

SIEMENS

SIMATIC HMI

WinCC flexible 2008 SP1/SP2 ADS protocol

Configuration Manual

Preface

Overview	1
Connect controller and establish connection	2
User data areas	3
Software commissioning	4
Enabling a CX1000 controller	5
Appendix	A
Abbreviations	B

Legal information

Warning notice system

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

⚠ DANGER
indicates that death or severe personal injury will result if proper precautions are not taken.
⚠ WARNING
indicates that death or severe personal injury may result if proper precautions are not taken.
⚠ CAUTION
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 product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation for the specific task, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

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

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.

Disclaimer of Liability

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

Preface

Purpose of this configuration manual

This configuration manual gives you an overview of the following topics:

- Communication between a controller from Beckhoff Automation GmbH and an HMI device from Siemens AG over an Ethernet connection.
 - Description of the software
 - Operation of the software
- Connectable Beckhoff controllers

This configuration manual is intended for:

- Project planning engineers
- Users

Pay particular attention to the section "Safety instructions (Page 9)".

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 required

General knowledge of automation technology and process communication is needed to understand this configuration manual.

It is also assumed that those using the operating instructions have experience with personal computers and knowledge of Microsoft operating systems.

You are familiar with software from Beckhoff GmbH, such as "TwinCAT controller Control".

Area of application

In conjunction with the WinCC flexible 2008 SP1 and WinCC flexible 2008 SP2 software packages, this configuration manual applies to the following controllers from Beckhoff Automation:

- CX1000
- BC9000
- Soft controller

Conventions

The following text notation will facilitate reading this manual:

Notation	Scope
"Add screen"	<ul style="list-style-type: none">• Terminology that appears in the user interface, for example dialog names, tabs, buttons, menu commands• Required input, for example, limits, tag values.• Path information
"File > Edit"	Operating sequences, for example, menu commands, shortcut menu commands
<F1>, <Alt+P>	Keyboard operation

Please observe notes labeled as follows:

Note

Notes contain important information concerning the product, its use or a specific section of the documentation to which you should pay particular attention.

Illustrations in this manual

This documentation includes illustrations associated with the product. These illustrations may differ from the factory state of the product.

Trademarks

The following designations marked with the symbol ® are registered trademarks of Siemens AG:

- HMI®
- SIMATIC®
- WinCC®

Technical Support

Technical support for the products covered in the manual is available in the Internet at:

- Technical Support (<http://support.automation.siemens.com>)
- Support Request (<http://support.automation.siemens.com/WW/view/en/16605654>)
- Service (<http://support.automation.siemens.com/WW/view/en/16604318>)
- Contacts and office locations (<http://www.automation.siemens.com/mcms/aspa-db/en/Pages/default.aspx>)
- Training center (<http://sitrain.automation.siemens.com/sitrain/default.aspx?AppLang=en>)

Additional information on SIMATIC products is available in the Internet at:

- Industry Portal (http://www.automation.siemens.com/_en/portal/index.htm)
- Overall SIMATIC documentation (http://www.automation.siemens.com/simatic/portal/html_76/techdoku.htm)

Table of contents

	Preface	3
1	Overview	7
1.1	Product overview	7
1.2	Communication partners	7
1.3	Communication between the HMI device and the PLC	8
2	Connect controller and establish connection	9
2.1	Safety instructions.....	9
2.2	Requirements of communication	9
2.3	Connecting PLCs and HMI devices	10
2.4	Establish communications	10
2.4.1	Overview	10
2.4.2	WinCC flexible project.....	11
2.4.2.1	Installing a communication driver.....	11
2.4.2.2	Selecting a protocol	11
2.4.2.3	Changing connection parameters	12
2.4.2.4	Initializing protocol parameters	12
2.4.2.5	Permissible data types.....	14
2.4.2.6	Optimizing a project	15
3	User data areas	17
3.1	Area pointer.....	17
3.1.1	General information on range pointers	17
3.1.2	"Screen number" range pointer.....	20
3.1.3	"Date/time" range pointer	21
3.1.4	"Date/time PLC" range pointer.....	22
3.1.5	"Coordination" range pointer.....	23
3.1.6	"Project ID" range pointer	24
3.1.7	"Job mailbox" range pointer	24
3.1.8	"Data record" area pointer	28
3.1.8.1	General information	28
3.1.8.2	Transferring without synchronization	29
3.1.8.3	Causes of error when transferring data records	30
3.1.8.4	Transferring when triggering by means of a configured function.....	30
3.1.8.5	Transferring with a PLC job	32
3.1.8.6	Transfer by the operator in the recipe view	34
3.1.8.7	Transferring with synchronization	35
3.2	Status alarms, error alarm, and acknowledgments	36
3.2.1	Overview	36
3.2.2	Creating a tag or an array	37
3.2.3	Configuring an alarm.....	38
3.2.4	Configuring the acknowledgment.....	40
3.3	Trend request and trend transfer	42
3.4	LED mapping	44

4	Software commissioning	45
4.1	Overview	45
4.2	Transferring the PLC program to the PLC	45
4.3	Transferring the project to the HMI device	46
5	Enabling a CX1000 controller	47
A	Appendix	51
A.1	ESD guideline	51
B	Abbreviations	53
	Glossary	54
	Index	55

Overview

1.1 Product overview

The "ADS" protocol enables communication between a SIMATIC HMI device and a controller from Beckhoff GmbH. Communication uses a cable protocol.

The following tasks are described in this configuration manual:

- Creating a WinCC flexible project for a SIMATIC HMI device
The WinCC flexible project contains the "ADS" driver for a controller from Beckhoff GmbH.
- Assigning parameters for the Beckhoff controller

Note

If you connect a SIMATIC HMI device to a Beckhoff controller and want to enable communication, you will need to follow the steps described in this configuration manual.

You enable the SIMATIC HMI device on the Beckhoff controller.

The SIMATIC HMI device is enabled on the Beckhoff CX1000 controller using the CERHOST.EXE program – see Section Enabling a CX1000 controller (Page 47).

For detailed information about Beckhoff hardware and software, refer to the corresponding documentation.

1.2 Communication partners

Introduction

This section describes communication between the HMI device and the controller. The controller from Beckhoff GmbH communicates with the HMI device using an ADS protocol.

Connectable Beckhoff controllers

The following Beckhoff controllers can be connected:

- BC9000
- CX1000
- TwinCAT Soft PLC, V2.10.0 and later

The connection is established using the ADS protocol.

Connectable devices

The following Siemens HMI devices can be connected to the indicated controllers:

Operating system	Stationary HMI device
Windows CE	<ul style="list-style-type: none"> • TP 177B 4" • TP 177 6" color PN/DP • MP 277 10" Touch and Key • MP 377 12" Touch and Key

Released communication types

The following physical connection types using the ADS protocol have been tested and released for Beckhoff controllers:

- Point-to-point link
From a SIMATIC HMI device to a Beckhoff controller using the ADS protocol.
- Multipoint link
From a SIMATIC HMI device with up to four Beckhoff controllers using the ADS protocol.

1.3 Communication between the HMI device and the PLC

The HMI device and the Beckhoff controller communicate using:

- Tags
- User data areas

Tags

The PLCs and the HMI device exchange data using tags. You must therefore create tags in your configuration that point to a specific data area on the controller. The HMI device reads the value from the specified data area and then displays it.

The operator can also make an input on the HMI device that is then written to the data area on the controller.

User data areas

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

For example, user data areas are required for:

- Job mailboxes
- Transferring data records
- Date/time synchronization
- Life-bit monitoring

During configuration in WinCC flexible, user data areas must be created, and the corresponding addresses must be assigned on the controller.

Connect controller and establish connection

2

2.1 Safety instructions

For all activities during configuration and use, pay proper attention to the safety instructions provided in the documentation supplied with the hardware and software in the product overview.

Note the following safety information:

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

2.2 Requirements of communication

Hardware requirements

The following are required for the connection:

- A Beckhoff controller
- A SIMATIC HMI device approved for communication via Ethernet

Software requirements

You require the WinCC flexible 2008 SP1 or SP2 Engineering System for the connection. You also require the following:

- A driver for the connection between the Beckhoff controller and SIMATIC HMI device
The driver for the ADS protocol is not supplied with WinCC flexible. This driver must be installed in a separate setup procedure.
- Matching interface parameters in WinCC flexible and in the controller.

