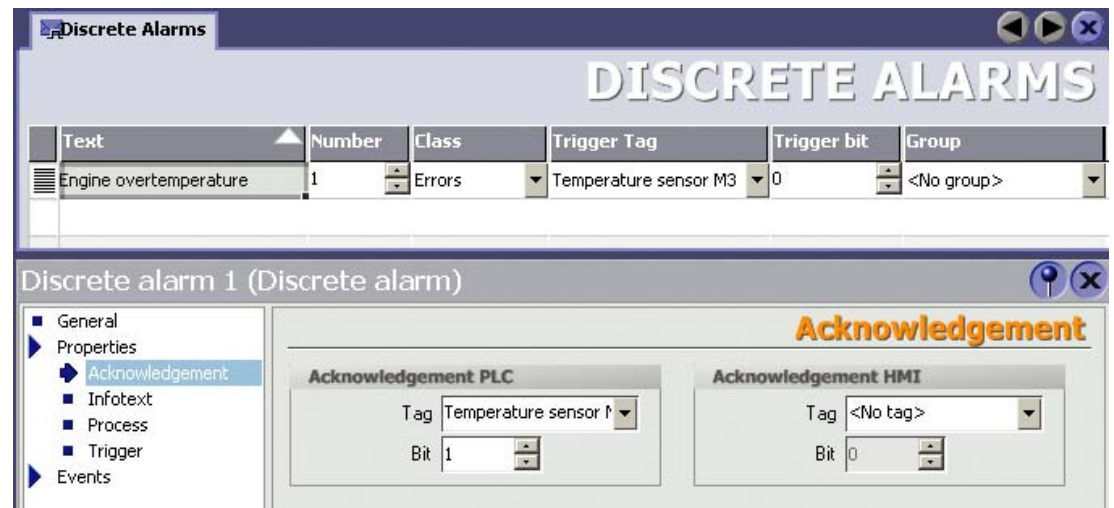| How the bit positions are counted | Flag byte 0 | | | | | | | | Flag byte 1 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Left byte | | | | | | | | Right byte | | | | | | | |
| in SIMATIC S5 controllers | 7 | | | | | | | 0 | 7 | | | | | | | 0 |
| In WinCC flexible, you configure: | 15 | | | | | | | 8 | 7 | | | | | | | 0 |

## Analog alarms

The only difference between discrete alarms and analog alarms is that you configure a limit value, rather than a bit number. If the limit value is violated, the alarm is triggered. The outgoing alarm is triggered when the low limit is violated, and where applicable, taking into account any configured hysteresis.

### 7.3.2.4 Step 3: Configure the acknowledgment

**Procedure**

Create suitable tags on the controller to acknowledge an alarm. You assign these tags to an alarm in the "Bit messages" editor. You make the assignment in "Properties ▸ Acknowledgment",

The following figure shows the dialog for configuring an acknowledgment.



The following distinctions are made for acknowledgments:

- Acknowledgment on the HMI device
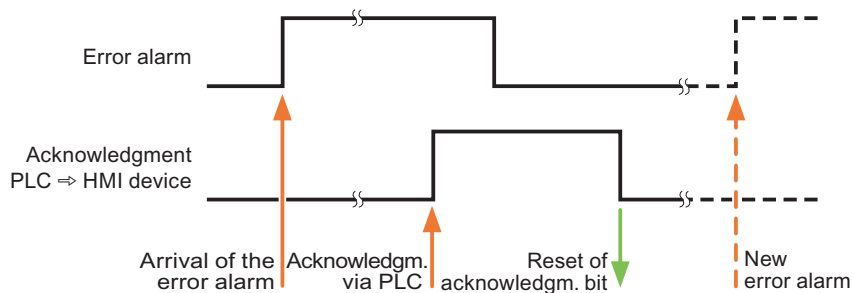- Acknowledgment by the PLC

## Acknowledgment by the PLC

In "Ack write tag", you configure the tag or the array tag and the bit number based on which the HMI device can recognize an acknowledgment by the controller.

A bit set in the tag acknowledges the assigned alarm bit on the HMI device and, therefore, has the same function as an acknowledgment from the HMI device, for example, by pressing the "ACK" key.

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

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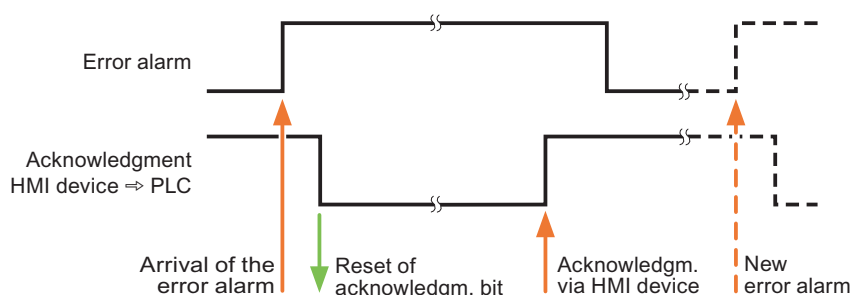


## Acknowledgment on the HMI device

In "Ack read tag", you configure the tag or the array tag and the bit number that is written to the controller after acknowledgment from the HMI device.

To ensure that a signal change always occurs when the acknowledgment bit is set, the HMI device first resets the acknowledgment bit assigned to an alarm as soon as the alarm bit is set. There is a certain time-based offset between these two operations, due to the processing time of the HMI device.

If the alarm is acknowledged on the HMI device, the bit is then set in the assigned acknowledgment tag on the controller. This allows the controller to recognize that the alarm has been acknowledged.

The figure below shows the pulse diagram.

## 7.3.3      Trend request and trend transfer_SIMATIC S5

### Function

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

### Time-triggered trends

The HMI device reads in the trend values cyclically at an interval specified in the configuration. Time-triggered trends are suitable for continuous processes, for example, the operating temperature of a motor.

### Bit-triggered trends

By setting a trigger bit in the tag trend transfer, the HMI device either reads in a trend value or an entire trend buffer. This is specified in the configuration. Bit-triggered trends are normally used to visualize rapidly changing values. One example might be the injection pressure in the production of plastic parts.

To trigger bit-triggered trends, create suitable external tags in the "Tags" editor of WinCC flexible and interconnect those to trend areas. The HMI device and controller then communicate with each other over these trend areas.

The following areas are available for trends:

- Trend request area

- Trend transfer area 1

- Trend transfer area 2 (required only with switch buffers)

Assign a bit to each trend. Valid are tags and array tags of the data type "KH" or "KF."

### Trend request area

If a screen is opened on the HMI device with one or more trends, the HMI device sets the corresponding bits in the trend request area. After deselecting the screen, the HMI device resets the relevant bits in the trend request area.
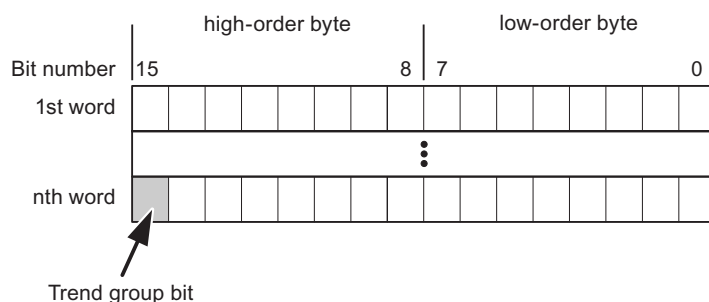
Using the trend request area, the controller can recognize which trend is currently displayed on the HMI device. Trends can also be triggered without evaluating the curve request area.

## Trend transfer area 1

This area is used to trigger trends. In your controller program, you set the bit assigned to the trend in the trend transfer area and set the trend group bit. The trend group bit is the last bit in the trend transfer area.

The HMI device recognizes the trigger and either reads a value or the entire buffer from the controller. It then resets the trend bit and the trend group bit.

The following picture shows the structure of a trend transfer area.



Trend group bit

The trend transfer area must not be modified by the controller 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. It is structured exactly the same as trend transfer area 1.

## Switch buffers

The switch buffer is a second buffer for the same trend that can be set up during configuration.

While the HMI device reads the values from buffer 1, the controller writes to buffer 2. If the HMI device is reading buffer 2, the controller writes to buffer 1. This prevents the trend values being overwritten by the controller while the trend is being read out by the HMI device.

## 7.3.4    LED mapping

### Function

The HMI devices Operator Panel (OP), Multi Panel (MP) and Panel PC have LEDs in their function keys. These LEDs can be activated by the controller. In a particular situation, this allows the controller to activate an LED, for example, to tell the operator which key to press.

### Requirement

To be able to activate an LED, an LED tag or an array tag must be set up on the controller and specified as an LED tag during configuration.

### LED assignment

The assignment of individual LEDs to the bits in the LED tags is specified when you configure the function keys. You specify the "LED tag" and the assigned "bit" in the properties window in the "General" group.

The bit number "bit" identifies the first of two consecutive bits that control the following LED states:

| Bit n+ 1 | Bit n | LED function |
|----------|-------|--------------|
| 0 | 0 | Off |
| 0 | 1 | Flashing |
| 1 | 0 | Flashing |
| 1 | 1 | On permanently |

## 7.4 Commissioning components

### 7.4.1 Commissioning components

### Transferring the PLC program to the controller

1. Connect the PC and CPU with the appropriate cable.

2. Download the program files to the CPU.

3. Then set the CPU to RUN.

### Transferring the project to the HMI device

1. To transfer the project, the HMI device must be in transfer mode.

   A distinction must be made between the following situations:

   – Initial startup

     When commissioning for the first time, there is no configuration on the HMI device. The project data and runtime software required for operation must be transferred from the configuration computer to the device: The HMI device automatically changes to transfer mode.
     The transfer dialog with a connection message appears on the HMI device.

   – Recommissioning

     When you recommission, you replace an existing configuration on the HMI device with another.

   For detailed information on how to do this, refer to the manual supplied with your HMI device.

2. In your project, check whether the alarm settings meet your requirements.

3. Before you transfer the project to the HMI device, select the "Project ▸ Transfer ▸ Transfer settings" menu and set the transfer parameters:

   – Select the interface to be used.

   – Select the transfer parameters "Mode", "Interface" and Baud rate".

   – Select the destination storage location "Flash" or RAM".

4. Start the transfer of the project with the "Transfer" button.

   – The project is compiled automatically

   – Each step of the compilation and transfer is logged in an output window.

   If the transfer was successful, the following message is displayed: "Transfer completed."

   The start screen appears on the HMI device.

## Connecting the controller and the HMI device

1. Connect the controller and the HMI device using a suitable cable.

2. The message "Connection to controller .... is established" is output to the HMI device. Please remember that the system alarm texts can be edited by the user in WinCC flexible.

---

### Notice

During commissioning make sure you are aware of and keep to the safety-related notices in the HMI device manual.

Remember that high-frequency emissions, for example from mobile phones, can cause unwanted operating states.

---

## 7.5 Connecting cables for SIMATIC S5
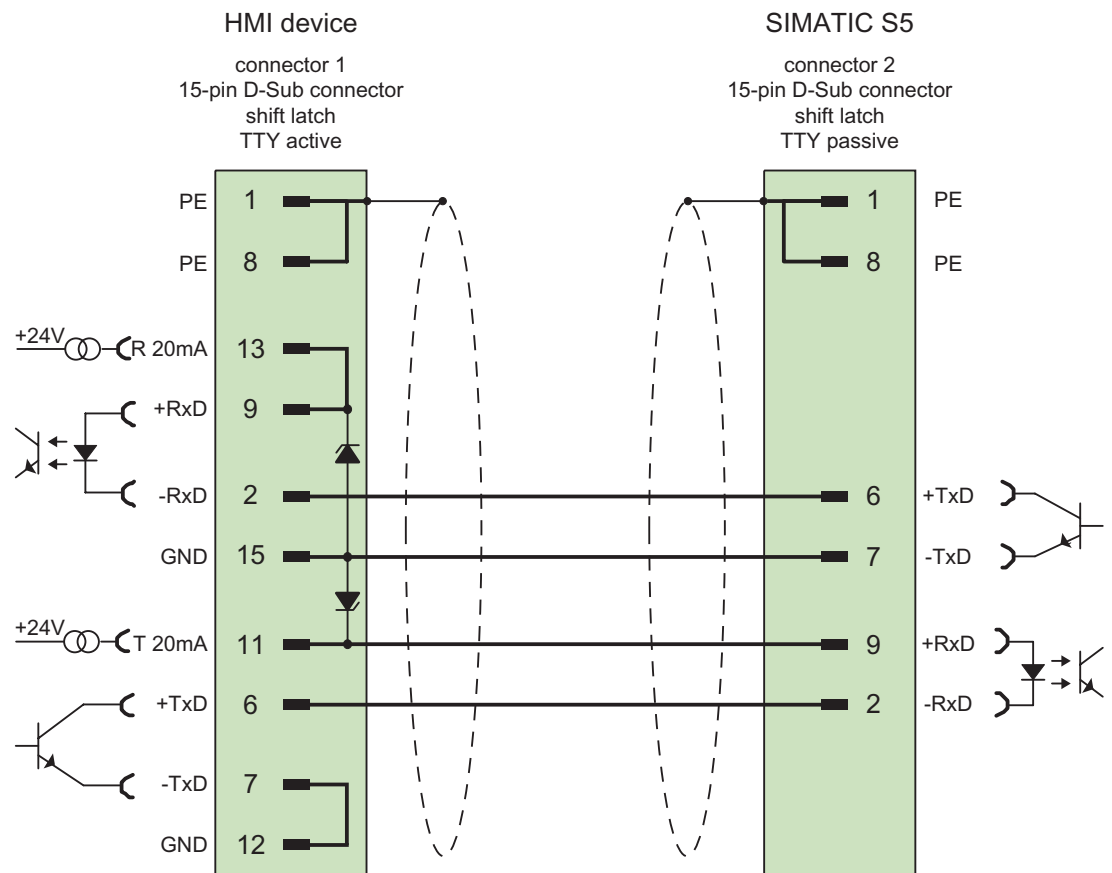
### 7.5.1 Connecting cable 6XV1 440-2A, TTY, for SIMATIC S5

**6XV1 440 - 2A _ _ _**

Length key, see catalog ST 80



Shield with large-area contact to casing at both ends
Cable: 5 x 0.14 mm2, shielded,
length: max. 1,000 m

For TTY cables with a special length > 10 m, 2 zener diodes BZX 55 C12 (12 V) must be soldered into the 15-pin connector to the HMI device (TTY active):

## 7.5.2 Connecting cable 6ES5 734-1BD20, TTY, for SIMATIC S5

### 6ES5 734-1BD20

To connect HMI device (RS-232, 9-pin sub D) - SIMATIC S5 (PG interface, TTY)

For the connection to the HMI device, a cable adapter 9-pin (female) > 25-pin (male) is required.
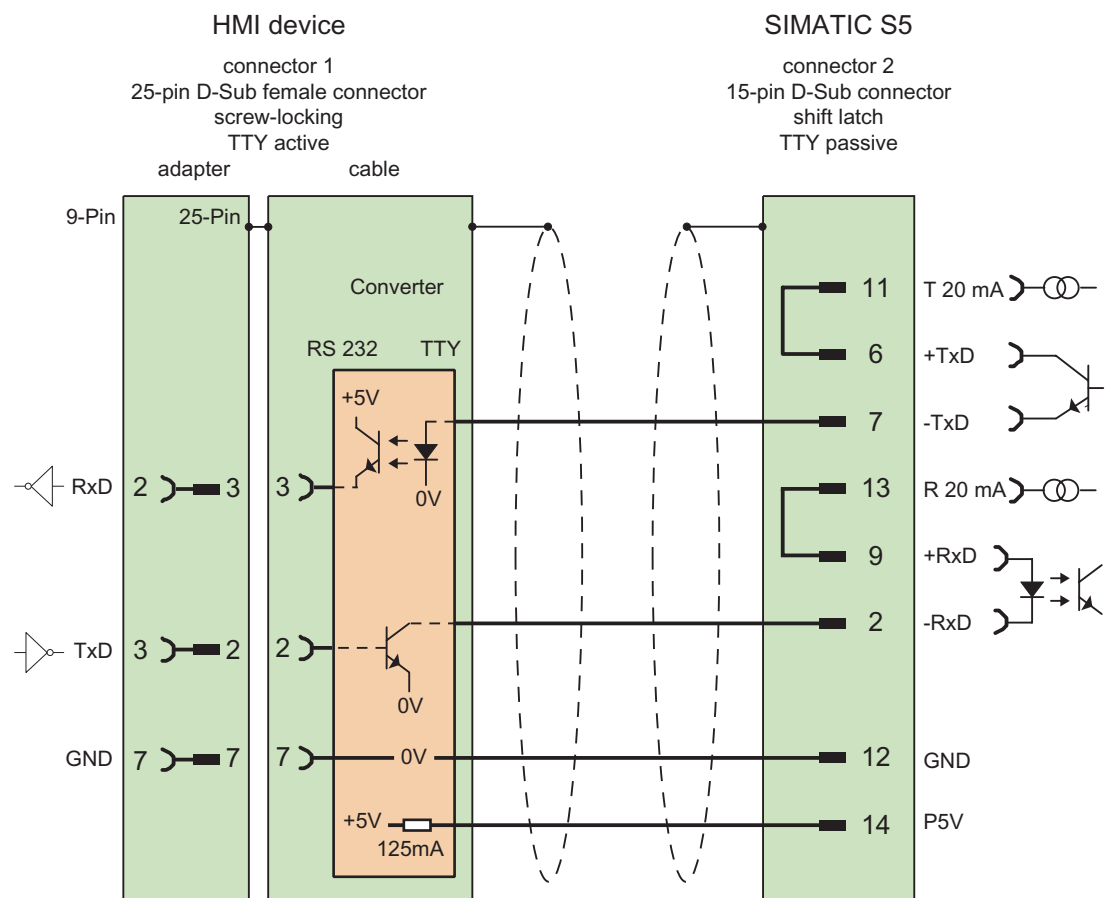


Shield with large-area contact to casing at both ends
Cable: 5 x 0.14 mm², shielded,
max. length 3.2 m

## 7.5.3    Connecting cable 6ES5 734-2BD20, TTY, for SIMATIC S5

**6ES5 734-2BD20**

To connect a Panel PC 670, 677, 870, 877 (TTY, 25-pin sub-D) – SIMATIC S5 (PG interface, TTY)



