ifm electronic

Supplementary device manual
Interface Ethernet/IP in the AS-i controllerE

ecomat300

AC1327

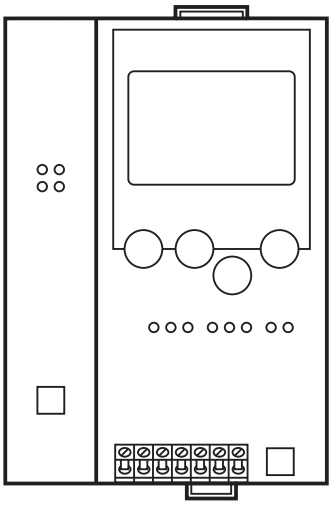
AC1337

Master profile: M4

Firmware version RTS 2.x

Target from 15

CoDeSys® version 2.3 or higher



7390706 / 02 02 / 2010

Contents

Contents

Contents

1 On this manual

Contents

In the additional "Programming Manual for CoDeSys ® V2.3" you will obtain more details about the use of the programming system "CoDeSys for Automation Alliance ™". This manual can be downloaded free of charge from **ifm's** website:
→ www.ifm.com > Select country/language > [Service] > [Download] > [Bus system AS-Interface]

Nobody is perfect. Send us your suggestions for improvements to this manual and you will receive a little gift from us to thank you.

1.1 What do the symbols and formats mean?

The following symbols or pictograms depict different kinds of remarks in our manuals:

⚠ DANGER

Death or serious irreversible injuries are to be expected.

⚠ WARNING

Death or serious irreversible injuries are possible.

⚠ CAUTION

Slight reversible injuries are possible.

NOTICE

Property damage is to be expected or possible.

NOTE

Important notes to faults and errors.

Info

Further hints.

► ...	Required action
> ...	Response, effect
→ ...	"see"
abc	Cross references (links)
[...]	Designations of keys, buttons or display

1.2 What devices are described in this manual?

This manual describes the AS-i device family controllerE of **ifm electronic gmbh**.

- according to AS-i master specification 3.0 (M4)
- with a firmware from version RTS 2.3 onwards
- with the target from 15 onwards
- with the option Ethernet fieldbus interface and the protocols Ethernet/IP and Modbus TCP

In this supplementary manual only the above-mentioned Ethernet fieldbus interface is described. Higher-level or general information → basic device manual.

In the "programming manual CoDeSys ® 2.3" you will find more details how to use the programming system "CoDeSys for Automation Alliance". This manual can be downloaded free of charge from **ifm's** website at:
→ www.ifm.com > Select country/language > [Service] > [Download] > [Bus system AS-Interface]

1.3 How is this documentation structured?

This documentation is a combination of different types of manuals. It is for beginners and also a reference for advanced users.

How to use this documentation:

- Refer to the table of contents to select a specific subject.
- At the beginning of a chapter we will give you a brief overview of its contents.
- Abbreviations and technical terms are listed in the glossary.
- The print version of the manual contains a search index in the annex.

In case of malfunctions or uncertainties please contact the manufacturer at:
→ www.ifm.com > Select country/language > [Contact]

We reserve the right to make alterations which can result in a change of contents of the documentation. You can find the current version on **ifm's** website at:
→ www.ifm.com > Select country/language > [Service] > [Download] > [Bus system AS-Interface]

2 Safety instructions

Contents

2.1 General

No characteristics are warranted with the information, notes and examples provided in this manual. The drawings, representations and examples imply no responsibility for the system and no application-specific particularities.

The manufacturer of the machine/equipment is responsible for the safety of the machine/equipment.

⚠ WARNING

Property damage or bodily injury possible when the notes in this manual are not adhered to!
ifm electronic gmbh does not assume any liability in this regard.

- ► The acting person must have read and understood the safety instructions and the corresponding chapters of this manual before performing any work on or with this device.
- ► The acting person must be authorised to work on the machine/equipment.
- ► Adhere to the technical data of the devices!
  You can find the current data sheet on **ifm's** homepage at:
  → www.ifm.com > Select country/language > [Data sheet direct] > (Article no.) > [Technical data in PDF format]
- ► Note the installation and wiring information as well as the functions and features of the devices!
  → supplied installation instructions or on **ifm's** homepage:
  → www.ifm.com > Select country/language > [Data sheet direct] > (Article no.) > [Operating instructions]

ATTENTION

The driver module of the serial interface can be damaged!

Disconnecting the serial interface while live can cause undefined states which damage the driver module.

- ► Do not disconnect the serial interface while live.

Start-up behaviour of the controller

The manufacturer of the machine/equipment must ensure with his application program that when the controller starts or restarts no dangerous movements can be triggered.

A restart can, for example, be caused by:

- voltage restoration after power failure
- reset after watchdog response because of too long a cycle time

2.2 What previous knowledge is required?

This document is intended for people with knowledge of control technology and PLC programming with IEC 61131-3.

If this device contents a PLC, in addition these persons should know the CoDeSys® software.

The document is intended for specialists. These specialists are people who are qualified by their training and their experience to see risks and to avoid possible hazards that may be caused during operation or maintenance of a product. The document contains information about the correct handling of the product.

Read this document before use to familiarise yourself with operating conditions, installation and operation. Keep the document during the entire duration of use of the device.

Adhere to the safety instructions.

3 System description

Contents

3.1 Information concerning the device

→ separate basic instructions of the device manual.

This manual describes the AS-i controllerE device family of **ifm electronic gmbh** with the option Ethernet interface.

3.2 Overview: where is what?

1. unlocking key for detaching the unit from a DIN rail
2. metal housing IP20
3. text/graphics display → basic manual
4. 4 operating keys → basic manual
5. status LEDs (here: 2 AS-i masters) → basic manual
6. 6-pole RJ11 socket of RS-232C as programming interface → basic manual
7. terminals for the voltage supply 24 V, AS-i interface(s) and protective wire → basic manual
8. Ethernet fieldbus interface
9. status LEDs of the fieldbus interface → status LEDs on the network connection (→ page 167)

Graphics: overview controllerE with interface Ethernet/IP

3.3 Information concerning the software

→ separate basic instructions of the device manual

3.4 Required accessories

Basic functions → separate basic instructions of the device manual.

For configuration and programming you also need:

- the software "CoDeSys for Automation Alliance™" version 2.3 or higher (→ CD),
- in case of direct connection of the controllerE to a PC with Ethernet interface (LAN):
  a cross-over CAT5 Ethernet patch cable with RJ45 plug on both sides:
  - 2 m e.g. art.-no. EC2080
  - 5 m e.g. art.-no. E30112
- in case of connection of the controllerE to a PC with Ethernet interface (LAN) via a hub or switch:
  - a common CAT5 Ethernet patch cable with RJ45 plug on both sides,
- in case of direct connection of the controllerE to a PC with serial interface:
  - programming cable article no. E70320.

4 Getting started

Contents

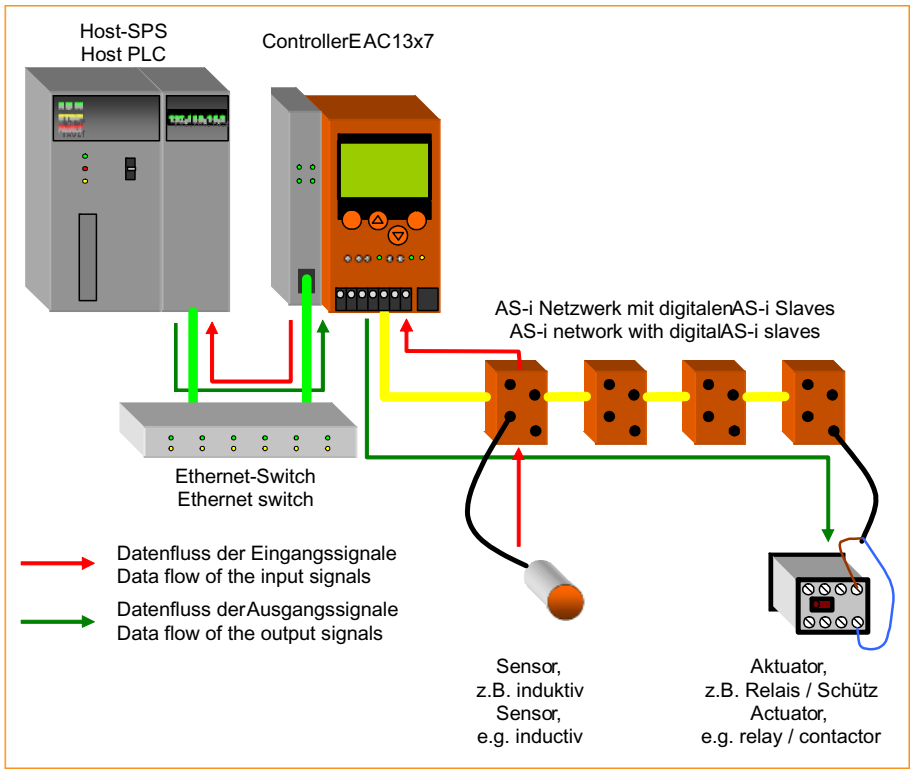
4.1 Overview

The chapter General set-up procedure (→ page 15) illustrates the general set-up procedure for the controllerE units AC1327 / AC1337 by means of 2 flowcharts. Possible error states and the corresponding corrective measures are described in additional tables in this chapter.

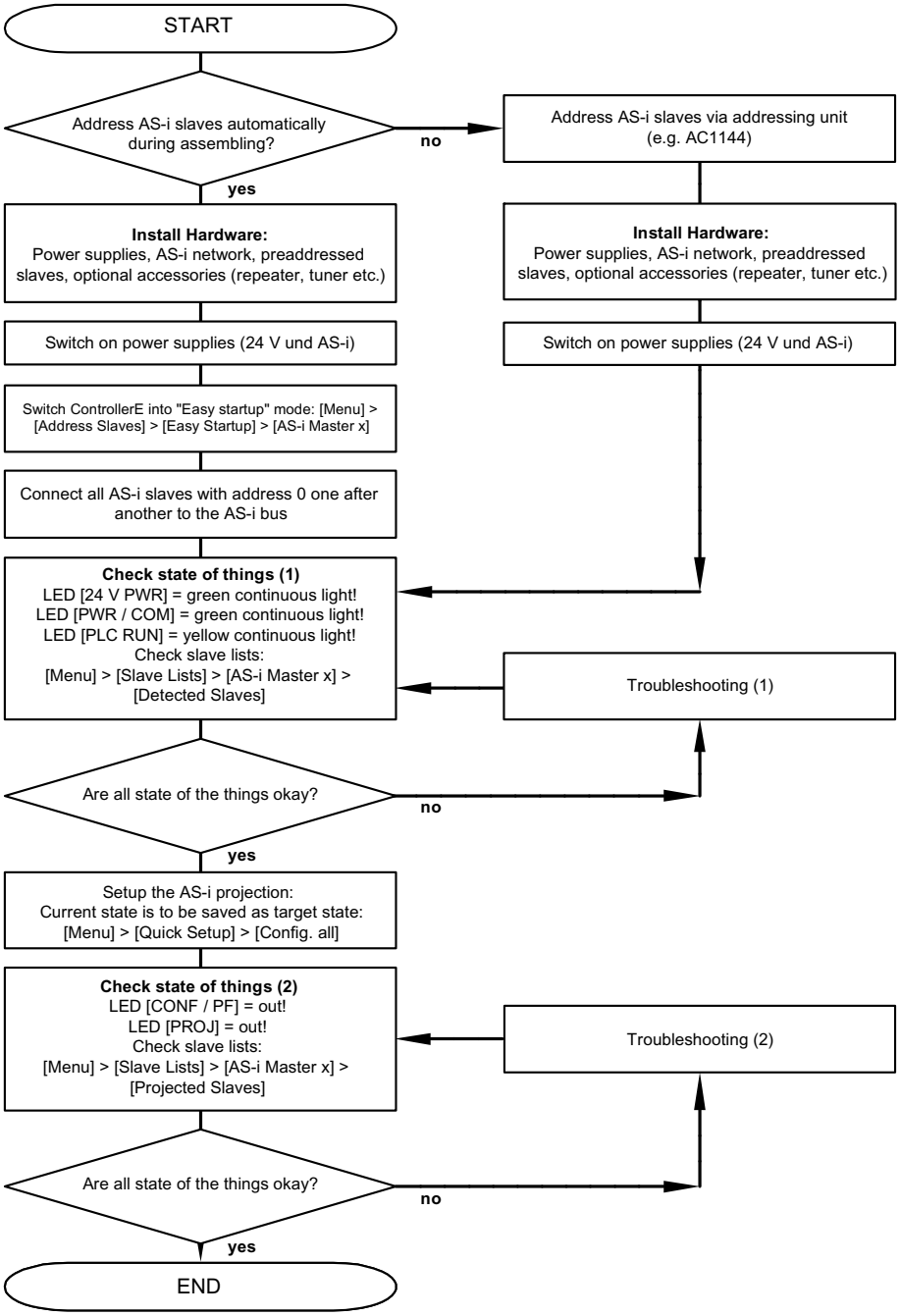
The chapters Connect Allen Bradley ControlLogix controller via Ethernet/IP (→ page 19) and Connection to a Schneider PL7 PLC via Modbus/TCP (→ page 26) show a configuration example of a connection between host PLCs and the controllerE. These quick instructions presuppose the following:

- 16 bytes digital input and output data respectively are to be exchanged between the connected host and the controllerE. Accordingly, the fieldbus modules 1 and 2 are both set to 16 bytes:
  → chapter Module 1: digital input master 1(A) (→ page 38),
  → chapter Module 2: digital output master 1(A) (→ page 40).
- The IP address and the subnet mask of the controllerE have been set as defined in the respective examples.
- The configuration files `ifm.cfg` and `onoffln.cfg` have been configured according to the selected connection protocol:
  → chapter The general configuration file ifm.cfg (→ page 64),
  → chapter The configuration file onoffln.cfg (→ page 66).

The following illustration is supposed to give an overview of the system structure and the corresponding data flow.



4.2 General setup



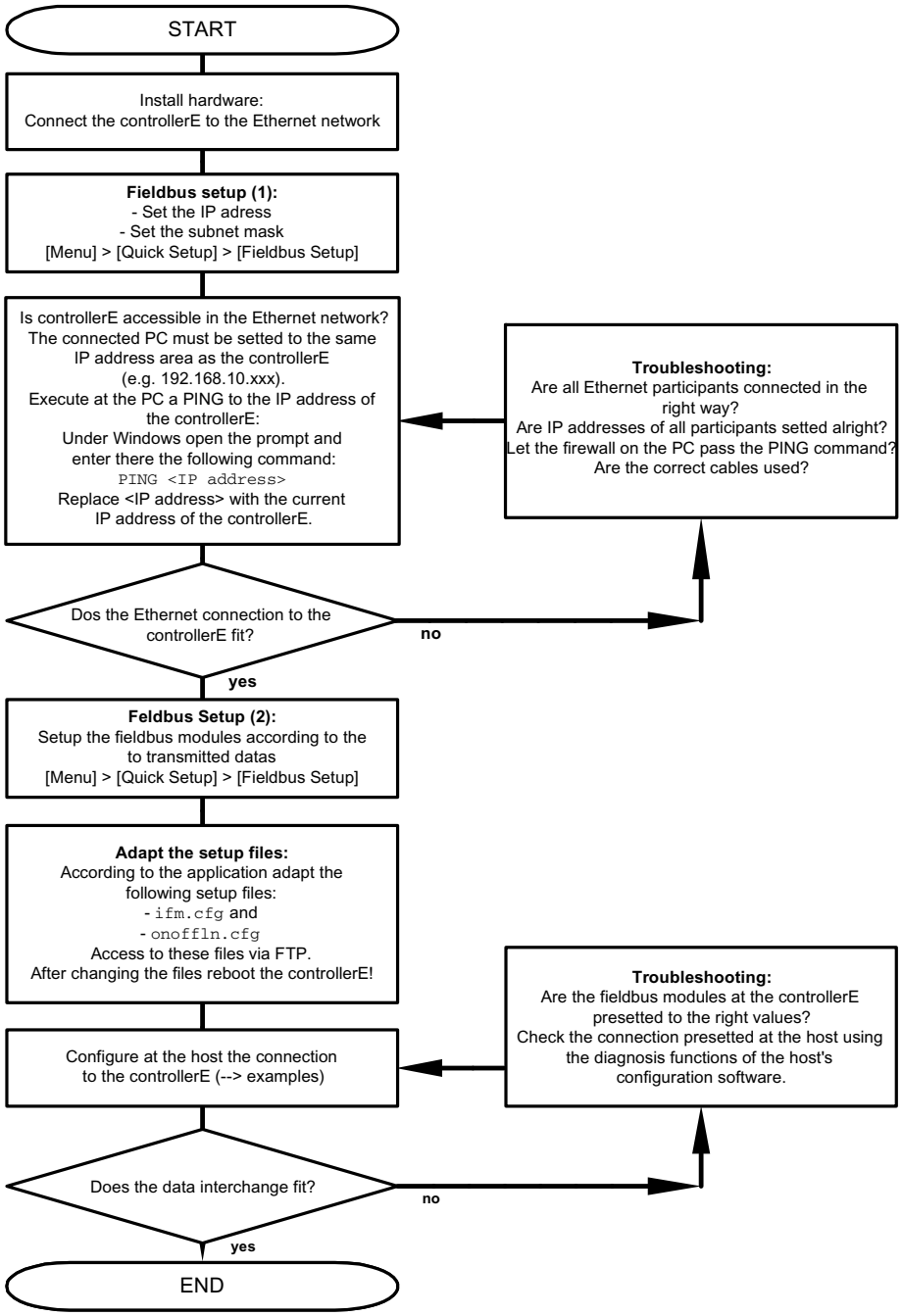
4.2.1 Troubleshooting (1)

Checkpoint	Condition	Possible cause	Remedy
LED [24 V PWR]	Off	24 V voltage supply not ok	► Check 24 V voltage supply!
LED [PWR / COM]	Off	AS-i voltage supply not ok	► Check AS-i voltage supply!
	Green flashing	AS-i voltage supply ok but no AS-i slave detected on the bus	► Check wiring of the AS-i network! Adhere to the maximum admissible cable lengths!
LED [PLC RUN]	Yellow flashing	ControllerE PLC is in the operating mode STOP	► Switch PLC to the operating mode RUN: [Menu] > [PLC Setup] > [PLC Settings] > [Run] ► If no change is possible: Is the project "ETIP_M4_xxx.pro" stored in the controllerE as boot project? [Menu] > [PLC Setup] > [PLC Info]
Slave lists (detected slaves)	The connected AS-i slaves are not detected correctly	Wiring fault in the AS-i network.	► Check wiring of the AS-i network! Adhere to the maximum admissible cable lengths!
		There is double addressing, i.e. two or more participants have been set the same AS-i address.	► Check the addresses of the connected AS-i slaves!

4.2.2 Troubleshooting (2)

Checkpoint	Condition	Possible cause	Remedy
LED [CONF / PF]	Red flashing	One of the connected AS-i slaves causes a peripheral fault	► Read the error messages on the display of the controllerE and determine the concerned slave address(es)! ► Check in the corresponding installation instructions of the concerned slaves what might cause a peripheral fault in the corresponding unit! ► Remove this cause!
	Red permanently lit (configuration error)	The list of activated slaves does not correspond to the list of projected slaves	► Check the wiring of the AS-i network, in particular the wiring of the slaves which are projected but not activated: [Menu] > [Slave Lists] > ... ► Adhere to the maximum admissible cable lengths!
		The configuration of the AS-i network was changed after executing the function "Config. all" (slave(s) added, slave(s) removed, slave(s) replaced by another type)	► Check the AS-i configuration! ► If the configuration is ok and the LED [CONF / PF] still is permanently lit: Repeat the function "Config. all": [Menu] > [Quick Setup] > [Config. all]
LED [PROJ]	Yellow flashing	The AS-i master is in the projection mode. Switching to the protected mode is not possible because at least one slave with the address 0 was detected the bus	► Correct the AS-i configuration according to your requests! ► Repeat the function "Config. all": [Menu] > [Quick Setup] > [Config. all]
	permanent yellow light	The AS-i master is in the projection mode	► Switch the AS-i master to the "protected mode": [Menu] > [Master Setup] > [AS-i Master x] > [Operation Mode] > [Protect. Mode]

4.3 Fieldbus setup (overview)



4.4 Connect Allen Bradley ControlLogix controller via Ethernet/IP

4.4.1 Step 1: Start RSLogix5000

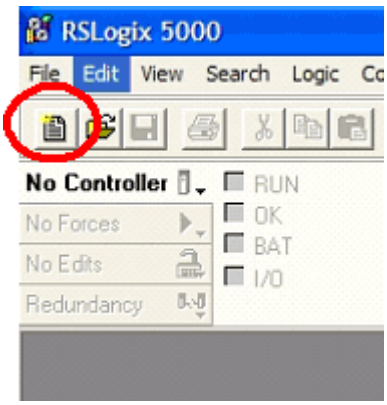
► Start the software on the PC.

4.4.2 Step 2: Create a new project

If there already is a project available to which an Ethernet/IP connection is to be added:

► Skip step 2 and continue with Step 3 (→ page 20).

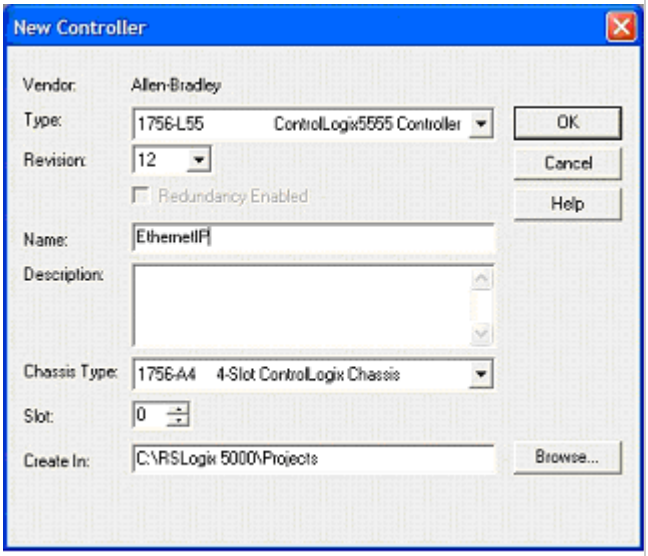
► Create a new project:
Click on the symbol [File new]
(→ screenshot)
or:
Select the menu [File] > [New...].