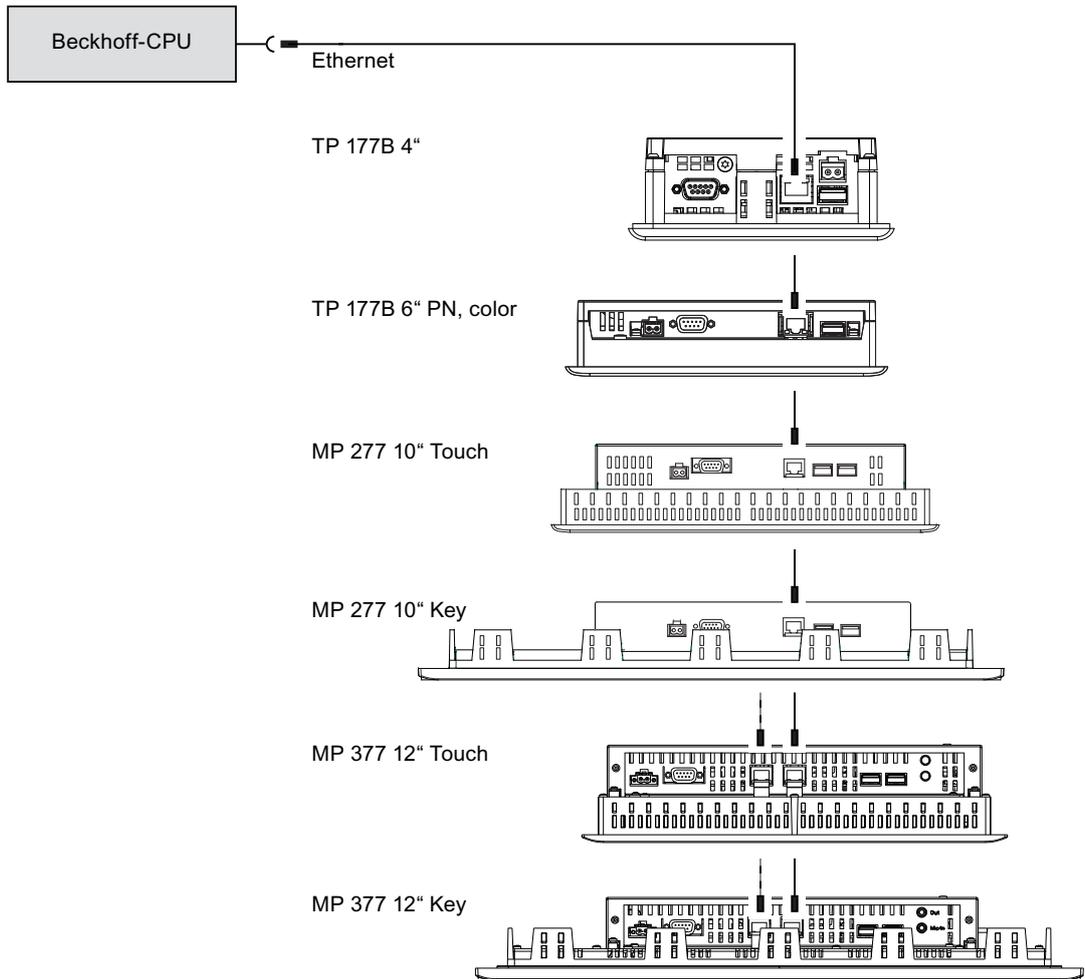
The connection does not require special blocks on the controller.

Connectable cables

Refer to the corresponding operating instructions to find out which interface must be used on the SIMATIC HMI device. The SIMATIC HMI device is connected directly to the Ethernet interface of the controller.

2.3 Connecting PLCs and HMI devices

The following configuration graphic shows which SIMATIC HMI devices you can connect to a Beckhoff controller and which interface you have to use.



2.4 Establish communications

2.4.1 Overview

Work through the following steps to establish an ADS connection to a SIMATIC HMI device:

- Set up a WinCC flexible project with an integrated ADS driver for a SIMATIC HMI device.
- Configure the Beckhoff controller

These work steps are described in the following sections.

2.4.2 WinCC flexible project

2.4.2.1 Installing a communication driver

The driver for the ADS protocol is not supplied with WinCC flexible. This driver must be installed using a setup procedure. The setup file is located in the "V1.3.0.0_0.01.0.1.zip" file or in a later version.

The file also contains this configuration manual.

2.4.2.2 Selecting a protocol

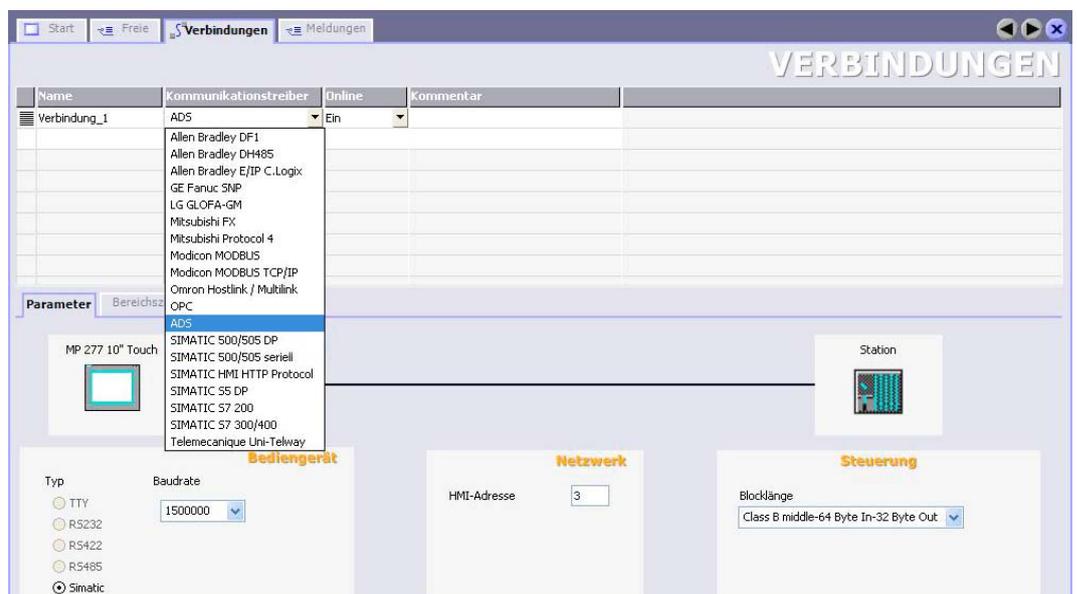
Requirement

When configuring in WinCC flexible, the "ADS" communication driver is required.

Procedure

Proceed as follows:

1. Double-click "Communication > Connections" in the project window.
2. Select the "ADS" driver in the "Communications driver" column.



After you have selected the driver, the content of the "HMI device" and "controller" changes.

2.4.2.3 Changing connection parameters

Procedure

Proceed as follows:

1. Double-click "Communication > Connections" in the project window.
2. Select the required connection.
3. Change the parameters as required in the Properties window.

2.4.2.4 Initializing protocol parameters

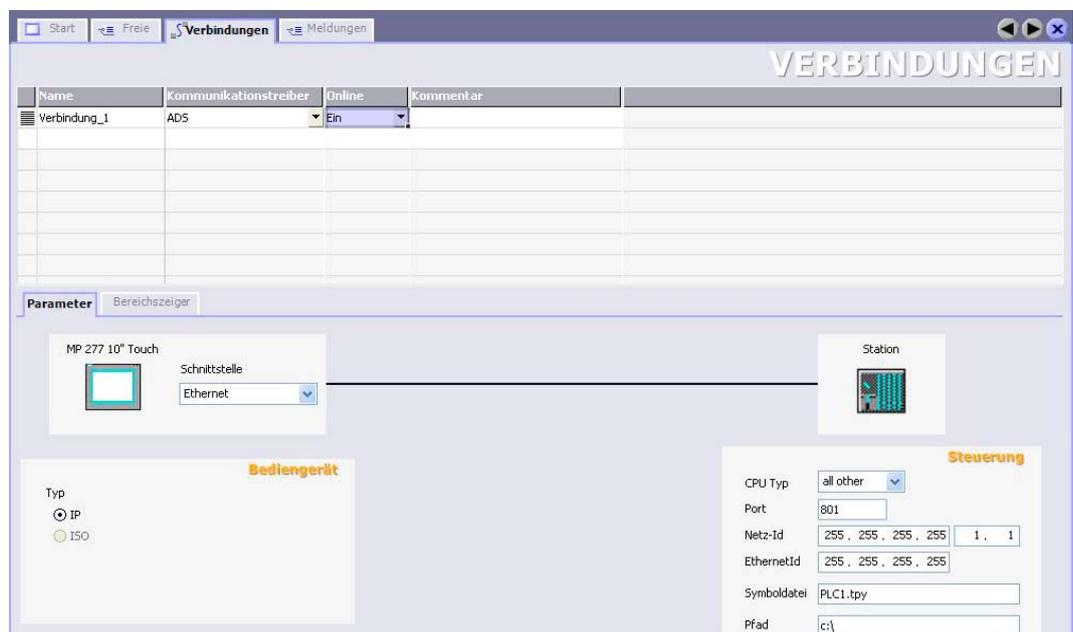
Introduction

You can enter or modify the protocol parameters in the "ADS" Properties window.