PE = casing shield
N.C. = not used
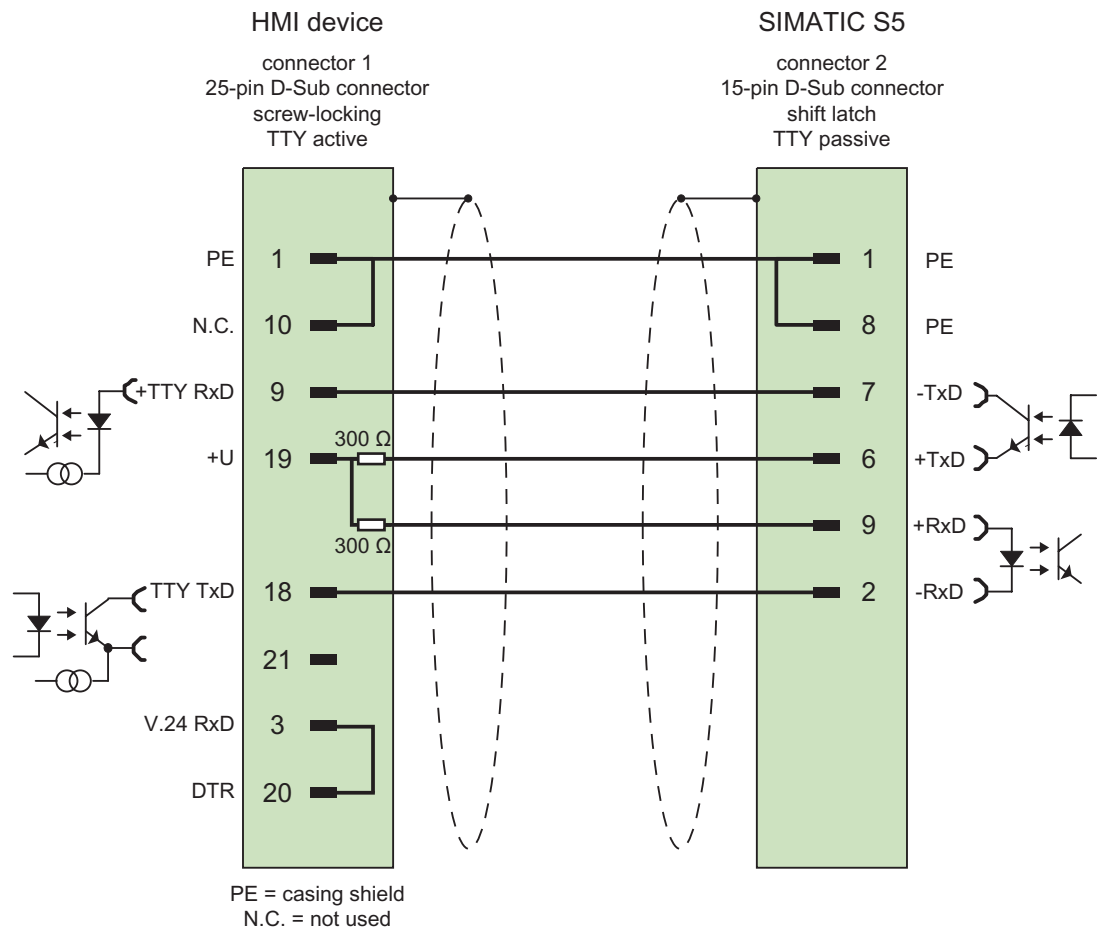
Shield with large-area contact to casing at both ends
Cable: 5 x 0.14 mm², shielded,
length: 3.2 m

---

**Note**

You can obtain lengths differing from the standard connecting cable (6ES5 734-2BD20) under order number 6ES5 734-2 _ _ _ 0, where "_" stands for the length key.

---

# Communication with SIMATIC 500/505 controllers

<div style="text-align: right; font-size: 3em;">8</div>

## 8.1 Communication with SIMATIC 500/505

### 8.1.1 Communication partner (SIMATIC 500/505)

**Introduction**

This section describes communication between HMI device and SIMATIC 500/505 controllers.

With SIMATIC 500/505, the connections use protocols that are specific to the controllers.

- NITP

  Serial connection over the RS-232/RS-422 interface with the NITP protocol (Non-intelligent terminal protocol)

- PROFIBUS DP

  PROFIBUS DP is master-slave fieldbus with up to 122 slaves. Normally, a PROFIBUS DP network is operated by one master. This master polls all slaves cyclically. The master could, for example, be a controller with interface module that is compatible with the DP standard. Each HMI device is a slave and is assigned permanently to a master.

**Released communication types**

The TP 170A can only connect to SIMATIC 500/505 over NITP.

**Known restrictions**

An RS 422 connection to the SIMATIC 575-VME is not currently supported.

When using the special function with the CPU 560-2120 and CPU 560-2820, access to the S memory data types (special user data types) is not possible. The standard data types can be used as normal.

## 8.1.2 Communication between HMI device and controller (SIMATIC 500/505)

### Communications principle

The HMI device and the controller communicate using

- Tags
- User data areas

### Tags

The controller and the HMI device exchange data using process values. In your configuration, create tags that point to an address on the controller. The HMI device reads the value from the specified address and displays it. The operator can also make an entry on the HMI device that is then written to the address on the controller.

### User data areas

User data areas are intended for the exchange of special data and are set up only when such data is used.

User data areas are, for example, required for the following:

- Job mailboxes
- Transferring data mailboxes
- Date/time synchronization
- Sign-of-life monitoring

During configuration in WinCC flexible, create user data areas and assign the corresponding addresses on the controller.

## 8.2 Configuring the communication driver SIMATIC 500/505

### 8.2.1 Communication driver for the NITP protocol

#### 8.2.1.1 Requirements of communication

### Connections

To connect the HMI device to the SIMATIC 500/505, no additional communications module is required. Communication can be handled via one of the standard ports in the system. This is the COM 1 or COM 2 port on the Panel PC and standard PC and the IF1 port on all other HMI devices. At the controller end, the HMI device must be connected to the programming interface of the CPU (RS-232 or RS-422).

Which port you should use on the HMI device is described in the relevant device manual.

## Cables

The following cables are available to connect the HMI device to the controller:

| Interface HMI devices | SIMATIC 500/505 | | | |
|---|---|---|---|---|
| | RS-232, 9-pin | RS-232, 25-pin | RS-422, 9-pin [1) | RS-422, 9-pin [2) |
| RS-232, 9-pin | PP1 | SIMATIC 505 standard cable PPX 260 109-0001 and adapter 25-pin. | Commercially available V.24/RS 422 converter | Commercially available V.24/RS 422 converter |
| RS 232, 15-pin [3) | 6XV1 440-2K _ _ _ | 6VX1 440-2L_ _ _ | – | – |
| RS-422, 9-pin | – | – | 6XV1 440-2M_ _ _ | 6XV1 440-1M_ _ _ |

'_' Length key

[1) For SIMATIC 505 (PLC 535, PLC 545-1101, PLC 565T)

[2) For SIMATIC 505 (PLC 545-1102, PLC 555)

[3) The 15-pin RS-232 interface is only relevant for MP 370.

### 8.2.1.2 Installing the communication driver

### Driver for the HMI device

The driver for connection to SIMATIC 500/505 controllers is supplied with WinCC flexible and is installed automatically.

No special blocks are required on the controller for the connection.

### 8.2.1.3 Configuring the controller type and protocol

### Select the controller

To connect to a SIMATIC 500/505 over the NITP protocol, double-click "Communication ▸ Connections" in the project view, and select the following protocol from the "Communications driver" column in the work area:

- SIMATIC 500/505

The properties window displays the parameters of the selected protocol.

To edit the parameters, double-click "Communication > Connections" in the project view on the HMI device, select the connection, and then edit the parameters in the properties window.

---

### Note

The settings on the HMI device and on the controller must match.

---

## 8.2.1.4 Configuring protocol parameters

### Parameters to be set

To edit the parameters, double-click "Communication > Connections" in the project window of the HMI device. "SIMATIC 500/505 serial" is selected in the "Communication drivers" column in the work area. Change the protocol parameters of the connection in the properties window.

### Device-dependent parameters

- Interface

  Here, you select the port on the HMI device over which the connection to the SIMATIC 500/505 will be established.

| HMI | Interface |
|---|---|
| Mobile Panel 170, Mobile Panel 177 DP | IF1A (RS 232)<br>IF1B (RS 422) |
| OP 77B, OP 170B, TP 170B, TP 177B, OP 177B, TP 270, OP 270 MP 270B, MP 370 | IF1A, IF1B [1] or IF2 |
| Panel PC, PC | COM1 or COM2 |
| [1] When using port "IF1B", also configure it by setting the DIL switch on the rear panel of the HMI. In this case, the RS 422 received data and the RTS signal are interchanged. | |

For further information, refer to the relevant hardware manuals.

- Type

  The type depends on the selected interface.

- Data bits

  Here, set "7."

- Parity

  Here, set "Odd."

- Stop bits

  Here, set "1."

- Baud rate

  Here, you set the transmission rate between the HMI device and the SIMATIC 505. Communication is possible with up to 38,400 bauds.

---

### Note

Make sure that the transmission parameters on the HMI device and the SIMATIC 500/505 match.

---

## 8.2.2 Communication driver for PROFIBUS DP

### 8.2.2.1 Requirements of communication

#### Hardware requirements

To include the HMI devices in an existing PROFIBUS DP network, you require the following hardware components:

- For Panel PC or Standard PC, a communications processor is required (for example CP5511, CP5611).

- On the controller: CP 5434-DP (annex card)

- For each device (HMI device or controller): PROFIBUS DP bus connector or other approved component (except for FSK bus terminal, see configurator in the SIMATIC HMI catalog ST80.1).

#### Software Requirements

For the PROFIBUS DP connection, you require the following software components:

- WinCC flexible engineering tool

- Specific configuration software (e.g. COM PROFIBUS) for assigning parameter values complying with the DP standard to the interface module in the controller.

### 8.2.2.2 Installing the communication driver

#### Driver for the HMI device

The driver for connection to SIMATIC 500/505 controllers is supplied with WinCC flexible and is installed automatically.

To connect the HMI device and the controller, a physical connection and a control program are necessary. A sample program is supplied with WinCC flexible.

### 8.2.2.3 Configuring the controller type and protocol

#### Select the controller

To connect to a SIMATIC 500/505 controller using the PROFIBUS DP protocol, double-click on "Communication ► Connections" in the project view and select the following protocol in the work area in the "Communications driver" column:

- SIMATIC 500/505 - DP

The properties window displays the parameters of the selected protocol.

To edit the parameters, double-click "Communication > Connections" in the project view on the HMI device, select the connection, and then edit the parameters in the properties window.

---

**Note**

The settings on the HMI device and on the controller must match.

---

## 8.2.2.4 Configuring protocol parameters

### Parameters to be set

To edit the parameters, double-click "Communication > Connections" in the project window of the HMI device. "SIMATIC 500/505 DP" is selected in the "Communication drivers" column in the work area. Change the protocol parameters of the connection in the properties window.

### Device-dependent parameters

- Interface

  Here, you select the port on the HMI device over which the connection to the SIMATIC 500/505 will be established. The type is set automatically depending on the selected interface.

| HMI device | Interface |
|---|---|
| Mobile Panel 170, Mobile Panel 177 | IF1B (connection box) |
| OP 77B, OP 170B, TP 170B, TP 177B, OP 177B, TP 270, OP 270 MP 270B, MP 370 | IF1B |
| Panel PC, PC | MPI/DP |

- Baud rate

  Here, you select the transmission rate at which the HMI device communicates with the SIMATIC 500/505 in the network configuration. The same baud rate must be set on all devices in the network.
  System default: 1500 Kbps.

### Controller-dependent parameters

- HMI address

  Here, you enter the PROFIBUS DP address of the HMI device.

  Values from 3 through 125 are permitted.

- Block length

  Here, you specify the assignment of the I/O area that will be used as the communication areas between the HMI device and controller. The size of the I/O area influences performance.

  The configuration is implemented according to class B (basic DP slave complying with EN 50170). You can select four different configurations:

  – Class B tiny

  – Class B small

  – Class B middle

  – Class B big

The assignment of the I/O areas is fixed with the four various settings. The following table shows the assignment of the I/O areas:

| Class | Inputs (bytes) | Outputs (bytes) |
|---|---|---|
| Class B tiny | 32 | 22 |
| Class B small | 42 | 22 |
| Class B middle | 64 | 32 |
| Class B big | 128 | 64 |

The settings in WinCC flexible must match the configuration data of the interface module, for example CP 5434 DP.

### Note

To transfer large amounts of data, it is advisable to set a large I/O area. This means that the displays on the HMI device are updated more quickly since the data is fetched in one cycle.

## CP 5434 DP interface module

To configure the CP 5434 DP, you require the COM PROFIBUS configuration package. GSD files for HMI device slaves are supplied with ProTool.

These GSD files are located in the "WinCC flexible Support\PlcProg\GSD" directory. Different GSD files must be used depending on the particular HMI device:

| GSD file | Vendor ID | HMI device |
|---|---|---|
| SIEM8076.GSD | 0x8076 | Panel PC, standard PC |
| SIEM80BE.GSD | 0x80BE | MP 370 |
| SIEM80E4.GSD | 0x80E4 | MP 270B, OP 270, TP 270 |
| SIEM80B3.GSD | 0x80B3 | TP 170B, OP 170B, Mobile Panel 170 |
| SIEM8107.GSD | 0x8107 | OP 77B |

If the GSD files in the COM PROFIBUS folder "WINCC flexible\PLCPROG\GSD" are older than the GSD files supplied with ProTool, or COM PROFIBUS does not yet support a new HMI, copy those files from ProTool to COM PROFIBUS. Then restart COM PROFIBUS and select "Read in GSD files".

If you have already created a COM PROFIBUS configuration with an older file and want to use the newer GSD files, you will have to recreate the configuration.

## Parameter

Before the CP 5434 DP and HMI device can communicate with each other, the following parameters must be set in COM PROFIBUS:

- Station type: HMI
- Station number: 3–

    The value entered here must match the OP address specified when you configured the HMI device.

- Desired configuration:

  The desired configuration is set by selecting the class and the symbolic name of the configuration. The following desired configurations can be set:

  – Class B tiny

  – Class B small

  – Class B middle

  – Class B big

- Address ID:

  The address ID is assigned automatically according to the selected configuration and must not be changed.

- I and Q address:

  The address must match the address stored in the PLC program.

### Integrating the COM PROFIBUS configuration into TISOFT

The TISOFT documentation describes in detail how to integrate the COM PROFIBUS configuration into the TISOFT program. The main steps are simply outlined below:

1. Create a binary file in COM PROFIBUS using the "Export" function.

2. Merge the binary file into your TISOFT program with "CONFIO" > "PRO-DP" > "MERGE."

3. Set the CPU to "ONLINE PLC" mode.

4. Specify the I/O addresses in TISOFT with "CONFIO" > "PRO-DP" > "CONFIG."

5. Transfer the program to the CPU with "UPDATE."

### Program for handling the DP protocol

To connect to PROFIBUS DP, you require a control program that handles the protocol. A sample program (written in LADDER) is supplied with WinCC flexible. You can adapt this program to suit your requirements. The sample program supports linear P addressing. The sample program is available in the "WINCC flexible\PLCPROG\SIMATIC505" folder.

The sample program was written for the CPU 545 and CPU 555 that connect to the PROFIBUS DP network via the CP 5434 DP (annex card). The sample program contains the following settings and these identical settings must also be made in the configuration.

| Program | Parameter | Value |
|---|---|---|
| WinCC flexible | HMI device | PC/ |
| | Protocol | SIMATIC 505 DP |
| | OP address | 3 |
| | Interface | DP/MPI |
| | Baud rate | 1.5 Kbps |
| | Block length | Class B middle |

| Program | Parameter | Value |
|---|---|---|
| **COM PROFIBUS** | Master station type | 505-CP5434-DP |
| | Type of addressing | Linear |
| | Slave address | 3 |
| | Station type | Appropriate HMI device |
| | Desired configuration | Class B middle |
| | Configured to I address | P000-P048 |
| | Configured to Q address | P000-P016 |
| **TISOFT** | I/O address | 0100 for WX32 and WY16 |
| | Area for data exchange | V900-V1020 |

## Setting the interface

For Panel PCs and standard PCs, the interface is set under Windows as follows: "Settings > Control Panel > Set PG/PC interface"

| Parameter | Value |
|---|---|
| Access point of the application | DPSONLINE |
| Module configuration used | PROFIBUS DP slave |

If you use the Windows CE operating system, you do not need to make any settings.

## Permitted data types

When configuring tags and area pointers, you can use the user data types listed in the following table. It is essential that these data areas are also set up for the CPU with TISOFT.

The following table lists the data types for the HMI device:

| Name | Type | Data type |
|---|---|---|
| Discrete input | X | BIT |
| Discrete output | Y | BIT |
| Control relay | C | BIT |
| Variable memory | V | BIT |
| Word input | WX | +/– INT |
| | | INT |
| Word output | WY | +/– DOUBLE |
| Constant memory | K | DOUBLE |
| Status word memory | STW | REAL |
| | | ASCII |
| Timer/counter preset | TCP | +/– INT |
| Timer/counter current | TCC | INT |
| Analog alarm/alarm acknowledge flags | AACK | +/–INT, INT |
| Analog alarm deadband | AADB | +/–INT, INT, REAL |
| Most significant word of analog alarm C flags | ACFH | +/–INT, INT |
| Least significant word of analog alarm C flags | ACFL | +/–INT, INT |
| Analog alarm error | AERR | +/–INT, INT, REAL |
| Analog alarm high alarm limit | AHA | +/–INT, INT, REAL |

| Name | Type | Data type |
|---|---|---|
| Analog alarm high-high alarm limit | AHHA | +/–INT, INT, REAL |
| Analog alarm low alarm limit | ALA | +/–INT, INT, REAL |
| Analog alarm low-low alarm limit | ALLA | +/–INT, INT, REAL |
| Analog alarm orange deviation alarm limit | AODA | +/–INT, INT, REAL |
| Analog alarm process variable | APV | +/–INT, INT, REAL |
| Analog alarm process variable high limit | APVH | REAL |
| Analog alarm process variable low limit | APVL | REAL |
| Analog alarm rate of change alarm limit | ARCA | REAL |
| Analog alarm setpoint | ASP | +/–INT, INT, REAL |
| Analog alarm SP high limit | ASPH | +/–INT, INT, REAL |
| Analog alarm SP low limit | ASPL | +/–INT, INT, REAL |
| Analog alarm sample rate | ATS | REAL |
| Analog alarm flags | AVF | +/– INTINT |
| Analog alarm yellow deviation alarm limit | AYDA | +/–INT, INT, REAL |
| Alarm peak elapsed time | APET | +/–INT, INT |
| Loop alarm/alarm acknowledge flags | LACK | +/–INT, INT |
| Loop alarm deadband | LADB | +/–INT, INT, REAL |
| Most significant word of loop C flags | LCFH | +/–INT, INT |
| Least significant word of loop C flags | LCFL | +/–INT, INT |
| Loop error | LERR | +/–INT, INT, REAL |
| Loop alarm high limit | LHA | +/–INT, INT, REAL |
| Loop alarm high-high limit | LHHA | +/–INT, INT, REAL |
| Loop gain | LKC | REAL |
| Loop derivative gain limiting coefficient | LKD | REAL |
| Loop low alarm limit | LLA | +/–INT, INT, REAL |
| Loop low-low alarm limit | LLLA | +/–INT, INT, REAL |
| Loop output | LMN | +/–INT, INT, REAL |
| Loop bias | LMX | +/–INT, INT, REAL |
| Loop orange deviation limit | LODA | +/–INT, INT, REAL |
| Loop process variable | LPV | +/–INT, INT, REAL |
| Loop PV high limit | LPVH | REAL |
| Loop PV low limit | LPVL | REAL |
| Loop rate of change alarm limit | LRCA | REAL |
| Loop ramp/soak flags | LRSF | +/–INT, INT |
| Loop ramp/soak step number | LRSN | +/–INT, INT |
| Loop setpoint | LSP | +/–INT, INT, REAL |
| Loop setpoint high point | LSPH | +/–INT, INT, REAL |
| Loop setpoint low limit | LSPL | +/–INT, INT, REAL |
| Loop rate | LTD | REAL |
| Loop reset | LTI | REAL |
| Loop sample rate | LTS | REAL |
| Loop V flags | LVF | +/–INT, INT |
| Loop yellow deviation alarm limit | LYDA | +/–INT, INT, REAL |

| Name | Type | Data type |
|------|------|-----------|
| Loop peak elapsed time | LPET | +/–INT, INT |
| SF program peak elapsed time | PPET | +/–INT, INT |
| SF subroutine peak elapsed time | SPET | +/–INT, INT |

## 8.2.3 Optimizing the configuration

### Acquisition cycle and update time

The acquisition cycles for the "Area pointers" and the acquisition cycles of the tags specified in the configuration software are decisive factors for the actual update times that can be achieved.