> The window [New Controller] appears
(→ screenshot).

► Enter the following data:

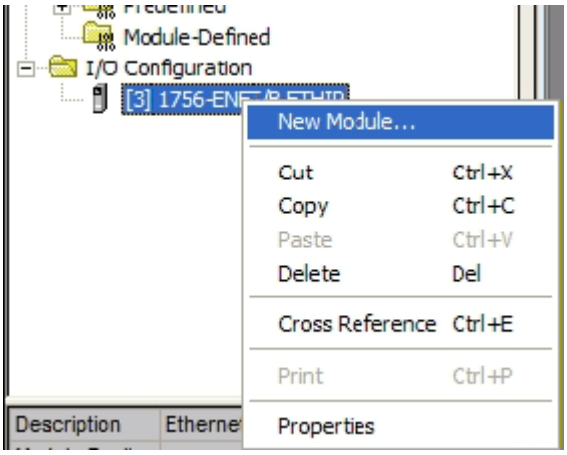
Term	Explanation
Type	Select CPU type
Revision	Select CPU version
Name	Project name
Description	optional project description
Chassis Type	Select the type of rack
Slot	Select the position of the CPU in the rack
Create In	Enter the project path in which the project is to be stored on the hard disk



4.4.3 Step 3: Add a new module in the directory "I/O Configuration"

► Right click on the directory "I/O Configuration".

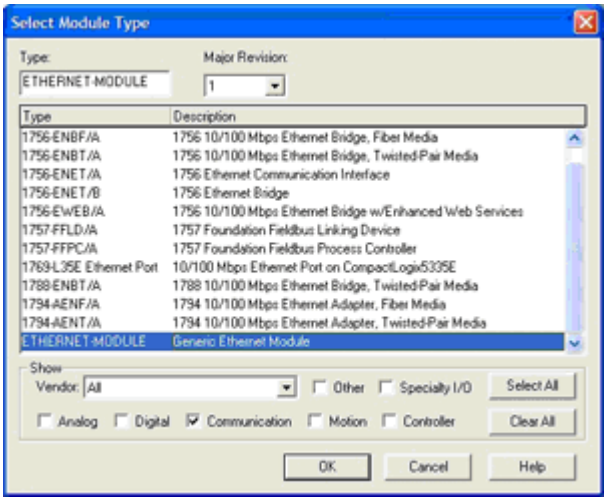
► Select [New Module...] in the context menu (→ screenshot).



> The window [Select Module Type] appears (→ screenshot).

► Select the used module type, here: Ethernet Bridge "1756-ENET/B".

► Confirm with [OK].

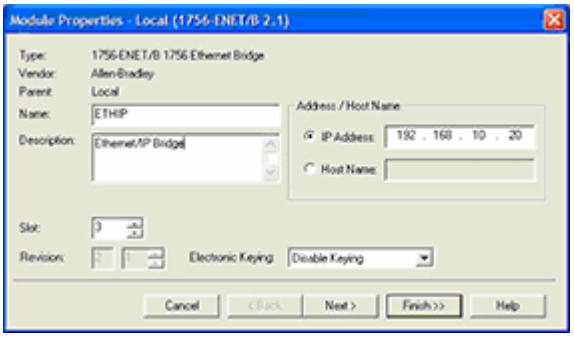


> The window [Module Properties] appears (→ screenshot).

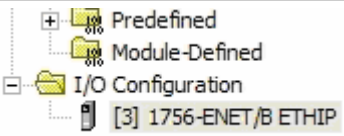
► Enter the following data:

Term	Explanation
Name	Name for the module, here: "ETHIP"
Description	Optional description
Address	IP address of the module
Slot	position of the Ethernet module in the rack

► Finish making entries with [Finish >>]

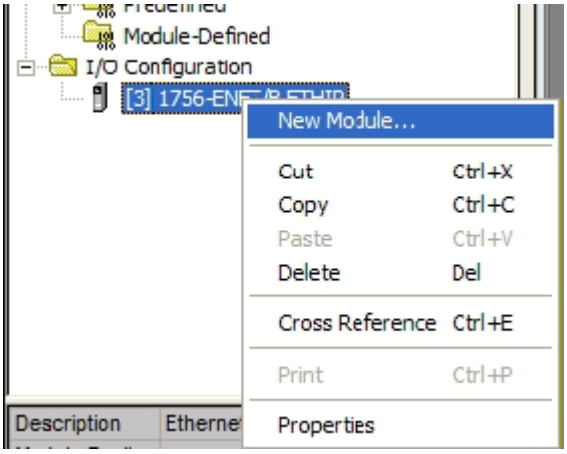
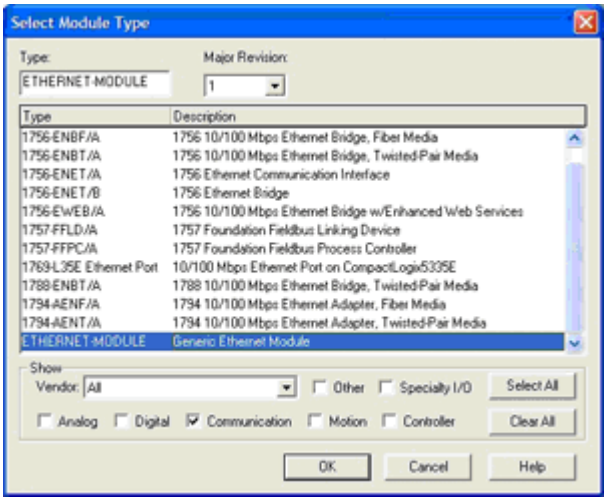
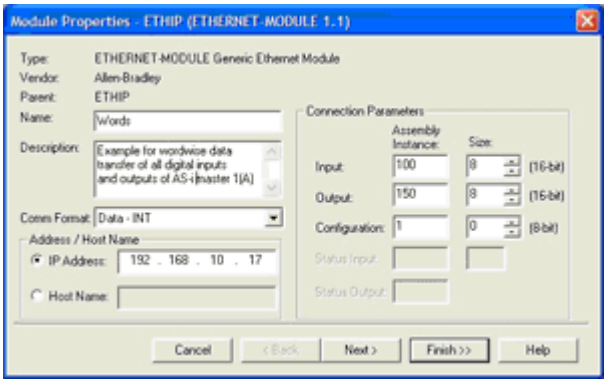


> Result display



4.4.4 Step 4: Insertion of a new module

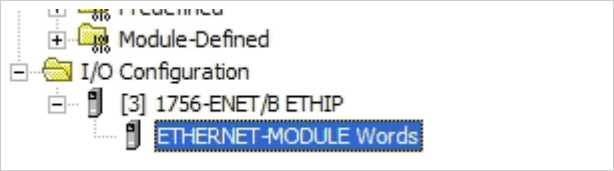
Example: Another module is to be added to the module "1756-ENET/B ETHIP".

► Right click on the directory "1756-ENET/B ETHIP" below "I/O Configuration". ► Select [New Module...] in the context menu (→ screenshot).	
> The dialogue "Select Module Type" appears (→ screenshot). ► Select the generic Ethernet module "ETHERNET MODULE". ► Confirm with [OK].	
> The window "Module Properties" appears (→ screenshot). ► Enter the data according to the following table:	

Term	Explanation
Name	Name for the connected controllerE. The name is also used for the marking of variables in the RSLogix project.
Description	Optional description
Comm Format	Select communication format from the list. Recommended: "Data-INT", to have direct access to all data in the RSLogix project which are exchanged with the controllerE. **IMPORTANT:** Subsequent changes to this setting are only possible by deleting the module and recreating it.
Address / Host Name	Enter the IP address of the controllerE
Connection Parameters	For controllerE AC13x7: • [Input Assembly Instance] = 100, • [Output Assembly Instance] = 150, • [Configuration Assembly Instance] is not supported, however, the value must be > 0. • [Configuration Size] = 0, nonetheless a data field of 400 bytes is created in the [Controller Tags]. • For [Input Size] max. 500 bytes = 250 words are permissible • For [Output Size] max. 496 bytes = 248 words are permissible

► Finish making entries with [Finish >>]

> Result display

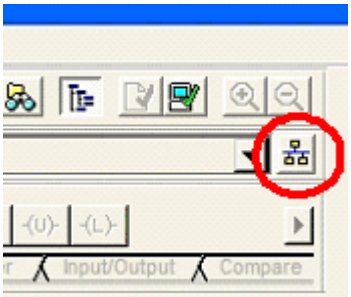
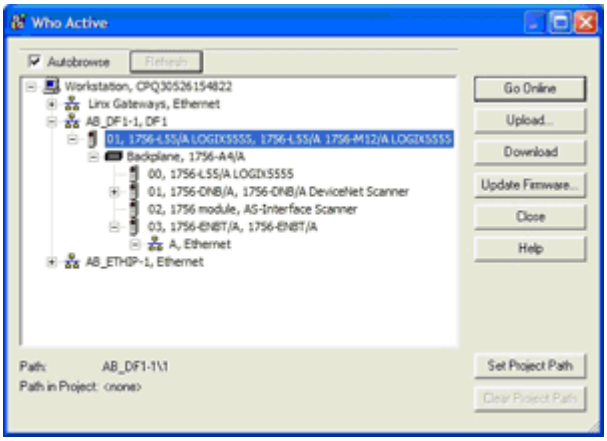


4.4.5 Step 5: Connect the configuration PC to the ControlLogix CPU

Here, select the communication path for your application.

If the communication path is already set:

► Skip step 5 and continue with Step 6 (→ page 24).

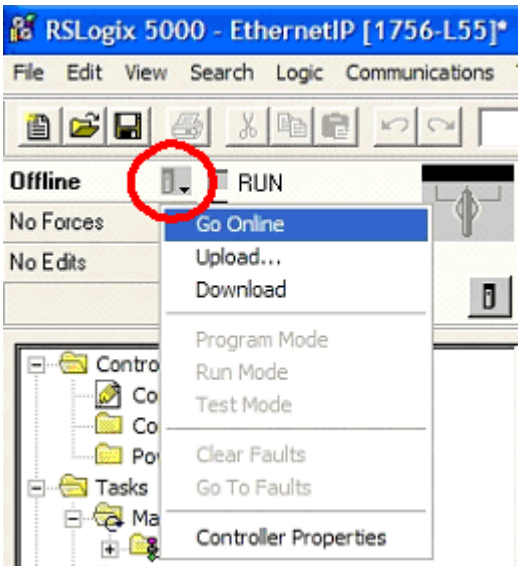
► Click on the symbol [Who Active]:	
> The window [Who Active] appears (→ screenshot). ► Select the right access path. ► Adopt the setting with [Set Project Path]. ► Close the dialogue with [Close].	

4.4.6 Step 6: Download of the created configuration to the ControlLogix PLC

► Click on the symbol [Controller Status] (→ screenshot).

> The selection menu appears.

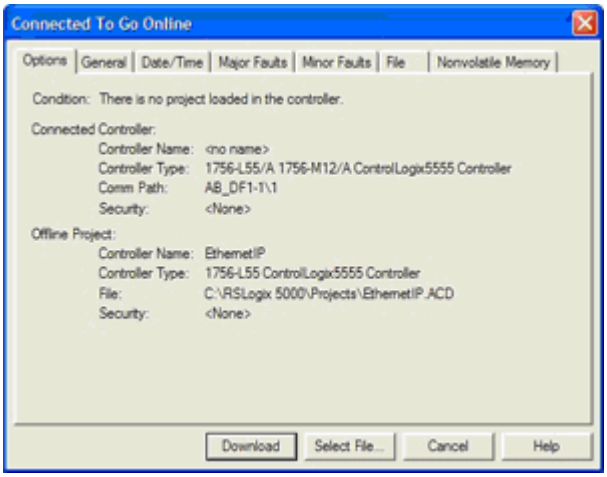
► Select [Go Online].



> The window [Connected To Go Online] appears (→ screenshot).

► Click on [Download].

If the key switch of the CPU is in the position "Prog" or "REM": → following screenshot.

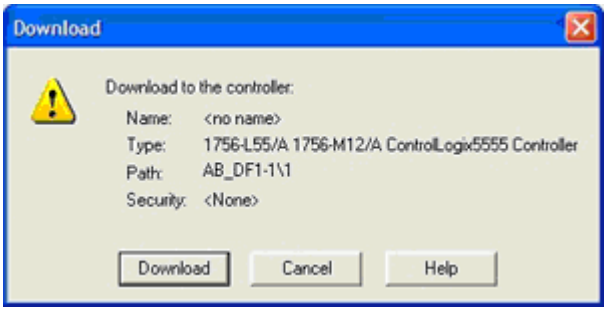


> The window [Download] appears (→ screenshot).

► Click on [Download].

> The download is running.

► Set the CPU to the mode "Run".



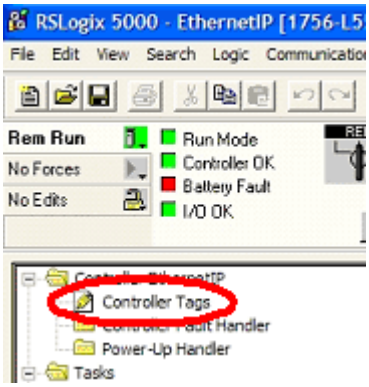
4.4.7 Step 7: Check the Ethernet connection

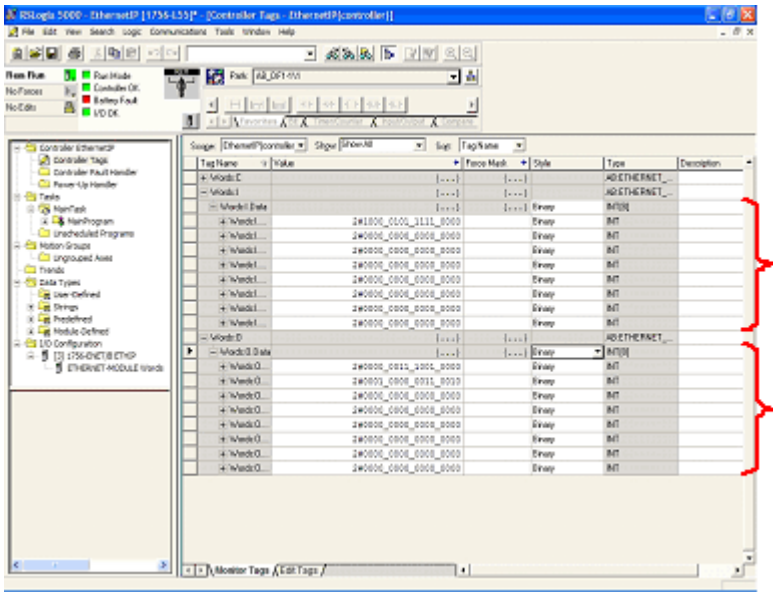
Can the connection between ControlLogix PLC and the controllerE be established?

> No message appears in the status line [Module Fault]: ⇒ connection is ok	
> A message appears in the status line [Module Fault]: ⇒ connection is not ok	

4.4.8 Step 8: check data exchange

Can ControlLogix PLC and controllerE data be exchanged?

► Double click on [Controller Tags] (→ screenshot). > Display of the data exchange (→ screenshot below).	



Input signals from the controllerE

Output signals to the controllerE

4.5 Connection to a Schneider PL7 PLC via Modbus/TCP

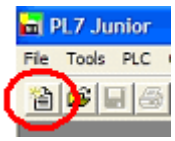
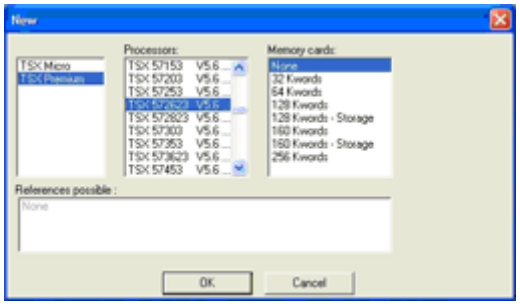
4.5.1 Step 1: Start the software

► Start the PL7 software on your computer!

4.5.2 Step 2: Create a new project

If there already is a project available to which a Modbus/TCP connection is to be added:

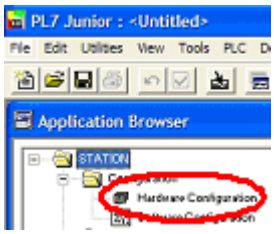
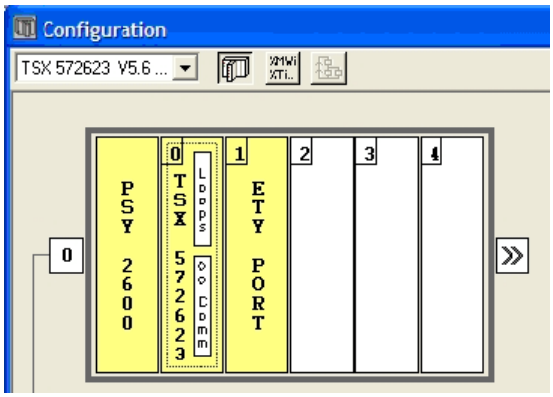
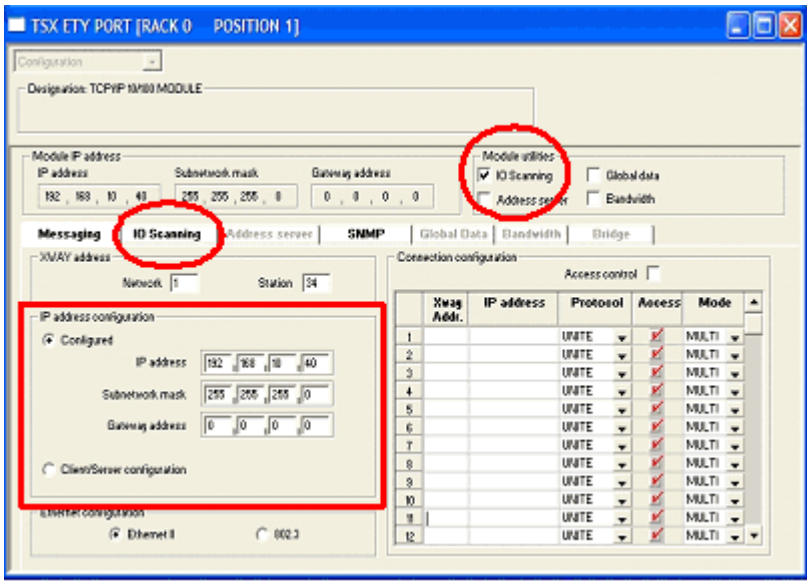
► Skip step 2 and continue with Step 3 (→ page 27).

► Create a new project: Click on the symbol [New File] (→ screenshot) or: Menu [File] > [New...].	
> The window [New] appears. ► Select CPU type, here: TSX Premium TSX 572623 V5.6 (→ screenshot). ► Confirm the selection with [OK].	

> The new project is created.

> The window [Application Browser] appears: (→ Step 3, → page 27)

4.5.3 Step 3: Configure the Ethernet connection

► Double click on [Hardware Configuration] (→ screenshot).	
> The window [Configuration] is opened. ► Double click on the requested Ethernet connection, here: module "ETY PORT".	
> The configuration window for this module is opened	

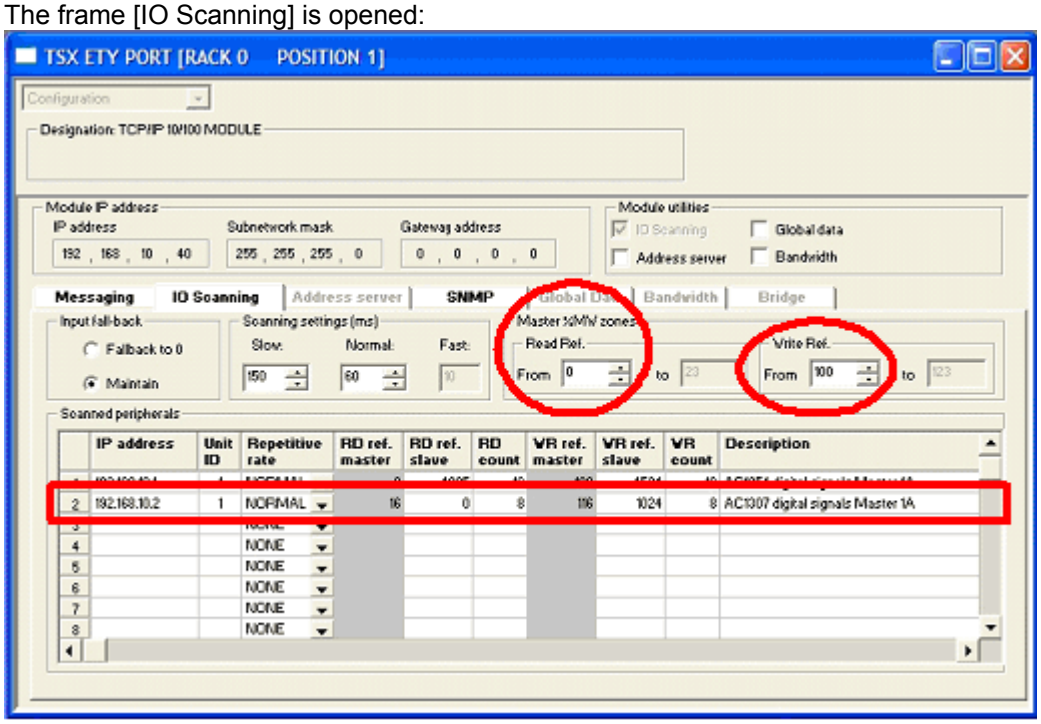
► Enter the IP address of the module and the subnet mask in the frame [IP address configuration].