Procedure

Proceed as follows:

1. Double-click "Communication > Connections" in the project window.
2. Select the "Communication drivers" column and then select ADS.



3. Set the following device-dependent parameters:

– Interface

Select the "Ethernet" interface. For additional information, refer to the operating instructions for the HMI device.

– Type

Select the interface used on the HMI device. You can only select "IP".

– CPU type

Select "BC, BX". Or select "all other" for all other Beckhoff controllers.

– Port

Specify the port to be used for communication with the controller. The default is 801 – you need to use 800 for the "BC" and "BX" controllers.

– Network ID

Specify the network ID for the Beckhoff controller. This is the IP address of the target controller with the extension ".1.1". Access using a network name is not possible. Select "fixed IP address" on the Beckhoff controller. Do not use a dynamic address.

NOTICE**Assigning parameters for the specific controller**

If you do not make the parameter settings as explained below, you may experience functional disruptions.

On some controllers, for example, "TwinCAT SoftPLC" or "CX", the HMI device must be shared to allow access. This does not apply to the BC9000 controller.

- Applies to TwinCAT SoftPLC

Here, you use the "TwinCAT System Control" program from Beckhoff GmbH.

Select the menu command "TwinCAT System Properties > AMS Router > Remote Computer > Add". Under "AMSNetID", enter the IP address of the HMI device with the extension ".1.1" and under Address, the IP address of the HMI device without an extension. For "Transport", select "TCP/IP".

Restart the TwinCAT computer.

- Applies to the CX series

Here, you use the CERHOST program from Beckhoff GmbH. See section Enabling a CX1000 controller (Page 47) for a description of the program.

– Symbol file and path

If you want to access a tag using a name, you can specify the file name and path of the TPY file. This file is created by TwinCAT for every project and contains the tag names used in the project. This means that tag names can be selected in the "Symbol" column of the tag grid with the Browser when you configure the HMI device.

2.4.2.5 Permissible data types

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

Name	Area	Data type
Bit memory area	%M	Bool, SInt, USInt/Byte, Int, UInt/Word, DInt, UDInt/DWord, Real, String
Input area	%I	Bool, SInt, USInt/Byte, Int, UInt/Word, DInt, UDInt/DWord, Real, String
Output area	%Q	Bool, SInt, USInt/Byte, Int, UInt/Word, DInt, UDInt/DWord, Real, String
Beckhoff tag name	Named	Bool, SInt, USInt/Byte, Int, UInt/Word, DInt, UDInt/DWord, Real, String

Please observe the following:

- Area pointers can only be created in the areas "%M", "%I", and "%Q".
- Tags of the areas "%M", "%I", and "%Q" and of the data types "Int" or "UInt/Word" are permitted as trigger tags for discrete alarms.
- Array tags may only be used for discrete alarms and trends.

Therefore, array tags can only be created as tags of the areas "%M", "%I", and "%Q" and of the data types "Int" or "UInt/Word".

Special Configuration Features

When configuring tags, the configuration engineer must pay particular attention to the following points:

- Beckhoff tag name "Named"

On controllers of the "BC/BX" series (for example, BC9000), access in the ADS protocol is not possible via the "Named" area.

- The tag name of a Beckhoff controller can be entered directly in the address dialog if "Named" was selected as address.

After clicking in the "Symbol" column, a tag name from the TwinCAT program can be selected. This is only possible if the file name and the path of the Beckhoff symbol file are set correctly in "Connection". The appropriate data type must be selected.

<p>NOTICE</p> <p>Tags are interpreted as a byte addresses</p> <p>Addresses entered in the areas "%M", "%I", and "%Q" are interpreted as a byte addresses.</p> <p>If you are using the "Word addressing" setting on a Beckhoff controller, you must multiply the byte address by 2 when configuring tags for the HMI device. With double words, the byte address must be multiplied by 4.</p>
--

Example

With "Word addressing" on the controller, "%MB40" is configured on the SIMATIC HMI device for the address "%MW20". The double word "%MD20" becomes "%MB80".

2.4.2.6 Optimizing a project

Acquisition cycle and update time

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

The update time is the sum of:

- Acquisition cycle
- Transmission time
- Processing time

For optimal update times during configuration, note the following:

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

You can reduce the update time by setting up one large data area instead of several small areas.

- Acquisition cycles that are too short compromise the overall performance unnecessarily.

Set the acquisition cycle according to the rate of change of the process values. The rate of temperature change at a reactor, for example, is significantly slower compared to the speed rate of an electric drive. A time of approx. 1 second is a standard value for the acquisition cycle.

- Put the tags of an alarm or a screen in one data area without gaps.
- Changes in the PLC can only be detected reliably if these are available at least for the duration of the actual acquisition cycle.
- Set the transmission rate to the highest possible value.

Discrete alarms

Use arrays to handle discrete alarms. Assign each alarm to one bit of the array tag instead of assigning these to any subelements. Tags of data areas "%M", "%I", and "%Q" and of data types "Int" or "UInt/Word" are permitted for discrete alarms and arrays.

Screens

The actual refresh rate of screens is determined by the type and volume of data to be visualized.

Configure short acquisition cycles only for objects that actually require shorter refresh cycles.

Trends

The HMI device always updates all bit-triggered trends whose group bit is set in the "Trend transfer area". Afterward, the bits are reset.

The group bit in the PLC program can only be set again after the HMI device has reset all bits.

Job mailboxes

A high rate and volume of PLC jobs transferred may lead to overload in communication between the HMI device and the PLC.

If the HMI device enters the value 0 in the first data word of the job mailbox, the HMI device has received the PLC job. The HMI device processes the job. This requires a certain amount of time.

It may take the HMI device some time to process a new PLC job that is transferred to the job mailbox. The next PLC job is only accepted if sufficient computing resources are available.

User data areas

3.1 Area pointer

3.1.1 General information on range pointers

Introduction

Area pointers are parameter fields from which WinCC flexible runtime obtains information about the location and size of data areas in the controller. The controller and HMI device interactively communicate to read and write data for these data areas. The controller and the HMI device trigger defined interactions based on the evaluation of stored data.

The area pointers reside in controller memory. Their addresses are configured in the "Range Pointers" dialog of the "Connections" editor.

WinCC flexible uses the following area pointers:

- Controller job
- Project ID
- Screen number
- Data record
- Date/time
- Date/time of controller
- Coordination

Application

Configure and enable the area pointer in "Communication > Connections" before you put it into use.

Parameter		Bereichszeiger				
Für alle Verbindungen						
Verbindung	Name	Adresse	Länge	Erfassungsart	Erfassungszyklus	Kommentar
<undefiniert>	Bildnummer		5	Zyklisch fortlaufend	<undefiniert>	
<undefiniert>	Datum/Uhrzeit Steuerung		6	Zyklisch fortlaufend	<undefiniert>	
<undefiniert>	Projektkennung		1	Zyklisch fortlaufend	<undefiniert>	
Für jede Verbindung getrennt						
Aktiv	Name	Adresse	Länge	Erfassungsart	Erfassungszyklus	Kommentar
Aus	Datensatz		5	Auf Anforderung	<undefiniert>	
Aus	Datum/Uhrzeit		6	Auf Anforderung	<undefiniert>	
Ein	Koordinierung	DB 1 DBW 0	1	Zyklisch fortlaufend	<undefiniert>	
Aus	Steuerungsauftrag		4	Zyklisch fortlaufend	<undefiniert>	

The contents of the columns have the following meaning:

- Active
The corresponding area pointer can be enabled or disabled in this table cell.
- Name
Contains the default name of the area pointer set by WinCC flexible.
- Address
Enter the tag address of the area pointer in the controller.
- Length
WinCC flexible defines the default length of the area pointer.
- Acquisition type
The acquisition type determines the mode and frequency at which the HMI device reads from or writes to the user data area. Select an acquisition type.
- Acquisition cycle
You must specify an acquisition cycle so that a area pointer will be read periodically by runtime. Note that an extremely short acquisition time may have a negative impact on HMI device performance.
- Comment
Use a comment, for example, to describe the purpose of the area pointer.