The update time is the sum of the acquisition cycle + transmission time + processing time.

To achieve optimum update times, remember the following points in your configuration:

- Keep the individual data areas as small as possible and as large as necessary.

- Define data areas that belong together as belonging together. The actual update time improves if you set up a large area rather than several smaller areas.

- If the acquisition cycles you select are too short, this is detrimental to the overall performance. Set the acquisition cycle to suit the rate of change of the process values. The temperature curve of a furnace, for example, changes much more slowly than the speed curve of an electric drive. As a general guideline, the acquisition cycle should be approx. 1 second.

- Put the tags of an alarm or a screen in one data area without gaps.

- To allow changes in the controller to be recognized reliably, these must be available at least during the actual acquisition cycle.

- Set the transmission rate to the highest possible value.

### Discrete alarms

For discrete alarms, use arrays and assign the individual alarms to one bit of the array tags themselves and not to the individual subelements. For discrete alarms and arrays, only tags of the data types "Int" and "+/- Int" are permitted.

### Screens

With screens, the update rate that can actually be achieved depends on the type and amount of data to be displayed.

To achieve shorter update times, during configuration you should make sure that you only configure short acquisition cycles for those objects that actually need to be updated quickly.

## Trends

When using bit-triggered trends, if the group bit is set in the "Trend transfer area", the HMI device always updates all the trends whose bit is set in this area. It then resets the bits again.

The group bit in the PLC program can only be set again after all bits have been reset by the HMI device.

## Job mailboxes

If large numbers of job mailboxes are sent in quick succession, this can lead to overload in the communication between the HMI device and controller.

If the HMI device enters the value 0 in the first data word of the job mailbox, the HMI device has received the job mailbox. The HMI device now processes the job for which it requires time. If a new job mailbox is entered again immediately in the jobs mailbox, it may take some time before the HMI device can process the next controller job. The next job mailbox will only be accepted when there is computing capacity available.

# 8.3 User data areas

## 8.3.1 Area pointer

### 8.3.1.1 General information on area pointers (SIMATIC 500/505)

## Introduction

Area pointers are parameter fields from which WinCC flexible Runtime obtains information about the location and size of data areas in the controller. During communication, the controller and the HMI device read and write information for each other in these data areas. By evaluating the data entered in these areas by the partner device, the controller and HMI device trigger defined actions.

Physically, the area pointers reside in the memory of the controller. Their addresses are set up during configuration in the "Connections" editor in "Coordination".

WinCC flexible uses the following area pointers:

- Control request
- Project ID
- Screen number
- Data set
- Date/time
- Date/time controller
- Coordination

## Application

Before an area pointer can be used, it must be set up and activated by selecting "Communication ▸ Connections."



Activating an area pointer based on a SIMATIC S7 controller

- Active

  Activates the area pointer.

- Name

  Default name of the area pointer set by WinCC flexible.

- Address

  Here, enter the tag address of the area pointer In the PLC.

- Length

  The default length of the area pointer is set by WinCC flexible.

- Acquisition cycle

  An acquisition cycle must be specified here, so that an area pointer is read periodically by Runtime. Make allowances for the fact that a shorter acquisition time may have a negative influence on HMI performance.

- Comment

  Here, you can enter a comment, for example to describe the use of the area pointer.

## Functional scope

Which area pointers are actually available depends on the HMI device you are using. The following table provides you with an overview of the range of functions of the various HMI devices:

| Area pointer | PC Panel PC | MP 270B MP 370 | TP 270 MP 270 OP 270 | Mobile Panel 170 Mobile Panel 177OP 77B TP 170B OP 170B TP 177B OP 177B | OP73 OP 77A OP 177A | OP 73micro TP 170micro TP 177micro TP 170A |
|---|---|---|---|---|---|---|
| Screen number | x | x | x | x | x | – |
| Data set | x | x | x | x | – | – |
| Date/time | x | x | x | x | x | – |
| Date/time controller | x | x | x | x | x | x |
| Coordination | x | x | x | x | x | – |
| Project ID | x | x | x | x | – | – |
| Control request | x | x | x | x | x | – |

[1]OP 73micro , TP 170micro and TP 170micro devices only support communication with an S7-200 PLC.

The following table shows how the controller and HMI device access the various data areas, reading (R) or writing (W).

| Data area | Required for | HMI device | Controller |
|---|---|---|---|
| Screen number | Evaluation by the controller to determine which screen is currently open | W | R |
| Data set | Transfer of data mailboxes with synchronization | R / W | R / W |
| Date/time | Transfer of date and time from the HMI device to the controller | W | R |
| Date/time controller | Transfer of date and time from the controller to the HMI device | R | W |
| Coordination | Querying the status of the HMI device in the controller program | W | R |
| Project ID | Runtime checks whether or not the WinCC flexible user version and the project on the controller are consistent | R | W |
| Control request | Triggering functions on the HMI device by the controller program | R / W | R / W |

The following sections explain the area pointers and the assigned job mailboxes.

## 8.3.1.2 "Screen number" area pointer

### Function

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

This allows information on the current screen content of the HMI device to be transferred to the controller and to trigger certain reactions from the controller, for example calling up a different screen.

### Application

Before the "Screen number" area pointer can be used, it must be set up and activated by selecting "Communication ▸ Connections." It can be created only in **one** controller and only **once** on this controller.

The screen number is transferred spontaneously to the controller; in other words, it is transferred whenever a new screen is selected on the HMI device. It is therefore unnecessary to configure an acquisition cycle.

### Layout

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

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1. Word | Current screen type | | | | | | | | | | | | | | | |
| 2. Word | Current screen number | | | | | | | | | | | | | | | |
| 3. Word | Reserved | | | | | | | | | | | | | | | |
| 4th word | Current field number | | | | | | | | | | | | | | | |
| 5. Word | Reserved | | | | | | | | | | | | | | | |

- Current screen type

  "1" for root screen or
  "4" for permanent window

- Current screen number

  1 to 65535

- Current field number

  1 to 65535

### 8.3.1.3 "Date/time" area pointer

### Function

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

The controller writes job mailbox "41" to the jobs mailbox.

When it evaluates the job mailbox, 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 data area has the following structure:

| Data word | Left byte | | | | | | Right byte | | | | | | |
|-----------|:---:|---|---|---|---|:---:|:---:|---|---|---|---|:---:|---|
| | 15 | | | | | 8 | 7 | | | | | 0 | |
| n+0 | Reserved | | | | | | Hour (0-23) | | | | | | Time |
| n+1 | Minute (0-59) | | | | | | Second (0-59) | | | | | | |
| n+2 | Reserved | | | | | | Reserved | | | | | | |
| n+3 | Reserved | | | | | | Weekday (1-7, 1=Sunday) | | | | | | Date |
| n+4 | Day (1-31) | | | | | | Month (1-12) | | | | | | |
| n+5 | Year (80-99/0-29) | | | | | | Reserved | | | | | | |

### Note

Note that when you enter the year, values 80-99 result in years 1980 through 1999 and the values 0-29 result in the years 2000 through 2029.

### 8.3.1.4 "Date/time controller" area pointer

### Function

This area pointer is used to transfer the date and time from the controller to the HMI device. This is useful when the controller is time master.

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

The HMI device reads the data cyclically at the configured acquisition interval and synchronizes itself.

### Note

Do not configure an acquisition cycle for the date/time area pointer that is too short, since this can be detrimental to the performance of the HMI device.
Recommendation: Acquisition cycle of 1 minute, if the process allows this.

The date/time data area has the following structure:

## DATE_AND_TIME format (in BCD code)

| Data word | Left byte | | | Right byte | | |
|---|---|---|---|---|---|---|
| | 15 | . . . . . . | 8 | 7 | . . . . . . | 0 |
| n+0 | Year (80-99/0-29) | | | Month (1-12) | | |
| n+1 | Day (1-31) | | | Hour (0-23) | | |
| n+2 | Minute (0-59) | | | Second (0-59) | | |
| n+3 | Reserved | | | Reserved | | Weekday (1-7, 1=Sunday) |
| n+4 [1] | Reserved | | | Reserved | | |
| n+5 [1] | Reserved | | | Reserved | | |

1)  The two data words must be present in the data area in order to ensure a correlation of the data format with WinCC flexible and to avoid the reading of the wrong information.

### Note

Note that when you enter the year, values 80-99 result in years 1980 through 1999 and the values 0-29 result in the years 2000 through 2029.

## 8.3.1.5    "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
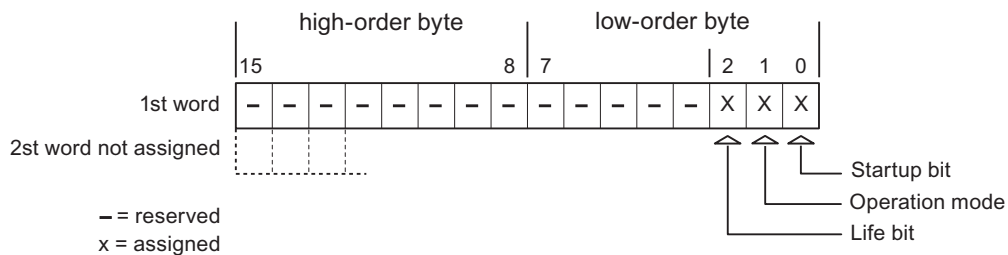- Detecting whether the HMI device is ready to communicate in the control program

Its length is two words.

### Application

### Note

Each time the area pointer is updated by the HMI device, the entire coordination area is always written.
For this reason, the PLC program must not make any changes in 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. After startup, the bit is set permanently to "1."

## Operating mode

As soon as the HMI device is switched offline by the user, the operating mode bit is set to 1. In normal operation of the HMI device, the state of the operating mode bit is "0." You can find out the current operating mode of the HMI device by querying this bit.

## Life bit

The life bit is inverted by the HMI device at intervals of approximately one second. By querying this bit in the controller program, you can check whether or not the connection to the HMI device still exists.

### 8.3.1.6 "User version" area pointer

## Function

When runtime starts, it is possible to check whether the HMI device is connected to the correct controller. This is important when several HMI devices are being used.

The HMI device compares a value stored on the controller with the value specified in the configuration data. This ensures the compatibility of the configuration data and the controller program. If there is a discrepancy, a system alarm is displayed on the HMI device and runtime is stopped.

## Application

If you want to use this area pointer, set up the following during configuration.

- Specify the version of the configuration data. Possible values between 1 and 65535.

  Enter the version in the "Device settings ▶ Device settings" editor in "Project ID."

- Data address of the value of the version stored on the controller:

  Enter the data address in the "Communication ▶ Connections" editor in "Address."

## 8.3.1.7 "Job mailbox" area pointer

### Function

Using the jobs mailbox, job mailboxes can be given to the HMI device so that actions are triggered on the HMI device. These functions include, for example:

- Display screen
- Set date and time

### Data structure

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

| Word | Left byte | Right byte |
|------|-----------|------------|
| n+0 | 0 | Job number |
| n+1 | Parameter 1 | |
| n+2 | Parameter 2 | |
| n+3 | Parameter 3 | |

If the first word of the job mailbox is a value other than 0, the HMI device evaluates the job mailbox. 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

Below, you can see all the job mailboxes and their parameters available on the various HMI devices. The "No." column contains the job number of the job mailbox. Job mailboxes can only be triggered by the controller when the HMI device is online.

#### Note

There are no job mailboxes for the TP 170A HMI device.

| No. | Function | | PC Panel PC | Panel [1] |
|-----|----------|--|-------------|-----------|
| 14 | Setting time (in BCD code) | | x | x |
| | Parameter 1 | Left byte: - <br> Right byte: hours (0-23) | | |
| | Parameter 2 | Left byte: minutes (0-59) <br> Right byte: seconds (0-59) | | |
| | Parameter 3 | - | | |

| No. | Function | | PC<br>Panel PC | Panel [1] |
|---|---|---|---|---|
| 15 | Setting date (in BCD code) | | x | x |
| | Parameter 1 | Left byte:  -<br>Right byte:  weekday<br>                    (1-7: Sunday-Saturday) | | |
| | Parameter 2 | Left byte:  day (1-31)<br>Right byte:  month (1-12) | | |
| | Parameter 3 | Left byte:  year | | |
| 23 | User login | | x | x |
| | 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 - 255 | | |
| | Parameter 2, 3 | - | | |
| 24 | User logoff | | x | x |
| | Logs off the currently logged on user.<br>The function corresponds to the "logoff" system function) | | | |
| | Parameter 1, 2, 3 | - | | |
| 40 | Transfer date/time to controller | | x | x |
| | (in the S7 DATE_AND_TIME format)<br>There should be at least 5 seconds between two jobs, otherwise the HMI device will be overloaded. | | | |
| | Parameter 1, 2, 3 | - | | |
| 41 | Transfer date/time to controller | | x | x |
| | (in the OP/MP format)<br>There should be at least 5 seconds between two jobs, otherwise the HMI device will be overloaded. | | | |
| | Parameter 1, 2, 3 | - | | |
| 46 | Updating tags | | x | x |
| | Causes the HMI device to read the current value of the tags from the controller whose update ID matches the value transferred in parameter 1.<br>(Function corresponds to the "UpdateTag" system function.) | | | |
| | Parameter 1 | 1 - 100 | | |
| 49 | Clear event buffer | | x | x |
| | Parameter 1, 2, 3 | - | | |
| 50 | Clear error alarm buffer | | x | x |
| | Parameter 1, 2, 3 | - | | |
| 51 | Display selection | | x | x |
| | Parameter 1 | Screen number | | |
| | Parameter 2 | - | | |
| | Parameter 3 | Field number | | |

| No. | Function | | PC Panel PC | Panel [1] |
|---|---|---|---|---|
| 69 | Read data mailbox from controller | | x | x |
| | Parameter 1 | Recipe number (1-999) | | |
| | Parameter 2 | Data mailbox number (1-65535) | | |
| | Parameter 3 | 0: Do not overwrite existing data record<br><br>1: Overwrite existing data mailbox | | |
| 70 | Write data mailbox to controller | | x | x |
| | Parameter 1 | Recipe number (1-999) | | |
| | Parameter 2 | Data mailbox number (1-65535) | | |
| | Parameter 3 | - | | |

[1] These include the HMI devices MP 370, MP 270B, TP 270, MP 270, OP 270, TP 170B, OP 170B, Mobile Panel 170 and OP 77B

### 8.3.1.8    "Data mailbox" area pointer

**"Data mailbox" area pointer**

**Function**

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

**Data transfer types**

There are two ways of transferring data mailboxes between the HMI device and controller:

- Transfer without synchronization

- Transfer with synchronization over the data mailbox

Data records are always transferred directly, i.e. tag values are directly read from the address or written to the address that was configured for the tag, without any detour through temporary storage.

**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 the data mailboxes can also be triggered by the controller.

- Triggering by configured functions

If the transfer of data mailboxes is triggered by a configured function or by a job mailbox, the recipe display on the HMI device remains operable since the data mailboxes 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.

## Transfer without synchronization

If you select asynchronous transfer of data mailboxes between the HMI device and controller, 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 partner.

- The controller does not require information about the recipe number and data mailbox number.

- The transfer of data records is triggered by the operator on the HMI device.

## Reading values

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

- Triggering by the operator in the recipe view:

  The values are loaded on the HMI device. There you can then edit them, for example, modify values, save etc.

- Triggering by a function or job mailbox:

  The values are saved to the data medium immediately.

## Writing values

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

- Triggering by the operator in the recipe view:

  The current values are written to the controller.

- Triggering by a function or job mailbox:

  The current values are written to the controller from the data medium.

## Transfer with synchronization (SIMATIC 500/505)

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 mailbox transfer can be a useful alternative, for example when

- the controller is the "active partner" in the transfer of data mailboxes,

- Information about the recipe number and data mailbox number needs to be evaluated on the controller.

- The transfer of data mailboxes is triggered by a job mailbox.

## Requirement

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

- An area pointer has been set up: "Communication ▸ Connections" editor in "Area pointer"

- The controller with which the HMI device synchronizes transfer of data mailboxes is specified in the recipe. "Recipes" editor, properties window of the recipe, "Properties" group in "Transfer".

## Structure of the data area

The data area has a fixed length of 5 words and is structured as follows:

| | 15 | 0 |
|---|---|---|
| 1. Word | Current recipe number (1 - 999) | |
| 2. Word | Current data mailbox 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 | Transfer active |
| 4 | 0000 0100 | Transfer completed without error |
| 12 | 0000 1100 | Transfer completed with error |

## Sequence of a transfer started by the operator in the recipe display

## Reading from the controller started by the operator in the recipe display

| Step | Action | |
|---|---|---|
| 1 | Check: Status word = 0? | |
| | yes | No |
| 2 | The HMI device enters the recipe number to be read and the status "Transfer active" in the data mailbox and sets the data mailbox number to 0. | Abort with system alarm. |
| 3 | The HMI device reads the values from the controller and displays them in the recipe display. If the recipes have synchronized tags, the values from the controller are also written to the tags. | |
| 4 | The HMI device sets the status "Transfer completed." | |
| 5 | To allow further transfers, the controller program must set the status word to 0 again. | |

## Writing to the controller started by the operator in the recipe display

| Step | Action | |
|---|---|---|
| | Check: Status word = 0? | |
| 1 | yes | No |
| | The HMI device enters the recipe and data mailbox number to be written and the status "Transfer active" in the data mailbox. | Abort with system alarm. |
| 2 | The HMI device writes the current values to the controller. If the recipes have synchronized tags, the changed values are synchronized between the recipe display and tags and then written to the controller. | |
| 3 | The HMI device sets the status "Transfer completed." | |
| 4 | If required, the controller program can now evaluate the transferred data. | |
| 5 | To allow further transfers, the controller program must set the status word to 0 again. | |

## Sequence of the transfer triggered by a job mailbox

Transfer of data mailboxes between the HMI device and controller can be initiated as follows:

- By the HMI device or
- By the controller

The two controller jobs No. 69 and No. 70 are available for this type of transfer.