► Activate the option [IO Scanning] in the frame [Module utilities].

> Access to the tab [IO Scanning] is enabled.

► Click on the tab [IO Scanning].

> The frame [IO Scanning] is opened:

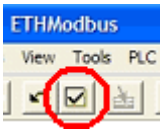


In this example two configured connections are shown:
- to a controllerE AC1327 (item 2, marked) and
- to a controllerE AC1354 (item 1).

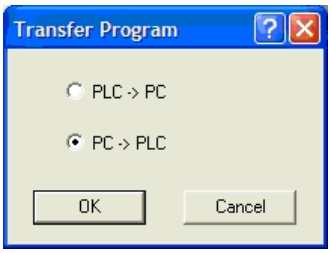
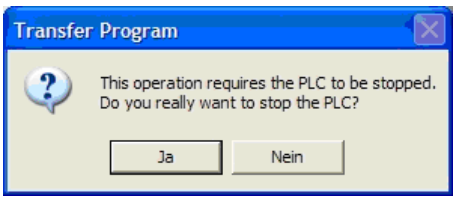
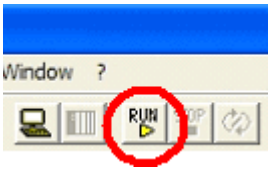
► Declare only the connection to the controllerE AC1327 in your configuration, using item 1 of the table.

Frame / column	Description
Input fall-back	Here, the behaviour of the input signals when they are no longer updated is defined, e.g. by "PLC in stop" or by "interrupted fieldbus connection". Fallback to 0 – The input signals are reset. Maintain – The input signals maintain their last status.
Scanning Settings	Here, the scanning intervals of the 3 given tasks are defined. The indications are in milliseconds. Fast – The scanning interval is fixed to 10 ms. Normal – The scanning interval can be changed in the range of 30...240 ms in increments of 15 ms. Slow – The scanning interval can be changed in the range of 150...1000 ms in increments of 50 ms. **NOTE:** The scanning interval of [Slow] must not be shorter than the scanning interval of [Normal].
Master %MW zones	Here it is defined which ranges of the %MW memory range are to be used for the I/O data. RD ref. – Read reference = the start address in the %MW range, as from which the read input data are to be stored. WR ref. – Write reference = the start address in the %MW range as from which the output data to be written are stored.

Frame / column	Description
IP address	IP addresses of the Ethernet units (here: controllerE) which are to be scanned by the Ethernet module.
Unit ID	The so-called Modbus slave address. For controllerE AC13x7: Unit ID = "1"
Repetitive rate	Indication in which scanning interval the data are to be exchanged with the said Ethernet unit (→ Scanning Settings, above). Possible setting values: None / Fast / Normal / Slow When selecting [None] the indicated memory ranges are reserved for this unit, but there is no data exchange.
RD ref slave	Modbus start address of the indicated Ethernet unit as from which the input data are to be read.
RD count	Length in words (16-bit information) which is to be read from the indicated Ethernet unit.
WR ref. slave	Modbus start address of the indicated Ethernet unit as from which the output data are to be written.
WR count	Length in words (16-bit information) which is to be written to the indicated Ethernet unit.

After all the necessary entries in the window [Configuration]: ► Click on the symbol [Confirm] to adopt the data.	

4.5.4 Step 4: Project download to Schneider PLC

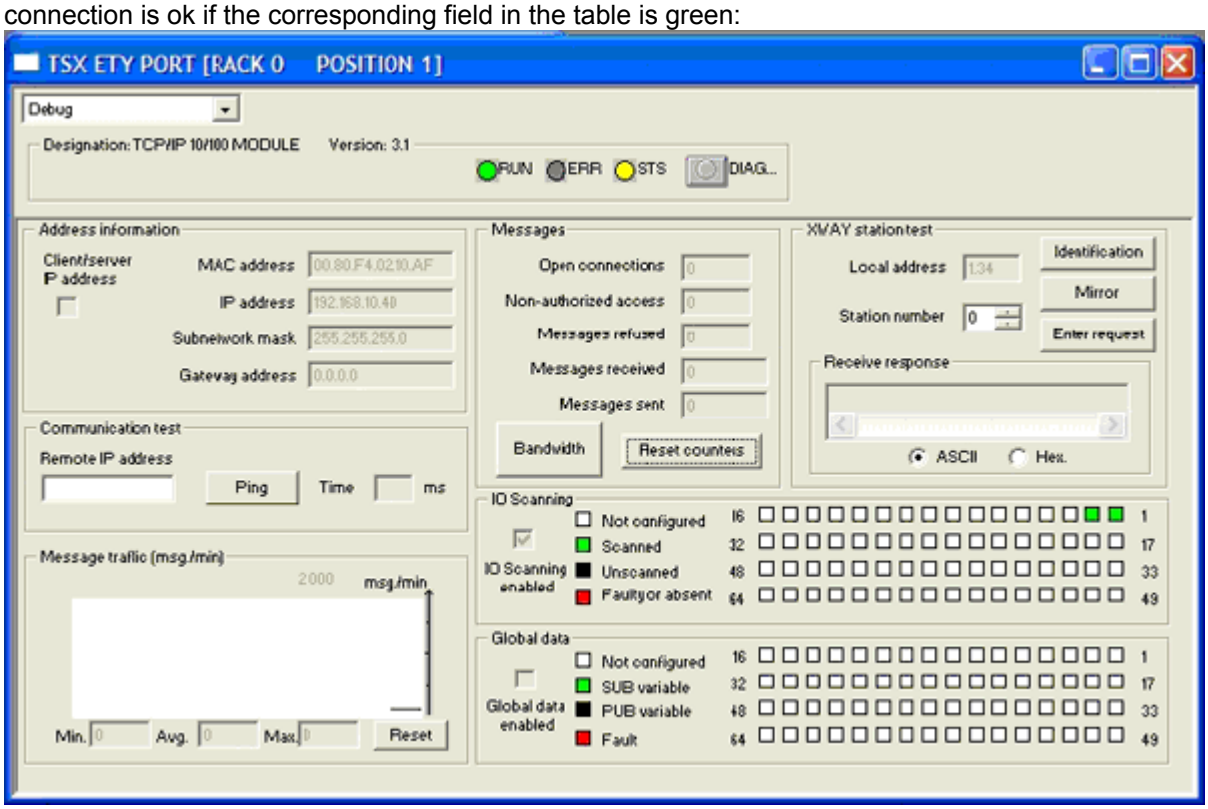
► Click on the symbol [Transfer] (→ screenshot). > The download of the project to the controller starts.	
If the connection between configuration PC and Schneider PLC is ok: > The window [Transfer Program] appears. ► Select the transfer direction [PC -> PLC]. ► Click on [OK]. ► Confirm the safety query.	
The PLC must be stopped for this process (→ screenshot). ► Confirm with [Yes].	
► Click on the symbol [Connect] (→ screenshot). > The PL7 software changes to the online mode.	
► Click on the symbol [Run] (→ screenshot). ► Confirm the safety query. > The PLC switches the mode from [Stop] to [Run].	

4.5.5 Step 5: Check the established connection

► Open the configuration window (→ Step 2, → page 26).

► Change the entry in the selection list field from [Configuration] to [Debug].

> The status of the defined connections is displayed in the box [IO Scanning] (→ screenshot). The connection is ok if the corresponding field in the table is green:



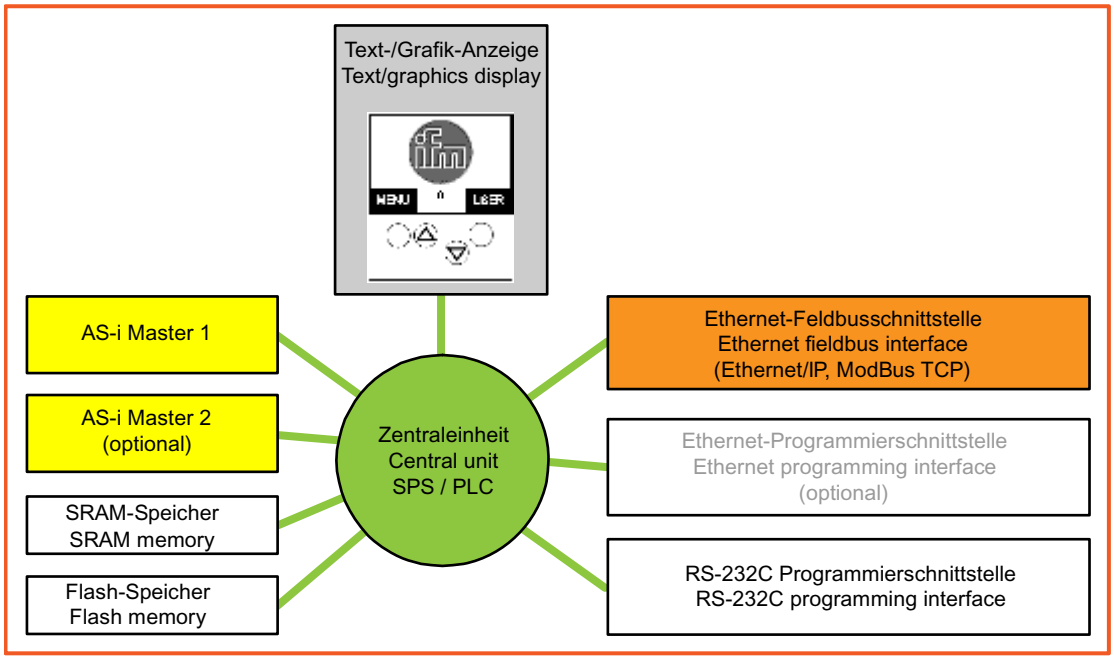
5 Function

Contents

Basic functions → separate basic instructions of the device manual

5.1 Data management

The controllerE consists of different units:



This manual exclusively describes the following subject:

- With the optional **Ethernet fieldbus interface**, (10/100 MBd, twisted pair, Ethernet/IP, Modbus TCP) the unit can be connected to other control systems.

5.2 The Ethernet fieldbus interface

The AS-i Ethernet controllerE contains a 10Mbit/100Mbit Ethernet fieldbus interface. Connection to Ethernet is made via a standard RJ45 Ethernet connection.

The data exchange between Ethernet fieldbus interface and the PLC function in the controllerE is carried out via a transfer memory (dual-ported RAM or short DPRAM) which contains maximum 512 bytes of input and output data respectively.

The following communication options are supported:

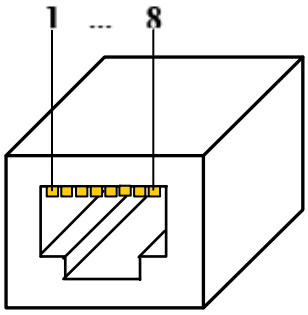
- Fieldbus protocol Modbus TCP (server),
- Fieldbus protocol Ethernet/IP (server),
- HTTP server,
- FTP Server,
- Telnet server,
- E-mail client (SMTP),
- IP access control

The following chapters explain the configuration steps for integration of the controllerE in an Ethernet network (protocols Ethernet/IP and Modbus TCP), sending of e-mails with the controllerE as well as use of the controllerE as web server in an intranet.

5.2.1 Connection of the hardware

The controllerE units AC1327 und AC1337 are equipped with RJ45 sockets for the connection of the units to the Ethernet.

Wiring diagram:

contact	signal	RJ45 socket
1	TD +	
2	TD –	
3	RD +	
4	not used	
5	not used	
6	RD –	
7	not used	
8	not used	

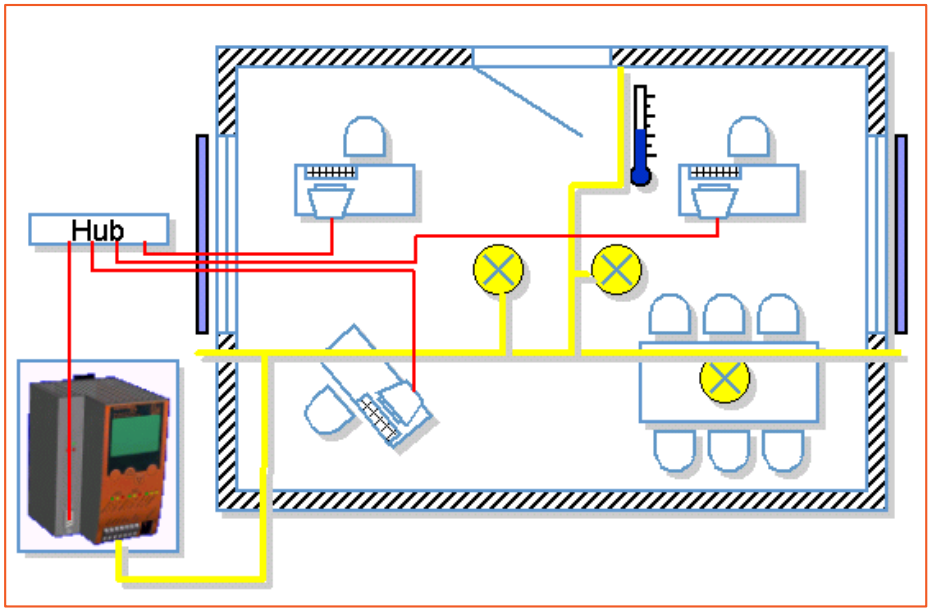
As connection cable, common twisted pair Ethernet cables can be used. A cable of the category 5 or higher is required for a 100 Mbit Ethernet interface. The maximum Ethernet cable length between the controllerE and the next connected unit (e.g. PC, hub, switch or router) is max. 100 m.

To connect the controllerE to a hub, switch or router, so-called patch cables must be used.

In case of a direct connection of two end units – called point-to-point connection – a connection cable with crossed data wires – a so-called cross-over patch cable – must be used in the normal case. If one of the two units has an Ethernet interface with automatic detection of the connection, connection can also be made using a standard patch cable.

Sample application with patch cables

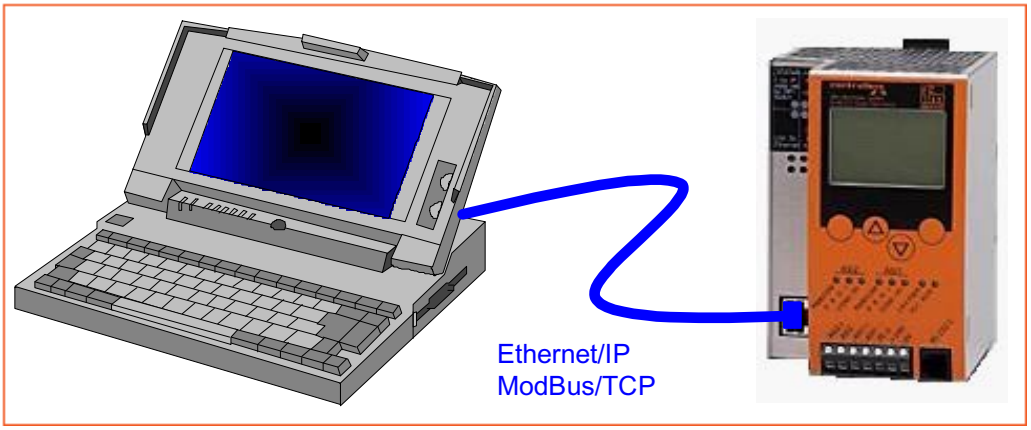
ControllerE as control unit and data server of light and air-conditioning technology in a network with 3 PC workstations:



red wires: Ethernet/IP patch cable
yellow wires: AS-Interface cable

Sample application with crossover cable

ControllerE as data server for visualisation on a PC implemented by means of an OPC server software and Modbus TCP with an Ethernet point-to-point connection.



If the shown PC has an Ethernet interface with automatic detection of the connection, connection can also be made using a standard patch cable.

5.2.2 Protocols supported by the Ethernet fieldbus interface

The Ethernet fieldbus interface enables the connection of the actuator-sensor interface (AS-i) to a higher-level fieldbus system for the bidirectional exchange of data. The interface supports standard protocols such as:

- HTTP (**H**yper**T**ext **T**ransfer **P**rotocol),
- FTP (**F**ile **T**ransfer **P**rotocol),
- SMTP (**S**imple **M**ail **T**ransfer **P**rotocol),
- Telnet (**Tel**etype**Net**work)

as well as special protocols for industrial communication, such as:

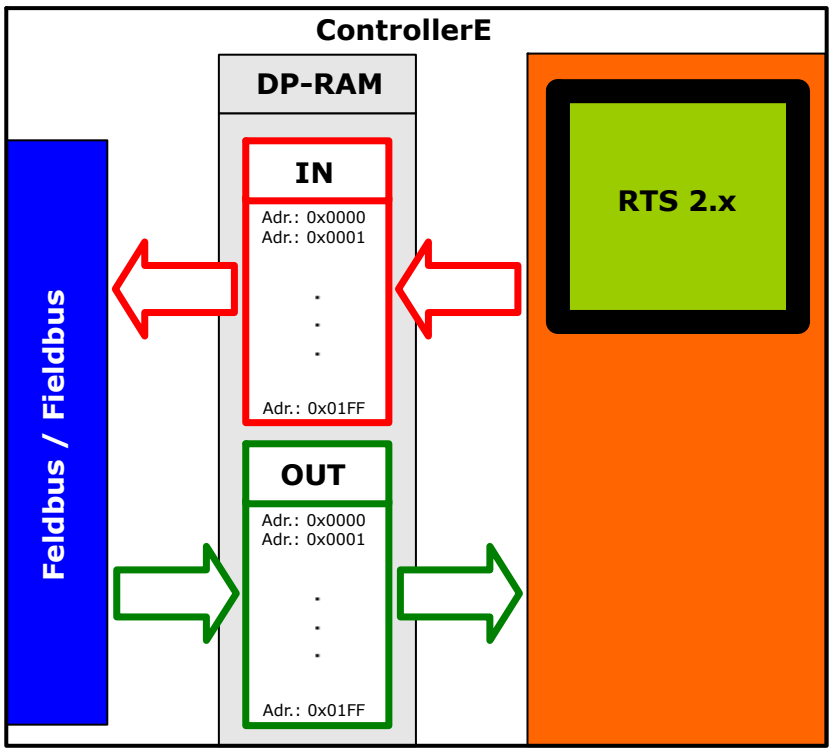
- Ethernet/IP (Ethernet **I**ndustrial **P**rotocol), only server functionality according to Ethernet/IP specification group 2 and 3,
- Modbus TCP (Modbus **T**ransmission **C**ontrol **P**rotocol), only server functionality according to Modbus TCP specification V1.0.

5.2.3 The dual-ported RAM

In order to understand the settings of the fieldbus interface it is important to understand the function of the dual-ported RAM. The dual-ported RAM, in the following called DP-RAM, is a memory range which constitutes the interface between the controllerE data and the data of the fieldbus interface. The DP-RAM consists of two different ranges:

- the so-called **IN** range which provides data from the controllerE to the fieldbus interface (controllerE output data),
- the so-called **OUT** range which provides data from the fieldbus interface to the controllerE (controllerE input data).

The following figure is supposed to illustrate the correlations of the data flow:



Both ranges have a size of 512 bytes respectively. Direct access to the addresses 0...511 (0x0000 to 0x01FF) in the IN range is always possible via the fieldbus interface. Access to addresses of the OUT range via the fieldbus interface always depends on the used type of access. The following table provides further information:

Type of access	Start address of the OUT range
Host connection via Modbus TCP	1024
HTTP (HTML page with Javascript applet "ModbusTCPClient.class")	512
SMPT client	0

5.3 The fieldbus modules

As with all controllerE units with fieldbus interface, the information to be exchanged is subdivided into logical blocks: the so-called fieldbus modules - in the following called modules. These modules often have a variable size (data length). The contents, i.e. the data, of the modules depend on the type of information to be transmitted. The modules can be set, activated / deactivated in the user menu [Fieldbus Setup].

When activating modules with controllerE output data (data from the controllerE to the fieldbus interface), these data are consistently copied in their set lengths and in the sequence of the activated module numbers into the IN range of the DP-RAM.

The activation of modules with controllerE input data (data from the fieldbus interface to the controllerE) specifies how the data of the DP-RAM OUT range are to be interpreted by the controllere. Here, the sequence of the activated module numbers and the set length are decisive again.

The Ethernet controllerE units provide 19 modules. The following table gives a quick overview of the modules and the setting options.

Module	Direction of data	Possible settings	information about the setting values	
Module 1: digital input master 1(A)	C ⇒ F	0…16	0	deactivated
Module 2: digital output master 1(A)	C ⇐ F		1…16	number of bytes
Module 3: digital input master 2(A)	C ⇒ F			
Module 4: digital output master 2(A)	C ⇐ F			
Module 5: digital input master 1(B)	C ⇒ F			
Module 6: digital output master 1(B)	C ⇐ F			
Module 7: digital input master 2(B)	C ⇒ F			
Module 8: digital output master 2(B)	C ⇐ F			
Module 9: analogue multiplexed input	C ⇔ F	0 / 1	0	deactivated
Module 10: analogue multiplexed output	C ⇔ F		1	activated
Module 11: fieldbus data command channel	C ⇔ F			
Module 12: fieldbus data PLC input	C ⇐ F	0…128	0	deactivated
Module 13: fieldbus data PLC output	C ⇒ F		1…128	number of bytes
Module 14: analogue input master 1	C ⇒ F	0…15	0	deactivated
Module 15: analogue output master 1	C ⇐ F		1…15	4 words analogue data respectively
Module 16: analogue input master 2	C ⇒ F			
Module 17: analogue output master 2	C ⇐ F			
Module 18: fieldbus data diagnosis	C ⇒ F	0 / 1 / 2	0	deactivated
			1	activated for master 1
			2	activated for master 1 + 2
Module 19: host command channel	C ⇔ F	0 / 1 / 2	0	deactivated
			1	activated (5 words)
			2	activated (18 words)

C ⇒ F	Data from controllerE to fieldbus interface (controllerE output data)
C ⇐ F	Data from fieldbus interface to controllerE (controllerE input data)
C ⇔ F	Bidirectional data (controllerE output data as well as controllerE input data)

5.3.1 Module 1: digital input master 1(A)

Data content	Binary input data of the digital single or A slaves of the AS-i master 1
Direction of data	Data from the controllerE to the fieldbus interface
Module settings	Value range: 0…16 [bytes] 0 = module is deactivated 1…16 = module is activated (details → data interpretation)
Data interpretation	In each transmitted byte, the digital signals of 2 AS-i slaves can be transmitted. The position of the data in this memory range depends on the AS-i address of the corresponding slave. Therefore the value to be set is based on the highest AS-i slave address of the used digital input slaves and not on the number of used slaves. The following table shows the allocation of AS-i slave addresses to the module settings. Given that the AS-i slave address 0 is not available for cyclical data exchange, this range is used for the transmission of status information of the AS-i master.