Functional scope

The HMI device used decides the number and type of area pointers that can be set. For information about the area pointers of each HMI device, refer to the "Communication" user manual.

You can set the following area pointers:

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

The table shows how the controller and HMI device handle read (R) and write (W) access to the data areas.

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

The next sections describe the area pointers and their associated job mailboxes.

3.1.2 "Screen number" range pointer

Function

The HMI device saves information about the screen called to the "Screen number" area pointer. This allows information about the current screen to be transferred to the controller. The controller can initiate specific responses such as calling a different screen.

Application

Configure and enable the area pointer in "Communication > Connections" before you put it into use.

Note

The area pointer can only be configured in the controller. One area pointer can be created for each controller.

The screen number is transferred automatically to the controller; that is, it is transferred each time a new screen is displayed on the HMI device. It is therefore unnecessary to configure an acquisition cycle.

Area pointer bit assignment

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

Data word	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
1. word	Current screen type														
2. word	Current screen number														
3. word	Reserved														
4. word	Current field number														
5. word	Reserved														

The following value ranges apply:

- Current screen type
"1" for the root screen or "4" for the permanent window
- Current screen number
1 to 65535
- Current field number
1 to 65535

3.1.3 "Date/time" range pointer

Function

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

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

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

Area pointer bit assignment

This range pointer has the following assignment:

Data word	Left byte							Right byte							
	15						8	7						0	
n + 0	Reserved							Hour (0 to 23)							Time
n + 1	Minute (0 to 59)							Second (0 to 59)							
n + 2	Reserved							Reserved							Date
n + 3	Reserved							Day of week (1 to 7, 1 = Sunday)							
n + 4	Day (1 to 31)							Month (1 to 12)							
n + 5	Year (80 to 99/0 to 29)							Reserved							

Note

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

3.1.4 "Date/time PLC" range pointer

Function

This range 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 range pointer. All definitions are coded in BCD format. The HMI device reads the data cyclically within the configured acquisition cycle and synchronizes itself.

Note

Set an acquisition cycle of sufficient length for the "Date/time" area pointer in order to avoid any negative impact on HMI device performance.

Recommended acquisition cycle: 1 minute, if the process allows this.

Area pointer bit assignment

The "Date/time" data area has the following structure:

Data word	Left byte							Right byte						
	15						8	7						0
n + 0	Year (80 to 99/0 to 29)							Month (1 to 12)						
n + 1	Day (1 to 31)							Hour (0 to 23)						
n + 2	Minute (0 to 59)							Second (0 to 59)						
n + 3	Reserved							Reserved			Day of week (1 to 7, 1 = Sunday)			

Note

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

3.1.5 "Coordination" range pointer

Function

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

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

The "Coordination" range pointer is two words long.

Note

Each time the HMI device updates the range pointer, the entire coordination area is always written.

The PLC program is therefore not permitted to make any changes in the coordination area.

Area pointer bit assignment

This range pointer has the following assignment:

Most significant byte								Least significant byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	-	-	-	X2	X1	X0
2. data word - unassigned								1. data word							

- Reserved
- X0 Startup bit
- X1 Operating mode
- X2 Life bit

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.

3.1.6 "Project ID" range pointer

Function

When runtime starts it can check to see if the HMI device is connected to the required controller. This is important particularly when operating with several HMI devices.

The HMI device compares a value stored on the controller with the configured value. This ensures that the project data will be compatible with the controller program. If discrepancy is detected, a system alarm issued on the HMI device and runtime is stopped.

Application

Make the following configuration settings in order to use this range pointer:

- Specify the project version.
Possible values between 1 and 65535.
Enter the version under "Project ID" in the "Device Settings".
- Specify the data address of the value for the version that is stored in the controller.
Enter the data address under "Address" in the "Communication ► Connections".

3.1.7 "Job mailbox" range pointer

Function

Controller jobs can be forwarded to the HMI device via the job mailbox. This makes it possible to initiate actions on the HMI device. These actions include, for example:

- Display screen
- Set date and time

Area pointer bit assignment

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

This range pointer has the following assignment:

Data word	Left byte								Right byte							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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 controller job. This means that the parameters must be entered in the job mailbox first, followed by the job number. When the HMI device accepts the controller job, the first word is set to 0 again.

The execution of the controller job is generally not completed at this point in time.

Controller job

All possible controller jobs on the different HMI devices are then listed, along with their parameters.

The "No." column contains the job number of the controller job. Job mailboxes can only be triggered by the controller when the HMI device is online.

No.	Function	
14	Set time (BCD)	
	Parameter 1	Left byte: – Right byte: Hours (0 to 23)
	Parameter 2	Left byte: Minutes (0 to 59) Right byte: Seconds (0 to 59)
	Parameter 3	–

No.	Function	
15	Set date (BCD)	
	Parameter 1	Left byte: – Right byte: Day of week (1 to 7: Sunday to Saturday)
	Parameter 2	Left byte: Day (1 to 31) Right byte: Month (1 to 12)
	Parameter 3	Left byte: Year

No.	Function	
23	User logon	
	Logs the user on with the name "PLC user" at the HMI device with the group number transferred in Parameter 1. The logon is possible only when the transferred group number exists in the project.	
	Parameter 1	Group number 1 to 255
	Parameter 2	–
	Parameter 3	–

No.	Function	
24	User logoff	
	Logs off the current user. This function corresponds to the "Logoff" system function.	
	Parameter 1	–
	Parameter 2	–
	Parameter 3	–

No.	Function	
40	Transfer date/time to controller	
	S7 format DATE_AND_TIME An interval of at least 5 seconds must be maintained between two successive jobs to prevent overload of the HMI device.	
	Parameter 1	–
	Parameter 2	–
	Parameter 3	–

No.	Function	
41	Transfer date/time to controller	
	OP/MP HMI device format An interval of at least 5 seconds must be maintained between two successive jobs to prevent overload of the HMI device.	
	Parameter 1	–
	Parameter 2	–
	Parameter 3	–

No.	Function	
46	Update tags	
	Causes the HMI device to read the current value of the controller tags whose update ID matches the value transferred in Parameter 1. This function corresponds to the "Update tag" system function.	
	Parameter 1	1 to 100

No.	Function	
46	Update tags	
	Parameter 2	–
	Parameter 3	–

No.	Function	
49	Clear alarm buffer	
	Parameter 1	–
	Parameter 2	–
	Parameter 3	–

No.	Function	
50	Clear error alarm buffer	
	Parameter 1	–
	Parameter 2	–
	Parameter 3	–

No.	Function	
51	Screen selection	
	Parameter 1	Screen number
	Parameter 2	–
	Parameter 3	–

No.	Function	
69	Read data record from controller	
	Parameter 1	Recipe number (1 to 999)
	Parameter 2	Data record number (1 to 65535)
	Parameter 3	0: Do not overwrite existing data record 1: Overwrite existing data record

No.	Function	
70	Write data record to controller	
	Parameter 1	Recipe number (1 to 999)
	Parameter 2	Data record number (1 to 65535)
	Parameter 3	–

3.1.8 "Data record" area pointer

3.1.8.1 General information

Introduction

When data records 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 records between the HMI device and controller:

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

Data records are always transferred directly. This means that tag values are read from or written to the address that is configured for the tag. It is not necessary to pass through the clipboard.

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 records can also be triggered by the controller.

- Triggering by configured functions

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

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

3.1.8.2 Transferring without synchronization

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

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

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

Reading values

When the transfer 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 downloaded to the HMI device, where they can undergo additional processing.
- Triggering by a function or controller job
The values are saved immediately to the data volume.

Writing values

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

3.1.8.3 Causes of error when transferring data records

Overview

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

- Tag address is not set up in the controller.
- Data records cannot be overwritten.
- Recipe number does not exist.
- Data record number does not exist.

NOTICE
Data consistency compromised
If you evaluate the recipe number and data record number in the controller before the status in the mailbox is set to "Transfer completed" or "Transfer completed with error", this can result in inconsistent data.
Therefore, do not evaluate the recipe number or data record number in the controller until after the status has been set.

Reaction to an aborted transfer due to errors

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

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

However, it is still possible to evaluate the status of the transfer by querying the status word in the mailbox.

3.1.8.4 Transferring when triggering by means of a configured function

The following transfer options are available:

- Reading from the controller using a configured function
- Writing to the controller by means of configured function

The functions are described in more detail below.