## No. 69: Read data mailbox from controller ("PLC → DAT")

Job mailbox no. 69 transfers data mailboxes from the controller 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 mailbox number (1-65,535) | |
| Word 4 | Do not overwrite existing data mailbox: 0 Overwrite existing data mailbox: 1 | |

## No. 70: Write data mailbox to controller ("DAT → PLC")

Job mailbox no. 70 transfers data mailboxes from the HMI device to the controller. 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 mailbox number (1-65,535) | |
| Word 4 | — | |

## Sequence when reading from the controller 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 mailbox number specified in the job and the status "Transfer active" in the data mailbox. | Abort without return message. |
| 3 | The HMI device reads the values and stores the values in the data mailbox specified in the job mailbox. | |
| 4 | • If "Overwrite" was selected in the job, an existing data mailbox is overwritten without any prompt for confirmation. <br><br> The HMI device sets the status "Transfer completed." <br><br> • If "Do not overwrite" was selected in the job, and the data mailbox already exists, the HMI device aborts the job and enters 0000 1100 in the status word of the data mailbox. | |
| 5 | To allow further transfers, the controller program must set the status word to 0 again. | |

## Sequence of writing to the PLC using PLC job "DAT → PLC" (no. 70)

| Step | Action | |
|---|---|---|
| 1 | Check: Status word = 0? | |
| | yes | No |
| 2 | The HMI device enters the recipe and data mailbox number specified in the job and the status "Transfer active" in the data mailbox. | Abort without return message. |
| 3 | The HMI device fetches the values of the data mailbox specified in the job from the data medium and writes the values to the controller. | |
| 4 | The HMI device sets the status "Transfer completed." | |
| 5 | The controller program can now evaluate the transferred data. To allow further transfers, the controller program must set the status word to 0 again. | |

## Sequence of the transfer when triggered by a configured function

## Reading from the controller 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 "Transfer active" in the data mailbox. | Abort with system alarm. |
| 3 | The HMI device reads the values from the controller and stores them in the data mailbox specified in the function. | |
| 4 | • If "Yes" was selected for the "Overwrite" function, an existing data mailbox is overwritten without any prompt for confirmation. The HMI device sets the status "Transfer completed." <br> • If "No" was selected for the "Overwrite" function and the data mailbox already exists, the HMI device aborts the job and enters 0000 1100 in the status word of the data mailbox. | |
| 5 | To allow further transfers, the controller program must set the status word to 0 again. | |

## Writing to the controller with 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 "Transfer active" 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 writes the values to the controller. | |
| 4 | The HMI device sets the status "Transfer completed." | |
| 5 | The controller program can now evaluate the transferred data. To allow further transfers, the controller program must set the status word to 0 again. | |

## Possible causes of error when transferring data records

## Possible causes of error

If the transfer of data records is terminated with error, this may be the result of one of the following causes:

- Tag address not set up on the controller
- Overwriting data records not possible
- Recipe number does not exist
- Data record number does not exist

### Note

To ensure data consistency, the evaluation of the recipe and data mailbox number on the controller must only be started after the status in the data mailbox is set to "Transfer completed" or "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 display

  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 nevertheless evaluate the status of the transfer by querying the status word in the data mailbox.

## 8.3.2 Events, alarms, and acknowledgments

### 8.3.2.1 General information on events, alarms, and acknowledgments

#### Function

Alarms provide the user with information on controller or HMI device operating states or problems by displaying them on the HMI device. The alarm texts consist of configurable texts and/or tags with actual values.

Alarms and events must be distinguished. The programmer defines what is an event and what is an alarm.

#### Events

An event indicates a status, for example

- Motor on

- Controller in manual mode

#### Alarm

Alarms indicate operational errors, for example

- Valve does not open

- Excess motor temperature

Alarms indicate exceptional operational states, and must therefore be acknowledged.

#### Acknowledgment

Alarms can be acknowledged

- By operator input at the HMI device

- By the controller setting an acknowledgment bit.

#### Triggering alarms

An alarm is triggered on the controller

- By setting a tag bit

- By a measured value violating a limit.

The location of tags, or of the tag array, is defined in WinCC flexible ES. The tag or array must be set up on the controller.

### Number of alarms

The table below shows the number of alarms and words available for the various HMI devices.

| HMI device | Events and alarms | |
|---|---|---|
| | Words in total | Alarms in total |
| Standard PC | 250 | 4000 |
| MP 370 | 250 | 4000 |
| TP 270, MP 270, OP 270, MP 270B | 250 | 4000 |
| TP 177B, OP 177B Mobile Panel 177 | | |
| TP 177micro | | 500 |
| TP 170B, OP 170B Mobile Panel 170 | 125 | 2000 |
| TP 170A [1] | 63 | 1000 |
| TP 170micro | | 500 |
| OP 77B | 63 | 1000 |

[1] Only events possible.

### 8.3.2.2 Step 1: Creating tags or an array

### Procedure

You create tags or arrays in the "Tags" editor. The dialog box is shown below.



- Specifying the names of the tags or the name of the array
- Select the connection to the controller.

  The connection must already be configured in the "Connections" editor.
- Select the data type

  The available data types depend on the controller being used. If you select an illegal data type, the tag will not be available in the "Discrete alarms" and "Analog alarms" editors.

The following data types are supported for SIMATIC 500/505 controllers:

| Controller | Permitted data types | |
|---|---|---|
| | Discrete alarms | Analog alarms |
| SIMATIC 500/505 | Int, +/-Int | Bit, Int, +/– Int, Double, +/– Double, Real |

- Specify the address

  The tag addressed here contains the bit that triggers the alarm

  As soon as the bit of the tag is set on the controller and is transferred to the HMI device in the configured acquisition cycle, the HMI device recognizes the alarm as "incoming."

  After the same bit is reset on the controller, the HMI device recognizes the alarm as "outgoing."

- Specify array elements

  If the number of array elements is increased, you can select more bit numbers in the "Bit messages" editor. An array with 3 words, for example, makes 48 alarm bits available.

### 8.3.2.3    Step 2: Configuring an alarm

#### Procedure

Alarms fall into the following categories:

- Discrete alarms
- Analog alarms

You create alarms in the "Discrete alarms" and "Analog alarms" editors.

#### Discrete alarms

The editor is shown in the figure below.



- Edit text

  Here, you enter the text that will be shown in runtime. It can be formatted in fonts and contain fields for the visualization of tags.

  The text appears, for example, when an alarm view was configured in the "Screens" editor.

- Specify number

  Every alarm has a number that must only occur once within the project. It is used to uniquely identify the alarm and is indicated with the alarm in runtime.

  The permitted range of values is 1 to 100,000.

  The numbers are assigned consecutively in the WinCC engineering system. If required, you can change these numbers, for example when you want to group numbers together.

- Specify the alarm class

  Available alarm classes:

  – Error alarms

    This class must be acknowledged.

  – Process events

    This class signals events with incoming and outgoing alarms.

- Assign trigger tag

  In the "Trigger tag" column, you link the configured alarm with the tag created in step 1. All tags with permitted data types are shown in the selection list.

- Specify the bit number

  In the "bit number" column, specify the relevant bit position in the created tag.

  Remember that the way the bit positions are counted depends on the particular controller. With SIMATIC 500/505 controllers, the bit positions are counted as follows:

| How the bit positions are counted | Left byte | | | | | | | Right byte | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| in SIMATIC 500/505 controllers | 1 | | | | | | 8 | 9 | | | | | | | 16 |
| In the WinCC flexible, you configure the following: | 15 | | | | | | 8 | 7 | | | | | | | 0 |

## Analog alarms

The only difference between discrete alarms and analog alarms is that you configure a limit value, rather than a bit number. If the limit value is violated, the alarm is triggered. The outgoing alarm is triggered when the low limit is violated, and where applicable, taking into account any configured hysteresis.

### 8.3.2.4 Step 3: Configure the acknowledgment

**Procedure**

Create suitable tags on the controller to acknowledge an alarm. You assign these tags to an alarm in the "Bit messages" editor. You make the assignment in "Properties ▸ Acknowledgment",

The following figure shows the dialog for configuring an acknowledgment.



The following distinctions are made for acknowledgments:

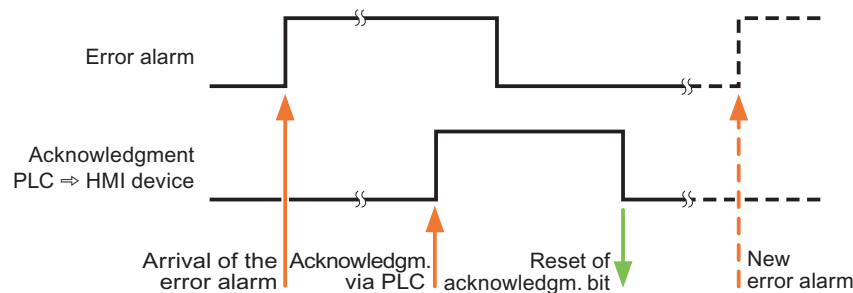- Acknowledgment on the HMI device
- Acknowledgment by the PLC

## Acknowledgment by the PLC

In "Ack write tag", you configure the tag or the array tag and the bit number based on which the HMI device can recognize an acknowledgment by the controller.

A bit set in the tag acknowledges the assigned alarm bit on the HMI device and, therefore, has the same function as an acknowledgment from the HMI device, for example, by pressing the "ACK" key.

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

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



## Acknowledgment on the HMI device

In "Ack read tag", you configure the tag or the array tag and the bit number that is written to the controller after acknowledgment from the HMI device.

To ensure that a signal change always occurs when the acknowledgment bit is set, the HMI device first resets the acknowledgment bit assigned to an alarm as soon as the alarm bit is set. There is a certain time-based offset between these two operations, due to the processing time of the HMI device.

If the alarm is acknowledged on the HMI device, the bit is then set in the assigned acknowledgment tag on the controller. This allows the controller to recognize that the alarm has been acknowledged.

The figure below shows the pulse diagram.

## 8.3.3 Trend request and trend transfer

### Function

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

### Time-triggered trends

The HMI device reads in the trend values cyclically at an interval specified in the configuration. Time-triggered trends are suitable for continuous processes, for example, the operating temperature of a motor.

### Bit-triggered trends

By setting a trigger bit in the trend request tag, the HMI device either reads in a trend value or an entire trend buffer. This is specified in the configuration. Bit-triggered trends are normally used to visualize rapidly changing values. One example might be the injection pressure in the production of plastic parts.

To trigger bit-triggered trends, create suitable external tags in the "Tags" editor of WinCC flexible and interconnect those to trend areas. The HMI device and controller then communicate with each other over these trend areas.

The following areas are available for trends:

- Trend request area
- Trend transfer area 1
- Trend transfer area 2 (required only with switch buffers)

Tags of the "V" or "K" types are permitted. They must be of the data type "Int", "+/- Int", or an array tag of the data type "Int", "+/-Int." During configuration you assign a bit to a trend. This sets a unique bit assignment for all areas.

### Trend request area

If a screen is opened on the HMI device with one or more trends, the HMI device sets the corresponding bits in the trend request area. After deselecting the screen, the HMI device resets the relevant bits in the trend request area.

Using the trend request area, the controller can recognize which trend is currently displayed on the HMI device. Trends can also be triggered without evaluating the curve request area.
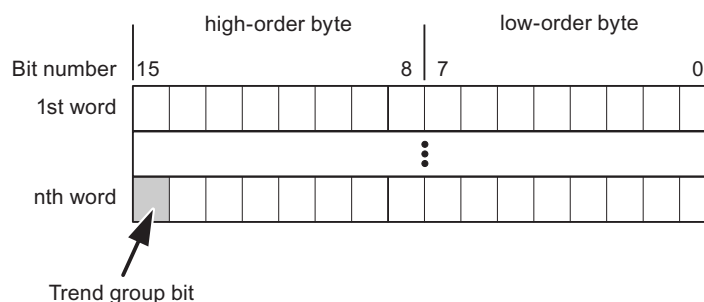
## Trend transfer area 1

This area is used to trigger trends. In your controller program, you set the bit assigned to the trend in the trend transfer area and set the trend group bit. The trend group bit is the last bit in the trend transfer area.

The HMI device recognizes the trigger and either reads a value or the entire buffer from the controller. 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 controller 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. It is structured exactly the same as trend transfer area 1.

## Switch buffers

The switch buffer is a second buffer for the same trend that can be set up during configuration.

While the HMI device reads the values from buffer 1, the controller writes to buffer 2. If the HMI device is reading buffer 2, the controller writes to buffer 1. This prevents the trend values being overwritten by the controller while the trend is being read out by the HMI device.

## 8.3.4 LED mapping

### Function

The HMI devices Operator Panel (OP), Multi Panel (MP) and Panel PC have LEDs in their function keys. These LEDs can be activated by the controller. In a particular situation, this allows the controller to activate an LED, for example, to tell the operator which key to press.

### Requirement

To be able to activate an LED, an LED tag or an array tag must be set up on the controller and specified as an LED tag during configuration.

### LED assignment

The assignment of individual LEDs to the bits in the LED tags is specified when you configure the function keys. You specify the "LED tag" and the assigned "bit" in the properties window in the "General" group.

The bit number "bit" identifies the first of two consecutive bits that control the following LED states:

| Bit n+ 1 | Bit n | LED function |
|:---:|:---:|:---:|
| 0 | 0 | Off |
| 0 | 1 | Flashing |
| 1 | 0 | Flashing |
| 1 | 1 | On permanently |

## 8.4 Commissioning components

## 8.4.1 Commissioning components

### Transferring the PLC program to the controller

1. Connect the PC and CPU with the appropriate cable.

2. Download the program files to the CPU.

3. Then set the CPU to RUN.

## Transferring the project to the HMI device

1. To transfer the project, the HMI device must be in transfer mode.

   A distinction must be made between the following situations:

   – Initial startup

   When commissioning for the first time, there is no configuration on the HMI device. The project data and runtime software required for operation must be transferred from the configuration computer to the device: The HMI device automatically changes to transfer mode.
   The transfer dialog with a connection message appears on the HMI device.

   – Recommissioning

   When you recommission, you replace an existing configuration on the HMI device with another.

   For detailed information on how to do this, refer to the manual supplied with your HMI device.

2. In your project, check whether the alarm settings meet your requirements.

3. Before you transfer the project to the HMI device, select the "Project ▶ Transfer ▶ Transfer settings" menu and set the transfer parameters:

   – Select the interface to be used.

   – Select the transfer parameters "Mode", "Interface" and Baud rate".

   – Select the destination storage location "Flash" or RAM".

4. Start the transfer of the project with the "Transfer" button.

   – The project is compiled automatically

   – Each step of the compilation and transfer is logged in an output window.

   If the transfer was successful, the following message is displayed: "Transfer completed."

   The start screen appears on the HMI device.

## Connecting the controller and the HMI device

1. Connect the controller and the HMI device using a suitable cable.

2. The message "Connection to controller .... is established" is output to the HMI device. Please remember that the system alarm texts can be edited by the user in WinCC flexible.

---

**Notice**

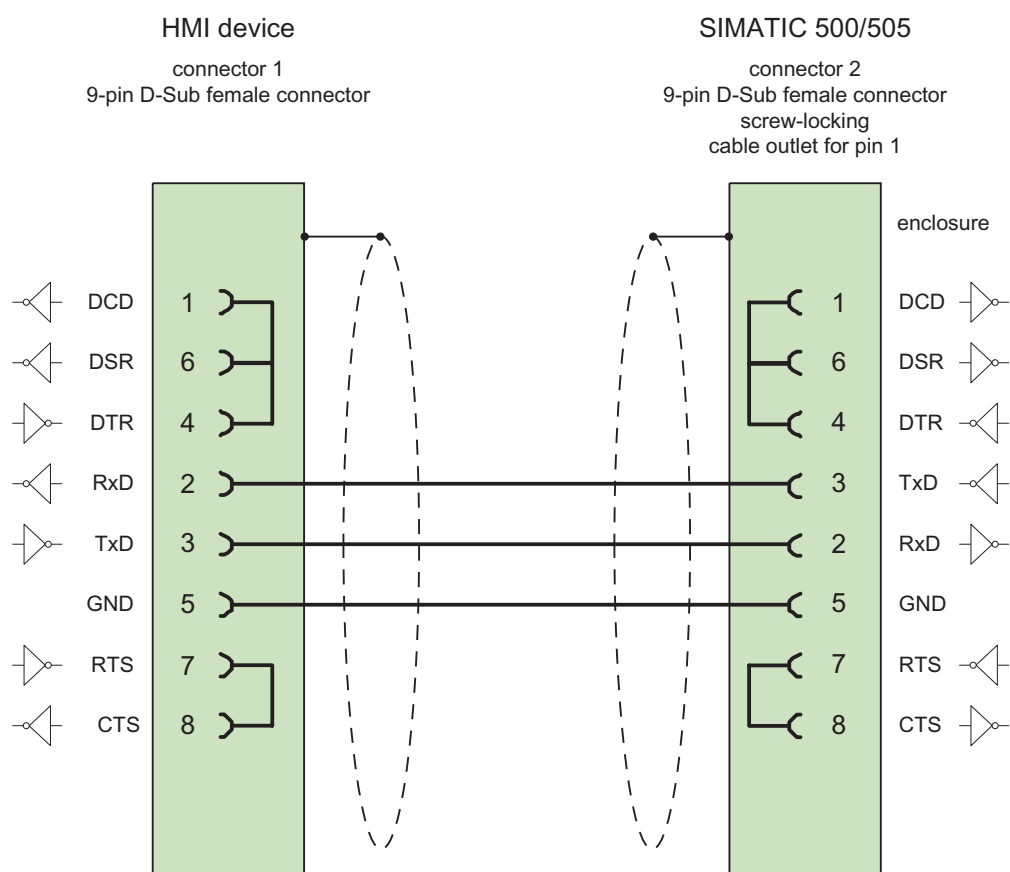During commissioning make sure you are aware of and keep to the safety-related notices in the HMI device manual.

Remember that high-frequency emissions, for example from mobile phones, can cause unwanted operating states.