Setting value [byte]	AS-i slave addresses							
1	0 (status master)				1			
2	2				3			
3	4				5			
4	6				7			
5	8				9			
6	10				11			
7	12				13			
8	14				15			
9	16				17			
10	18				19			
11	20				21			
12	22				23			
13	24				25			
14	26				27			
15	28				29			
16	30				31			
Bit ⇒	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Status information AS-i master			
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**
reserved	configuration error in the AS-i circuit or AS-i voltage too low	AS-i master is offline (AS-i data invalid)	peripheral fault in the AS-i circuit

Examples for module 1:

Task 1:	The digital input signals of the AS-i slaves 1...3 are to be transmitted. To which value must module 1 be set at least?
Solution:	The highest used AS-i slave address is 3. According to the table, the data of the AS-i slave 3 **are stored in byte 2** of the module. Therefore, module 1 must be at least set to the value **2.**
Task 2:	The digital input signals of the AS-i slaves 2, 13 and 28 are to be transmitted. To which value must module 1 be set at least? Where can the data of slave 13 be found?
Solution:	The highest used AS-i slave address is 28. According to the table, the data of the AS-i slave 28 **are stored in byte 15** of the module. Therefore, module 1 must be at least set to the value **15**. The data of slave 13 are stored in byte 7 in the bits 0...3.

5.3.2 Module 2: digital output master 1(A)

Data content	Binary output data of the digital single or A slaves of AS-i master 1
Direction of data	Data from the fieldbus interface to the controllerE
Module settings	Value range: 0…16 [bytes] 0 = module is deactivated 1…16 = module is activated (details → data interpretation)
Data interpretation	In each transmitted byte, the digital signals of 2 AS-i slaves can be transmitted. The position of the data in this memory range depends on the AS-i address of the corresponding slave. Therefore the value to be set is based on the highest AS-i slave address of the used digital output slaves and not on the number of used slaves. The following table shows the allocation of AS-i slave addresses to the module settings. Given that the AS-i slave address 0 is not available for cyclical data exchange, this range is used for the transmission of control bits which are valid for both AS-i masters.

Setting value [byte]	AS-i slave addresses							
1	0 (control bits)				1			
2	2				3			
3	4				5			
4	6				7			
5	8				9			
6	10				11			
7	12				13			
8	14				15			
9	16				17			
10	18				19			
11	20				21			
12	22				23			
13	24				25			
14	26				27			
15	28				29			
16	30				31			
Bit ⇒	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Control bits AS-i master 1 + 2			
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**
reserved	reserved	reset of the stored diagnostic data	transfer of the stored diagnostic data

Examples for module 2:

Task 1:	The digital output signals of the AS-i slaves 1 and 2 are to be transmitted. To which value must module 2 be set ?
Solution:	The highest used AS-i slave address is 2. According to the table, the data of the AS-i slave 2 are stored **in byte 2** of the module. Therefore, module 2 must be at least set to the value **2**.
Task 2:	The digital output signals of the AS-i slaves 5, 17 and 30 are to be transmitted. To which value must module 2 be set ?
Solution:	The highest used AS-i slave address is 30. According to the table, the data of the AS-i slave 30 **are stored in byte 16** of the module. Therefore, module 2 must be set to the value **16**.

5.3.3 Module 3: digital input master 2(A)

Data content	Binary input data of the digital single or A slave of AS-i master 2
Direction of data	Data from the controllerE to the fieldbus interface
Module settings	Value range: 0…16 [bytes] 0 = module is deactivated 1…16 = module is activated (details → data interpretation)
Data interpretation	→ Module 1: digital input master 1(A) (→ page 38)

5.3.4 Module 4: digital output master 2(A)

Data content	Binary output data of the digital single or A slaves of AS-i master 2
Direction of data	Data from the fieldbus interface to the controllerE
Module settings	Value range: 0…16 [bytes] 0 = module is deactivated 1…16 = module is activated (details → data interpretation)
Data interpretation	→ Module 2: digital output master 1(A) (→ page 40), however the control information in the range of the AS-i slave address 0 is not used in module 4. Switching to the stored diagnostic data and reset of the stored diagnostic data must therefore also be carried out via the control bits in module 2 for AS-i master 2.

5.3.5 Module 5: digital input master 1(B)

Data content	Binary input data of the digital B slaves of AS-i master 1.
Direction of data	Data from the controllerE to the fieldbus interface
Module settings	Value range: 0…16 [bytes] 0 = module is deactivated 1…16 = module is activated (details → data interpretation)
Data interpretation	In each transmitted byte, the digital signals of 2 AS-i slaves can be transmitted. The position of the data in this memory range depends on the AS-i address of the corresponding slave. Therefore the value to be set is based on the highest AS-i slave address of the used digital input slaves and not on the number of used slaves. The following table shows the allocation of AS-i slave addresses to the module settings. The data range of the AS-i slave address 0 is not used.
Examples	→ Module 1: digital input master 1(A) (→ page 38)

Setting value [byte]	AS-i slave addresses							
1	0				1			
2	2				3			
3	4				5			
4	6				7			
5	8				9			
6	10				11			
7	12				13			
8	14				15			
9	16				17			
10	18				19			
11	20				21			
12	22				23			
13	24				25			
14	26				27			
15	28				29			
16	30				31			
Bit ⇒	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

5.3.6 Module 6: digital output master 1(B)

Data content	Binary output data of the digital B slaves of AS-i master 1
Direction of data	Data from the fieldbus interface to the controllerE
Module settings	Value range: 0…16 [bytes] 0 = module is deactivated 1…16 = module is activated (details → data interpretation)
Data interpretation	In each transmitted byte, the digital signals of 2 AS-i slaves can be transmitted. The position of the data in this memory range depends on the AS-i address of the corresponding slave. Therefore the value to be set is based on the highest AS-i slave address of the used digital output slaves and not on the number of used slaves. The following table shows the allocation of AS-i slave addresses to the module settings. The data range of the AS-i slave address 0 is not used.

Setting value [byte]	AS-i slave addresses							
1	0				1			
2	2				3			
3	4				5			
4	6				7			
5	8				9			
6	10				11			
7	12				13			
8	14				15			
9	16				17			
10	18				19			
11	20				21			
12	22				23			
13	24				25			
14	26				27			
15	28				29			
16	30				31			
Bit ⇒	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Examples	→ Module 2: digital output master 1(A) (→ page 40)

5.3.7 Module 7: digital input master 2(B)

Data content	Binary input data of the digital B slaves of AS-i master 2
Direction of data	Data from the controllerE to the fieldbus interface
Module settings	Value range: 0…16 [bytes] 0 = module is deactivated 1…16 = module is activated (details → data interpretation)
Data interpretation	→ Module 5: digital input master 1(B) (→ page 42)

5.3.8 Module 8: digital output master 2(B)

Data content	Binary output data of the digital B slaves of AS-i master 2
Direction of data	Data from the fieldbus interface to the controllerE
Module settings	Value range: 0…16 [bytes] 0 = module is deactivated 1…16 = module is activated (details → data interpretation)
Data interpretation	→ Module 6: digital output master 1(B) (On page 43)

5.3.9 Additional notes on the modules 1...8

If the transmission of the data between controllerE and the host application is carried out word-by-word, the data are shown as follows *):

Host data	**Data of the AS-i slave addresses ...**															
Word $_{n}$	(Slave 0)				Slave 1				Slave 2				Slave 3			
Word $_{n+1}$	Slave 4				Slave 5				Slave 6				Slave 7			
Word $_{n+2}$	Slave 8				Slave 9				Slave 10				Slave 11			
Word $_{n+3}$	Slave 12				Slave 13				Slave 14				Slave 15			
Word $_{n+4}$	Slave 16				Slave 17				Slave 18				Slave 19			
Word $_{n+5}$	Slave 20				Slave 21				Slave 22				Slave 23			
Word $_{n+6}$	Slave 24				Slave 25				Slave 26				Slave 27			
Word $_{n+7}$	Slave 28				Slave 29				Slave 30				Slave 31			
Bit ⇒	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

*) correct setting of the byte order in the configuration file `ifm.cfg` provided
→ chapter The general configuration file ifm.cfg (→ page 64)

It is recommended not to set the setting values of the modules 1...8 to uneven values in case of word-by-word data transfer between the controllerE and the host application, because this might result in byte offsets in following modules (→ following example):

Example:

setting value module 1 = 3
→ 3 bytes digital input data of AS-i master 1

setting value module 14 = 1
→ 4 bytes digital input data of AS-i master 1

This results in the following division of data:

bad example:																
Host data	**Data of the AS-i slave addresses ...**															
Word $_{n}$	(Slave 0)				Slave 1				Slave 2				Slave 3			
Word $_{n+1}$	Slave 4				Slave 5				analogue value 1 (high byte)							
Word $_{n+2}$	analogue value 1 (low byte)								analogue value 2 (high byte)							
Word $_{n+3}$	analogue value 2 (low byte)								analogue value 3 (high byte)							
Word $_{n+4}$	analogue value 3 (low byte)								analogue value 4 (high byte)							
Word $_{n+5}$	analogue value 4 (low byte)															
Bit ⇒	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

As you can see in the table above, the uneven setting value of module 1 caused the analogue values to be "torn apart".

If the setting value of module 1 is changed from "3" to "4", you are given a more clear data view (→ table below). Now, direct access to the analogue data in the host application is possible:

Good example:																
Host data	**Data of the AS-i slave addresses ...**															
Word $_{n}$	(Slave 0)				Slave 1				Slave 2				Slave 3			
Word $_{n+1}$	Slave 4				Slave 5				Slave 6				Slave 7			
Word $_{n+2}$	analogue value 1															
Word $_{n+3}$	analogue value 2															
Word $_{n+4}$	analogue value 3															
Word $_{n+5}$	analogue value 4															
Bit ⇒	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

5.3.10 Module 9: analogue multiplexed input

Data content	Analogue input data of the slaves of the AS-i masters 1 + 2
Note	The data of analogue output slaves with the following AS-i slave addresses can be directly written via the modules 14 and 16: • 1...15 (setting 4 channels per slave), • 1…30 (setting 2 channels per slave), • 1...31 (setting 1 channel per slave). So, module 9 only has to be used if the data cannot directly be written via the modules 14 or 16. → chapter Module 14: analogue input master 1 (→ page 52) → chapter Module 16: analogue input master 2 (→ page 56)
Direction of data	Bidirectional (2 words = 4 bytes in both directions)
Module settings	Value range: 0 and 1 0 = module is deactivated 1 = module is activated (details see data interpretation)
Data interpretation	Using module 9, analogue input data of an AS-i slave with any AS-i address can be retrieved. The information which channel of which AS-i slave on which master is to be read must be given to the controllerE via the fieldbus interface. The controllerE replies to such a request with a copy of the request data and the corresponding analogue value. As a result, only one specific analogue value can be transmitted at a time by module 9. This process is called multiplexing. **Syntax** **Request of host to controllerE** 2 words from the fieldbus interface to the controllerE Word 1:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MM		ST	SLA					0	0	0	0	0	0	CC	

MM	master number (1 or 2)
ST	0 = single slave or A slave 1 = B slave (addition of 20_{hex} or 32_{dec} to the slave address)
SLA	5 bit slave number (1…31)
CC	channel number (0...3) corresponds to the effective channel designations 1...4 (labelling on the unit)

Word 2: not used

Response from controllerE to host

2 words from the controllerE to the fieldbus interface

Word 1: Copy of word 1 of the request

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MM		ST	SLA					E	E	E	E	0	0	CC	

E_4 = the selected channel is invalid (NOT valid flag),

E_5 = channel overflow (overflow flag),

E_6 = reserved,

E_7 = data exchange error with the slave (NOT transfer valid flag).

Word 2: Analogue value (integer)

Example for module 9:

Task:	Channel 2 (according to the labelling on the unit) of the analogue input slave with the AS-i address 21 on master 2 is to be read.
Solution:	as follows:

Request of host to controllerE

Word 1:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	1	Value
	Master 2		↓	Slave 21											channel 2		Meaning
			single slave														

Word 2: not used

Response from controllerE to host:

Word 1: Copy of word 1 of the request

Word 2: Analogue value (integer)

5.3.11 Module 10: analogue multiplexed output

Data content	Analogue output data of the slaves of the AS-i masters 1 + 2
Note	The data of analogue output slaves with the following AS-i slave addresses can be directly written via the modules 15 and 17: • 16…31 (setting 4 channels per slave), • 1...30 (setting 2 channels per slave), • 1...31 (setting 1 channel per slave) So, module 10 only has to be used if the data cannot directly be written via the modules 15 or 17. If an analogue output is written simultaneously via the modules 10 and 15 or 17, the modules 15 or 17 have priority. → chapter Module 15: analogue output master 1 (→ page 54) → chapter Module 17: analogue output master 2 (→ page 56)
Direction of data	Bidirectional (2 words = 4 bytes in both directions)
Module settings	Value range: 0 and 1 0 = module is deactivated 1 = module is activated (details → data interpretation)
Data interpretation	Using module 10, analogue output data of an AS-i slave with any AS-i address can be retrieved. The information which channel of which AS-i slave on which master is to be written must be given to the controllerE via the fieldbus interface, in addition to the analogue value. The controllerE replies to such a request with a copy of the request data. As a result, only one specific analogue value can be transmitted at a time by module 10. This process is called multiplexing. **Syntax:** **Request of host to controllerE** 2 words from the fieldbus interface to the controllerE Word 1:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MM		ST	SLA					0	0	0	0	0	0	CC	

MM	master number (1 or 2)
ST	0 = single or A slave 1 = B slave (addition of 20_{hex} or 32_{dec} to the slave address)
SLA	5 bit slave number (1…31)
CC	channel number (0...3) corresponds to the effective channel designations 1...4 (labelling on the unit)

Word 2: Analogue value (integer)

Response from controllerE to host

2 words from the controllerE to the fieldbus interface

Word 1: Copy of word 1 of the request:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MM		ST	SLA					E	E	E	E	0	0	CC	

E_4 = the selected channel is invalid (NOT valid flag),

E_5 = reserved,

E_6 = the output value is not ok (NOT output valid flag),

E_7 = data exchange error with the slave (NOT transfer valid flag).

Word 2: Analogue value (integer), copy of word 2 of the request

Example for module 10:

Task:	Channel 4 (according to the labelling on the unit) of the analogue output slave with the AS-i address 12 on master 1 is to be set to the value 5 000.
Solution:	as follows:

Request of host to controllerE

Word 1:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	0	1	0	0	1	1	0	0	0	0	0	0	0	0	1	1	Value
	Master 1		↓	Slave 12											channel 4		Meaning
			single slave														

Word 2: analogue value = 5000

Response from controllerE to host:

Word 1: Copy of word 1 of the request

Word 2: Copy of word 2 of the request

5.3.12 Module 11: Fieldbus data command channel

Data content	Command channel data of the AS-i masters 1 + 2
Note	For a detailed description of the handling of the fieldbus data command channel and the different commands → chapter Commands in the fieldbus data command channel (module 11) (→ page 74).
Direction of data	Bidirectional (2 words = 4 bytes in both directions)
Module settings	Value range: 0 and 1 0 = module is deactivated 1 = module is activated (details → data interpretation)
Data interpretation	The command channel gives the user the opportunity to read different data from the controllerE or access defined functions of the controllerE via the fieldbus interface. The following table provides an overview of the available commands.

Command no.	Description
1	read master flags
2	change operating mode
3	change current slave configuration
4	read projected slave configuration
5	change projected slave configuration
6	read current slave parameters
7	change projected slave parameters (default values)
8	read LAS (list of active slaves)
9	read LDS (list of detected slaves)
10	read LPF (list of slaves with peripheral fault)
11	read LPS (list of projected slaves)
12	- reserved -
13	read telegram error counter of a slave
14	read configuration error counter
15	read AS-i cycle counter
16	change current slave parameters
17	- reserved -
18	- reserved -
19	project all
20	- reserved -
21	save configuration in the flash memory
22	reset telegram error counter

Examples	Examples for the handling of the "fieldbus data command channel" → chapter Commands in the fieldbus data command channel (module 11) (→ page 74).

5.3.13 Module 12: fieldbus data PLC input

Data content	Up to 128 bytes freely definable data
Direction of data	Data from the fieldbus interface to the controllerE
Module settings	Value range: 0…128 [bytes] 0 = module is deactivated 1…128 = module is activated (details → data interpretation)
Data interpretation	Module 12 "Fieldbus data PLC input" contains the input data from the controllerE PLC's point of view, i.e. data which are for example `sent` by a higher-level PLC to the controllerE. These data can be accessed via the PLC application program of the controllerE. Access in the application program is carried out via the variables `PLC_Input[0]` to `PLC_Input[127]`.

Example for module 12:

Task:	Process data (temperature, pressure, counter values etc.) with a total length of 14 words are to be transmitted from a higher-level PLC to the controllerE. To which value must module 12 be set?
Solution:	14 words = 28 bytes. Module 12 must be set to a length of at least 28 bytes in order to transmit all data. In case of space between the different process data in the transmitted range of the higher-level PLC, this must be taken into account for the data length in addition .

5.3.14 Module 13: fieldbus data PLC output

Data content	Up to 128 bytes freely definable data
Direction of data	Data from the controllerE to the fieldbus interface
Module settings	Value range: 0…128 [bytes] 0 = module is deactivated 1…128 = module is activated (details → data interpretation)
Data interpretation	Module 13 "Fieldbus Data PLC Output" contains output data from the controllerE PLC's point of view, i.e. data transmitted by the controllerE e.g. to a higher-level PLC or a PC. These data can be accessed via the PLC application program of the controllerE. Access in the application program is carried out via the variables `PLC_Output[0]` to `PLC_Output[127]`.

Example for module 13:

Task:	Process data with a total length of 50 bytes are to be transmitted by the controllerE to a higher-level PLC. To which value must module 13 be set?
Solution:	The data length is 50 bytes. Module 13 must be set to a length of at least 50 bytes in order to transmit all data.

5.3.15 Module 14: analogue input master 1

Data content	Analogue input data of the analogue slaves to AS-i master 1
Note	With module 14 the data of the analogue input slaves on AS-i master 1 with the following AS-i slave addresses can be directly read: • 1…15 (setting 4 channels per slave), • 1…30 (setting 2 channels per slave), • 1…31 (setting 1 channel per slave) The setting how many channels per analogue slave are to be transmitted is made via the configuration file `ifm.cfg` (→ chapter The general configuration file ifm.cfg, → page 64). **IMPORTANT!** With the setting 1 or 2 channels per slave, no data are transmitted from analogue B slaves! If an analogue input channel is to be read outside the ranges indicated above, module 9 must be used for reading these data (→ chapter Module 9: analogue multiplexed input, → page 46).
Direction of data	Data from the controllerE to the fieldbus interface
Module settings	Value range: 0…15 [4 words] 0 = module is deactivated 1…15 = module is activated (details → data interpretation)
Data interpretation	Each increment of the value range stands for the reservation of 4 words analogue output data. For the setting 1 channel per slave consider the note below!

Value range	Sum of words	Word	4 channels per slave		2 channels per slave		1 channel per slave	
			AS-i addr.	channel	AS-i addr.	channel	AS-i addr.	channel
1	4	0	1	1	1	1	1	1
		1		2		2	2	1
		2		3	2	1	3	1
		3		4		2	4	1
2	8	4	2	1	3	1	5	1
		5		2		2	6	1
		6		3	4	1	7	1
		7		4		2	8	1
3	12	8	3	1	5	1	9	1
		9		2		2	10	1
		10		3	6	1	11	1
		11		4		2	12	1
4	16	12	4	1	7	1	13	1
		13		2		2	14	1
		14		3	8	1	15	1
		15		4		2	16	1
5	20	16	5	1	9	1	17	1
		17		2		2	18	1
		18		3	10	1	19	1
		19		4		2	20	1
6	24	20	6	1	11	1	21	1
		21		2		2	22	1
		22		3	12	1	23	1
		23		4		2	24	1
7	28	24	7	1	13	1	25	1
		25		2		2	26	1
		26		3	14	1	27	1
		27		4		2	28	1
8	32	28	8	1	15	1	29	1
		29		2		2	30	1
		30		3	16	1	31	1
		31		4		2	- *)	- *)
9	36	32	9	1	17	1	- *)	- *)
		33		2		2	- *)	- *)
		34		3	18	1	- *)	- *)
		35		4		2	- *)	- *)

10	40	36	10	1	19	1	- *)	- *)
		37		2		2	- *)	- *)
		38		3	20	1	- *)	- *)
		39		4		2	- *)	- *)
11	44	40	11	1	21	1	- *)	- *)
		41		2		2	- *)	- *)
		42		3	22	1	- *)	- *)
		43		4		2	- *)	- *)
12	48	44	12	1	23	1	- *)	- *)
		45		2		2	- *)	- *)
		46		3	24	1	- *)	- *)
		47		4		2	- *)	- *)
13	52	48	13	1	25	1	- *)	- *)
		49		2		2	- *)	- *)
		50		3	26	1	- *)	- *)
		51		4		2	- *)	- *)
14	56	52	14	1	27	1	- *)	- *)
		53		2		2	- *)	- *)
		54		3	28	1	- *)	- *)
		55		4		2	- *)	- *)
15	60	56	15	1	29	1	- *)	- *)
		57		2		2	- *)	- *)
		58		3	30	1	- *)	- *)
		59		4		2	- *)	- *)

*) These areas are NOT transferred to the interface. For example for a setting value of 15 and the setting 1 channel per slave not the expected 60 words are copied to the interface but maximum 31 words. Activated fieldbus modules which may follow start with their data without a gap with the following word.