Reading from the controller using a configured function

Step	Action	
1	Test: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe number and data record number specified in the function, as well as "Transfer active" status, in the data mailbox.	Abort with system alarm
3	The HMI device reads the values from the controller. These values are stored in the data record that is specified in the controller job.	
4	<ul style="list-style-type: none"> If "Yes" was selected for the "Overwrite" function, an existing data record is overwritten without any prompt for confirmation. <p>The HMI device sets the status "Transfer completed".</p> <ul style="list-style-type: none"> If "No" was selected for the "Overwrite" function and the data record exists, the HMI device aborts the job. <p>The value 0000 1100 is entered in the status word of the data mailbox.</p>	
5	To allow further transfers, the controller program must set the status word to 0 again.	

Writing to the controller by means of configured function

Step	Action	
1	Test: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe number and data record number specified in the function, as well as "Transfer active" status, in the data mailbox.	Abort with system alarm
3	The HMI device calls the values from the data record specified in the job. These values are written to the controller.	
4	The HMI device sets the status "Transfer completed".	
5	<p>The controller program can now evaluate the transferred data.</p> <p>To allow further transfers, the controller program must set the status word to 0 again.</p>	

3.1.8.5 Transferring with a PLC job

Function

Transfer of data records between the HMI device and controller can be initiated by the following devices:

- HMI device
- Controller

Job mailboxes no. 69 and no. 70 are available for this transfer.

Read data record from controller "PLC → DAT" (no. 69)

Controller job no. 69 transfers data records from the controller to the HMI device.

The controller job is structured as follows:

Data word	Left byte	Right byte
Word 1	0	69
Word 2	Recipe number (1 to 999)	
Word 3	Data record number (1 to 65,535)	
Word 4	Do not overwrite existing data record: 0 Overwrite existing data record: 1	

Write data record to controller "DAT → PLC" (no 70)

Controller job no. 70 transfers data records from the HMI device to the controller. The controller job is structured as follows:

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

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

Step	Action	
1	Test: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe number and data record number specified in the job, as well as the "Transfer active" status, in the data mailbox.	Abort with system alarm
3	The HMI device reads the values from the controller. These values are stored in the data record that is specified in the controller job.	
4	<ul style="list-style-type: none"> If "Overwrite" was selected in the job, an existing data record is overwritten without any prompt for confirmation. <p>The HMI device sets the status "Transfer completed".</p> <ul style="list-style-type: none"> If "Do not overwrite" was selected and the data record exists, the HMI device aborts the job. <p>The value 0000 1100 is entered in the status word of the data mailbox.</p>	
5	To allow further transfers, the controller program must set the status word to 0 again.	

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

Step	Action	
1	Test: Status word = 0?	
	Yes	No
2	The HMI device enters the recipe number and data record number specified in the job, as well as the "Transfer active" status, in the data mailbox.	Abort with system alarm
3	The HMI device calls the values from the data record specified in the job. These values are written to the controller.	
4	The HMI device sets the status "Transfer completed".	
5	<p>The controller program can now evaluate the transferred data.</p> <p>To allow further transfers, the controller program must set the status word to 0 again.</p>	

3.1.8.6 Transfer by the operator in the recipe view

Reading from the controller started by the operator in the recipe view

Step	Action	
1	Test: 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. The data record number is set to 0.	Abort with system alarm
3	The HMI device reads the values from the controller and displays them in the recipe view. 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 view

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

3.1.8.7 Transferring with synchronization

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

Application

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

- The controller is the "active peer" in the transfer of data records,
- The controller evaluates information about the recipe number and data record number.
- The transfer of data records is triggered by a controller job.

Requirement

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

- A range pointer has been set up:
"Communication > Connections" editor in "Range pointer"
- The recipe specifies the controller used by the HMI device to synchronize the data record transfer:
"Recipes" editor, properties view of the recipe, "Properties" group in "Transfer".

Area pointer bit assignment

The data area has a fixed length of five words with the following structure:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1. word	Current recipe number (1 to 999)															
2. word	Current data record number (0 to 65,535)															
3. word	Reserved															
4. word	Status (0, 2, 4, 12)															
5. word	Reserved															

The status word (4th word) can assume the following values:

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

3.2 Status alarms, error alarm, and acknowledgments

3.2.1 Overview

Function

Messages provide users with information about operating states or operating faults on the controller or HMI device. The alarm texts consist of configurable texts and/or tags with actual values.

A basic distinction is made between status alarms and error alarms. The programmer defines what a status alarm and what an error alarm is.

Status alarm

A status alarm indicates a status, for example:

- Motor on
- Controller in manual mode

Error alarm

An error alarm indicates an operating fault, for example:

- Valve does not open
- Excess motor temperature

Error alarms indicate exceptional operational states, and must therefore be acknowledged.

Acknowledging error alarms

Error alarms are acknowledged by:

- Operator input on the HMI device
- Setting of an acknowledgment bit by the controller

Initiating status alarms and error alarms

A message is initiated in the controller by:

- Setting a tag bit
- Violation of a measured value

The location of the tags or tag array is defined in WinCC flexible 2008 Engineering System. The tag or tag array must be set up on the controller.

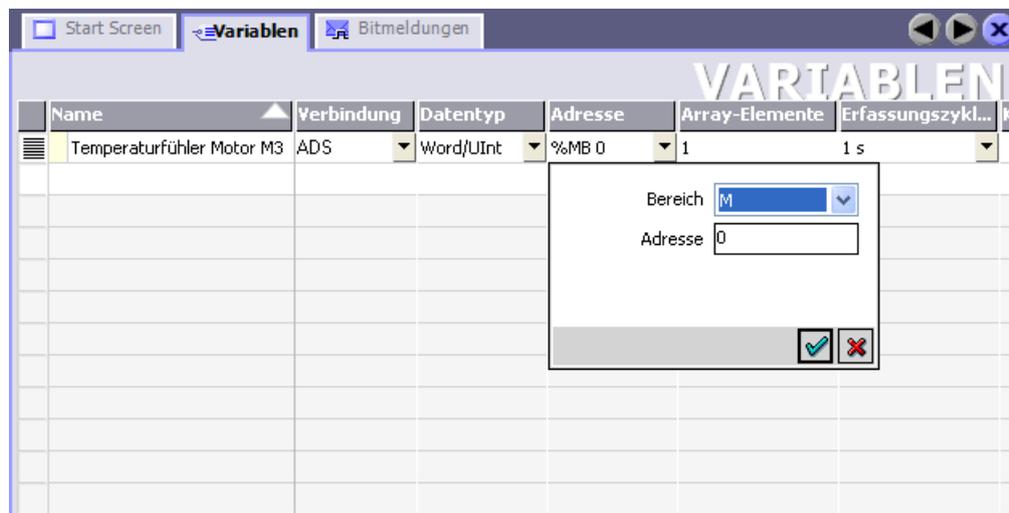
Messages with WinCC flexible

The number of alarms and words varies for each different HMI device class. For information about specification of the "Alarms" object, refer to the "Overview" section of the relevant operating instructions.

3.2.2 Creating a tag or an array

Requirement

You create a tag or an array in the "Tags" editor. The dialog box is shown below.



Procedure

Proceed as follows:

1. Define the tag and array names.
2. Select the connection to the controller.

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

3. 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:

Permitted data types	
Discrete alarms	Analog alarms
Int, UInt/Word	Bool, Sint, UInt/Byte, Int, UInt/Word, DInt, UInt/DWord, Real, String

4. Specify the address.

The addressed tag contains the alarm-triggering bit. 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".

5. Specify the array elements.

If the number of array elements is increased, you can select more bit numbers in the "Discrete alarms" editor. An array with a length of three words provides 48 alarm bits, for example.