---

# 8.5 Connecting cables for SIMATIC 500/505
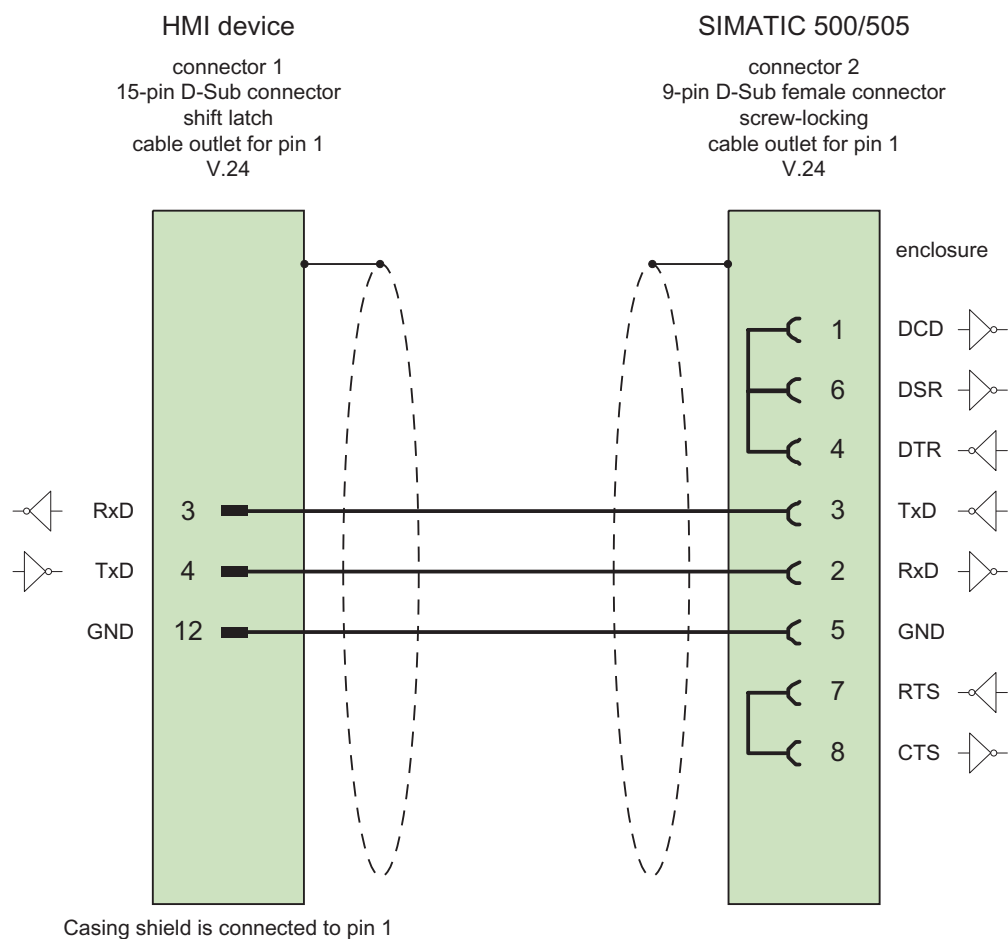
## 8.5.1 Connecting cable PP1, RS-232, for SIMATIC 500/505

### PP1 connecting cable

| HMI device | SIMATIC 500/505 |
|---|---|
| connector 1 | connector 2 |
| 9-pin D-Sub female connector | 9-pin D-Sub female connector |
| | screw-locking |
| | cable outlet for pin 1 |

| | | | | |
|---|---|---|---|---|
| DCD | 1 | | 1 | DCD |
| DSR | 6 | | 6 | DSR |
| DTR | 4 | | 4 | DTR |
| RxD | 2 | | 3 | TxD |
| TxD | 3 | | 2 | RxD |
| GND | 5 | | 5 | GND |
| RTS | 7 | | 7 | RTS |
| CTS | 8 | | 8 | CTS |

enclosure

Shield with large-area contact to casing
Cable: 3 x 0.14 mm², shielded,
max. length 15 m

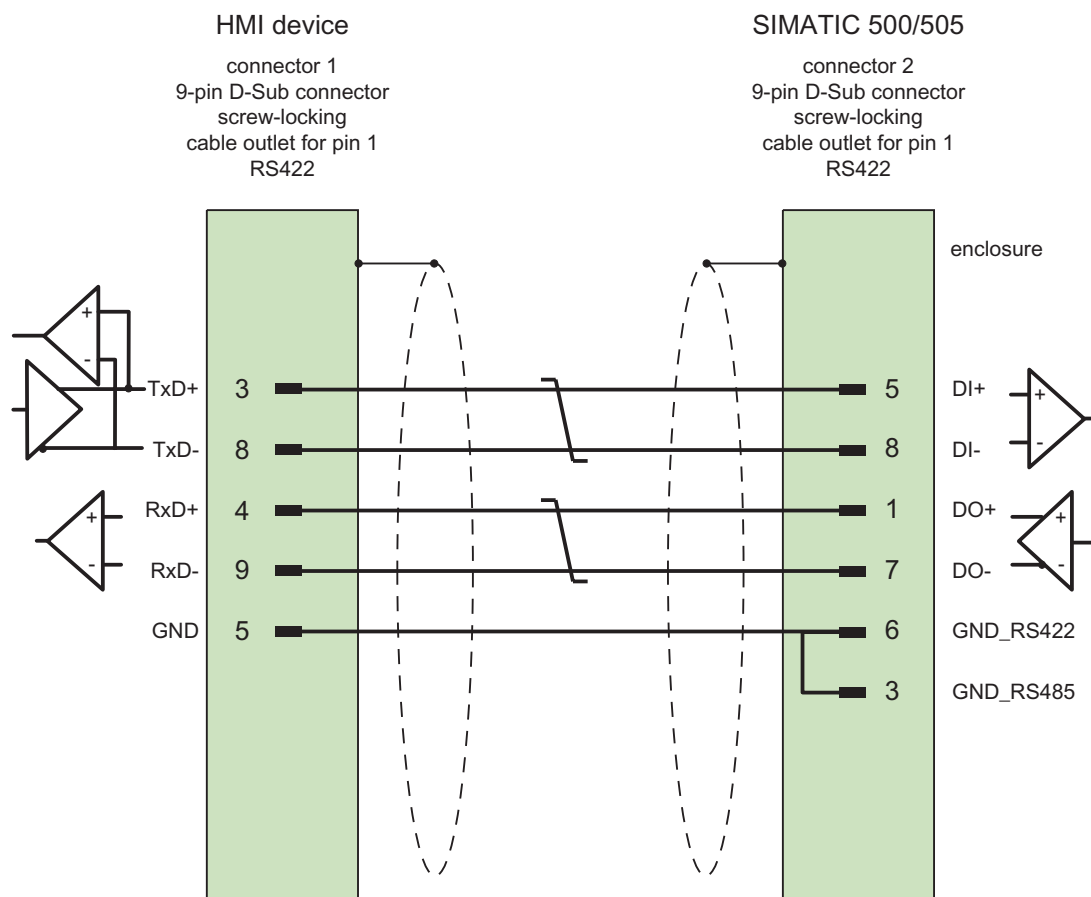## 8.5.2 Connecting cable 6XV1440-2K, RS-232, for SIMATIC 500/505

### 6XV1440 –K...

| HMI device | SIMATIC 500/505 |
|---|---|
| connector 1 | connector 2 |
| 15-pin D-Sub connector | 9-pin D-Sub female connector |
| shift latch | screw-locking |
| cable outlet for pin 1 | cable outlet for pin 1 |
| V.24 | V.24 |

enclosure

| | | | | | |
|---|---|---|---|---|---|
| | | | 1 | DCD | |
| | | | 6 | DSR | |
| | | | 4 | DTR | |
| RxD | 3 | | 3 | TxD | |
| TxD | 4 | | 2 | RxD | |
| GND | 12 | | 5 | GND | |
| | | | 7 | RTS | |
| | | | 8 | CTS | |

Casing shield is connected to pin 1

Shield with large-area contact to casing
Cable: 5 x 0.14 mm$^2$, shielded,
max. length 15 m

## 8.5.3 Connecting cable 6XV1440-2L, RS-232, for SIMATIC 500/505

### 6XV1440 –L... ...

| HMI device | SIMATIC 500/505 |
|---|---|
| connector 1<br>15-pin D-Sub connector<br>shift latch<br>cable outlet for pin 1<br>V.24 | connector 2<br>25-pin D-Sub connector<br>screw-locking<br>cable outlet for pin 1<br>V.24 |

| | | | | | |
|---|---|---|---|---|---|
| PE | 1 | | | | enclosure |
| | 8 | | | | |
| RxD | 3 | | 2 | TxD | |
| TxD | 4 | | 3 | RxD | |
| RTS | 10 | | 4 | RTS | |
| CTS | 5 | | 5 | CTS | |
| GND | 12 | | 7 | GND | |
| | | | 6 | DSR | |
| | | | 20 | DTR | |
| | | | 8 | DCD | |

Casing shield is connected to pin 8          Casing shield is connected to pin 25
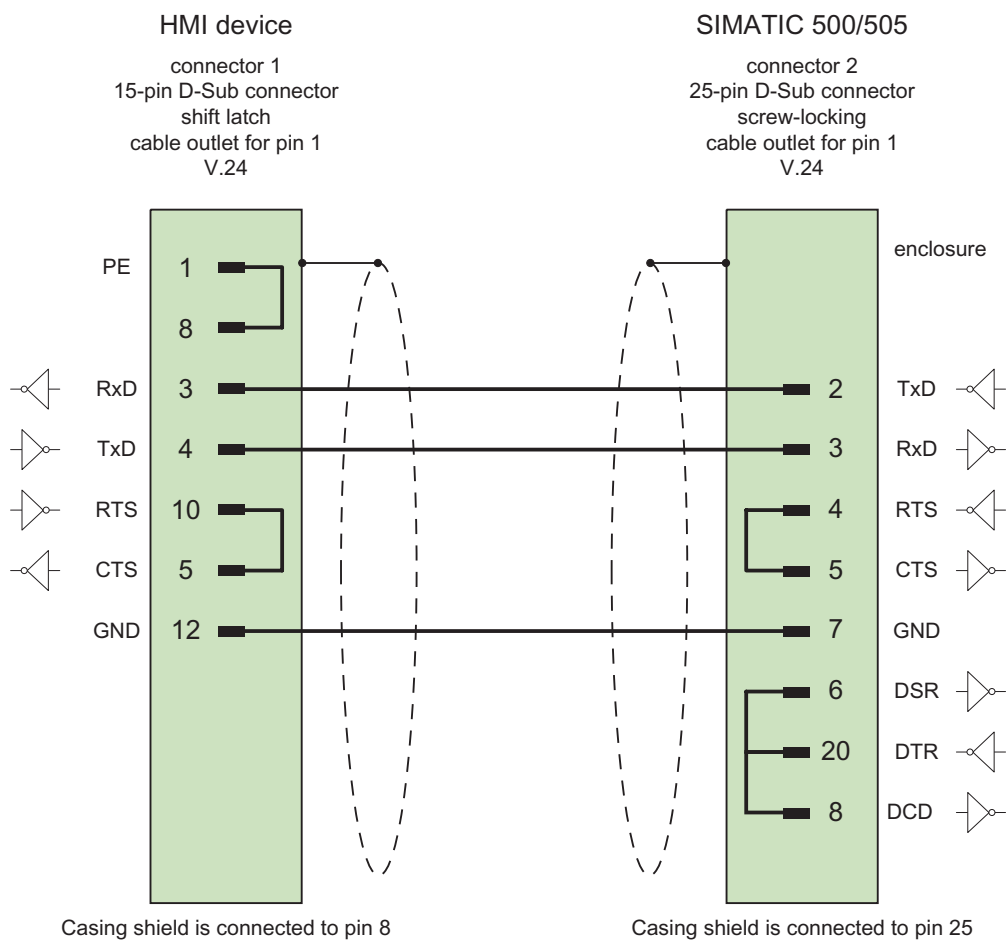
Shield with large-area contact to casing
Cable: 5 x 0.14 mm², shielded,
max. length 15 m

## 8.5.4 Connecting cable 6XV1440-1M, RS-422, for SIMATIC 505

### 6XV1440 –M... (PLC 545 / CPU 1102, PLC 555))



| HMI device | SIMATIC 505 |
|---|---|
| connector 1 | connector 2 |
| 9-pin D-Sub connector | 9-pin D-Sub female connector |
| screw-locking | screw-locking |
| cable outlet for pin 1 | cable outlet for pin 1 |
| RS422 | RS422 |
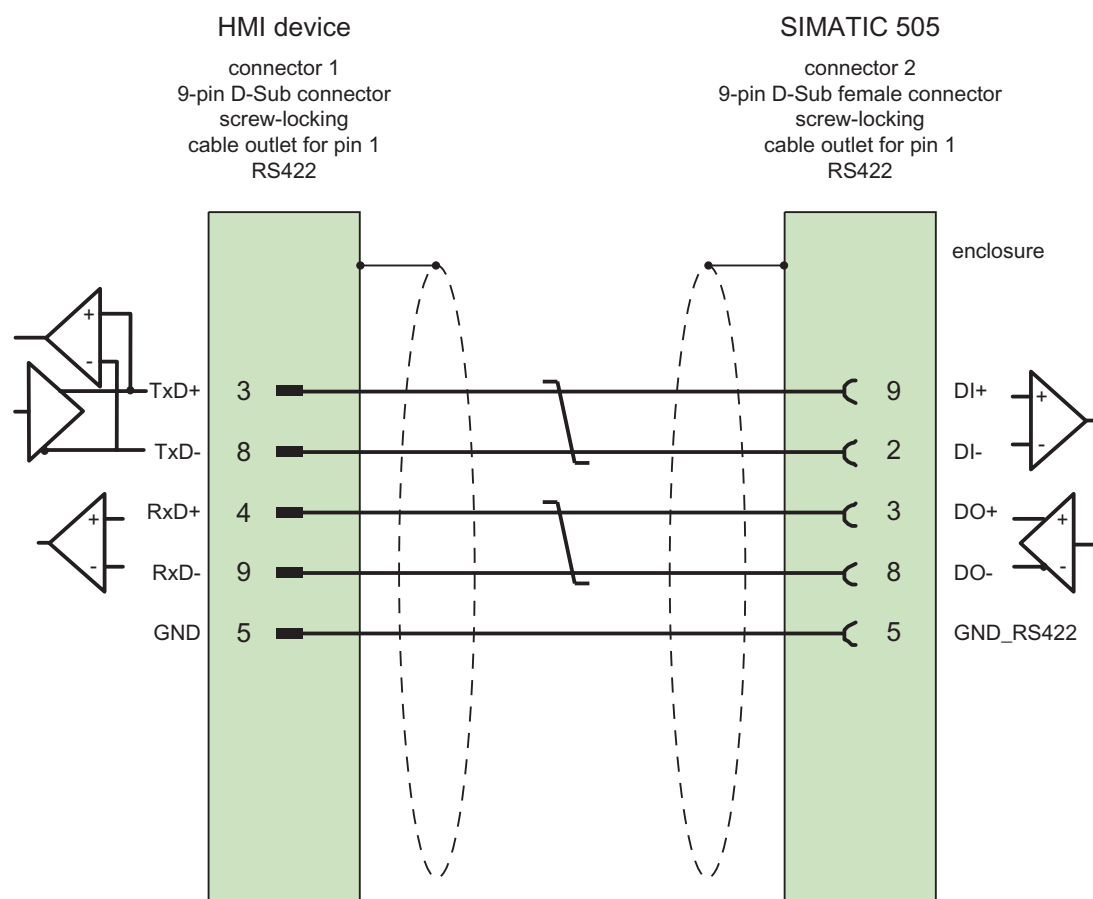
Shield with large-area contact to casing
Cable: 3 x 2 x 0.14 mm², shielded,
max. length 300 m

## 8.5.5 Connecting cable 6XV1440-2M, RS-422, for SIMATIC 500/505

### 6XV1440 –M... (PLC 525, 545 / CPU 1101, PLC 565T) )

HMI device

connector 1
9-pin D-Sub connector
screw-locking
cable outlet for pin 1
RS422

SIMATIC 500/505

connector 2
9-pin D-Sub connector
screw-locking
cable outlet for pin 1
RS422

enclosure

| | | | |
|---|---|---|---|
| TxD+ | 3 | 5 | DI+ |
| TxD- | 8 | 8 | DI- |
| RxD+ | 4 | 1 | DO+ |
| RxD- | 9 | 7 | DO- |
| GND | 5 | 6 | GND_RS422 |
| | | 3 | GND_RS485 |

Shield with large-area contact to casing
Cable: 3 x 2 x 0.14 mm$^2$, shielded,
max. length 300 m

# Appendix 9

## 9.1 System alarms

### Introduction

System alarms on the HMI device provide information about internal states of the HMI device and PLC.

The overview below shows the causes of system alarms and how to eliminate the cause of error.

Some of the system alarms described in this section are relevant to individual HMI devices based on their range of features.

---

### Note

System alarms are only indicated if an alarm window was configured. System alarms are output in the language currently set on your HMI device.