Example for module 14:

Task 1:	The value 12 is specified for module 14. The setting for channels per slave in the file ifm.cfg is 2. What is the highest AS-i slave address whose data can be transmitted with this setting and how many words are transmitted in total?
Solution:	The highest AS-i slave address is 24. 48 words are transmitted. → line "value range = 12" in the table columns "2 channels per slave" or "sum of words".
Task 2:	To which minimum value must module 14 be set so that the data of the analogue input slave with the AS-i address 10 can be read (setting 4 channels per slave)? In which word in the range can the data of channel 3 of the said slave be found?
Solution:	The value to be set for module 14 is 10. The data of slave 10, channel 3 can be found in word 38 of the range.
Task 3:	Channel 3 of the AS-i slave address 17 is to be read. Which settings are required for the fieldbus modules?
Solution:	Channel 3 of the AS-i slave address 17 cannot be read directly (via module 14). In this case, the data have to be read via the module 9 (→ chapter Module 9: analogue multiplexed input, → page 46).

5.3.16 Module 15: analogue output master 1

Data content	Analogue output data of the analogue slaves to AS-i master 1
Note	With module 15 the data of the analogue input slaves on AS-i master 1 with the following AS-i slave addresses can be directly read: • 16…30 (setting 4 channels per slave), • 1…30 (setting 2 channels per slave), • 1…31 (setting 1 channel per slave) The setting how many channels per analogue slave are to be transmitted is made via the configuration file `ifm.cfg` (→ chapter The general configuration file ifm.cfg, → page 64). **IMPORTANT!** With the setting 1 or 2 channels per slave, no data are transmitted from analogue B slaves! If an analogue output channel outside the ranges indicated above is to be written, module 10 is to be used for writing these data (→ chapter Module 10: analogue multiplexed output, → page 48).
Direction of data	Data from the fieldbus interface to the controllerE
Module settings	Value range: 0…15 [4 words] 0 = module is deactivated 1…15 = module is activated (details → data interpretation)
Data interpretation	Each increment of the value range stands for the reservation of 4 words analogue output data. For the setting 1 channel per slave consider the note below!

Value range	Sum of words	Word	4 channels per slave		2 channels per slave		1 channel per slave	
			AS-i addr.	channel	AS-i addr.	channel	AS-i addr.	channel
1	4	0	16	1	1	1	1	1
		1		2		2	2	1
		2		3	2	1	3	1
		3		4		2	4	1
2	8	4	17	1	3	1	5	1
		5		2		2	6	1
		6		3	4	1	7	1
		7		4		2	8	1
3	12	8	18	1	5	1	9	1
		9		2		2	10	1
		10		3	6	1	11	1
		11		4		2	12	1
4	16	12	19	1	7	1	13	1
		13		2		2	14	1
		14		3	8	1	15	1
		15		4		2	16	1
5	20	16	20	1	9	1	17	1
		17		2		2	18	1
		18		3	10	1	19	1
		19		4		2	20	1
6	24	20	21	1	11	1	21	1
		21		2		2	22	1
		22		3	12	1	23	1
		23		4		2	24	1
7	28	24	22	1	13	1	25	1
		25		2		2	26	1
		26		3	14	1	27	1
		27		4		2	28	1
8	32	28	23	1	15	1	29	1
		29		2		2	30	1
		30		3	16	1	31	1
		31		4		2	- *)	- *)
9	36	32	24	1	17	1	- *)	- *)
		33		2		2	- *)	- *)
		34		3	18	1	- *)	- *)
		35		4		2	- *)	- *)

10	40	36	25	1	19	1	- *)	- *)
		37		2		2	- *)	- *)
		38		3	20	1	- *)	- *)
		39		4		2	- *)	- *)
11	44	40	26	1	21	1	- *)	- *)
		41		2		2	- *)	- *)
		42		3	22	1	- *)	- *)
		43		4		2	- *)	- *)
12	48	44	27	1	23	1	- *)	- *)
		45		2		2	- *)	- *)
		46		3	24	1	- *)	- *)
		47		4		2	- *)	- *)
13	52	48	28	1	25	1	- *)	- *)
		49		2		2	- *)	- *)
		50		3	26	1	- *)	- *)
		51		4		2	- *)	- *)
14	56	52	29	1	27	1	- *)	- *)
		53		2		2	- *)	- *)
		54		3	28	1	- *)	- *)
		55		4		2	- *)	- *)
15	60	56	30	1	29	1	- *)	- *)
		57		2		2	- *)	- *)
		58		3	30	1	- *)	- *)
		59		4		2	- *)	- *)

*) These areas are NOT transferred from the interface. For example for a setting value of 15 and the setting 1 channel per slave not the expected 60 words are copied from the interface but maximum 31 words. Activated fieldbus modules which may follow start with their data without a gap with the following word.

Example for module 15:

Task 1:	The value 11 is specified for module 15. The setting for channels per slave in the file ifm.cfg is 2. What is the highest AS-i slave address whose data can be transmitted with this setting and how many words are transmitted in total?
Solution:	The highest AS-i slave address is 22. 44 words are transmitted. → line "value range = 11" in the columns "2 channels per slave" and "sum of words".
Task 2:	To which minimum value must module 15 be set so that data can be written to the analogue output slave with the AS-i address 19 (setting 4 channels per slave)? In which word in the range can the data of channel 2 of the said slave be found?
Solution:	The value to be set for module 15 is 4. The data of slave 19, channel 2 can be found in word 13 of the range.
Task 3:	Channel 1 of the AS-i slave address 31 is to be written. Which settings are required for the fieldbus modules?
Solution:	Channel 1 of the AS-i slave address 31 cannot be written directly (via module 15). In this case, the data have to be read via the module 10 (→ chapter Module 10: analogue multiplexed output, → page 48).

5.3.17 Module 16: analogue input master 2

Data content	Analogue input data of the analogue slaves to AS-i master 2
Note	With module 16 the data of the analogue input slaves on AS-i master 2 with the following AS-i slave addresses can be directly read. • 1...15 (setting 4 channels per slave), • 1...30 (setting 2 channels per slave), • 1...31 (setting 1 channel per slave) The setting how many channels per analogue slave are to be transmitted is made via the configuration file `ifm.cfgd` (→ chapter The general configuration file ifm.cfg, → page 64). If an analogue input channel is to be read outside the ranges indicated above, module 9 must be used for reading these data (→ chapter Module 9: analogue multiplexed input, → page 46).
Direction of data	Data from the controllerE to the fieldbus interface
Module settings	Value range: 0…15 [4 words] 0 = module is deactivated 1…15 = module is activated (details → data interpretation)
Data interpretation	→ Module 14: analogue input master 1 (→ page 52)

5.3.18 Module 17: analogue output master 2

Data content	Analogue output data of the analogue slaves to AS-i master 2
Note	With module 17 the data of the analogue input slaves on AS-i master 2 with the following AS-i slave addresses can be directly written: • 16...30 (setting 4 channels per slave), • 1...30 (setting 2 channels per slave), • 1...31 (setting 1 channel per slave). The setting how many channels per analogue slave are to be transmitted is made via the configuration file `ifm.cfg` (→ chapter The general configuration file ifm.cfg, → page 64). If an analogue output channel outside the ranges indicated above is to be written, module 10 is to be used for writing these data (→ chapter Module 10: analogue multiplexed output, → page 48).
Direction of data	Data from the fieldbus interface to the controllerE
Module settings	Value range: 0…15 [4 words] 0 = module is deactivated 1…15 = module is activated (details → data interpretation)
Data interpretation	→ Module 15: analogue output master 1 (→ page 54)

5.3.19 Module 18: fieldbus diagnostic data

Data content	Diagnostic data of the AS-i masters 1 and 2
Note	Using the control information bit 4 (→ module 2), you can choose whether the current diagnosis or the stored diagnosis is to be transmitted. Furthermore, the stored diagnostic data can be reset using bit 5 (→ module 2) → Module 2: digital output master 1(A) (→ page 40)
Direction of data	Data from the controllerE to the fieldbus interface
Module settings	Value range: 0…2 0 = module is deactivated 1 = 13 words diagnostic data from AS-i master 1 2 = 13 words diagnostic data from AS-i masters 1 and 2 respectively
Data interpretation	General overview of the total diagnostic range

Word	Description
0	AS-i master 1: master flags
1...4	AS-i master 1: list of detected slaves (LDS)
5...8	AS-i master 1: list of configuration faults
9...12	AS-i master 1: list of peripheral faults (LPF)
13	AS-i master 2: master flags
14...17	AS-i master 2: list of detected slaves (LDS)
18...21	AS-i master 2: list of configuration faults
22...25	AS-i master 2: list of peripheral faults (LPF)

Details master flags

Bit	Name according to AS-i specification	Description
0	-	Reserved
1	Configuration_Active	AS-i master is in the projection mode
2	LDS.0	one slave with the address 0 was detected
3	AS-i_Power_Fail	The AS-i voltage is too low.
4	NOT Periphery_OK	peripheral fault
5	-	Reserved
6	NOT Config_OK	configuration error
7	-	Reserved
8...15	-	Reserved

Details LDS, configuration error, peripheral fault (LPF) → following tables

list of detected slaves:	"1" at the corresponding position of an AS-i slave means: this slave is detected.
Configuration errors:	"1" at the corresponding position of an AS-i slave means: this slave has caused a configuration error.
Peripheral fault:	"1" at the corresponding position of an AS-i slave means: this slave has caused a peripheral fault.

AS-i master 1: list of detected slaves (LDS)

Word no.	bit (AS-i slave address)															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	15(A)	14(A)	13(A)	12(A)	11(A)	10(A)	9(A)	8(A)	7(A)	6(A)	5(A)	4(A)	3(A)	2(A)	1(A)	0 *
2	31(A)	30(A)	29(A)	28(A)	27(A)	26(A)	25(A)	24(A)	23(A)	22(A)	21(A)	20(A)	19(A)	18(A)	17(A)	16(A)
3	15B	14B	13B	12B	11B	10B	9B	8B	7B	6B	5B	4B	3B	2B	1B	res
4	31B	30B	29B	28B	27B	26B	25B	24B	23B	22B	21B	20B	19B	18B	17B	16B

AS-i master 2: list of detected slaves (LDS)

Word no.	bit (AS-i slave address)															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
14	15(A)	14(A)	13(A)	12(A)	11(A)	10(A)	9(A)	8(A)	7(A)	6(A)	5(A)	4(A)	3(A)	2(A)	1(A)	0 *
15	31(A)	30(A)	29(A)	28(A)	27(A)	26(A)	25(A)	24(A)	23(A)	22(A)	21(A)	20(A)	19(A)	18(A)	17(A)	16(A)
16	15B	14B	13B	12B	11B	10B	9B	8B	7B	6B	5B	4B	3B	2B	1B	res
17	31B	30B	29B	28B	27B	26B	25B	24B	23B	22B	21B	20B	19B	18B	17B	16B

AS-i master 1: list of configuration faults

Word no.	bit (AS-i slave address)															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
5	15(A)	14(A)	13(A)	12(A)	11(A)	10(A)	9(A)	8(A)	7(A)	6(A)	5(A)	4(A)	3(A)	2(A)	1(A)	0 *
6	31(A)	30(A)	29(A)	28(A)	27(A)	26(A)	25(A)	24(A)	23(A)	22(A)	21(A)	20(A)	19(A)	18(A)	17(A)	16(A)
7	15B	14B	13B	12B	11B	10B	9B	8B	7B	6B	5B	4B	3B	2B	1B	res
8	31B	30B	29B	28B	27B	26B	25B	24B	23B	22B	21B	20B	19B	18B	17B	16B

AS-i master 2: list of configuration faults

Word no.	bit (AS-i slave address)															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
18	15(A)	14(A)	13(A)	12(A)	11(A)	10(A)	9(A)	8(A)	7(A)	6(A)	5(A)	4(A)	3(A)	2(A)	1(A)	0 *
19	31(A)	30(A)	29(A)	28(A)	27(A)	26(A)	25(A)	24(A)	23(A)	22(A)	21(A)	20(A)	19(A)	18(A)	17(A)	16(A)
20	15B	14B	13B	12B	11B	10B	9B	8B	7B	6B	5B	4B	3B	2B	1B	res
21	31B	30B	29B	28B	27B	26B	25B	24B	23B	22B	21B	20B	19B	18B	17B	16B

AS-i master 1: list of peripheral faults (LPF)

Word no.	bit (AS-i slave address)															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
9	15(A)	14(A)	13(A)	12(A)	11(A)	10(A)	9(A)	8(A)	7(A)	6(A)	5(A)	4(A)	3(A)	2(A)	1(A)	0 *
10	31(A)	30(A)	29(A)	28(A)	27(A)	26(A)	25(A)	24(A)	23(A)	22(A)	21(A)	20(A)	19(A)	18(A)	17(A)	16(A)
11	15B	14B	13B	12B	11B	10B	9B	8B	7B	6B	5B	4B	3B	2B	1B	res
12	31B	30B	29B	28B	27B	26B	25B	24B	23B	22B	21B	20B	19B	18B	17B	16B

AS-i master 2: list of peripheral faults (LPF)

Word no.	bit (AS-i slave address)															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
22	15(A)	14(A)	13(A)	12(A)	11(A)	10(A)	9(A)	8(A)	7(A)	6(A)	5(A)	4(A)	3(A)	2(A)	1(A)	0 *
23	31(A)	30(A)	29(A)	28(A)	27(A)	26(A)	25(A)	24(A)	23(A)	22(A)	21(A)	20(A)	19(A)	18(A)	17(A)	16(A)
24	15B	14B	13B	12B	11B	10B	9B	8B	7B	6B	5B	4B	3B	2B	1B	res
25	31B	30B	29B	28B	27B	26B	25B	24B	23B	22B	21B	20B	19B	18B	17B	16B

* Only for LDS and list of configuration errors, otherwise not used.

5.3.20 Module 19: Host command channel

Data content	Host command channel data of the AS-i masters 1 + 2
Note	Details → chapter The host command channel (→ page 105)
Direction of data	Bidirectional (5/18 words in both directions)
Module settings	Value range: 0…2 0 = module is deactivated 1 = 5 words 2 = 18 words
Data interpretation	The host command channel gives the user the opportunity to read different data from the controllerE and access defined functions of the controllerE. The following table provides an overview of the available commands.

Command no.	Description
0	no execution of a command
1	write parameters to a connected AS-i slave
3	adopt and save currently connected AS-i slaves in the configuration
4	change the list of projected AS-i slaves (LPS)
5	set the operating mode of the AS-i master
6	readdress a connected AS-i slave
7	set the auto addressing mode of the AS-i master
9	change the extended ID code 1 in the connected AS-i slave
10...20	adopt and save currently connected AS-i slaves in the configuration
28	deactivation of the slave reset when changing to the protected mode
31	one-time execution of the Extended safety monitor protocol in the Safety-at-Work monitor
21	read the ID string of an AS-i slave with profile S-7.4
33	read the diagnosis string of an AS-i slave with profile S-7.4
34	read the parameter string of an AS-i slave with profile S-7.4
35	write parameter string of an AS-i slave with the profile S-7.4
50	read current configuration AS-i slaves 0(A)...15(A)
51	read current configuration AS-i slaves 16(A)...31(A)
52	read current configuration AS-i slaves 0...15B
53	read current configuration AS-i slaves 16B...31B
54	read current parameters of a connected AS-i slave

	55	read current AS-i slave lists
	56	read projected configuration AS-i slaves 1(A)...15(A)
	57	read projected configuration AS-i slaves 16(A)...31(A)
	58	read projected configuration AS-i slaves 1B...15B
	59	read projected configuration AS-i slaves 16B...31B
	96	save data non-volatilely in the flash memory of the controllerE
	97	carry out various settings in the controllerE
	102	retrieve the status of the controllerE display
	105	read the device properties of the controllerE
Examples	examples for the different commands → chapter commands in the host command channel (→ page 107)	

5.4 The file system of the web server

The Ethernet interface of the controllerE has a memory area with a hierarchical directory structure. There files can be stored up to a total volume of 1.4 Mbytes and grouped in directories.

The file system has two safety levels. Depending on these safety levels different users can be given different types of access to directories and files.

The file system can be accessed via the Ethernet interface via FTP (**F**ile **T**ransfer **P**rotocol), Telnet or HTTP (**H**yper**T**ext **T**ransfer **P**rotocol).

Info

The file system differentiates between capital and small letters.
Example: `Test.TXT` is a different file than `Test.txt`

File names can have a length of max. 48 characters. The permissible length of path names including file names is 256 characters.

The file size is only limited by the memory capacity of 1.4 Mbytes.

The Ethernet interface used is an Anybus-S PCB type ABS-EIP-2 from HMS Industrial Networks GmbH. A current, detailed description of all functions and of the control via the PLC of the controllerE via the mailbox interface can be downloaded from the HMS website.

The structures and procedures that are important for interface handling are described below. Further information is given in the above-mentioned documentation. But basic knowledge of Ethernet, TCP/IP, FTP, Telnet, e-mail server and Modbus/TCP is required to use the corresponding services. There is a multitude of technical literature available in the book trade about these topics.

5.4.1 The structure of the file system of the web server

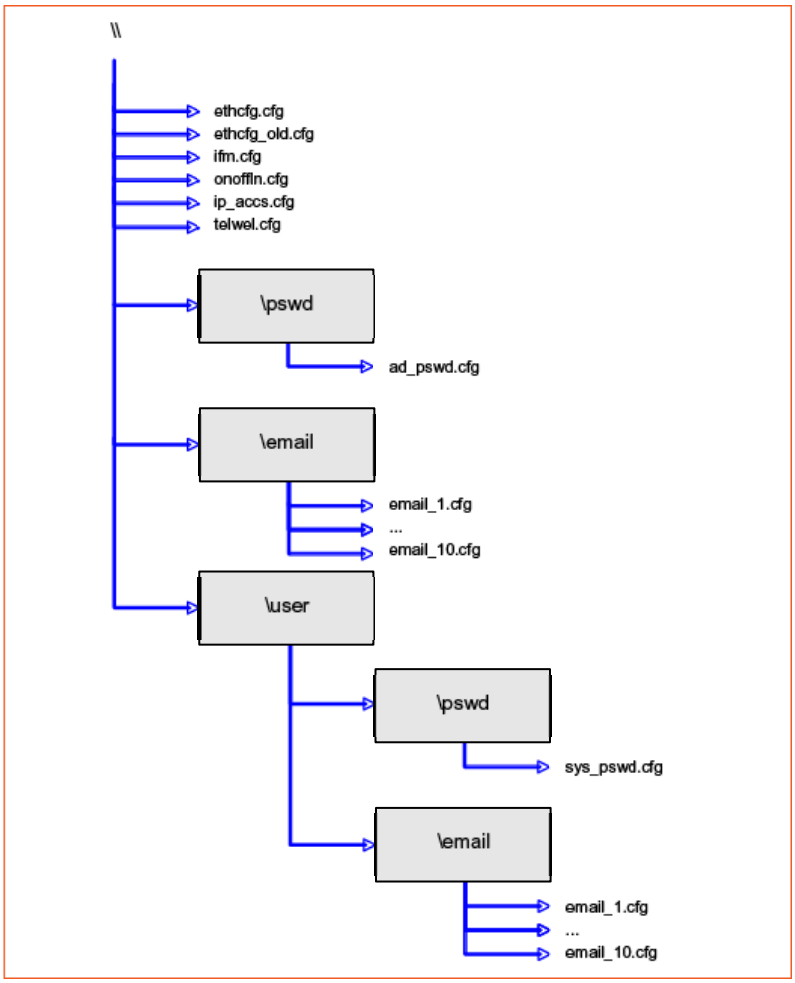
The structure of the file system shows in which directories the different configuration data must be stored so that they can be correctly identified and processed by the controllerE. On delivery of the controllerE each user has unlimited access via Telnet or FTP to all files and directories of the controllerE web server. If access is to be restricted, you can create corresponding user accounts by means of the configuration files `\pswd\ad_pswd.cfg` and `\user\pswd\sys_pswd.cfg` (→ chapter The configuration files ad_pswd.cfg and sys_pswd.cfg, → page 68).

NOTE

Danger for the data caused by unauthorised participants in the network!

► Restrict the access to the web server of the controllerE, because otherwise the entire file system will be open to all network participants!

This password protection does not refer to access to HTML sites with a browser but only to the FTP or Telnet access to the file system.



Graphics: Structure of the file system

The two directories `\pswd` are protected.

5.4.2 The configuration files

The configuration files serve for the non-volatile storage of configuration data of the controllerE. If configuration files are to be edited, they always have to be copied from the web server to the PC first.

NOTE

► Restart the controllerE after a change to a configuration file (voltage off/on).

The Ethernet configuration file ethcfg.cfg and its backup file ethcfg_old.cfg

In the file `ethcfg.cfg` the network settings of the controllerE are non-volatilely stored. The following table shows the format of this file and provides a short description of the different parameters. Changes made to the file `ethcfg.cfg` will become effective on reboot of the controllerE.