3.2.3 Configuring an alarm

Introduction

Alarms are divided into the categories:

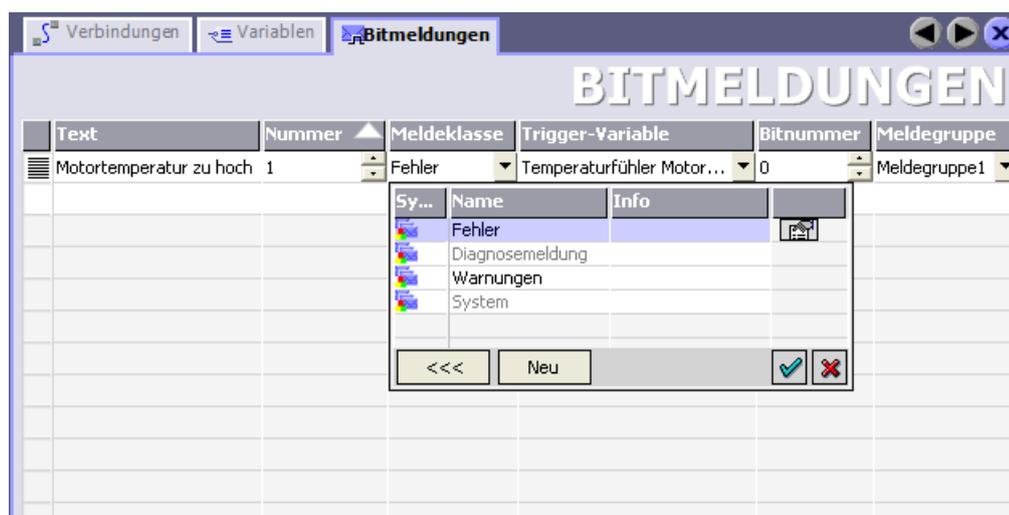
- Discrete alarms
- Analog alarms

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

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

Requirement

The editor is shown in the figure below.



Procedure

Proceed as follows:

1. Edit the 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 is output to the alarm view if this view was configured in the "Screens" editor.

2. Define the number.

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

Values between 1 and 100,000 are allowed.

Consecutive alarm numbers are assigned by WinCC flexible. You can change the alarm numbers if the numbers are to be divided into groups, for example.

3. Define the alarm class.

Available alarm classes:

– Error alarms

This class must be acknowledged.

– Status alarms

This class signals events with incoming and outgoing alarms.

4. Assign the trigger tag.

In the "Trigger tag" column, the configured alarm can be linked to the tag created in step 1. The selection list returns all tags with valid data type.

5. Define 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. The following bit-position counting method is used for a Beckhoff ADS controller:

Counting method	Left byte								Right byte							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Beckhoff controller	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WinCC flexible project	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

3.2.4 Configuring the acknowledgment

Requirements

Create suitable tags on the controller to acknowledge an error alarm. You assign these tags to an alarm in the "Bit messages" editor.

You make the assignment in "Properties > Acknowledgment".

The following figure shows the dialog for configuring an acknowledgment.



Acknowledgment is possible on the following devices:

- Controller
- HMI device

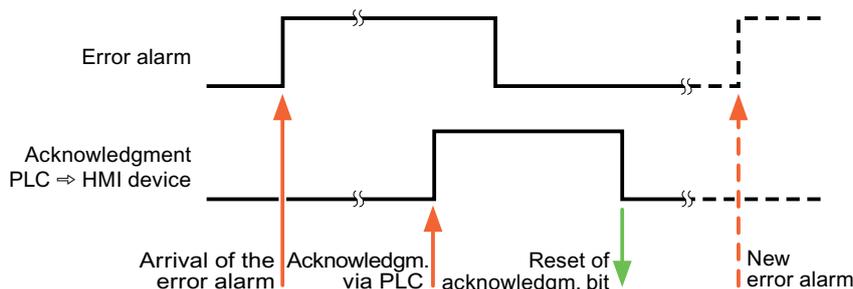
Acknowledgment via the controller

In "Acknowledgment controller 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 returns acknowledgement of the assigned error alarm bit on the HMI device, thus performing a function similar to acknowledgment on the HMI device triggered, for example, by pressing the "ACK" button.

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

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



Acknowledgment on the HMI device

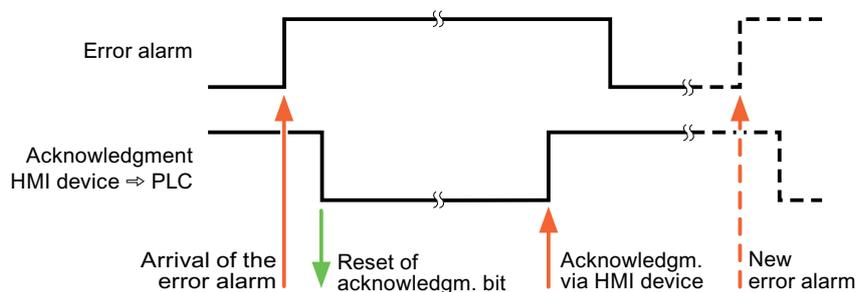
In "Ack read tag", you configure the tag or the array tag and the bit number that is written to the controller after acknowledgment from the HMI device.

The HMI devices always behaves in the following manner in order to generate a signal change every time the acknowledgment bit is set:

As soon as an error alarm bit is set, the HMI device resets the acknowledgment bit that is assigned to the error alarm. There is a time-based offset between these two operations, due to the processing time in the HMI device.

If the error 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 error alarm has been acknowledged.

The figure below shows the pulse diagram.



3.3 Trend request and trend transfer

Introduction

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

Time-triggered trends

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

Bit-triggered trends

By setting a trigger bit in the "Trend request" tag, the HMI device either reads in a trend value or an entire trend buffer. The configuration determines whether the trend values is read in its entirety or only partially. Bit-triggered trends are normally used to represent fast-changing values. One example might be the injection pressure in the production of plastic parts.

To trigger bit-triggered trends, create external tags in the "Tags" editor and interconnect them 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 "%M", "%I", and "%Q" "Area" are permitted. The tags must be of the "Int" or "UInt/Word" data type. The data types "Int" or "UInt/Word" are valid for an array tag. 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 evaluation of the trend 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.

If the HMI device recognizes the trigger, either a value or the entire buffer is read from the controller. The HMI device then resets the trend bit and the trend group bit.

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

Data word	Most significant byte								Least significant byte							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1																
2																
up to																
n	x															

x Trend indicator 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. The switch buffer has the exact same structure 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 in 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.

3.4 LED mapping

Introduction

This section applies to HMI devices with light diodes (LEDs) in their function keys. These LEDs can be controlled by the controller. This functionality can be used to activate an LED in order to tell the operator which key to press in a specific situation, for example.

Requirement

In order to control an LED, an LED tag or an array tag must be set up in the controller. The array tag must be defined as an LED tag during configuration.

LED assignment

Assign the LEDs to the LED tag bits when you configure the function keys. Define the "LED tag" and the corresponding "bit" for each function key in the "General" group of the "Properties" window.

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

Bit n+ 1	Bit n	LED status
0	0	Off
0	1	Flash
1	0	Flash
1	1	On permanently

Software commissioning

4.1 Overview

Commissioning the program involves the following activities:

- Transfer the controller program to the controller.
- Transfer project to the HMI device.

These work steps are described below.

4.2 Transferring the PLC program to the PLC

Procedure

Proceed as follows:

1. Connect the controller to the data network over an Ethernet cable.

The controller and HMI device must be located in the same data network. The controller can also be connected directly to the HMI device using a crossover cable.

2. Enable the Ethernet address of the HMI device on the Beckhoff controller.

Note

If you connect a SIMATIC HMI device to a Beckhoff controller and want to enable communication, you will need to follow the steps described in this configuration manual.

The SIMATIC HMI device is enabled on the Beckhoff CX1000 controller using the CERHOST.EXE program – see Section Enabling a CX1000 controller (Page 47).

The AMS router is enabled via a remote computer and for the CX1000 controller using a registry entry.

3. Start the controller.
4. Download the program files to the CPU.
5. Set the CPU to "RUN".

The CPU starts.

4.3 Transferring the project to the HMI device

Requirement

The HMI device must be in transfer mode to transfer the project.

The following cases are distinguished:

- Initial startup

The HMI device does not yet contain any configuration data in the initial commissioning phase. The project data and runtime software required for operation must be transferred from the configuration computer to the device: The HMI device automatically goes to transfer mode.

The transfer dialog with a connection alarm is displayed on the HMI device.

- Recommissioning

During recommissioning, an existing project on the HMI device is replaced by another project.

A guide to this procedure is included in the operating instructions for your HMI device.

Procedure

Proceed as follows:

1. Check whether the alarm settings in your project meet your requirements.
2. Initialize the transfer parameters by selecting "Project > Transfer > Transfer Settings" from the menu.

The "Select Device for Transfer" dialog opens.

Initialize the required parameters. For additional information, refer to the Online Help.

3. Click on the "Transferieren" button.

The project transfer begins.

- The project is compiled automatically.
- All compilation and transfer steps are logged in the output window.

If the transfer was successful, the message "Transfer completed successfully" is displayed on the configuring PC. The start screen appears on the HMI device. The message "Connection to PLC is established" is then displayed.

Note

The system message text in WinCC flexible can be edited by the user.