---

### System alarm parameters

System alarms may contain encrypted parameters which are relevant to troubleshooting because they provide a reference to the source code of the Runtime software. These parameters are output after the text "Error code:"

### Meaning of the system alarms

| Number | Effect/cause | Remedy |
|--------|-------------|--------|
| 10000 | The print job could not be started or was canceled due to an unknown error. Faulty printer setup. Or: No permission is available for accessing the network printer. Power supply failure during data transfer. | Check the printer settings, cable connections and the power supply. Set up the printer once again. Obtain a network printer authorization. If the error persists, contact the Hotline! |
| 10001 | No printer is installed or a default printer has not been set up. | Install a printer and/or select it as the default printer. |
| 10002 | Overflow of the graphics buffer for printing. Up to two images are buffered. | Allow sufficient intervals between successive print jobs. |
| 10003 | Images can now be buffered again. | — |
| 10004 | Overflow of the buffer for printing lines in text mode (e.g.alarms). Up to 1000 lines are buffered. | Allow sufficient intervals between successive print jobs. |
| 10005 | Text lines can now be buffered again. | — |

| Number | Effect/cause | Remedy |
|--------|-------------|--------|
| 10006 | The Windows printing system reports an error. Refer to the output text and the error ID to determine the possible causes. Nothing is printed or the print is faulty. | Repeat the action if necessary. |
| 20010 | An error has occurred in the specified script line. Execution of the script was therefore aborted. Note the system alarm that may have occurred prior to this. | Select the specified script line in the configuration. Ensure that the tags used are of the allowed types. Check system functions for the correct number and types of parameters. |
| 20011 | An error has occurred in a script that was called by the specified script. Execution of the script was therefore aborted in the called script. Take the system alarm that may have occurred prior to this into account. | In the configuration, select the script that has been called directly or indirectly by the specified script. Ensure that the tags used are of the allowed types. Check the system functions for the correct number and type of parameters. |
| 20012 | The configuration data is inconsistent. The script could therefore not be generated. | Recompile the configuration. |
| 20013 | The scripting component of WinCC flexible Runtime is not correctly installed. Therefore, no scripts can be executed. | Reinstall WinCC flexible Runtime. |
| 20014 | The system function returns a value that is not written in any return tag. | Select the specified script in the configuration. Check if the script name has been assigned a value. |
| 20015 | Too many successive scripts have been triggered in short intervals. When more than 20 scripts are queued for processing, any subsequent scripts are rejected. In this case, the script indicated in the alarm is not executed. | Find what is triggering the scripts. Extend the times, e.g. the polling time of the tags which trigger the scripts. |
| 30010 | The tag could not accept the function result, e.g. when it has exceeded the value range. | Check the tag type of the system function parameter. |
| 30011 | A system function could not be executed because the function was assigned an invalid value or type in the parameter. | Check the parameter value and tag type of the invalid parameter. If a tag is used as a parameter, check its value. |
| 40010 | The system function could not be executed since the parameters could not be converted to a common tag type. | Check the parameter types in the configuration. |
| 40011 | The system function could not be executed since the parameters could not be converted to a common tag type. | Check the parameter types in the configuration. |
| 50000 | The HMI device is receiving data faster than it is capable of processing. Therefore, no further data is accepted until all current data have been processed. Data exchange then resumes. | — |
| 50001 | Data exchange has been resumed. | — |
| 60000 | This alarm is generated by the "DisplaySystemAlarms" function. The text to be displayed is transferred to the function as a parameter. | — |
| 60010 | The file could not be copied in the direction defined because one of the two files is currently open or the source/target path is not available. It is possible that the Windows user has no access rights to one of the two files. | Restart the system function or check the paths of the source/target files. Using Windows NT/2000/XP: The user executing WinCC flexible Runtime must be granted access rights for the files. |

| Number | Effect/cause | Remedy |
|--------|--------------|--------|
| 60011 | An attempt was made to copy a file to itself.<br>It is possible that the Windows user has no access rights to one of the two files. | Check the path of the source/target file.<br>Using Windows NT/2000/XP with NTFS: The user executing WinCC flexible Runtime must be granted access rights for the files. |
| 70010 | The application could not be started because it could not be found in the path specified or there is insufficient memory space. | Check if the application exists in the specified path or close other applications. |
| 70011 | The system time could not be modified.<br>The error alarm only appears in connection with area pointer "Date/time PC". Possible causes:<br>• An invalid time was transferred in the job mailbox.<br>• The Windows user has no right to modify the system time.<br>If the first parameter in the system alarm is displayed with the value 13, the second parameter indicates the byte containing the incorrect value. | Check the time which is to be set.<br>Using Windows NT/2000/XP: The user executing WinCC flexible Runtime must be granted the right to change the system time of the operating system. |
| 70012 | An error occurred when executing the function "StopRuntime" with the option "Runtime and operating system".<br>Windows and WinCC flexible Runtime are not closed.<br>One possible cause is that other programs cannot be closed. | Close all programs currently running.<br>Then close Windows. |
| 70013 | The system time could not be modified because an invalid value was entered. Incorrect separators may have been used. | Check the time which is to be set. |
| 70014 | The system time could not be modified. Possible causes:<br>• An invalid time was transferred.<br>• The Windows user has no right to modify the system time.<br>• Windows rejects the setting request. | Check the time which is to be set.<br>Using Windows NT/2000/XP: The user executing WinCC flexible Runtime must be granted the right to change the system time of the operating system. |
| 70015 | The system time could not be read because Windows rejects the reading function. | — |
| 70016 | An attempt was made to select a screen by means of a system function or job. This is not possible because the screen number specified does not exist.<br>Or: A screen could not be generated due to insufficient system memory. | Check the screen number in the function or job with the screen numbers configured.<br>Assign the number to a screen if necessary. |
| 70017 | Date/Time is not read from the area pointer because the address set in the controller is either not available or has not been set up. | Change the address or set up the address in the controller. |
| 70018 | Acknowledgment that the password list has been successfully imported. | — |
| 70019 | Acknowledgment that the password list has been successfully exported. | — |
| 70020 | Acknowledgment for activation of alarm reporting. | — |
| 70021 | Acknowledgment for deactivation of alarm reporting. | — |

| Number | Effect/cause | Remedy |
|--------|--------------|--------|
| 70022 | Acknowledgment to starting the Import Password List action. | — |
| 70023 | Acknowledgment to starting the Export Password List action. | — |
| 70024 | The value range of the tag has been exceeded in the system function. The calculation of the system function is not performed. | Check the desired calculation and correct it if necessary. |
| 70025 | The value range of the tag has been exceeded in the system function. The calculation of the system function is not performed. | Check the desired calculation and correct it if necessary. |
| 70026 | No other screens are stored in the internal screen memory. No other screens can be selected. | — |
| 70027 | The backup of the RAM file system has been started. | — |
| 70028 | The files from the RAM have been copied in the Flash memory. The files from the RAM have been copied in the Flash memory. Following a restart, these saved files are copied back to the RAM file system. | — |
| 70029 | Backup of the RAM file system has failed. No backup copy of the RAM file system has been made. | Check the settings in the "Control Panel > OP" dialog and save the RAM file system using the "Save Files" button in the "Persistent Storage" tab. |
| 70030 | The parameters configured for the system function are faulty. The connection to the new controller was not established. | Compare the parameters configured for the system function with the parameters configured for the controllers and correct them as necessary. |
| 70031 | The controller configured in the system function is not an S7 controller. The connection to the new controller was not established. | Compare the S7 controller name parameter configured for the system function with the parameters configured for the controller and correct them as necessary. |
| 70032 | The object configured with this number in the tab order is not available in the selected screen. The screen changes but the focus is set to the first object. | Check the number of the tab order and correct it if necessary. |
| 70033 | An e-mail cannot be sent because a TCP/IP connection to the SMTP server no longer exists. This system alarm is generated only at the first attempt. All subsequent unsuccessful attempts to send an e-mail will no longer generate a system alarm. The event is regenerated when an e-mail has been successfully sent in the meantime. The central e-mail component in WinCC flexible Runtime attempts, in regular intervals (1 minute), to establish the connection to the SMTP server and to send the remaining e-mails. | Check the network connection to the SMTP server and re-establish it if necessary. |
| 70034 | Following a disruption, the TCP/IP connection to the SMTP server could be re-established. The queued e-mails are then sent. | — |

| Number | Effect/cause | Remedy |
|---|---|---|
| 70036 | No SMTP server for sending e-mails is configured. An attempt to connect to an SMTP server has failed and it is not possible to send e-mails. WinCC flexible Runtime generates the system alarm after the first attempt to send an e-mail. | Configure an SMTP server: In WinCC flexible Engineering System using "Device settings > Device settings" In the Windows CE operating system using "Control Panel > Internet Settings > E-mail > SMTP Server" |
| 70037 | An e-mail cannot be sent for unknown reasons. The contents of the e-mail are lost. | Check the e-mail parameters (recipient etc.). |
| 70038 | The SMTP server has rejected sending or forwarding an e-mail because the domain of the recipient is unknown to the server or because the SMTP server requires authentication. The contents of the e-mail are lost. | Check the domain of the recipient address or deactivate the authentication on the SMTP server if possible. SMTP authentication is currently not used in WinCC flexible Runtime. |
| 70039 | The syntax of the e-mail address is incorrect or contains illegal characters. The contents of the e-mail are discarded. | Check the e-mail address of the recipient. |
| 70040 | The syntax of the e-mail address is incorrect or contains illegal characters. | — |
| 70041 | The import of the user management was aborted due to an error. Nothing was imported. | Check your user management or transfer it again to the panel. |
| 80001 | The log specified is filled to the size defined (in percent) and must be stored elsewhere. | Store the file or table by executing a 'move' or 'copy' function. |
| 80002 | A line is missing in the specified log. | — |
| 80003 | The copying process for logging was not successful. In this case, it is advisable to check any subsequent system alarms, too. | — |
| 80006 | Since logging is not possible, this causes a permanent loss of the functionality. | In the case of databases, check if the corresponding data source exists and start up the system again. |
| 80009 | A copying action has been completed successfully. | — |
| 80010 | Since the storage location was incorrectly entered in WinCC flexible, this causes a permanent loss of the functionality. | Configure the storage location for the respective log again and restart the system when the full functionality is required. |
| 80012 | Log entries are stored in a buffer. If the values are read to the buffer faster than they can be physically written (using a hard disk, for example), overloading may occur and recording is then stopped. | Archive fewer values. Or: Increase the logging cycle. |
| 80013 | The overload status no longer applies. Archiving resumes the recording of all values. | — |
| 80014 | The same action was triggered twice in quick succession. Since the process is already in operation, the action is only carried out once. | — |
| 80015 | This system alarm is used to report DOS or database errors to the user. | — |
| 80016 | The logs are separated by the system function "CloseAllLogs" and the incoming entries exceed the defined buffer size. All entries in the buffer are deleted. | Reconnect the logs. |

| Number | Effect/cause | Remedy |
|--------|-------------|--------|
| 80017 | The number of incoming events cause a buffer overflow. his can be caused, for example, by several copying actions being activated at the same time.<br>All copy jobs are deleted. | Stop the copy action. |
| 80019 | The connection between WinCC flexible and all logs were closed, for example, after executing the system function "CloseAllLogs".<br>Entries are written to the buffer and are then written to the logs when a connection is re-established.<br>There is no connection to the storage location and the storage medium may be replaced, for example. | — |
| 80020 | The maximum number of simultaneously copy operations has been exceeded. Copying is not executed. | Wait until the current copying actions have been completed, then restart the last copy action. |
| 80021 | An attempt was made to delete a log which is still busy with a copy action. Deletion has not been executed. | Wait until the current copying actions have been completed, then restart the last action |
| 80022 | An attempt was made to start a sequence log, which is not a sequence log, from a log using the system function "StartSequenceLog". No sequence log file is created. | In the project, check<br>• if the "StartSequenceLog" system function was properly configured<br>• if the tag parameters are properly provided with data on the HMI device |
| 80023 | An attempt was made to copy a log to itself.<br>The log is not copied. | In the project, check<br>• if the "CopyLog" system function was properly configured<br>• if the tag parameters are properly provided with data on the HMI device |
| 80024 | The "CopyLog" system function does not allow copying when the target log already contains data ("Mode" parameter). The log is not copied. | Edit the "CopyLog" system function in the project if necessary. Before you initiate the system function, delete the destination log file. |
| 80025 | You have canceled the copy operation.<br>Data written up to this point are retained. The destination log file (if configured) is not deleted.<br>The cancellation is reported by an error entry $RT_ERR$ at the end of the destination log. | — |
| 80026 | This alarm is output after all logs are initialized. Values are written to the logs from then on. Prior to this, no entries are written to the logs, irrespective whether WinCC flexible Runtime is active or not. | — |
| 80027 | The internal Flash memory has been specified as the storage location for a log. This is not permissible.<br>No values are written to this log and the log file is not created. | Configure "Storage Card" or a network path as the storage location. |
| 80028 | The alarm returns a status report indicating that the logs are currently being initialized. No values are logged until the alarm 80026 is output. | — |
| 80029 | The number of logs specified in the alarm could not be initialized. The logs are initialized.<br> The faulty log files are not available for logging jobs. | Evaluate the additional system alarms , related to this alarm which is also generated.<br>Check the configuration, the ODBC (Open Database Connectivity) and the specified drive. |

| Number | Effect/cause | Remedy |
|--------|-------------|--------|
| 80030 | The structure of the existing log file does not match the expected structure. Logging is stopped for this log. | Delete the existing log data manually, in advance. |
| 80031 | The log in CSV format is corrupted. The log cannot be used. | Delete the faulty file. |
| 80032 | Logs can be assigned events. These are triggered as soon as the log is full. If WinCC flexible Runtime is started and the log is already full, the event is not triggered. The log specified no longer logs data because it is full. | Close WinCC flexible Runtime, delete the log, then restart WinCC flexible Runtime. Or: Configure a button which contains the same actions as the event and press it. |
| 80033 | "System Defined" is set in the data log file as the data source name. This causes an error. No data is written to the database logs, whereas the logging to the CSV logs works. | Install MSDE again. |
| 80034 | An error has occurred in the initialization of the logs. An attempt has been made to create the tables as a backup. This action was successful. A backup has been made of the tables of the corrupted log file and the cleared log was restarted. | No action is necessary. However, it is recommended to save the backup files or delete them in order to make the space available again. |
| 80035 | An error has occurred in the initialization of the logs. An attempt has been made to create backups of the tables and this has failed. No logging or backup has been performed. | It is recommended to save the backups or to delete them in order to release memory. |
| 80044 | The export of a log was interrupted because Runtime was closed or due to a power failure. It was detected that the export needed to be resume when Runtime restarted. | The export resumes automatically. |
| 80045 | The export of a log was interrupted due to an error in the connection to the server or at the server itself. | The export is repeated automatically. Check: <br> • the connection to the server <br> • if the server is running <br> • if there is enough free space on the server. |
| 80046 | The destination file could not be written while exporting the log. | Check if there is enough space on the server and it you have permission to create the log file. |
| 80047 | The log could not be read while exporting it. | Check if the storage medium is correctly inserted. |
| 80048 | — | — |
| 80049 | The log could not be renamed while preparing to export it. The job can not be completed." | Check if the storage medium is correctly inserted and if there is sufficient space on the medium. |
| 80050 | The log which shall be exported is not closed. The job can not be completed. | Make sure the "CloseAll Logs" system function is called before using the "ExportLog" system function. Change the configuration as required. |
| 90024 | No operator actions can be logged due to lack of space on the storage medium for log. The operator action will therefore not be executed. | Make more space available by inserting an empty storage medium or swapping out the log files on the server using "ExportLog". |
| 90025 | Because of error state of the archive no user actions can be logged. Therefore the user action will not be executed. | Check if the storage medium is correctly inserted. |
| 90026 | No operator actions can be logged because the log is closed. The operator action will therefore not be executed. | Before further operator actions are carried out, the log must be opened again using the system function "OpenAllLogs". Change the configuration as required. |

| Number | Effect/cause | Remedy |
|--------|-------------|--------|
| 90029 | Runtime was closed during ongoing operation (perhaps due to a power failure) or an a storage medium in use is incompatible with Audit Trail. An Audit Trail is not suitable if it belongs to another project or has already be archived. | Ensure that you are using the correct storage medium. |
| 90030 | Runtime was closed during ongoing operation (perhaps due to a power failure). | — |
| 90031 | Runtime was closed during ongoing operation (perhaps due to a power failure). | — |
| 90032 | Running out of space on the storage medium for log. | Make more space available by inserting an empty storage medium or swapping out the log files on the server using "ExportLog". |
| 90033 | No more space on the storage medium for log. As of now, no more operator actions requiring logging will be executed. | Make more space available by inserting an empty storage medium or swapping out the log files on the server using "ExportLog". |
| 90040 | Audit Trail is switched off because of a forced user action. | Activate the "Audit Trail" again using the system function "StartLog". |
| 90041 | A user action which has to be logged has been executed without a logged on user. | A user action requiring logging should only be possible with permission. Change the configuration by setting a required permission for the input object. |
| 90044 | A user action which has to be confirmed was blocked, because there is another user action pending. | Repeat the user action if necessary. |
| 110000 | The operating mode was changed. "Offline" mode is now set. | — |
| 110001 | The operating mode was changed. "Online" mode is now set. | — |
| 110002 | The operating mode was not changed. | Check the connection to the controllers. Check if the address area for the area pointer 88 "Coordination" in the controller is available. |
| 110003 | The operating mode of the specified controller was changed by the system function "SetConnectionMode". The operating mode is now "offline". | — |
| 110004 | The operating mode of the specified controller has been changed by the system function "SetConnectionMode". The operating mode is now "online". | — |
| 110005 | An attempt was made to use the system function SetConnectionMode to switch the specified controller to "online" mode, although the entire system is in "offline" mode. This changeover is not allowed. The controller remains in "offline" mode. | Switch the complete system to "online" mode, then execute the system function again. |
| 110006 | The content of the "project version" area pointer does not match the user version configured in WinCC flexible. WinCC flexible Runtime is therefore closed. | Check: <br> • the project version entered on the controller <br> • the project version entered in WinCC flexible |
| 120000 | The trend is not displayed because you configured an incorrect axis to the trend or an incorrect trend. | Change the configuration. |
| 120001 | The trend is not displayed because you configured an incorrect axis to the trend or an incorrect trend. | Change the configuration. |

| Number | Effect/cause | Remedy |
|--------|-------------|--------|
| 120002 | The trend is not displayed because the tag assigned attempts to access an invalid controller address. | Check if the data area for the tag exists in the controller, the configured address is correct and the value range for the tag is correct. |
| 130000 | The action was not executed. | Close all other programs. Delete files no longer required from the hard disk. |
| 130001 | The action was not executed. | Delete files no longer required from the hard disk. |
| 130002 | The action was not executed. | Close all other programs. Delete files no longer required from the hard disk. |
| 130003 | No data medium found. The operation is canceled. | Check, for example, if <br> • the correct data medium is being accessed <br> • the data medium is inserted |
| 130004 | The data medium is write-protected. The operation is canceled. | Check if access has been made to the correct data carrier. Remove the write protection. |
| 130005 | The file is read only. The operation is canceled. | Check if access has been made to the correct file. Edit the file attributes if necessary. |
| 130006 | Access to file failed. The operation is canceled. | Check, for example, if <br> • the correct file is being accessed <br> • the file exists <br> • another action is preventing simultaneous access to the file |
| 130007 | The network connection is interrupted. Data records cannot be saved or read over the network connection. | Check the network connection and eliminate the cause of error. |
| 130008 | The storage card is not available. Data records cannot be saved to / read from the storage card. | Insert the storage card. |
| 130009 | The specified folder does not exist on the storage card. Any files saved to this directory are not backed up when you switch off the HMI device. | Insert the storage card. |
| 130010 | The maximum nesting depth can be exhausted when, for example, a value change in a script results in the call of another script and the second script in turn has a value change that results in the call of yet a further script etc. The configured functionality is not supported. | Check the configuration. |
| 140000 | An online connection to the controller is established. | — |
| 140001 | The online connection to the controller was shut down. | — |
| 140003 | No tag updating or writing is executed. | Check the connection and if the controller is switched on. Check the parameter definitions in the Control Panel using "Set PG/PC interface". Restart the system. |
| 140004 | No tag update or write operations are executed because the access point or the module configuration is faulty. | Verify the connection and check if the controller is switched on. Check the access point or the module configuration (MPI, PPI, PROFIBUS) in the Control Panel with "Set PG/PC interface". Restart the system. |