File format	Description
[IP address] `xxx.xxx.xxx.xxx`	IP address of the controllerE
[Subnet Mask] `xxx.xxx.xxx.xxx`	Subnet mask of the controllerE
[Gateway address] `xxx.xxx.xxx.xxx`	Gateway address of the standard gateway in the network. If there is no standard gateway in the network the setting 0.0.0.0 should be entered as gateway address.
[SMTP address] `xxx.xxx.xxx.xxx`	SMTP address (SMTP = **S**imple **M**ail **T**ransfer **P**rotocol) `xxx.xxx.xxx.xxx` = IP address of the SMTP server
[DHCP/BOOTP] `ON / OFF`	DHCP = **D**ynamic **H**ost **C**onfiguration **P**rotocol The allocation of the IP address is carried out automatically from an address pool of an address server in the network. `ON` = activated (= IP address retrieved from the address server) `OFF` = deactivated (locally stored IP address)
[Speed] `Auto / 100 / 10`	Ethernet baud rate `Auto` = automatic detection `100` = 100 Mbit/s fixed `10` = 10 Mbit/s fixed
[Duplex] `Auto / Full / Half`	`Auto` = automatic detection `Full` = full duplex fixed `Half` = half duplex fixed

[SMTP username]
`username`
[SMTP password]
`password`
	SMTP = **S**imple **M**ail **T**ransfer **P**rotocol
	The indications `user name` and `password` only need to be made if the defined SMTP server requires these.

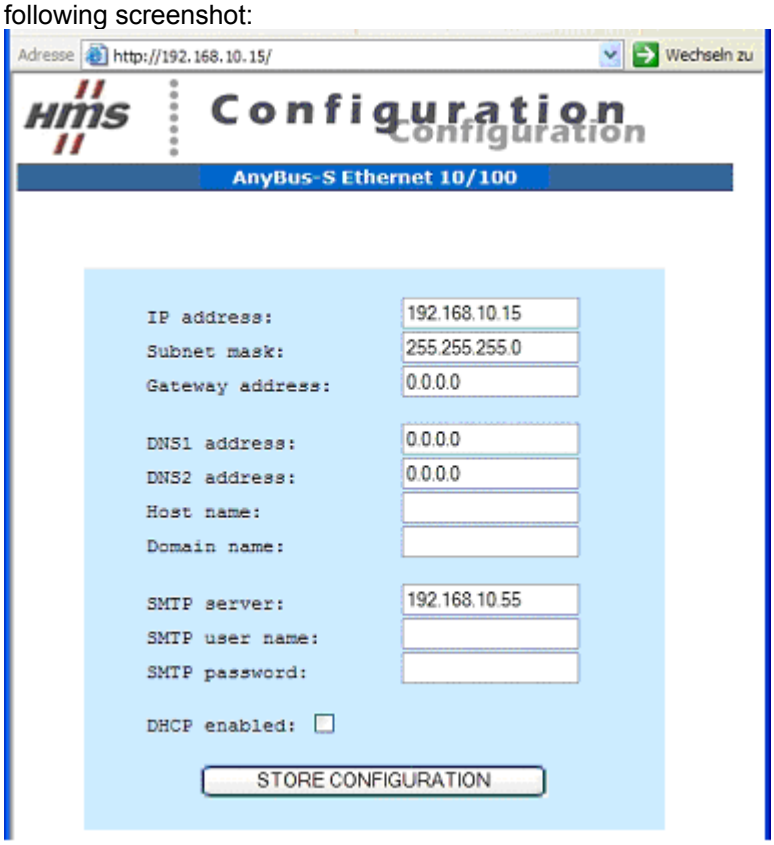
[DNS1 address] `xxx.xxx.xxx.xxx` [DNS2 address] `xxx.xxx.xxx.xxx`	DNS = **D**omain **N**ame **S**ystem The DNS is one of the most important services on the internet. The main task is the translation of names into IP addresses, i.e. answering to name enquiries with the corresponding IP address.
[Domain name] `domain`	Domain Name if the indicated Host Name is ambiguous.

File format	Description
[Host name] `hostname`	Host name

Of all entries in the configuration file `ethcfg.cfg` only the entries [IP address] and [subnet mask] can be changed via the operating menu of the controllerE. All other entries can only be configured and stored via the integrated web server of the controllerE. The configuration of the fieldbus interface can be accessed by means of a web browser.

► To do so, enter the following in the address line of the web browser:
`http://<IP-address>`
<IP address> corresponds to the IP address of the controllerE.

> After pressing the button [Enter] the browser displays the configuration mask shown in the following screenshot:



► Adjust the configuration entries according to the requirements.

► Save the new configuration by clicking on [STORE CONFIGURATION].

> The configuration file `ethcfg.cfg` is then overwritten with the new settings.

► Restart the controllerE.

> The changes made become active.

If the IP address or the subnet mask is changed via the operating menu of the controllerE the last valid Ethernet configuration data will be stored in the file `ethcfg_old.cfg`. The format of the files `ethcfg.cfg` and `ethcfg_old.cfg` is identical.

The general configuration file ifm.cfg

By means of the file `ifm.cfg` general settings regarding the operating behaviour of the controllerE can be made and stored non volatilely. Changes made to the file `ifm.cfg` will become effective on reboot of the controllerE.

File format	**Description** (* = preset value)

`[Channels per analogue input slave]`
`M1: x`
`M2: x`
	Number of input channels of an analogue slave to be evaluated. Data illustration according to the description of fieldbus module 14
	(→ chapter Module 14: analogue input master 1, → page 52).

	Number of output channels of an analogue slave to be written. Data illustration according to the description of the fieldbus module 15 (→ chapter Module 15: analogue output master 1, → page 54). `M1` = AS-i master 1 `M2` = AS-i master 2 `x` = 1 / 2 / **4***
`[clear/freeze outputs]` `0|1`	This setting defines how the outputs (analogue and digital) of the controllerE should behave in case of a detected fieldbus interruption. **0*** = clear = outputs are reset `1` = freeze = the outputs maintain their latest status
`[Byteorder]` `EIP|Modbus`	This setting defines the byte order of the data to be transmitted. **EIP*** = Ethernet/IP `Modbus` = Modbus/TCP

`[AC1309/10 mode]`
`on/off: 0`
`OS_In: 2`
`OS_Out: 2`
`AI_M1_1st: 1`
`AI_M1_Num: 15`
`AO_M1_1st: 16`
`AO_M1_Num: 15`
`AI_M2_1st: 1`
`AI_M2_Num: 15`
`AO_M2_1st: 16`
`AO_M2_Num: 15`
`OS_Diag: 310`
	By means of this setting, a compatibility mode can be activated in which the controllerE units AC1327/37 behave like the units AC1309/10. If the compatibility mode is activated, the controllerE ignores the following settings in the configuration file `ifm.cfg`:
	`[Channels per analogue input slave]`
	`[Channels per analogue output slave]`
	`[clear/freeze outputs]`
	`[Byteorder]`
	The controllerE then operates with the following settings internally:
	`[Channels per analogue input slave] = 4`
	`[Channels per analogue output slave] = 4`
	`[clear/freeze outputs] = 0`
	`[Byteorder] = Modbus`
	
	`on/off:` **0*** AC1309/10 mode deactivated
	1 AC1309/10 mode activated
	`OS_In:` offset for the input range
	`OS_Out:` offset for the output range
	`AI_M1_1st:` first analogue input slave on M1
	`AI_M1_Num:` number of analogue input slaves on M1
	`AO_M1_1st:` first analogue output slave on M1
	`AO_M1_Num:` number of analogue output slaves on M1

File format	Description (* = preset value)
	`AI_M2_1st:` first analogue input slave on M2 `AI_M2_Num:` first analogue input slave on M2 `AO_M2_1st:` first analogue output slave on M2 `AO_M2_Num:` number of analogue output slaves on M2 `OS_Diag:` offset for diagnostic data The parameters indicated under `[AC1309/10 mode]` are only effective if the AC1309/10 compatibility mode has been activated via the parameters `on/off`. More information about the different parameters (→ old device manual `GHBCone4.pdf`)

* = preset value for the file `ifm.cfg` and behaviour of the controllerE, when the file `ifm.cfg` is not available on the web server of the controllerE.

Table: Format of the file `ifm.cfg` and short description of the different parameters

NOTICE

Danger caused by wrong output signals!

If the entry [clear/freeze outputs] is set to 1, the digital as well as the analogue outputs are NOT set to "0" in case of a fault (interrupted bus communication).

► Use the function "Freeze Outputs" for targeted tests only!

NOTE

The function of the entry [clear/freeze outputs] is only ensured if an interruption of the Ethernet connection is correctly detected. So, the correct setting of the configuration file `onoffln.cfg` is absolutely necessary. (→ chapter The configuration file onoffln.cfg, → page 66)

This entire documentation, in particular the description of the fieldbus modules (→ chapter), always preassumes the correct setting of the entry [Byteorder] for the connected host PLC

Example:
If the controllerE is operated on an Ethernet/IP network, the byte order should also be set to `"EIP"`.

The configuration file onoffln.cfg

The file `onoffln.cfg` indicates how the existing Ethernet connection is to be monitored.

If the file `onoffln.cfg` is not available in the root directory of the web server, the connection is monitored as "Link": As soon as there is a connection to Ethernet, the connection to the connected host system is also regarded as "ok".

Monitoring of the connection via the file `onoffln.cfg` alone is not sufficient in order to clearly define the behaviour of AS-i outputs (digital and analogue) in case of an interruption of the connection.

► Enter in the configuration file `ifm.cfg` using the entry [clear/freeze outputs] how the output signals are to behave in case of an interrupted connection!

NOTE

Monitoring of the connection via the file `onoffln.cfg` does not have any influence on the input data from the point of view of the host application.

Here it has to be defined - provided that the host system has such a setting option - how the input information should behave in case of an interruption of the connection.

If there is no such setting option in the host system the communication must be monitored and it has to be specifically reacted to a fault.

Changes made to the file will become active after a restart of the controllerE.

File format	**Description** (* = preset value)
`[ON/OFF-line trigger]` `Link|EIP|Modbus`	Trigger `Link` = general Ethernet connection `EIP*` = Ethernet/IP connection `Modbus` = ModbusTCP connection
`[Timeout]` **) `x`	Timeout value Only used for [ON/OFF-line trigger] = `Modbus`. `x` = 1...4*…65 535 [100 ms each]
`[Commands]` **) `ALL|a[, b, c, ...]`	Modbus commands Only used for [ON/OFF-line trigger] = `Modbus`. `ALL*` = each Modbus command of a received Modbus frame received within the defined timeout period marks the connection as ok. `a[, b, c, ...]` = list of the Modbus commands of which at least one must be received within the defined timeout period for the connection to be marked as OK.
`[ON-line method]` `x`	Memory behaviour of the Ethernet interface module `x` = 1 `x` = 2*

* = preset value for the file `onoffln.cfg`.

**) These parameters are only indicated if the parameter [ON/OFF-line trigger] is set to Modbus.

Table: Format of the file `onoffln.cfg` and short description of the different parameters

NOTE

► When setting the parameter [ON-line method] = 1 in the file `onoffln.cfg` the following has to be taken into account:

If AS-i outputs have been triggered via an Ethernet host via the protocol Ethernet/IP or ModbusTCP and an interruption of this connection occurs, the triggered outputs will be reset (in case of correct setting of the configuration files `ifm.cfg` and `onoffln.cfg`). If now another access to the controllerE takes place via the same protocol, the previously triggered outputs are reactivated even if the connection to the original host is interrupted.

We therefore recommend not to change the presetting of the parameter [ON-line method] = 2!

The configuration file ip_accs.cfg

In the file `ip_accs.cfg`, the supported web services (web, FTP, Telnet, ModbusTCP, Ethernet/IP) can be approved for individual network participants or groups of participants. An asterisk (*) can be used as placeholder for groups of IP addresses.

Changes made to the file will become active after a restart of the controllerE.

File format	Description
[Web] `xxx.xxx.xxx.xxx` or `[xxx.xxx.xxx.xxx ... xxx.xxx.xxx.xxx]`	Enabled IP addresses for HTTP access `xxx.xxx.xxx.xxx` or IP address list: `[xxx.xxx.xxx.xxx ... xxx.xxx.xxx.xxx]` *)
[FTP] `xxx.xxx.xxx.xxx` or `[xxx.xxx.xxx.xxx ... xxx.xxx.xxx.xxx]`	Enabled IP addresses for FTP access `xxx.xxx.xxx.xxx` or IP address list: `[xxx.xxx.xxx.xxx ... xxx.xxx.xxx.xxx]` *)
[Telnet] `xxx.xxx.xxx.xxx` or `[xxx.xxx.xxx.xxx ... xxx.xxx.xxx.xxx]`	Enabled IP addresses for Telnet access `xxx.xxx.xxx.xxx` or IP address list: `[xxx.xxx.xxx.xxx ... xxx.xxx.xxx.xxx]` *)
[Modbus/TCP] `xxx.xxx.xxx.xxx` or `[xxx.xxx.xxx.xxx ... xxx.xxx.xxx.xxx]`	Enabled IP addresses for Modbus TCP access `xxx.xxx.xxx.xxx` or IP address list: `[xxx.xxx.xxx.xxx ... xxx.xxx.xxx.xxx]` *)
[Ethernet/IP] `xxx.xxx.xxx.xxx` or `[xxx.xxx.xxx.xxx ... xxx.xxx.xxx.xxx]`	Enabled IP addresses for Ethernet/IP access `xxx.xxx.xxx.xxx` or IP address list: `[xxx.xxx.xxx.xxx ... xxx.xxx.xxx.xxx]` *)
[All] `xxx.xxx.xxx.xxx` or `[xxx.xxx.xxx.xxx ... xxx.xxx.xxx.xxx]`	This entry is used if one or several of the previously described entries have been left out. If the entry [All] is not written, a connection is accepted by every IP address, irrespective of the setting of the other entries. `xxx.xxx.xxx.xxx` or IP address list: `[xxx.xxx.xxx.xxx ... xxx.xxx.xxx.xxx]` *)

* The total number of IP addresses to be entered in the file `ip_accs.cfg` is only limited by the available memory.

Table: Format of the file `ip_accs.cfg` and short description of the different parameters

The configuration file telwel.cfg

The configuration file `telwel.cfg` creates a defined welcome text in case of a Telnet access to the controllerE. To do so, the text can either be directly written in the file `telwel.cfg` or the entry [File path] can be used to refer to a corresponding text file on the web server.

Changes made to the file will become active after a restart of the controllerE.

File format	Description
`[File path]` `\ directory\file name.txt`	Directory path in which the text file to be displayed is stored on the web server. (entry is optional)

The configuration files ad_pswd.cfg and sys_pswd.cfg

On delivery of the controllerE each user has unlimited access via Telnet or FTP to all files and directories of the controllerE web server. If access is to be restricted, you can create corresponding user accounts by means of the files `\pswd\ad_pswd.cfg` and `\user\pswd\sys_pswd.cfg`.

NOTE

Danger for the data caused by unauthorised participants in the network!

► Restrict the access to the web server of the controllerE, because otherwise the entire file system will be open to all network participants!

This password protection does not refer to access to HTML sites with a browser but only to the FTP or Telnet access to the file system.

In general, two different types of user accounts are distinguished:

- User accounts for so-called normal operation,
- User accounts for the so-called administrator mode.

User accounts for **normal operation** are defined in the file `\user\pswd\sys_pswd.cfg` and enable access to the directory `\user` and to the corresponding subdirectories for these users. The file `sys_pswd.cfg` of course only makes sense if the administrator access has also been restricted.

User accounts for the **administrator mode** are defined in the file `\pswd\ad_pswd.cfg` and allow these users the unrestricted access to the entire file system of the controllerE.

Changes made to the files will become active after a restart of the controllerE.

The following table shows the format of these files and provides a short description of the setting options.

File format	Description

`username1:password1`
`username2:password2`
`username3:password3`
`...`
	User account with user name and password
	`usernameX` = user name
	`passwordX` = password
	For `usernameX` and `passwordX` characters, numbers and special characters can be used.
	If only the user name is indicated for a user account (the colon and the password are left out), the password is identical to the user name.
	The total number of user accounts to be entered in the files ip_accs.cfg is only limited by the available memory.

If user accounts are defined, an FTP access to the controllere web server can be carried out according to the following syntax (unless the used browser automatically opens a registration window):

`ftp://<username>:<password>@<IP-address>`

the following applies:

`<username>` = user name,
`<password>` = password,
`<IP-address>` = IP address of the controllerE.

The configuration files email_1.cfg ... email_10.cfg

E-mail templates can be defined using the files `email_x.cfg`. When defined trigger conditions occur, these files are sent from the controllerE to an SMTP server. A prerequisite for sending e-mails is that the IP address of the SMTP server is entered in the Ethernet settings of the controllerE (→ chapter The Ethernet configuration file ethcfg.cfg and its backup file ethcfg_old.cfg (→ page 62).

In total, up to 20 different e-mail templates can be stored in the controllerE.
- 10 templates in the directory `\email\` and
- 10 further templates in the directory `\user\email\`.

In the administrator mode all e-mail templates can be accessed via FTP, in normal operation only the 10 templates which are stored in the directory `\user\email\`.

File format	Description
[Register] `range , offset, type`	In the entry [Register] the range and DP-RAM address from which the trigger signal for the e-mail is to come are defined. Furthermore the data type is defined here, i.e. which data length is to be read from the indicated address. In the following, the value defined here will be called register value. The register value is updated every 500 ms. So, for triggering an e-mail it has to be ensured that the required register value is available in the DP-RAM for more than 500 ms. `range` = IN or OUT range of the DP-RAM → chapter The dual-ported RAM (→ page 36) `Offset` [1]) = address offset [bytes] in the address range defined under "range", e.g. 0x0001 `Type` = data type of the address to be read: byte|word|long
[Register Match] `Value, mask, operand`	In the entry [Register Match] it is defined in which ratio the register value has to be to the defined reference value, to trigger sending of the e-mail. `Value` [1]) = comparison value, e.g. 0x0001 `Mask` [1]) bit mask to which the register value has a binary logical AND conjunction. The resulting value is called result in the following. `Operand` = the operand defines how the result is to be compared with the reference value. If the result of the comparison is TRUE, the e-mail is triggered. Possible operands are "<", ">" or "=". Example: decimal | binary

	Read register value	=	125	1111101
	Mask	=	63	0111111
	Result from logical AND conjunction	=	61	0111101
	Operand	=	>	
	Comparison value	=	50	0110010
	Result	=	61 > 50 = TRUE ⇒ E-mail is triggered	
[To] `receiver1 [; receiver 2 ; ...; receiver n]`	E-mail address(es) of the recipient(s). ²) In case of several recipient addresses they are separated by semicolons.			
[From] `Transmitter`	E-mail address of the sender (= controllerE). ²)			
[Subject] `Subject`	The subject line of the e-mail. ²)			
[Message] `Message`	Message text of the e-mail. ²)			

¹) These values can be indicated as decimal values (e.g. 10) or hexadecimal values (e.g. 0x000A).

²) In this entry dynamical text can also be used via SSI functions (→ page 198).

Table: Format of the files `email_1.cfg` ... `email_10.cfg` and brief description of the different parameters

Changes made to the entries [Register] or [Register Match] in the files `emailx_cfg` will become effective on reboot of the controllere. All other changes are adopted by the controllerE without reboot.

5.4.3 The FTP server

The file system can be managed, e.g. creating and deleting directories or files, via a PC connected to the controllerE, which has an installed web browser.

► Enter the following in the address line of the web browser and confirm with [Enter]:
`ftp://<username>:<password>@<IP-address>`

The following applies:

`<username>` = user name,
`<password>` = password,
`<IP-address>` = IP address of the controllerE.
If no user accounts have been created on the controllerE, the entry `<username>:<password>@` is not necessary.
(→ chapter The configuration files ad_pswd.cfg and sys_pswd.cfg, → page 68)

> The web browser now shows the structure of the file system in the controllerE.

5.4.4 The Telnet server

The Telnet protocol (Telnet = **Tel**ecommunication **Net**work) is a network protocol widely used on the internet. A Telnet interface is part of the Windows operating system. This is a command line platform similar to DOS used for working with the file system.

To start the Telnet service from the start menu of the PC, click on [Execute...] and enter `"Telnet: Hostname"` (`hostname` = IP address of the controllerE). The program HyperTerminal (also part of the Windows operating system) opens and now tries to establish a connection to the controllerE.

After setup of the connection the following commands are available:

General Telnet commands

command	`admin`
Syntax	`admin`
Description	Enables the input of the administrator's user name and the corresponding password, and thus activation of the administrator mode.

command	`help`
Syntax	`help [general|diagnostic|filesystem]`

Description	If called without argument, the following message is displayed:
	`General commands:`
	`help - Help with menus`
	`version - Display version information`
	`exit - Exit station program`
	`Also try 'help [general|diagnostic|filesystem]'`

command	`version`
Syntax	`version`
Description	This command shows the version information, serial number and MAC ID of the device.

command	`exit`
Syntax	`exit`
Description	Terminates the network connection to the controllerE.

Telnet diagnosis commands

The following commands are listed with "`help diagnostic`".

command	`arps`
Syntax	`arps`
Description	Shows the ARP status.

command	`iface`
Syntax	`iface`
Description	(iface → interface) Shows the interface status.

command	`sockets`
Syntax	`sockets`
Description	(socket = Anschluss) Shows the connection list.

command	`routes`
Syntax	`routes`
Description	Shows the IP routing table.

Telnet file system commands

For commands which use file names, directory names or paths as arguments, these can be entered with or without apostrophes. However, names with a space must be placed in apostrophes. Relative path names with "`.`", "`\`" and "`..`" are permitted.

command	`append`
Syntax	`append [[path][file name]] ["text to be appended”]`
Description	Appends a defined text to the content of a file.

command	`cd`
Syntax	`cd [path]`
Description	(cd = change directory) Changes to another directory. Path = "`..`" → changes to the higher-level directory.

command	`copy`
Syntax	`copy [[source path][source file]] [[target path][target file]]`
Description	Copies a file.

command	`del`
Syntax	`del [[path][filename]]`
Description	(del = delete) Deletes a file.

command	`dir`
Syntax	`dir[path]`
Description	(dir = directory) Lists the contents of a directory. If no path is specified, the contents of the current directory are listed.

command	`df`
Syntax	`df`
Description	Shows the file system information.

command	`format`
Syntax	`format`
Description	Formats the file system. This command can only be called in the administrator mode. **NOTE!** After execution of this command all data stored in the file system of the controllerE will be irretrievably deleted!

command	`md`
Syntax	`md [path][directory name]`
Description	(md = make directory) If no path is entered, the directory is created under the current directory.

command	`mkfile`
Syntax	`mkfile [[path][file name]]`
Description	(mkfile = make file) Creates an empty file. If no path is indicated, the file will be created in the current directory.

command	`move`
Syntax	`move [[source path][source file]] [[target path]]`
Description	Moves a file or a directory.

command	`rd`
Syntax	`rd [directory]`
Description	(rd = remove directory) Deletes an empty directory.

command	`ren`
Syntax	`ren [[path][old name]] [[path][new name]]`
Description	(ren = rename) Renames a file or a directory.

command	`type`
Syntax	`type [[path][file name]]`
Description	Lists the contents of a file.