Enabling a CX1000 controller

Introduction

If you use a controller of the type CX1000, you require the CERHOST.EXE program. The CERHOST.EXE program is used to enable access from the SIMATIC HMI device to the connected controller. Enabling access is done on the controller.

Requirement

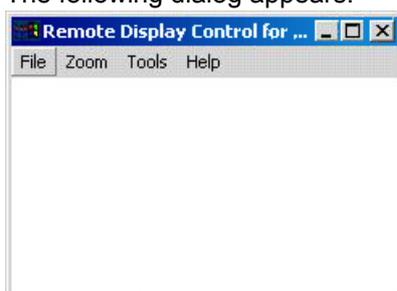
The controller is turned on.

Procedure

Proceed as follows:

1. Start the CERHOST.EXE program.

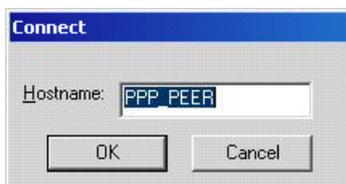
The following dialog appears.



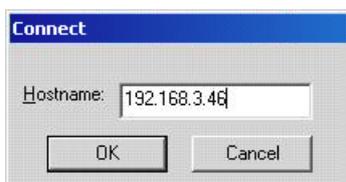
2. Select "Connect" in the "File" menu.



3. The following dialog appears.

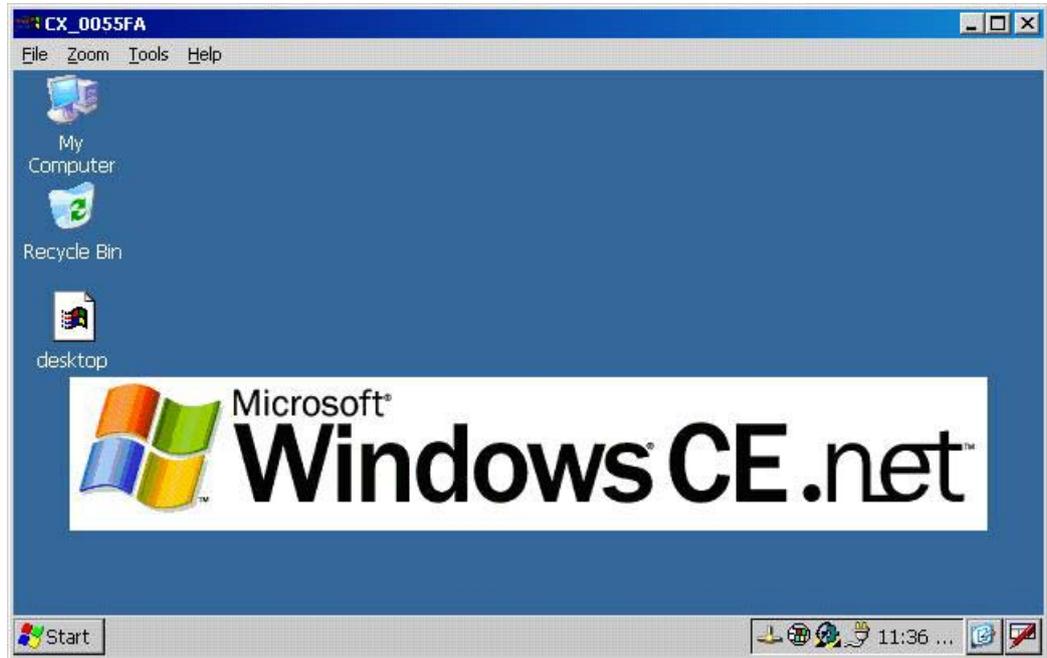


4. Enter the IP address of the CX1000 controller.



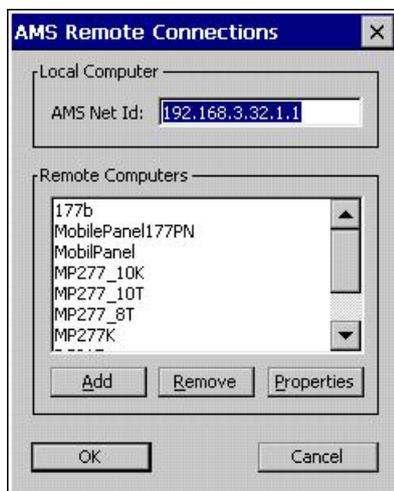
5. Click on the "OK" button.

The following screen appears on the desktop.



6. Select "My Computer > Hard Disk > System".
7. Launch the TcAmsRemoteMgr.exe program.

The following dialog appears.



The window shows the HMI devices that have already been enabled.

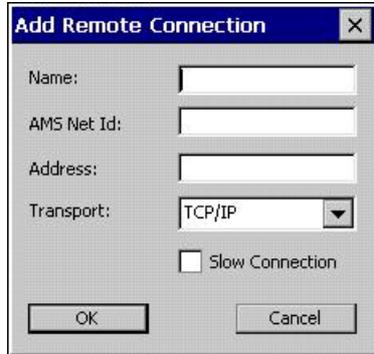
8. Enter the AMSNetID.

Note

Make sure that the entry for the "AMSNetID" has the extension ".1.1".

- Click the "Add" button, if the program is being used for the first time.

The following dialog appears.



- Create a node.

Enter a random name in the "Name" input box. Under "Address" enter the IP address of the SIMATIC HMI device and under "AMS Net Id" enter the IP address of the SIMATIC HMI device with the extension ".1.1".

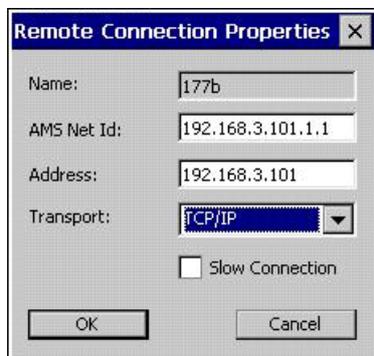
- Click on the "OK" button.

The dialog box closes.

<p>NOTICE</p> <p>Check the IP address and AMSNetId</p> <p>If the information does not match up, this will cause functional disruptions.</p> <p>Make sure that the "IP address" matches the "AMSNetID" and that the ".1.1" extension is present. This applies in particular to the SIMATIC HMI device as shown in the following example:</p> <p>IP address 192.168.3.45 → AMSNetID 192.168.3.45.1.1</p> <p>For the Beckhoff controller, this restriction applies only to the ".1.1" extension. The addresses themselves do not need to match.</p>
--

- Click the "Properties" button in the "AMS Remote Connections" dialog if you want to check or alter the settings.

The following dialog appears.



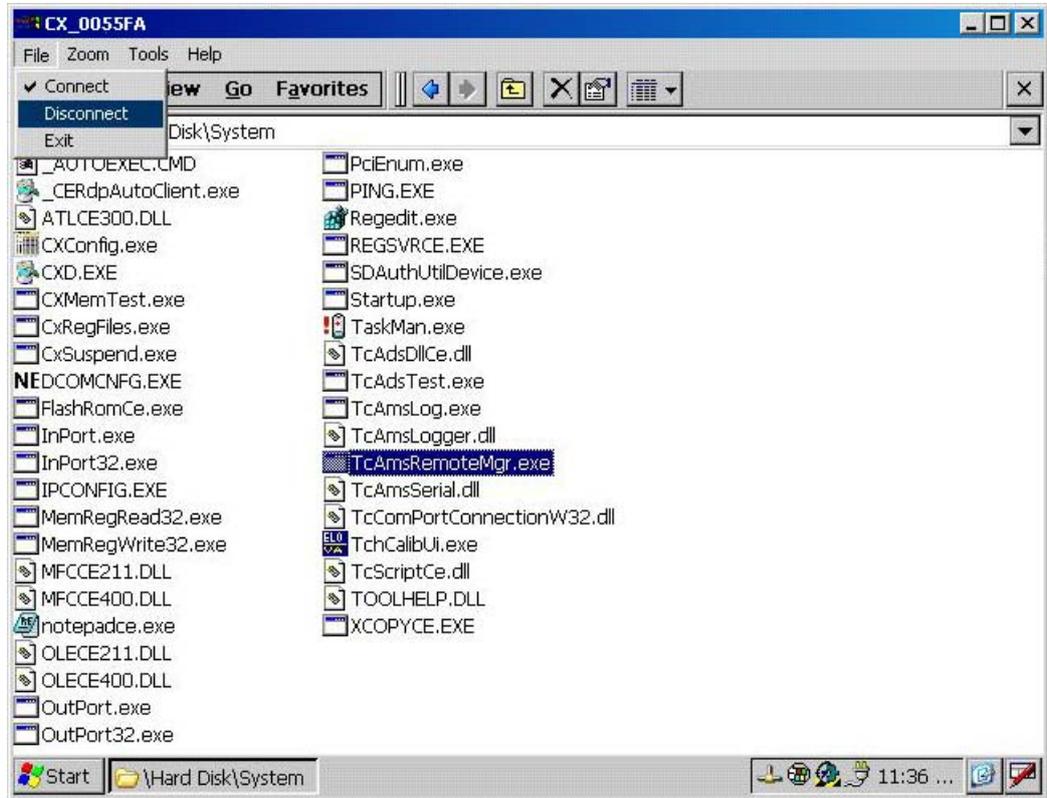
- Click on the "OK" button.