| Number | Effect/cause | Remedy |
|--------|-------------|--------|
| 140005 | No tag updating or writing is executed because the HMI device address is incorrect (possibly too high). | Use a different HMI device address.<br>Verify the connection and check ifthe controller is switched on.<br>Check the parameter definitions in the Control Panel using "Set PG/PC interface".<br>Restart the system. |
| 140006 | No tag updating or writing is executed because the baud rate is incorrect. | Select a different baud rate in WinCC flexible (according to module, profile, communication peer, etc.). |
| 140007 | Tag are not updated or written because the bus profile is incorrect (see %1).<br>The following parameter could not be written to the registry::<br>1: Tslot<br>2: Tqui<br>3: Tset<br>4: MinTsdr<br>5: MaxTsdr<br>6: Trdy<br>7: Tid1<br>8: Tid2<br>9: Gap Factor<br>10: Retry Limit | Check the user-defined bus profile.<br>Check the connection and if the controller is switched on.<br>Check the parameter definitions in the Control Panel using "Set PG/PC interface".<br>Restart the system. |
| 140008 | No tag updating or writing is executed because baud rate is incorrect. The following parameters could not be written to the registry:<br>0: General error<br>1: Wrong version<br>2: Profile cannot be written to the registry.<br>3: The subnet type cannot be written to the registry.<br>4: The Target Rotation Time cannot be written to the registry.<br>5: Faulty Highest Address (HSA). | Check the connection and if the controller is switched on.<br>Check the parameter definitions in the Control Panel using "Set PG/PC interface".<br>Restart the system. |
| 140009 | Tags are not updated or written because the module for S7 communication was not found. | Reinstall the module in the Control Panel using "Set PG/PC interface". |
| 140010 | No S7 communication partner found because the controller is shut down.<br>DP/T:<br>The option "PG/PC is the only master" is not set in the Control Panel under "Set PG/PC interface." | Switch the controller on.<br>DP/T:<br>If only one master is connected to the network, disable "PG/PC is the only master" in "Set PG/PC interface".<br>If several masters are connected to the network, enable these. Do not change any settings, for this will cause bus errors. |
| 140011 | No tag updating or writing is executed because communication is down. | Check the connection and that the communication partner is switched on. |
| 140012 | There is an initialization problem (e.g. when WinCC flexible Runtime was closed in Task Manager).<br>Or:<br>Another application (e.g.STEP7) with different bus parameters is active and the driver cannot be started with the new bus parameters (transmission rate, for example). | Restart the HMI device.<br>Or:<br>Run WinCC flexible Runtime, then start your other applications. |

| Number | Effect/cause | Remedy |
|---|---|---|
| 140013 | The MPI cable is disconnected and, thus, there is no power supply. | Check the connections. |
| 140014 | The configured bus address is in already in use by another application. | Edit the HMI device address in the controller configuration. |
| 140015 | Wrong transmission rate<br>Or:<br>Faulty bus parameters (e.g.HSA)<br>Or:<br>OP address > HSA or: Wrong interrupt vector (interrupt does not arrive at the driver) | Correct the relevant parameters. |
| 140016 | The hardware does not support the configured interrupt. | Change the interrupt number. |
| 140017 | The set interrupt is in use by another driver. | Change the interrupt number. |
| 140018 | The consistency check was disabled by SIMOTION Scout. Only a corresponding note appears. | Enable the consistency check with SIMOTION Scout and once again download the project to the PLC. |
| 140019 | SIMOTION Scout is downloading a new project to the controller. Connection to the controller is canceled. | Wait until the end of the reconfiguration. |
| 140020 | The version in the controller and that of the project (FWX file) do not match.<br>Connection to the controller is canceled | The following remedies are available:<br>Download the current version to the PLC using SIMOTION Scout.<br>Regenerate the project using WinCC flexible ES, close WinCC flexible Runtime and restart with a new configuration. |
| 150000 | No more data is read or written. Possible causes:<br>• The cable is defective.<br>• The PLC does not respond, is defective, etc.<br>• The wrong port is used for the connection.<br>• System overload. | Ensure that the cable is plugged in, the controller is operational, the correct interface is being used.<br>Restart the system if the system alarm persists. |
| 150001 | Connection is up because the cause of the interruption has been eliminated. | — |
| 160000 | No more data is read or written. Possible causes:<br>• The cable is defective.<br>• The PLC does not respond, is defective, etc.<br>• The wrong port is used for the connection.<br>• System overload. | Ensure that the cable is plugged in, the controller is operational, the correct interface is being used.<br>Restart the system if the system alarm persists. |
| 160001 | Connection is up because the cause of the interruption has been eliminated. | — |
| 160010 | No connection to the server because the server identification (CLS-ID) cannot be determined. Values cannot be read or written. | Check access rights. |
| 160011 | No connection to the server because the server identification (CLS-ID) cannot be determined. Values cannot be read or written. | Check, for example, if<br>• the server name is correct<br>• the computer name is correct<br>• the server is registered |

| Number | Effect/cause | Remedy |
|--------|--------------|--------|
| 160012 | No connection to the server because the server identification (CLS-ID) cannot be determined. Values cannot be read or written. | Check, for example, if<br>• the server name is correct<br>• the computer name is correct<br>• the server is registered<br>Note for advanced users:<br>Interpret the value from HRESULT. |
| 160013 | The specified server was started as InProc server. This has not been released and may possibly lead to incorrect behavior because the server is running in the same process area as the WinCC flexible Runtime software. | Configure the server as OutProc Server or Local Server. |
| 160014 | Only one OPC server project can be started on a PC/MP. An alarm is output when an attempt is made to start a second project.<br>The second project has no OPC server functionality and cannot be located as an OPC server by external sources. | Do not start a second project with OPC server functionality on the computer. |
| 170000 | S7 diagnostics events are not indicated because it is not possible to log in to the S7 diagnostics functions at this device. The service is not supported. | — |
| 170001 | The S7 diagnostics buffer cannot be viewed because communication with the controller is shut down. | Set the controller to online mode. |
| 170002 | The S7 diagnostics buffer cannot be viewed because reading of the diagnostics buffer (SSL) was canceled with error. | — |
| 170003 | An S7 diagnostics event cannot be visualized. The system returns internal error %2. | — |
| 170004 | An S7 diagnostics event cannot be visualized. The system returns an internal error of error class %2, error number %3. | — |
| 170007 | It is not possible to read the S7 diagnostics buffer (SSL) because this operation was canceled with an internal error of class %2 and error code %3. | — |
| 180000 | A component/OCX received configuration data with a version ID which is not supported. | Install a newer component. |
| 180001 | System overload because too many actions running in parallel. Not all the actions can be executed, some are rejected. | Several remedies are available:<br>• Increase the configured cycle times or basic clock.<br>• Generate the alarms at a slower rate (polling).<br>• Initiate scripts and functions at greater intervals.<br>If the alarm appears more frequently:<br>Restart the HMI device. |
| 180002 | The on-screen keyboard could not be activated. Possible causes:<br>"TouchInputPC.exe" was not registered due to a faulty Setup. | Reinstall WinCC flexible Runtime. |
| 190000 | It is possible that the tag is not updated. | — |

| Number | Effect/cause | Remedy |
|--------|-------------|--------|
| 190001 | The tag is updated after the cause of the last error state has been eliminated (return to normal operation). | — |
| 190002 | The tag is not updated because communication with the controller is down. | Select the system function "SetOnline" to go online. |
| 190004 | The tag is not updated because the configured tag address does not exist. | Check the configuration. |
| 190005 | The tag is not updated because the configured controller type does not exist for this tag. | Check the configuration. |
| 190006 | The tag is not updated because it is not possible to map the controller type in the data type of the tag. | Check the configuration. |
| 190007 | The tag value is not modified because the connection to the controller is interrupted or the tag is offline. | Set online mode or reconnect to the controller. |
| 190008 | The threshold values configured for the tag have been violated, for example, by<br>• a value entered<br>• a system function<br>• a script | Observe the configured or current threshold values of the tag. |
| 190009 | An attempt has been made to assign the tag a value which is outside the permitted range of values for this data type.<br>For example, a value of 260 was entered for a byte tag or a value of -3 for an unsigned word tag. | Observe the range of values for the data type of the tags. |
| 190010 | Too many values are written to the tag (for example, in a loop triggered by a script).<br>Values are lost because only up to 100 actions are saved to the buffer. | Increase the time interval between multiple write actions. |
| 190011 | Possible cause 1:<br>The value entered could not be written to the configured controller tag because the high or low limit was exceeded.<br>The system discards the entry and restores the original value.<br>Possible cause 2:<br>The connection to the controller was interrupted. | Make sure that the value entered lies within the range of values of the control tags.<br><br><br><br>Check the connection to the PLC. |
| 190012 | It is not possible to convert a value from a source format to a target format, for example:<br>An attempt is being made to assign a value to a counter that is outside the valid, PLC-specific value range.<br>A tag of the type Integer should be assigned a value of the type String. | Check the range of values or the data type of the tags. |

| Number | Effect/cause | Remedy |
|--------|-------------|--------|
| 190100 | The area pointer is not updated because the address configured for this pointer does not exist. Type<br>1 Warning alarms<br>2 Error alarms<br>3 Controller acknowledgment<br>4 HMI device acknowledgment<br>5 LED mapping<br>6 Trend request<br>7 Trend transfer 1<br>8 Trend transfer 2<br>No.:<br>consecutive number shown in WinCC flexible ES. | Check the configuration. |
| 190101 | The area pointer is not updated because it is not possible to map the PLC type to the area pointer type.<br>Parameter type and no.:<br>see alarm 190100 | — |
| 190102 | The area pointer is updated after the cause of the last error state has been eliminated (return to normal operation). Parameter type and no.: See alarm 190100. | — |
| 200000 | Coordination is not executed because the address configured in the controller does not exist/is not set. | Change the address or set up the address in the controller. |
| 200001 | Coordination is canceled because the write access to the address configured in the PLC is not possible. | Change the address or set the address in the controller at an area which allows write access. |
| 200002 | Coordination is not carried out at the moment because the address format of the area pointer does not match the internal storage format. | Internal error |
| 200003 | Coordination can be executed again because the last error state is eliminated (return to normal operation). | — |
| 200004 | The coordination may not be executed. | — |
| 200005 | No more data is read or written. Possible causes:<br>• The cable is defective.<br>• The PLC does not respond, is defective, etc.<br>• System overload. | Ensure that the cable is plugged in and the controller is operational.<br>Restart the system if the system alarm persists. |
| 200100 | Coordination is not executed because the address configured in the controller does not exist/is not set. | Change the address or set up the address in the controller. |
| 200101 | Coordination is canceled because the write access to the address configured in the PLC is not possible. | Change the address or set the address in the controller at an area which allows write access. |
| 200102 | Coordination is not carried out at the moment because the address format of the area pointer does not match the internal storage format. | Internal error |
| 200103 | Coordination can be executed again because the last error state is eliminated (return to normal operation). | — |
| 200104 | The coordination may not be executed. | — |

| Number | Effect/cause | Remedy |
|---|---|---|
| 200105 | No more data is read or written. Possible causes:<br>• The cable is defective.<br>• The PLC does not respond, is defective, etc.<br>• System overload. | Ensure that the cable is plugged in and the controller is operational.<br>Restart the system if the system alarm persists. |
| 210000 | Jobs are not processed because the address configured in the controller does not exist/has not been set up. | Change the address or set up the address in the controller. |
| 210001 | Jobs are not processed because read/write access to the address configured in the controller is not possible. | Change the address or set up the address in the controller in an area which allows read/write access. |
| 210002 | Jobs are not executed because the address format of the area pointer does not match the internal storage format. | Internal error |
| 210003 | The job buffer is processed again because the last error status has been eliminated (return to normal operation). | — |
| 210004 | It is possible that the job buffer will not be processed. | — |
| 210005 | A control request with an illegal number was initiated. | Check the controller program. |
| 210006 | An error occurred while attempting to execute the control request. As a result, the control request is not executed. Observe the next/previous system alarms. | Check the parameters of the control request. Recompile the configuration. |
| 220001 | The tag is not downloaded because the associated communication driver / HMI device does not support the download of Boolean/discrete data types. | Change the configuration. |
| 220002 | The tag is not downloaded because the associated communication driver / HMI device does not support write access to the data type BYTE. | Change the configuration. |
| 220003 | The communication driver cannot be loaded. The driver may not be installed. | Install the driver by reinstalling WinCC flexible Runtime. |
| 220004 | Communication is down and no update data is transferred because the cable is not connected or defective etc. | Check the connection. |
| 220005 | Communication is up. | — |
| 220006 | The connection between the specified PLC and the specified port is active. | — |
| 220007 | The connection to the specified controller is interrupted at the specified port. | Check if<br>• the cable is plugged in<br>• the controller is OK<br>• the correct port is used<br>• the configuration is OK (interface parameters, protocol settings, PLC address).<br>Restart the system if the system alarm persists. |

| Number | Effect/cause | Remedy |
|--------|--------------|--------|
| 220008 | The communication driver cannot access or open the specified port. The port may be in use by another application or the port used is not available on the destination device.<br>There is no communication with the controller. | Close all the applications which access this port and restart the computer.<br>Use another port of the system. |
| 230000 | The value entered could not be accepted. The system discards the entry and restores the previous value.<br>Either<br>• the value range has been exceeded<br>• illegal characters have been entered<br>• the maximum permitted number of users has been exceeded | Enter a practical value or delete any unneeded users. |
| 230002 | The currently logged in user is not granted write access rights. The system therefore discards the input and restored the previous value. | Log on as a user with appropriate rights. |
| 230003 | Changeover to the specified screen failed because the screen is not available/configured. The current screen remains selected. | Configure the screen and check the screen selection function. |
| 230005 | The value range of the tag has been exceeded in the IO field.<br>The original value of the tag is retained. | Observe the range of values for the tag when entering a value. |
| 230100 | During navigation in the web browser, the system returned a message which may be of interest to the user.<br>The web browser continues to run but may not (fully) show the new page. | Navigate to another page. |
| 230200 | The connection to the HTTP channel was interrupted due to an error. This error is explained in detail by another system alarm.<br>Data is no longer exchanged. | Check the network connection.<br>Check the server configuration. |
| 230201 | The connection to HTTP channel was established. Data is exchanged. | — |

| Number | Effect/cause | Remedy |
|--------|--------------|--------|
| 230202 | WININET.DLL has detected an error. This error usually occurs when an attempt to connect to the server fails or the server refuses to connect because the client lacks the proper authorization. An unknown server certificate may also be the cause if the connection is encrypted by means of SSL.<br>The alarm text provides details.<br>This text is always in the language of the Windows installation because it is returned by the Windows OS.<br>Process values are no longer exchanged. | Depending on the cause:<br>When an attempt to connect fails or a timeout error occurs:<br>• Check the network connection and the network.<br>• Check the server address.<br>• Check if the web server is actually running on the destination computer.<br>Faulty authorization:<br>• The configured user name and/or password do not match those on the server. Establish consistency.<br>When the server certificate is rejected:<br>Certificate signed by an unknown CA ( ):<br>• Either ignore this item in your project, or<br>• Install a certificate that has been signed with a root certificate known to the client computer.<br>The date of the certificate is invalid:<br>• Either ignore this item in your project, or<br>• Install a certificate with a valid date on the server.<br>Invalid CN (Common Name or Computer Name):<br>• Either ignore this item in your project, or<br>• Install a certificate with a name that corresponds to that of the server address. |
| 230203 | Although a connection can be made to the server, the HTTP server refuses to connect because<br>• WinCC flexible Runtime is not running on the server, or<br>• the HTTP channel is not supported (503 Service unavailable).<br>Other errors can only occur if the web server does not support the HTTP channel. The language of the alarm text depends on the web server.<br>Data is not exchanged. | Error 503 Service unavailable: Check if<br>WinCC flexible Runtime is running on the server<br>the HTTP channel is supported. |
| 230301 | An internal error has occurred. An English text explains the error in more detail. This may be caused by insufficient memory.<br>OCX does not work. | — |
| 230302 | The name of the remote server cannot be resolved.<br>The attempt to connect failed. | Check the configured server address.<br>Check if the DNS service is available on the network. |
| 230303 | The remote server is not running on the addressed computer.<br>Wrong server address.<br>The attempt to connect failed | Check the configured server address.<br>Check if the remote server is running on the target computer. |
| 230304 | The remote server on the addressed computer is incompatible to VNCOCX.<br>The attempt to connect failed. | Use a compatible remote server. |
| 230305 | The authentication has failed because the password is incorrect.<br>The attempt to connect failed. | Configure the correct password. |

| Number | Effect/cause | Remedy |
|--------|-------------|--------|
| 230306 | Error in the connection to the remote server. This may occur as a result of network problems. The attempt to connect failed. | Check if<br>• the bus cable is plugged in<br>• there are network problems |
| 230307 | The connection to the remote server was shut down because<br>• the remote server was shut down, or<br>• the user instructed the server to close all connections.<br>The connection is closed. | — |
| 230308 | This alarm provides information on the connection status.<br>An attempt is made to connect. | — |
| 240000 | WinCC flexible Runtime is operating in demo mode. You have no authorization or your authorization is corrupted. | Install the authorization. |
| 240001 | WinCC flexible Runtime is operating in demo mode. Too many tags are configured for the installed version. | Load an adequate authorization / powerpack. |
| 240002 | WinCC flexible Runtime is operating with a time-limited emergency authorization. | Restore the full authorization. |
| 240003 | Authorization failed.<br>Without authorization, WinCC will run in demo mode. | Restart WinCC flexible Runtime or reinstall it. |
| 240004 | Error while reading the emergency authorization. WinCC flexible Runtime is operating in demo mode. | Restart WinCC flexible Runtime, install the authorization or repair the authorization (see Commissioning Instructions Software Protection). |
| 240005 | The Automation License Manager has detected an internal system fault.<br>Possible causes:<br>• A corrupt file<br>• A defective installation<br>• No free space for the Automation License Manager etc. | Reboot the HMI device or PC. If this does not solve the problem, remove the Automation License Manager and install it again. |
| 250000 | The tag in the specified line in "Status force" is not updated because the address configured for this tag is not available. | Check the set address and then verify that the address is set up in the controller. |
| 250001 | The tag in the specified line in "Status force" is not updated because the controller type configured for this tag does not exist. | Check the set address. |
| 250002 | The tag in the specified line in "Status force" is not updated because it is not possible to map the controller type in the tag type. | Check the set address. |
| 250003 | An attempt to connect to the PLC failed. The tags are not updated. | Check the connection to the PLC. Check that the controller is switched on and is online. |
| 260000 | An unknown user or an unknown password has been entered in the system.<br>The current user is logged off from the system. | Log on to the system as a user with a valid password. |
| 260001 | The logged in user does not have sufficient authorization to execute the protected functions on the system. | Log on to the system as a user with sufficient authorization. |