6 Commands in the fieldbus data command channel (module 11)

Contents

→ chapter Module 11: Fieldbus data command channel (→ page 50)

6.1 Module 11, command 1: read master flags

6.1.1 Request of fieldbus master to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	MM		0						0	0	Command number = 01_{hex} = 01_{dec}					
2	Not used *)								Not used *)							

Legend:

MM	master no.	2 bits	01_{bin} = 1_{dec} = master 1 10_{bin} = 2_{dec} = master 2

*) **IMPORTANT:** For the query read only the required bytes!
Unused bytes can contain information of previous queries.

6.1.2 Response from controllerE to fieldbus master

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Copy of the request								E	B	Reflected command number					
2	MF2								MF1							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used
MF1, MF2	Master flags	1 byte	→ table Master flags (→ page 75)

6.1.3 Master flags in module 11

Byte	Bit	If bit D6 = TRUE, then:
MF1	0	periphery of all connected slaves is ok (no peripheral fault)
	1	automatic addressing is enabled
	2	exchange of data with the slaves is active
	3...7	reserved
MF2	0	AS-i configuration is ok
	1	a slave 0 is detected
	2	automatic addressing is enabled
	3	automatic addressing is active
	4	configuration mode is active
	5	normal mode is active
	6	AS-i voltage fault has occurred
	7	offline phase completed

6.2 Module 11, command 2: change operating mode

6.2.1 Request of fieldbus master to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	MM		0						0	0	Command number = 02_{hex} = 02_{dec}					
2	Not used *)								Mod							

Legend:

MM	master no.	2 bits	01_{bin} = 1_{dec} = master 1 10_{bin} = 2_{dec} = master 2
Mod	preset operating mode	1 byte	00 = protected mode 01 = projection mode

*) **IMPORTANT:** For the query read only the required bytes!
Unused bytes can contain information of previous queries.

6.2.2 Response from controllerE to fieldbus master

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Copy of the request								E	B	Reflected command number					
2	Not used *)								Copy of the request							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used

*) **IMPORTANT:** For the query read only the required bytes!
Unused bytes can contain information of previous queries.

6.3 Module 11, command 3: read current slave configuration

6.3.1 Request of fieldbus master to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	MM		ST	SLA					0	0	Command number = 03_{hex} = 03_{dec}					
2	Not used *)								Not used *)							

Legend:

MM	master no.	2 bits	01_{bin} = 1_{dec} = master 1 10_{bin} = 2_{dec} = master 2
ST	slave type	1 bit	0 = single slave or A slave 1 = B slave (addition of 20_{hex} or 32_{dec} to the slave address)
SLA	slave address	5 bits	00…$1F_{hex}$ = 0…31_{dec}

*) **IMPORTANT:** For the query read only the required bytes!
Unused bytes can contain information of previous queries.

6.3.2 Response from controllerE to fieldbus master

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Copy of the request								E	B	Reflected command number					
2	ID code				IO configuration				extended ID code 2				extended ID code 1			

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used

6.3.3 Example: read current slave configuration of slave 7B on AS-i master 1

Request of fieldbus master to controllerE

Word no.	Value [hex.]	Meaning
1	6703	(sllave no. **7**) + (master no. **1** * **64**) + (**32**, if B slave) = 103_{dec} = $\mathbf{67}_{hex}$ 03 = command 3
2	0000	Not used

Response from controllerE to fieldbus master

Word no.	Value [hex.]	Meaning
1	6703	Copy of the request Command processed, no error occurred
2	03EF	0 = ID code 3 = IO configuration E = extended ID code 2 F = extended ID code 1

(corresponds to slave profile S 3.0.E = 2I/2O module with peripheral fault detection)

6.4 Module 11, command 4: read projected slave configuration

6.4.1 Request of fieldbus master to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	MM		ST	SLA					0	0	Command number = 04_{hex} = 04_{dec}					
2	Not used *)								Not used *)							

Legend:

MM	master no.	2 bits	01_{bin} = 1_{dec} = master 1 10_{bin} = 2_{dec} = master 2
ST	slave type	1 bit	0 = single slave or A slave 1 = B slave (addition of 20_{hex} or 32_{dec} to the slave address)
SLA	slave address	5 bits	00…$1F_{hex}$ = 0…31_{dec}

*) **IMPORTANT:** For the query read only the required bytes!
Unused bytes can contain information of previous queries.

6.4.2 Response from controllerE to fieldbus master

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Copy of the request								E	B	Reflected command number					
2	ID code				IO configuration				extended ID code 2				extended ID code 1			

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used

6.4.3 Example: read projected slave configuration of slave 16(A) on AS-i master 1

Request of fieldbus master to controllerE

Word no.	Value [hex.]	Meaning
1	5004	(slave no. **16**) + (master no. **1 * 64**) + (32, if B slave) = 80_{dec} = $\mathbf{50}_{hex}$ 04 = command 4
2	0000	Not used

Request of fieldbus master to controllerE

Word no.	Value [hex.]	Meaning
1	5004	Copy of the request Command processed, no error occurred
2	37EF	3 = ID code 7 = IO configuration E = extended ID code 2 F = extended ID code 1

(corresponds to slave profile S 7.3.E = analogue input module with 4 inputs)

6.5 Module 11, command 5: change projected slave configuration

6.5.1 Request of fieldbus master to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	MM		ST	SLA					0	0	Command number = 05_{hex} = 05_{dec}					
2	ID code				IO configuration				extended ID code 2				extended ID code 1			

Legend:

MM	master no.	2 bits	01_{bin} = 1_{dec} = master 1 10_{bin} = 2_{dec} = master 2
ST	slave type	1 bit	0 = single slave or A slave 1 = B slave (addition of 20_{hex} or 32_{dec} to the slave address)
SLA	slave address	5 bits	00…$1F_{hex}$ = 0…31_{dec}

6.5.2 Response from controllerE to fieldbus master

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Copy of the request								E	B	Reflected command number					
2	Copy of the request								Copy of the request							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used

6.5.3 Example: change projected slave configuration of slave 1(A) on master 2

Request of fieldbus master to controllerE

Word no.	Value [hex.]	Meaning
1	8105	(slave no. **1**) + (master no. **2 * 64**) + (32, if B slave) = 129_{dec} = $\mathbf{81}_{hex}$ 05 = command 5
2	376F	3 = ID code 7 = IO configuration 6 = extended ID code 2 F = extended ID code 1

Response from controllerE to fieldbus master

Word no.	Value [hex.]	Meaning
1	8105	Copy of the request Command processed, no error occurred
2	376F	Copy of the request

6.6 Module 11, command 6: read slave parameters

6.6.1 Request of fieldbus master to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	MM		ST	SLA					0	0	Command number = 06_{hex} = 06_{dec}					
2	Not used *)								Not used *)							

Legend:

MM	master no.	2 bits	01_{bin} = 1_{dec} = master 1 10_{bin} = 2_{dec} = master 2
ST	slave type	1 bit	0 = single slave or A slave 1 = B slave (addition of 20_{hex} or 32_{dec} to the slave address)
SLA	slave address	5 bits	00…$1F_{hex}$ = 0…31_{dec}

*) **IMPORTANT:** For the query read only the required bytes!
Unused bytes can contain information of previous queries.

6.6.2 Response from controllerE to fieldbus master

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Copy of the request								E	B	Reflected command number					
2	current parameter								projected parameter							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used

6.6.3 Example: read slave parameters of slave 2(A) on AS-i master 1

Request of fieldbus master to controllerE

Word no.	Value [hex.]	Meaning
1	4206	(slave no. **2**) + (master no. **1 * 64**) + (32, if B slave) = 66_{dec} = **42_{hex}** 06 = command 6
2	0000	Not used

Response from controllerE to fieldbus master

Word no.	Value [hex.]	Meaning
1	4206	Copy of the request Command processed, no error occurred

2	0F03	0F = current parameter 03 = projected parameter

6.7 Module 11, command 7: change projected slave parameters

Info

The projected parameters can only be changed if the AS-i master operates in the projected mode.
Activation → chapter Module 11, command 2: change operating mode (→ page 76)

6.7.1 Request of fieldbus master to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	MM		ST	SLA					0	0	Command number = 07_{hex} = 07_{dec}					
2	Not used *)								projected parameter							

Legend:

MM	master no.	2 bits	01_{bin} = 1_{dec} = master 1 10_{bin} = 2_{dec} = master 2
ST	slave type	1 bit	0 = single slave or A slave 1 = B slave (addition of 20_{hex} or 32_{dec} to the slave address)
SLA	slave address	5 bits	00…$1F_{hex}$ = 0…31_{dec}

*) **IMPORTANT:** For the query read only the required bytes!
Unused bytes can contain information of previous queries.

6.7.2 Response from controllerE to fieldbus master

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Copy of the request								E	B	Reflected command number					
2	Not used *)								Copy of the request							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used

*) **IMPORTANT:** For the query read only the required bytes!
Unused bytes can contain information of previous queries.

6.7.3 Example: change projected slave parameters of slave 7B on AS-i master 1

Request of fieldbus master to controllerE

Word no.	Value [hex.]	Meaning
1	8707	(slave no. **7**) + (master no. **1 * 64**) + (**32**, if B slave) = 135_{dec} = $\mathbf{87}_{hex}$ 07 = command 7
2	000F	00 = Not used 0F = projected parameter

Response from controllerE to fieldbus master

Word no.	Value [hex.]	Meaning
1	8707	Copy of the request Command processed, no error occurred
2	000F	Copy of the request

6.8 Module 11, command 8: read LAS (list of active slaves)

6.8.1 Slave group in module 11

The feedback word can only give information about max. 16 slaves. Therefore the slaves are divided in 4 groups (→ table).

When querying the slave lists any slave address from the requested slave group is to be indicated.

Slave group	Bit (AS-i slave address)															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	15(A)	14(A)	13(A)	12(A)	11(A)	10(A)	9(A)	8(A)	7(A)	6(A)	5(A)	4(A)	3(A)	2(A)	1(A)	0 *)
2	31(A)	30(A)	29(A)	28(A)	27(A)	26(A)	25(A)	24(A)	23(A)	22(A)	21(A)	20(A)	19(A)	18(A)	17(A)	16(A)
3	15B	14B	13B	12B	11B	10B	9B	8B	7B	6B	5B	4B	3B	2B	1B	res
4	31B	30B	29B	28B	27B	26B	25B	24B	23B	22B	21B	20B	19B	18B	17B	16B

*) LAS and LPS have no slave 0, therefore this bit is set to 0!

6.8.2 Request of fieldbus master to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	MM		ST	SLA					0	0	Command number = 08_{hex} = 08_{dec}					
2	Not used *)								Not used *)							

Legend:

MM	master no.	2 bits	01_{bin} = 1_{dec} = master 1 10_{bin} = 2_{dec} = master 2
ST	slave type	1 bit	0 = single slave or A slave 1 = B slave (addition of 20_{hex} or 32_{dec} to the slave address)
SLA	slave address	5 bits	00…$1F_{hex}$ = 0…31_{dec}

*) **IMPORTANT:** For the query read only the required bytes!
Unused bytes can contain information of previous queries.

6.8.3 Response from controllerE to fieldbus master

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1									E	B						
2	addresses of the active slaves in this slave group (→ page 86)															

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used

*) **IMPORTANT:** For the query read only the required bytes!
Unused bytes can contain information of previous queries.

6.8.4 Example: read LAS (list of active slaves) of slave group 1 on master 1

Request of fieldbus master to controllerE

Word no.	Value [hex.]	Meaning
1	4208	(slave no. **2**) → slave group 1 + (master no. **1 * 64**) + (32, if B slave) = 66_{dec} = $\mathbf{42}_{hex}$ 08 = command 8
2	0000	Not used

Response from controllerE to fieldbus master

Word no.	Value [hex.]	Meaning
2	4208	Copy of the request Command processed, no error occurred
2	03FE	03hex = 00000011bin → table Slave group (→ page 86), slave group 1 ⇒ slaves 8(A) and 9(A) are active FEhex = 11111110bin → table Slave group (→ page 86), slave group 1 ⇒ slaves 1(A) to 7(A) are active

6.9 Module 11, command 9: read LDS (list of detected slaves)

The feedback word can only give information about max. 16 slaves. Therefore the slaves are divided in 4 groups (→ table Slave group, → page 86).

6.9.1 Request of fieldbus master to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	MM		ST	SLA					0	0	Command number = 09_{hex} = 09_{dec}					
2	Not used *)								Not used *)							

Legend:

MM	master no.	2 bits	01_{bin} = 1_{dec} = master 1 10_{bin} = 2_{dec} = master 2
ST	slave type	1 bit	0 = single slave or A slave 1 = B slave (addition of 20_{hex} or 32_{dec} to the slave address)
SLA	slave address	5 bits	00…$1F_{hex}$ = 0…31_{dec}

*) **IMPORTANT:** For the query read only the required bytes!
Unused bytes can contain information of previous queries.

6.9.2 Response from controllerE to fieldbus master

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Copy of the request								E	B	Reflected command number					
2	addresses of the detected slaves in this slave group (→ page 86)															

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used

6.9.3 Example: read LDS (list of detected slaves) of slave group 3 on AS-i master 2

Request of fieldbus master to controllerE

Word no.	Value [hex.]	Meaning
1	A509	(slave no. **5**) → slave group 3 + (master no. **2 * 64**) + (**32**, if B slave) = 165_{dec} = $\mathbf{A5}_{hex}$ 09 = command 9
2	0000	Not used

Response from controllerE to fieldbus master

Word no.	Value [hex.]	Meaning
1	A509	Copy of the request Command processed, no error occurred
2	03FE	03_{hex} = 00000011_{bin} → table Slave group (→ page 86), slave group 3: ⇒ slaves 8B and 9B were detected FE_{hex} = 11111110_{bin} → table Slave group (→ page 86), slave group 3: ⇒ slaves 1B to 7B were detected

6.10 Module 11, command 10_{dec} ($0A_{hex}$): read LPF (list of slaves with peripheral fault)

The feedback word can only give information about max. 16 slaves. Therefore the slaves are divided in 4 groups (→ table Slave group, → page 86).

6.10.1 Request of fieldbus master to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	MM		ST	SLA					0	0	Command number = $0A_{hex}$ = 10_{dec}					
2	Not used *)								Not used *)							

Legend:

MM	master no.	2 bits	01_{bin} = 1_{dec} = master 1 10_{bin} = 2_{dec} = master 2
ST	slave type	1 bit	0 = single slave or A slave 1 = B slave (addition of 20_{hex} or 32_{dec} to the slave address)
SLA	slave address	5 bits	00…$1F_{hex}$ = 0…31_{dec}

*) **IMPORTANT:** For the query read only the required bytes!
Unused bytes can contain information of previous queries.

6.10.2 Response from controllerE to fieldbus master

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Copy of the request								E	B	Reflected command number					
2	addresses of the slaves with peripheral faults in this slave group (→ page 86)															

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used

6.10.3 Example: read LPF (list of slaves with peripheral fault) of slave group 2 on AS-i master 1

Request of fieldbus master to controllerE

Word no.	Value [hex.]	Meaning
1	540A	(slave no. **20**) → slave group 2 + (master no. **1 * 64**) + (32, if B slave) = 84_{dec} = $\mathbf{54}_{hex}$ 0A = command 10
2	0000	Not used

Response from controllerE to fieldbus master

Word no.	Value [hex.]	Meaning
1	540A	Copy of the request Command processed, no error occurred
2	0220	02_{hex} = 00000010_{bin} → table Slave group (→ page 86), slave group 2: ⇒ slave 25(A) indicates peripheral fault 20_{hex} = 00100000_{bin} → table Slave group (→ page 86), slave group 2: ⇒ slave 21(A) indicates peripheral fault

6.11 Module 11, command 11_{dec} ($0B_{hex}$): read LPS (list of projected slaves)

The feedback word can only give information about max. 16 slaves. Therefore the slaves are divided in 4 groups (→ table Slave group, → page 86).

6.11.1 Request of fieldbus master to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	MM		ST	SLA					0	0	Command number = $0B_{hex}$ = 11_{dec}					
2	Not used *)								Not used *)							

Legend:

MM	master no.	2 bits	01_{bin} = 1_{dec} = master 1 10_{bin} = 2_{dec} = master 2
ST	slave type	1 bit	0 = single slave or A slave 1 = B slave (addition of 20_{hex} or 32_{dec} to the slave address)
SLA	slave address	5 bits	00…$1F_{hex}$ = 0…31_{dec}

*) **IMPORTANT:** For the query read only the required bytes!
Unused bytes can contain information of previous queries.

6.11.2 Response from controllerE to fieldbus master

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Copy of the request								E	B	Reflected command number					
2	addresses of the projected slaves in this slave group (→ page 86)															

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used

6.11.3 Example: read LPS (list of projected slaves) of slave group 2 on AS-i master 1

Request of fieldbus master to controllerE

Word no.	Value [hex.]	Meaning
1	540B	(slave no. **20**) → slave group 2 + (master no. **1 * 64**) + (32, if B slave) = 84_{dec} = $\mathbf{54}_{hex}$ 0B = command 11
2	0000	Not used

Response from controllerE to fieldbus master

Word no.	Value [hex.]	Meaning
1	540B	Copy of the request Command number
2	02FE	02_{hex} = 00000010_{bin} → table Slave group (→ page 86), slave group 2: ⇒ slave 25(A) is projected FE_{hex} = 11111110_{bin} → table Slave group (→ page 86), slave group 2: ⇒ slaves 17(A) to 23(A) are projected

6.12 Module 11, command 13_{dec} ($0D_{hex}$): read telegram error counter

6.12.1 Request of fieldbus master to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	MM		ST	SLA					0	0	Command number = $0D_{hex}$ = 13_{dec}					
2	Not used *)								Not used *)							

Legend:

MM	master no.	2 bits	01_{bin} = 1_{dec} = master 1 10_{bin} = 2_{dec} = master 2
ST	slave type	1 bit	0 = single slave or A slave 1 = B slave (addition of 20_{hex} or 32_{dec} to the slave address)
SLA	slave address	5 bits	00…$1F_{hex}$ = 0…31_{dec}

*) **IMPORTANT:** For the query read only the required bytes!
Unused bytes can contain information of previous queries.

6.12.2 Response from controllerE to fieldbus master

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Copy of the request								E	B	Reflected command number					
2	number of errors during the exchange of data between the slave and the master since power on or reset															

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used

6.12.3 Example: read telegram error counter of slave 1 on AS-i master 1

Request of fieldbus master to controllerE

Word no.	Value [hex.]	Meaning
1	410D	(slave no. **1**) + (master no. **1** * **64**) + (32, if B slave) = 65_{dec} = $\mathbf{41}_{hex}$ 0D = command 13
2	0000	Not used

Response from controllerE to fieldbus master

Word no.	Value [hex.]	Meaning
1	410D	Copy of the request Command processed, no error occurred
2	0020	error counter = 0020_{hex} = 0032_{dec} ⇒ Since the last power on of the controllerE or the reset of the counter, 32 faulty telegrams have occurred during data exchange.

6.13 Module 11, command 14_{dec} ($0E_{hex}$): read configuration error counter

6.13.1 Request of fieldbus master to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	MM		ST	SLA					0	0	Command number = $0E_{hex}$ = 14_{dec}					
2	Not used *)								Not used *)							

Legend:

MM	master no.	2 bits	01_{bin} = 1_{dec} = master 1 10_{bin} = 2_{dec} = master 2
ST	slave type	1 bit	0 = single slave or A slave 1 = B slave (addition of 20_{hex} or 32_{dec} to the slave address)
SLA	slave address	5 bits	00…$1F_{hex}$ = 0…31_{dec}

*) **IMPORTANT:** For the query read only the required bytes!
Unused bytes can contain information of previous queries.

6.13.2 Response from controllerE to fieldbus master

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Copy of the request								E	B	Reflected command number					
2	number of configuration errors of the master since power on or reset															

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used

6.13.3 Example: read configuration error counter on AS-i master 2

Request of fieldbus master to controllerE

Word no.	Value [hex.]	Meaning
1	800E	(slave no. **1**) + (master no. **1 * 64**) + (32, if B slave) = 65_{dec} = $\mathbf{41}_{hex}$ 0E = command 14
2	0000	Not used

Response from controllerE to fieldbus master

Word no.	Value [hex.]	Meaning
1	800E	Copy of the request Command processed, no error occurred
2	0003	error counter = 0003_{hex} = 0003_{dec} ⇒ Since the last power on of the controllerE or the reset of the counter, 3 configuration errors have occurred.

6.14 Module 11, command 15_{dec} ($0F_{hex}$): read AS-i cycle counter

6.14.1 Request of fieldbus master to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	MM		0	0					0	0	Command number = $0F_{hex}$ = 15_{dec}					
2	Not used *)								Not used *)							

Legend:

MM	master no.	2 bits	01_{bin} = 1_{dec} = master 1 10_{bin} = 2_{dec} = master 2

*) **IMPORTANT:** For the query read only the required bytes!
Unused bytes can contain information of previous queries.

6.14.2 Response from controllerE to fieldbus master

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Copy of the request								E	B	Reflected command number					
2	number of AS-i cycles of the master since power on															

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used

By carrying out several measurements the number of cycles per time unit can be measured.