The dialog box closes.

14. Click the "OK" button.

The TcAmsRemoteMgr.exe program closes.

15. Select "Disconnect" in the "File" menu.



NOTICE

Exiting the CERHOST.EXE program

If you do not exit the CERHOST.EXE program with "File > Disconnect", you will lose all your entries.

For new entries or changes, exit the CERHOST.EXE program using "File > Disconnect" only.

16. Restart the controller.

Result

The HMI device is enabled for communication with the CX1000 controller.

Appendix

A

A.1 ESD guideline

What does ESD mean?



An electronic module is equipped with highly integrated electronic components. Due to their design, electronic components are highly sensitive to overvoltage and thus to the discharge of static electricity. Such electronic components are labeled as electrostatic sensitive devices (ESD).

The following abbreviations are commonly used for electrostatic sensitive devices:

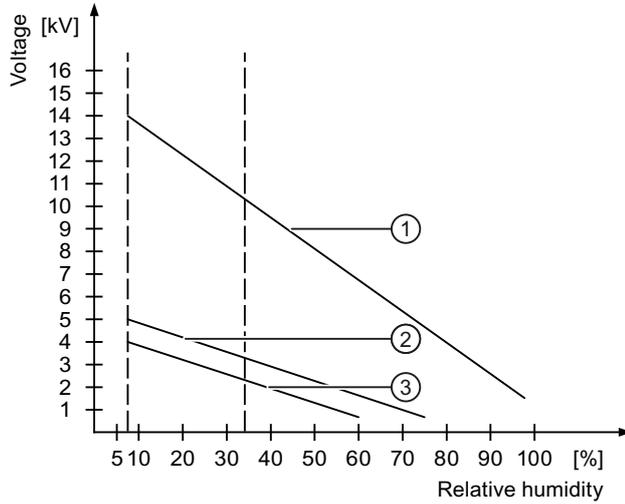
- ESD – Electrostatic Sensitive Device
- ESD – Electrostatic Sensitive Device (internationally recognized term)

Electrostatic charge

CAUTION
Electrostatic charge ESDs may be destroyed by voltages far below the level perceived by human beings. If you are not discharged electrostatically, the voltage that you transfer when touching a component or the contact points of a module can already cause damage. The damage to an ESD caused by overvoltage is usually not recognized immediately. The damage only becomes apparent after a long period of operation. Discharge any electrostatic charge of your body before you touch the ESD.

Anyone who is not connected conductively to their surroundings is subject to electrostatic charge.

The following diagram shows the maximum voltage values to which a person can be charged electrostatically. The values depend on the material and humidity. The shown values are in conformity with the specifications of EN 61000-4-2.



- ① Synthetic materials
- ② Wool
- ③ Antistatic materials such as wood or concrete

Protective measures against discharge of static electricity

CAUTION
<p>Grounding measures</p> <p>There is no equipotential bonding without grounding. An electrostatic charge is not discharged and may damage the ESD.</p> <p>When working with electrostatic sensitive devices, make sure that the person and the workplace are properly grounded.</p>

Note the following:

- Only touch the ESD if it is absolutely necessary.
- When you touch ESD modules, avoid touching the pins or the PCB tracks.
This precaution reduces the risk of damaging an ESD.
- Discharge electrostatic electricity from your body if you are performing measurements on an ESD.
To do so, touch a grounded metal object before you carry out the measurement.
- Always use grounded measuring instruments.

Abbreviations

BCD	Binary Coded Digit
CD	Compact Disk
CPU	Central Processor Unit
DAT	Dynamic Address Translation
EN	European standard
GmbH	Limited liability company
HMI	Human Machine Interface
IP	Internet Protocol
LED	Light Emitting Diode
PDF	Portable Document Format
PC	Personal Computer
PLC	Programmable Logic Controller
PLC	Programmable Logic Controller

Glossary

"Transfer" mode

Transfer is the transfer of an image or an HMI project from a configuring PC to an HMI device.

Controller

Controller is a general term for devices and systems with which the HMI device communicates, e.g. SIMATIC S7 controller.

EMC

Electromagnetic compatibility is the ability of electrical equipment to function properly in its electromagnetic environment without influencing this environment.

HMI device

HMI devices are used to operate and monitor machinery and plants. The machine or plant states are visualized on the HMI device by means of graphic objects or signal lamps. The operator controls of the HMI device enable intervention in machine or plant processes and sequences.

Parameters

In computer programming, a parameter is a variable element of a program or routine. This element is formally created in the program (formal parameter). During a call, the element is set to this one specific value (actual parameter). However, the element can also be a "switch" that controls sequences in the program - such as the command-line parameter.

Plant

General term referring to machinery, processing centers, systems, plants and processes which are operated and monitored using an HMI device.

Project

A project is the result of a configuration using an engineering software. The project usually contains several screens with embedded system-specific objects, basic settings and alarms.

Index

A

- Acknowledge
 - On the HMI device, 41
 - Via the controller, 41
- Acquisition cycle, 15
- ADS protocol, 7, 11
- AMS router, 45
- AMSNetID, 48
- Analog alarm, 38
- Area of application, 3
- Area pointer, 14
 - Application, 18
 - Functional scope, 19

B

- Basic knowledge
 - Required, 3
- BC9000, 7
- Beckhoff controller
 - Connectable, 7
- Bit assignment
 - "Date/time" range pointer,
 - Controller job area pointer, 25
 - Coordination range pointer, 23
 - Date/time controller range pointer, 22
 - Screen number area pointer, 20
- Bit message, 15
- Byte address, 14

C

- Cable
 - Connectable, 9
- CERHOST.EXE, 47
- Charge
 - Electrostatic, 51
- Communication type
 - Released, 8
- Configuration
 - Specialities, 14
- Configuration manual
 - Purpose, 3
- Connection
 - Multipoint, 8

- Point-to-point, 8
- Contact person, 4
- Controller
 - CX series, 13
 - CX1000, 7, 45, 47
 - TwinCAT SoftPLC, 13
- Controller job, 25
- Conventions, 4
- CX series, 13
- CX1000, 7, 45, 47

D

- Data record
 - Abort due to errors, 30
 - Initiating transfer, 28
 - Reading from controller, 32
 - Transfer mode, 28
 - Write to controller, 32
- Devices
 - Connectable, 8
- Documentation
 - SIMATIC complete, 4
- Drivers
 - ADS, 11

E

- Electrostatic charge, 51
- Error alarm, 36
 - Acknowledge, 36
 - Initiating, 36
 - Specification, 36

F

- Function
 - Reading from controller, 31
 - Write to controller, 31

H

- Hardware requirements, 9

I

Industry Portal, 4

L

Life bit, 23

M

Multipoint link, 8

O

Office location, 4
Operating mode, 23

P

Parameters
 Device-dependent, 12
PLC job, 16
Point-to-point link, 8
Protective measure
 Static electricity, 52
Protocol
 ADS, 11
Purpose of this configuration manual, 3

R

Reading from controller
 By operator in recipe view, 34
Requirements
 Hardware, 9
 Software, 9

S

Screen, 20
Sequence
 Reading from controller, 33
 Write to controller, 33
Service, 4
Software requirements, 9
Startup bit, 23
Static electricity
 Protective measures, 52
Status alarm, 36
 Initiating, 36

 Specification, 36
Support
 Technical, 4
Support Request, 4

T

Tag, 14
Tags, 8
TcAmsRemoteMgr.exe, 48, 50
Technical support, 4
Technical Support, 4
TPY file, 13
Training center, 4
Trend, 16
 Bit-triggered, 42
 Time-triggered, 42
Trend request area, 42
TwinCAT program, 14
TwinCAT SoftPLC, 13
TwinCAT Soft-PLC, 7

U

Update time, 15
User data area, 8

V

Value
 Read, 29
 stick, 29

W

Windows CE, 8
Write to controller
 By operator in recipe view, 34