| Number | Effect/cause | Remedy |
|--------|--------------|--------|
| 260002 | This alarm is triggered by the system function "TrackUserChange". | — |
| 260003 | The user has logged off from the system. | — |
| 260004 | The user name entered into the user view already exists in the user administration. | Select another user name because user names have to be unique in the user administration. |
| 260005 | The entry is discarded. | Use a shorter user name. |
| 260006 | The entry is discarded. | Use a shorter or longer password. |
| 260007 | The login timeout value entered is outside the valid range of 0 to 60 minutes.<br>The new value is discarded and the original value is retained. | Enter a login timeout value between 0 and 60 minutes. |
| 260008 | An attempt was made to read a PTProRun.pwl file created with ProTool V 6.0 in WinCC flexible.<br>Reading the file was canceled due to incompatibility of the format. | — |
| 260009 | You have attempted to delete the user "Admin" or "PLC User". These users are fixed components of the user management and cannot be deleted. | If you need to delete a user, because perhaps you have exceeded the maximum number permitted, delete another user. |
| 260012 | The passwords entered in the "Change Password" dialog and the confirmation field are not identical.<br>The password has not been changed. User will be logged off. | You have to log on to the system again. Then enter the identical password twice to be able to change the password. |
| 260013 | The password entered in the "Change Password" dialog is invalid because it is already in use.<br>The password has not been changed. User will be logged off. | You have to log on to the system again. Then enter a new password that has not been used before. |
| 260014 | User has unsuccessfully attempted to log on 3 times in succession.<br>The user has been locked out and assigned to group no. 0. | You can log on to the system with your correct password. Only an administrator can change the assignment to a group. |
| 270000 | A tag is not indicated in the alarm because it attempts to access an invalid address in the controller. | Check if the data area for the tag exists in the controller, the configured address is correct and the value range for the tag is correct. |
| 270001 | There is a device-specific limit as to how many alarms may be queued for output (see the operating instructions). This limit has been exceeded.<br>The view no longer contains all the alarms.<br>However, all alarms are written to the alarm buffer. | — |
| 270002 | The view shows alarms of a log for which there is no data in the current project.<br>Wildcards are output for the alarms. | Delete older log data if necessary. |
| 270003 | The service cannot be set up because too many devices want to use this service.<br>A maximum of four devices may execute this action. | Reduce the number of HMI devices which want to use the service. |
| 270004 | Access to persistent buffer is not possible. Alarms cannot be restored or saved. | If the problems persist at the next startup, contact Customer Support (delete Flash). |
| 270005 | Persistent buffer damaged: Alarms cannot be restored. | If the problems persist at the next startup, contact Customer Support (delete Flash). |

| Number | Effect/cause | Remedy |
|--------|-------------|--------|
| 270006 | Project modified: Alarms cannot be restored from the persistent buffer. | The project was generated and transferred new to the HMI device; The error should no longer occur when the device starts again. |
| 270007 | A configuration problem is preventing the restore (a DLL is missing, a directory is unknown, etc.). | Update the operating system and then transfer your project again to the HMI device. |
| 280000 | Connection is up because the cause of the interruption has been eliminated. | — |
| 280001 | No more data is read or written. Possible causes:<br>• The cable is defective.<br>• The PLC does not respond, is defective, etc.<br>• The wrong port is used for the connection.<br>• System overload. | Check if<br>• the cable is plugged in<br>• the controller is OK<br>• the correct port is used<br>Restart the system if the system alarm persists. |
| 280002 | The connection used requires a function block in the controller.<br>The function block has responded. Communication is now enabled. | — |
| 280003 | The connection used requires a function block in the controller.<br>The function block has not responded. | Check if<br>• the cable is plugged in<br>• the controller is OK<br>• the correct port is used<br>Restart the system if the system alarm persists.<br>Remedy depends on the error code:<br>1: The function block must set the COM bit in the response container.<br>2: The function block must not set the ERROR bit in the response container.<br>3: The function block must respond within the specified time (timeout).<br>4: Go online to the PLC. |
| 280004 | The connection to the controller is interrupted. There is no data exchange at present. | Check the connection parameters in WinCC flexible. Ensure that the cable is plugged in, the controller is operational, the correct interface is being used.<br>Restart the system if the system alarm persists. |
| 290000 | The recipe tag could not be read or written. It is assigned the start value.<br>The alarm can be entered in the alarm buffer for up to four more failed tags if necessary. After that, alarm 290003 is output. | Check in the configuration that the address has been set up in the controller. |
| 290001 | An attempt has been made to assign a value to a recipe tag which is outside the value range permitted for this type.<br>The alarm can be entered in the alarm buffer for up to four more failed tags if necessary. After that, alarm 290004 is output. | Observe the value range for the tag type. |
| 290002 | It is not possible to convert a value from a source format to a target format.<br>The alarm can be entered in the alarm buffer for up to four more failed recipe tags if necessary. After that, alarm 290005 is output. | Check the value range or type of the tag. |

| Number | Effect/cause | Remedy |
|--------|-------------|--------|
| 290003 | This alarm is output when alarm number 290000 is triggered more than five times.<br>In this case, no further separate alarms are generated. | Check in the configuration that the tag addresses have been set up in the controller. |
| 290004 | This alarm is output when alarm number 290001 is triggered more than five times.<br>In this case, no further separate alarms are generated. | Observe the value range for the tag type. |
| 290005 | This alarm is output when alarm number 290002 is triggered more than five times.<br>In this case, no further separate alarms are generated. | Check the value range or type of the tag. |
| 290006 | The threshold values configured for the tag have been violated by values entered. | Observe the configured or current threshold values of the tag. |
| 290007 | There is a difference between the source and target structure of the recipe currently being processed. The target structure contains an additional data recipe tag which is not available in the source structure.<br>The data recipe tag specified is assigned its start value. | Insert the specified data recipe tag in the source structure. |
| 290008 | There is a difference between the source and target structure of the recipe currently being processed. The source structure contains an additional data recipe tag which is not available in the target structure and therefore cannot be assigned.<br>The value is rejected. | Remove the specified data recipe tag in the specified recipe from the project. |
| 290010 | The storage location configured for the recipe is not permitted.<br>Possible causes:<br>Illegal characters, write protection, data carrier out of space or does not exist. | Check the configured storage location. |
| 290011 | The data record with the specified number does not exist. | Check the source for the number (constant or tag value). |
| 290012 | The recipe with the specified number does not exist. | Check the source for the number (constant or tag value). |
| 290013 | An attempt was made to save a data record under a data record number which already exists.<br>The action is not executed. | The following remedies are available:<br>• Check the source for the number (constant or tag value).<br>• First, delete the data record.<br>• Change the "Overwrite" function parameter. |
| 290014 | The file specified to be imported could not be found. | Check:<br>• The file name<br>• Ensure that the file is in the specified directory. |
| 290020 | Message reporting that the download of data records from the HMI device to the PLC has started. | — |
| 290021 | Message reporting that the download of data records from the HMI device to the PLC was completed. | — |

| Number | Effect/cause | Remedy |
|--------|-------------|--------|
| 290022 | Message reporting that the download of data records from the HMI device to the PLC was canceled due to an error. | Check in the configuration whether:<br>• The tag addresses are configured in the controller<br>• The recipe number exists<br>• The data record number exist<br>• The "Overwrite" function parameter is set |
| 290023 | Message reporting that the download of data records from the PLC to the HMI device has started. | — |
| 290024 | Message reporting that the download of data records from the PLC to the HMI device was completed. | — |
| 290025 | Message reporting that the download of data records from the PLC to the HMI device was canceled due to an error. | Check in the configuration whether:<br>• The tag addresses are configured in the controller<br>• The recipe number exists<br>• The data record number exist<br>• The "Overwrite" function parameter is set |
| 290026 | An attempt has been made to read/write a data record although the data record is not free at present.<br>This error may occur in the case of recipes for which downloading with synchronization has been configured. | Set the data record status to zero. |
| 290027 | Unable to connect to the controller at present. As a result, the data record can neither be read nor written.<br>Possible causes:<br>No physical connection to the controller (no cable plugged in, cable is defect) or the controller is switched off. | Check the connection to the PLC. |
| 290030 | This alarm is output after you selected screen which contains a recipe view in which a data record is already selected. | Reload the data record from the storage location or retain the current values. |
| 290031 | While saving, it was detected that a data record with the specified number already exists. | Overwrite the data record or cancel the action. |
| 290032 | While exporting data records it was detected that a file with the specified name already exists. | Overwrite the file or cancel the process. |
| 290033 | Confirmation request before deleting data records. | — |
| 290040 | A data record error with error code %1 that cannot be described in more detail occurred.<br>The action is canceled.<br>It is possible that the data record was not installed correctly on the controller. | Check the storage location, the data record, the "Data record" area pointer and if necessary, the connection to the controller.<br>Restart the action after a short time.<br>If the error persists, contact Customer Support. Forward the relevant error code to Customer Support. |
| 290041 | A data record or file cannot be saved because the storage location is full. | Delete files no longer required. |
| 290042 | An attempt was made to execute several recipe actions simultaneously. The last action was not executed. | Trigger the action again after waiting a short period. |
| 290043 | Confirmation request before storing data records. | — |

| Number | Effect/cause | Remedy |
|--------|--------------|--------|
| 290044 | The data store for the recipe has been destroyed and is deleted. | — |
| 290050 | Message reporting that the export of data records has started. | — |
| 290051 | Message reporting that the export of data records was completed. | — |
| 290052 | Message reporting that the export of data records was canceled due to an error. | Ensure that the structure of the data records at the storage location and the current recipe structure on the HMI device are identical. |
| 290053 | Message reporting that the import of data records has started. | — |
| 290054 | Message reporting that the import of data records was completed. | — |
| 290055 | Message reporting that the import of data records was canceled due to an error. | Ensure that the structure of the data records at the storage location and the current recipe structure on the HMI device are identical. |
| 290056 | Error when reading/writing the value in the specified line/column.<br>The action was canceled. | Check the specified line/column. |
| 290057 | The tags of the recipe specified were toggled from "offline" to "online" mode.<br>Each change of a tag in this recipe is now immediately downloaded to the controller. | — |
| 290058 | The tags of the specified recipe were toggled from "offline" to "online" mode.<br>Modifications to tags in this recipe are no longer immediately transferred to the controller but must be transferred there explicitly by downloading a data record. | — |
| 290059 | Message reporting that the specified data record was saved. | — |
| 290060 | Message reporting that the specified data record memory was cleared. | — |
| 290061 | Message reporting that clearing of data record memory was canceled due to an error. | — |
| 290062 | The data record number is above the maximum of 65536.<br>This data record cannot be created. | Select another number. |
| 290063 | This occurs with the system function "ExportDataRecords" when the parameter "Overwrite" is set to No.<br>An attempt has been made to save a recipe under a file name which already exists.<br>The export is canceled. | Check the "ExportDataRecords" system function. |
| 290064 | Message reporting that the deletion of data records has started. | — |
| 290065 | Message reporting that the deletion of data records has successfully completed. | — |
| 290066 | Confirmation request before deleting data records. | — |

| Number | Effect/cause | Remedy |
|--------|-------------|--------|
| 290068 | Security request to confirm if all data records in the recipe should be deleted. | — |
| 290069 | Security request to confirm if all data records in the recipe should be deleted. | — |
| 290070 | The data record specified is not in the import file. | Check the source of the data record number or data record name (constant or tag value). |
| 290071 | During the editing of data record values, a value was entered which exceeded the low limit of the recipe tag.<br>The entry is discarded. | Enter a value within the limits of the recipe tag. |
| 290072 | When editing data record values, a value was entered which exceeds the high limit of the recipe tag.<br>The entry is discarded. | Enter a value within the limits of the recipe tag. |
| 290073 | An action (e.g. saving a data record) failed due to an unknown error.<br>The error corresponds to the status alarm IDS_OUT_CMD_EXE_ERR in the large recipe view. | — |
| 290074 | While saving, it was detected that a data record with the specified number already exists but under another name. | Overwrite the data record, change the data record number or cancel the action. |
| 290075 | A data record with this name already exists.<br>The data record is not saved. | Please select a different data record name. |
| 300000 | Faulty configuration of process monitoring (e.g. using PDiag or S7-Graph): More alarms are queued than specified in the specifications of the CPU. No further ALARM_S alarms can be managed by the PLC and reported to the HMI devices. | Change the controller configuration. |
| 300001 | ALARM_S is not registered on this controller. | Select a controller that supports the ALARM_S service. |
| 310000 | An attempt is being made to print too many reports in parallel.<br>Only one log file can be output to the printer at a given time; the print job is therefore rejected. | Wait until the previous active log was printed.<br>Repeat the print job if necessary. |
| 310001 | An error occurred on triggering the printer. The report is either not printed or printed with errors. | Evaluate the additional system alarms related to this alarm.<br>Repeat the print job if necessary. |
| 320000 | The movements have already been indicated by another device.<br>The movements can no longer be controlled. | Deselect the movements on the other display units and select the motion control screen on the required display unit. |
| 320001 | The network is too complex.<br>The faulty addresses cannot be indicated. | View the network in STL. |
| 320002 | No diagnostics alarm selected.<br>The unit belonging to the alarm could not be selected. | Select a diagnostics alarm from the ZP_ALARM alarm screen. |
| 320003 | No alarms exists for the selected unit. The detail view cannot visualize any networks. | Select the defective unit from the overview screen. |
| 320004 | The required signal states could not be read by the PLC. The faulty addresses cannot be found. | Check the consistency between the configuration on the display unit and the PLC program. |

| Number | Effect/cause | Remedy |
|---|---|---|
| 320005 | The project contains ProAgent elements which are not installed. ProAgent diagnostic functions cannot be performed. | In order to run the project, install the optional ProAgent package. |
| 320006 | You have attempted to execute a function which is not supported in the current constellation. | Check the type of the selected unit. |
| 320007 | No error-triggering addresses were found on the networks.<br>ProAgent cannot indicate any faulty addresses. | Switch the detail screen to STL layout mode and check the status of the addresses and exclusion addresses. |
| 320008 | The diagnostic data stored in the configuration are not synchronized with those in the PLC.<br>ProAgent can only indicate the diagnostic units. | Download the project to the HMI device again. |
| 320009 | The diagnostic data stored in the configuration are not synchronized with those in the PLC. The diagnostic screens can be operated as usual.<br>ProAgent may be unable to show all diagnostic texts. | Download the project to the HMI device again. |
| 320010 | The diagnostic data stored in the configuration are not synchronized with those in STEP7.<br>The ProAgent diagnostics data is not up-to-date. | Download the project to the HMI device again. |
| 320011 | A unit with the corresponding DB number and FB number does not exist.<br>The function cannot be executed. | Check the parameters of the "SelectUnit" function and the units selected in the project. |
| 320012 | The "Step sequence mode" dialog is no longer supported. | Use the ZP_STEP step sequence screen from the corresponding standard project for your project. Instead of calling the Overview_Step_Sequence_Mode function, call the "FixedScreenSelection" function using ZP_STEP as the screen name. |
| 320014 | The selected controller cannot be evaluated for ProAgent.<br>The alarm view assigned to the "EvaluateAlarmDisplayFault" system function could not be found. | Check the parameters of the "EvaluateAlarmDisplayFault" system function. |
| 330022 | Too many dialogs are open on the HMI device. | Close all dialogs you do not require on the HMI device. |

## 9.2 Abbreviations

### Abbreviations

The abbreviations and acronyms used in the manual have the following meaning:

| | |
|---|---|
| AL | Alarm |
| ANSI | American National Standards Institute |
| AS 511 | Programming device interface to SIMATIC S5 protocol |
| ASCII | American Standard Code for Information Interchange |
| AT | Accept button |
| CCFL | Cold Cathode Fluorescence Lamp |
| CF | Compact Flash |
| CPU | Central Processing Unit |
| CS | Configuration System |
| CSA | Customer Specified Articles |
| CSV | Comma Separated Values |
| CTS | Clear To Send |
| DC | Direct Current |
| DCD | Data Carrier Detect |
| DI | Digital Input |
| DIP | Dual In–Line Package |
| DP | Distributed I/O |
| DRAM | Dynamic Random Access Memory |
| DSN | Data Source Name |
| DSR | Data Set Ready |
| DTR | Data Terminal Ready |
| EGB | Electrostatic Discharge, the components and modules endangered by such |
| EMC | Electromagnetic Compatibility |
| EN | European standard |
| ESD | Electrostatic Sensitive Device |
| GND | Ground |
| HF | High Frequency |
| HMI | Human Machine Interface |
| IEC | International Electronic Commission |
| IF | Interface |
| LCD | Liquid Crystal Display |
| LED | Light Emitting Diode |
| MOS | Metal Oxide Semiconductor |
| MP | Multi Panels |
| MPI | Multipoint Interface (SIMATIC S7) |
| MTBF | Mean Time Between Failures |
| OP | Operator Panel |
| PC | Personal Computer |

| | |
|---|---|
| PCL | Printer Control Language |
| PG | Programming device |
| PLC | AG |
| PLC | Programmable logic controller |
| PPI | Point to Point Interface (SIMATIC S7) |
| RAM | Random Access Memory |
| RTS | Request To Send |
| RxD | Receive Data |
| SELV | Safety Extra Low Voltage |
| SM | Events |
| SP | Service pack |
| SRAM | Static Random Access Memory |
| STN | Super Twisted Nematic |
| Sub-D | Subminiature D (plug) |
| TCP/IP | Transmission Control Protocol/Internet Protocol |
| TFT | Thin Film Transistor |
| TP | Touch Panel |
| TTL | Transistor Transistor Logic |
| TxD | Transmit Data |
| UL | Underwriter's Laboratory |
| UPS | Uninterruptible Power Supply |
| UPS | Uninterruptible power supply |
| VGA | Video Graphic Array |

# 9.3 Glossary

### Alarm

Indicates particularly serious operating states that must be acknowledged.

### Alarm cleared (also known as outgoing)

Time at which an alarm is cleared by the controller.

### Alarm generated (also known as incoming)

Time at which an alarm is triggered by the controller or the HMI device.

### Alarm level

Operational level of the HMI device in which the alarms are displayed.

## Alarm reporting

Printout of alarms and system messages at the same time as the screen display.

## Area pointer

Required for data exchange between HMI device and controller. It contains information on the location of data areas on the controller.

## Combo box

Box for setting a parameter (one value can be selected from those displayed).

## Configuration computer

Generic term for programming devices and PCs on which configurations are created.

## Configuring

Specifying basic settings, alarms, and screens for a specific plant using the ProTool configuration software.

## Control request

Triggering of a function by the controller.

## Display duration

Time between an alarm being generated and being cleared.

## Display function

Function that causes a change in the displayed information, for example alarm level display, alarm buffer display, screen display

## Events

Indicates certain statuses of the machine or plant connected to the controller.

## Fault time

Time between generation and clearance of an alarm.

## Field

Reserved area in configured or fixed texts for output and/or input of values.

## Flash memory

Programmable memory that can be erased quickly and then written to again.

## Forced printout

Automatic printout of the alarms and system messages deleted when there is a buffer overflow.

## Hardcopy

Printout of the content of the display on a printer.

## Infotext

Configurable additional information on alarms, screens, screen entries and boxes.

## Loop through mode

Mode on the HMI device Includes normal operation and also allows communication between the configuration computer and controller over the second interface of the HMI device. This mode is possible only when the connection to the controller uses the AS511 protocol.

## Normal operation

Mode on the HMI device in which alarms are displayed and input can made in screens.

## Output mode

Box for displaying a process value.

## Screen

Display of logically related process data that can be displayed together on the HMI device and that can be individually changed.

## Screen entry

Element of a screen - consists of the entry number, texts, and tags.

## Screen level

Processing level on the HMI device at which pictures can be viewed and manipulated.

## Self Test

Test of the status of the CPU and the memory each time the power is turned on.

## Softkey

Key with a variable assignment (dependent on the displayed screen entry)

## System alarms

Indicates an internal state on the HMI device and on the controller.

## System to be monitored

General term referring to machines, processing centers, systems, plants and processes which are operated and monitored on an HMI device.

## Transfer mode

Mode on the HMI device in which data is transferred from the configuration computer to the HMI device.

# Index

# X