6.14.3 Example: read AS-i cycle counter of AS-i master 1

Request of fieldbus master to controllerE

Word no.	Value [hex.]	Meaning
1	400F	(master no. **1 * 64**) = 64_{dec} = $\mathbf{40}_{hex}$ 0F = command 15
2	0000	Not used

Response from controllerE to fieldbus master

Word no.	Value [hex.]	Meaning
1	400F	Copy of the request Command processed, no error occurred
2	04CA	cycle counter = $04CA_{hex}$ = $1\,226_{dec}$ ⇒ Since the last power on of the controllerE, 1 226 cycles have been performed in the AS-i master 1.

6.15 Module 11, command 16_{dec} (10_{hex}): change current slave parameters

6.15.1 Request of fieldbus master to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	MM		ST	SLA					0	0	Command number = 10_{hex} = 16_{dec}					
2	Not used *)								Not used *)							

Legend:

MM	master no.	2 bits	01_{bin} = 1_{dec} = master 1 10_{bin} = 2_{dec} = master 2
ST	slave type	1 bit	0 = single slave or A slave 1 = B slave (addition of 20_{hex} or 32_{dec} to the slave address)
SLA	slave address	5 bits	00…$1F_{hex}$ = 0…31_{dec}

*) **IMPORTANT:** For the query read only the required bytes!
Unused bytes can contain information of previous queries.

6.15.2 Response from controllerE to fieldbus master

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Copy of the request								E	B	Reflected command number					
2	Not used *)								parameter feedback value (can be different from preset value)							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used

*) **IMPORTANT:** For the query read only the required bytes!
Unused bytes can contain information of previous queries.

6.15.3 Example: change slave parameter of slave 7 on AS-i master 1 to the value "F"

Request of fieldbus master to controllerE

Word no.	Value [hex.]	Meaning
1	4710	(slave no. **7**) + (master no. **1** * **64**) + (32, if B slave) = 71_{dec} = $\mathbf{47}_{hex}$ 10 = command 16
2	0000	Not used

Response from controllerE to fieldbus master

Word no.	Value [hex.]	Meaning
1	4710	Copy of the request Command processed, no error occurred
2	000F	0F = parameter feedback value (can be different from the preset value)

6.16 Module 11, command 19_{dec} (13_{hex}): project all

6.16.1 Request of fieldbus master to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	MM		0	0					0	0	Command number = 13_{hex} = 19_{dec}					
2	Not used *)								Not used *)							

Legend:

MM	master no.	2 bits	01_{bin} = 1_{dec} = master 1 10_{bin} = 2_{dec} = master 2

*) **IMPORTANT:** For the query read only the required bytes!
Unused bytes can contain information of previous queries.

6.16.2 Response from controllerE to fieldbus master

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Copy of the request								E	B	Reflected command number					
2	Not used *)								status							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used
status	status	1 byte	80_{hex} = process completed 00 = in all other cases

*) **IMPORTANT:** For the query read only the required bytes!
Unused bytes can contain information of previous queries.

6.16.3 Example: project all on AS-i master 1

Request of fieldbus master to controllerE

Word no.	Value [hex.]	Meaning
1	4013	(master no. **1 * 64**) = 64_{dec} = $\mathbf{40}_{hex}$ 13 = command 19
2	0000	Not used

Response from controllerE to fieldbus master

Word no.	Value [hex.]	Meaning
1	4013	Copy of the request Command processed, no error occurred
2	0080	80_{hex} = status: process completed

6.17 Module 11, command 21_{dec} (15_{hex}): save configuration in flash

6.17.1 Request of fieldbus master to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	MM		0	0					0	0	Command number = 15_{hex} = 21_{dec}					
2	Not used *)								Not used *)							

Legend:

MM	master no.	2 bits	01_{bin} = 1_{dec} = master 1 10_{bin} = 2_{dec} = master 2

*) **IMPORTANT:** For the query read only the required bytes!
Unused bytes can contain information of previous queries.

6.17.2 Response from controllerE to fieldbus master

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Copy of the request								E	B	Reflected command number					
2	Not used *)								Not used *)							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used

*) **IMPORTANT:** For the query read only the required bytes!
Unused bytes can contain information of previous queries.

6.17.3 Example: save AS-i configuration in flash for AS-i master 1

Request of fieldbus master to controllerE

Word no.	Value [hex.]	Meaning
1	4015	(master no. **1 * 64**) = 64_{dec} = $\mathbf{40}_{hex}$ 15 = command 21
2	0000	Not used

Response from controllerE to fieldbus master

Word no.	Value [hex.]	Meaning
1	4015	Copy of the request Command processed, no error occurred
2	0000	Not used

6.18 Module 11, command 22_{dec} (16_{hex}): reset telegram error counter of a slave

6.18.1 Request of fieldbus master to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	MM		ST	SLA					0	0	command number = 16_{hex} = 22_{dec}					
2	Not used *)								Not used *)							

Legend:

MM	master no.	2 bits	01_{bin} = 1_{dec} = master 1 10_{bin} = 2_{dec} = master 2
ST	slave type	1 bit	0 = single slave or A slave 1 = B slave (addition of 20_{hex} or 32_{dec} to the slave address)
SLA	slave address	5 bits	00…$1F_{hex}$ = 0…31_{dec}

*) **IMPORTANT:** For the query read only the required bytes!
Unused bytes can contain information of previous queries.

6.18.2 Response from controllerE to fieldbus master

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Copy of the request								E	B	Reflected command number					
2	Not used *)								Not used *)							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used

*) **IMPORTANT:** For the query read only the required bytes!
Unused bytes can contain information of previous queries.

6.18.3 Example: reset telegram error counter of slave 7(A) on AS-i master 2

Request of fieldbus master to controllerE

Word no.	Value [hex.]	Meaning
1	8716	(slave no. **7**) + (master no. **2** * **64**) + (32, if B slave) = 135_{dec} = $\mathbf{87}_{hex}$ 16 = command 22
2	0000	Not used

Response from controllerE to fieldbus master

Word no.	Value [hex.]	Meaning
1	8716	Copy of the request Command processed, no error occurred
2	0000	Not used

7 The host command channel

Contents

Module 19 (→ chapter Module 19: Host command channel, → page 59) contains an extended command channel which can have a length of 5 or 18 words. A ModbusTCP or an Ethernet/IP client can be used as host system. The commands are always triggered by the host by a corresponding entry in its output data range. The controllerE responds then in the input data area of the host system.

7.1 Syntax of the host command channel

7.1.1 Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	res. = 0		M	UID					Command number							
2	res. = 0		ST	SLA					res. = 0		Number of data bytes to be transmitted					
3...18	Parameter data of the command to be executed															

Legend:

M	master no.	1 bit	0 = master 1 1 = master 2
UID	user ID	5 bits	$00…1F_{hex} = 0…31_{dec}$ (a change to the user ID starts the command call)
ST	slave type	1 bit	0 = single slave or A slave 1 = B slave (addition of 20_{hex} or 32_{dec} to the slave address)
SLA	slave address	5 bits	$00…1F_{hex} = 0…31_{dec}$

2nd word: reserved for 7.4 commands
3...18th word: command data

NOTE

- ► If a command is to be executed, the user ID must be changed!
  Changing the command number does not start the execution.
- ► If a command is to be executed several times, the user ID must be changed accordingly, e.g. by counting up.
- ► Do not count up the user ID until the preceding command has been completed
  (to do so, check the bits E + B in the 1st word of the response).

7.1.2 Response from controllerE to host

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2	res.	ST	reflected SLA					F	res. = 0		Number of received data bytes					
3...18	command data															

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used
M	master no.	1 bit	0 = master 1 1 = master 2
ST	slave type	1 bit	0 = single slave or A slave 1 = B slave (addition of 20_{hex} or 32_{dec} to the slave address)
F	Error bit	1 bit	0 = no error detected 1 = error when executing the command

2nd word: reserved for 7.4 commands
3...18th word: command data

7.2 Commands in the host command channel

7.2.1 Module 19, command 00_{dec} (00_{hex}): no execution of a command

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	res. = 0		M	UID					Command number = 00_{hex} = 00_{dec}							
2...18	Not used								Not used							

Legend:

M	master no.	1 bit	0 = master 1 1 = master 2
UID	user ID	5 bits	00…$1F_{hex}$ = 0…31_{dec} (a change to the user ID starts the command call)

Response from controllerE to host

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2...18	Not changed								Not changed							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used
M	master no.	1 bit	0 = master 1 1 = master 2

Example: no execution of a command

Request of host to controllerE

Word no.	Value [hex.]	Meaning
1	0300	M = 0: AS-i master 1 UID = 03: user ID changes to 3 00 = command 0
2…18	0000	Not used

Response from controllerE to host

Word no.	Value [hex.]	Meaning
1	0300	Copy of the request Command processed, no error occurred
2…18	0000	Not changed

7.2.2 Module 19, command 01_{dec} (01_{hex}): write parameters to an AS-i slave (change current slave parameters)

Requirement: The addressed AS-i master must be in the normal mode.

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	res. = 0		M	UID					Command number = 01_{hex} = 01_{dec}							
2	Reserved = 0								Reserved = 0							
3	Reserved = 0								res. = 0		ST	SLA				
4	Reserved = 0								parameter value to be written							
5...18	Not used								Not used							

Legend:

M	master no.	1 bit	0 = master 1 1 = master 2
UID	user ID	5 bits	00…$1F_{hex}$ = 0…31_{dec} (a change to the user ID starts the command call)
ST	slave type	1 bit	0 = single slave or A slave 1 = B slave (addition of 20_{hex} or 32_{dec} to the slave address)
SLA	slave address	5 bits	00…$1F_{hex}$ = 0…31_{dec}

Response from controllerE to host in the normal case

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3	Reserved = 0								parameter value read back							
4...18	Not changed								Not changed							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used
M	master no.	1 bit	0 = master 1 1 = master 2

Response from controllerE to host in case of a fault

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3	Reserved = 0								Error code							
4...18	Not changed								Not changed							

Possible error codes:

Value [hex.]	Meaning
01	No slave response or: master is in the offline mode when calling the command
0A	The slave is not in the LAS
0B	Parameter or address invalid
14	Master is in the wrong operating mode *)

*) here: is not in the normal mode

Example: change parameter of slave 4B on AS-i master 1 to the value 03

Request of host to controllerE

Word no.	Value [hex.]	Meaning
1	0901	M = 0: AS-i master 1 UID = 09: user ID changes to 9 01 = command 1
2	0000	Reserved
3	0024	(slave no. **4**) + (**32**, if B slave) = 36_{dec} = $\mathbf{24}_{hex}$
4	0003	parameter value to be written
5…18	0000	Not used

Response from controllerE to host in the normal case

Word no.	Value [hex.]	Meaning
1	0901	Copy of the request Command processed, no error occurred
2	0000	Reserved
3	0003	parameter value read back; may differ from the value to be written (so-called reflected parameter)
4…18	0000	Not changed

Response from controllerE to host in case of a fault

Word no.	Value [hex.]	Meaning
1	8901	Copy of the request E=1: Error when executing the command
2	0000	Reserved
3	000A	0A = error code: slave is not in LAS
4…18	0000	Not changed

7.2.3 Module 19, command 03_{dec} (03_{hex}): adopt and save currently connected AS-i slaves in the configuration

Requirement: The addressed AS-i master must be in the projection mode.

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	res. = 0		M	UID					Command number = 03_{hex} = 03_{dec}							
2...18	Not used								Not used							

Legend:

M	master no.	1 bit	0 = master 1 1 = master 2
UID	user ID	5 bits	00…$1F_{hex}$ = 0…31_{dec} (a change to the user ID starts the command call)

Response from controllerE to host in the normal case

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2...18	Not changed								Not changed							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used
M	master no.	1 bit	0 = master 1 1 = master 2

Response from controllerE to host in case of a fault

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3	16#00								Error code = 16#14							
4...18	Not changed								Not changed							

Possible error codes:

Value [hex.]	Meaning
14	Master is in the wrong operating mode *)

*) here: is not in the projection mode

Example: adopt and save currently connected AS-i slaves in the configuration

Request of host to controllerE

Word no.	Value [hex.]	Meaning
1	0C03	M = 0: AS-i master 1 UID = 0C: user ID changes to 12 03 = command 3
2…18	0000	Not used

Response from controllerE to host in the normal case

Word no.	Value [hex.]	Meaning
1	0C03	Copy of the request Command processed, no error occurred
2…18	0000	Not changed

Response from controllerE to host in case of a fault

Word no.	Value [hex.]	Meaning
1	8C03	Copy of the request E=1: Error when executing the command
2	0000	Reserved
3	0017	17 = error code: master is not in the projection mode
4…18	0000	Not changed

7.2.4 Module 19, command 04_{dec} (04_{hex}): change the list of projected AS-i slaves (LPS)

Requirement: The addressed AS-i master must be in the projection mode.

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	res. = 0		M	UID					Command number = 04_{hex} = 04_{dec}							
2	Reserved = 0								Reserved = 0							
3	15(A)	14(A)	13(A)	12(A)	11(A)	10(A)	9(A)	8(A)	7(A)	6(A)	5(A)	4(A)	3(A)	2(A)	1(A)	0
4	31(A)	30(A)	29(A)	28(A)	27(A)	26(A)	25(A)	24(A)	23(A)	22(A)	21(A)	20(A)	19(A)	18(A)	17(A)	16(A)
5	15B	14B	13B	12B	11B	10B	9B	8B	7B	6B	5B	4B	3B	2B	1B	0
6	31B	30B	29B	28B	27B	26B	25B	24B	23B	22B	21B	20B	19B	18B	17B	16B
7...18	Not used								Not used							

Legend:

M	master no.	1 bit	0 = master 1 1 = master 2
UID	user ID	5 bits	$00…1F_{hex}$ = $0…31_{dec}$ (a change to the user ID starts the command call)

► Set the bit corresponding to the slave address to TRUE in the words 3...6 for each slave to be projected.

Response from controllerE to host in the normal case

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2...18	Not changed								Not changed							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used
M	master no.	1 bit	0 = master 1 1 = master 2

Response from controllerE to host in case of a fault

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3	Reserved = 0								Error code							
4...18	Not changed								Not changed							

Possible error codes:

Value [hex.]	Meaning
14	Master is in the wrong operating mode *)

*) here: is not in the projection mode

Example: adopt and save currently connected AS-i slaves in the configuration

Request of host to controllerE

Word no.	Value [hex.]	Meaning
1	0204	M = 0: AS-i master 1 UID = 02: user ID changes to 2 04 = command 4
2	0000	Reserved
3	003E	slaves 1 to 5 are to be projected
4	8000	slave 31A is to be projected
5	0002	slave 1B is to be projected
6	0001	slave 16B is to be projected
7…18	0000	Not used

Response from controllerE to host in the normal case

Word no.	Value [hex.]	Meaning
1	0204	Copy of the request Command processed, no error occurred
2…18	0000	Not changed

Response from controllerE to host in case of a fault

Word no.	Value [hex.]	Meaning
1	8204	Copy of the request E=1: Error when executing the command
2	0000	Reserved
3	0014	14 = error code: master is not in the projection mode
4…18	0000	Not changed

7.2.5 Module 19, command 05_{dec} (05_{hex}): set the operating mode of the AS-i master

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	res. = 0		M	UID					Command number = 05_{hex} = 05_{dec}							
2	Reserved = 0								Reserved = 0							
3	Reserved = 0								MOD							
4...18	Not used								Not used							

Legend:

M	master no.	1 bit	0 = master 1 1 = master 2
UID	user ID	5 bits	00…$1F_{hex}$ = 0…31_{dec} (a change to the user ID starts the command call)
MOD	Operating mode	1 byte	00 = set master to the normal mode (protected mode) 01 = set master to the projection mode

Response from controllerE to host in the normal case

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2...18	Not changed								Not changed							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used
M	master no.	1 bit	0 = master 1 1 = master 2

Response from controllerE to host in case of a fault

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3	Reserved = 0								Error code							
4...18	Not changed								Not changed							

Possible error codes:

Value [hex.]	Meaning
03	Slave with address 0 connected

Example: set AS-i master 1 to projection mode

Request of host to controllerE

Word no.	Value [hex.]	Meaning
1	0105	M = 0: AS-i master 1 UID = 01: user-ID changes to 1 05 = command 5
2	0000	Reserved
3	0001	1 = set master to the projection mode
4…18	0000	Not used

Response from controllerE to host in the normal case

Word no.	Value [hex.]	Meaning
1	0105	Copy of the request Command processed, no error occurred
2…18	0000	Not changed

Response from controllerE to host in case of a fault

Word no.	Value [hex.]	Meaning
1	8105	Copy of the request E=1: Error when executing the command
2	0000	Reserved
3	0003	03 = error code: one slave with the address 0 is connected
4…18	0000	Not changed

7.2.6 Module 19, command 06_{dec} (06_{hex}): readdress a connected AS-i slave

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	res. = 0		M	UID					Command number = 06_{hex} = 06_{dec}							
2	Reserved = 0								Reserved = 0							
3	Reserved = 0								res. = 0		ST	old SLA				
4	Reserved = 0								res. = 0		ST	new SLA				
5...18	Not used								Not used							

Legend:

M	master no.	1 bit	0 = master 1 1 = master 2
UID	user ID	5 bits	00…$1F_{hex}$ = 0…31_{dec} (a change to the user ID starts the command call)
ST	slave type	1 bit	0 = single slave or A slave 1 = B slave (addition of 20_{hex} or 32_{dec} to the slave address)
SLA	slave address	5 bits	00…$1F_{hex}$ = 0…31_{dec}

Response from controllerE to host in the normal case

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2...18	Not changed								Not changed							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used
M	master no.	1 bit	0 = master 1 1 = master 2

Response from controllerE to host in case of a fault

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3	Reserved = 0								Error code							
4...18	Not changed								Not changed							

Possible error codes:

Value [hex.]	Meaning
01	No slave response or: master is in the offline mode when calling the command
02	No slave with the old address found
03	Slave with address 0 connected
04	No slave with the new address found
05	Error when deleting the old address
06	Error when reading the IO configuration
07	Error when writing the new address or extended ID code 1
08	New address could only be saved temporarily
09	Extended ID code 1 could only be saved temporarily
0B	Parameter or address invalid
14	Master is in the wrong operating mode *)

*) here: is not in the normal mode

Example: readdress AS-i slave 9B to 11A

Request of host to controllerE

Word no.	Value [hex.]	Meaning
1	0806	M = 0: AS-i master 1 UID = 08: user ID changes to 8 06 = command 6
2	0000	Reserved
3	0029	29 = 20_{hex} (for B slaves) + 9_{hex} = old slave address 9B
4	000B	B_{hex} = 11_{dec} = new slave adress 11A
5…18	0000	Not used

Response from controllerE to host in the normal case

Word no.	Value [hex.]	Meaning
1	0806	Copy of the request Command processed, no error occurred
2…18	0000	Not changed

Response from controllerE to host in case of a fault

Word no.	Value [hex.]	Meaning
1	8806	Copy of the request E=1: Error when executing the command
2	0000	Reserved
3	0003	03 = error code: one slave with the address 0 is connected
4…18	0000	Not changed

7.2.7 Module 19, command 07_{dec} (07_{hex}): set the auto address mode of the AS-i master

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	res. = 0		M	UID					Command number = 07_{hex} = 07_{dec}							
2	Reserved = 0								Reserved = 0							
3	Reserved = 0								AutoAd							
4...18	Not used								Not used							

Legend:

M	master no.	1 bit	0 = master 1 1 = master 2
UID	user ID	5 bits	00…$1F_{hex}$ = 0…31_{dec} (a change to the user ID starts the command call)
AutoAd	Automatic addressing	1 byte	00 = deactivate automatic addressing 01 = activate automatic addressing

Response from controllerE to host

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2...18	Not changed								Not changed							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used
M	master no.	1 bit	0 = master 1 1 = master 2

Example: activate automatic addressing AS-i master 1

Request of host to controllerE

Word no.	Value [hex.]	Meaning
1	0407	M = 0: AS-i master 1 UID = 04: user ID changes to 4 07 = command 7
2	0000	Reserved
3	0001	01 = activate automatic addressing
4…18	0000	Not used

Response from controllerE to host

Word no.	Value [hex.]	Meaning
1	0407	Copy of the request Command processed, no error occurred
2…18	0000	Not changed

7.2.8 Module 19, command 09_{dec} (09_{hex}): change extended ID code 1 in the AS-i slave

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	res. = 0		M	UID					Command number = 09_{hex} = 09_{dec}							
2	Reserved = 0								Reserved = 0							
3	Reserved = 0								res. = 0		ST	SLA				
4	Reserved = 0								new "extended ID code 1"							
5...18	Not used								Not used							

Legend:

M	master no.	1 bit	0 = master 1 1 = master 2
UID	user ID	5 bits	00…1F_{hex} = 0…31_{dec} (a change to the user ID starts the command call)
ST	slave type	1 bit	0 = single slave or A slave 1 = B slave (addition of 20_{hex} or 32_{dec} to the slave address)
SLA	slave address	5 bits	00…1F_{hex} = 0…31_{dec}

Response from controllerE to host in the normal case

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2...18	Not changed								Not changed							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used
M	master no.	1 bit	0 = master 1 1 = master 2

Response from controllerE to host in case of a fault

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3	Reserved = 0								Error code							
4...18	Not changed								Not changed							

Possible error codes:

Value [hex.]	Meaning
01	No slave response or: master is in the offline mode when calling the command
02	No slave with the old address found
03	Slave with address 0 connected
07	Error when writing the new address or extended ID code 1
09	Extended ID code 1 could only be saved temporarily
0B	Parameter or address invalid

Example: change "extended ID code 1" in AS-i slave 17(A) to "8"

Request of host to controllerE

Word no.	Value [hex.]	Meaning
1	0F09	M = 0: AS-i master 1 UID = 0F: user ID changes to 15 09 = command 9
2	0000	Reserved
3	0011	$11_{hex} = 17_{dec}$ = slave address 17(A)
4	0008	new "Extended ID Code 1" is 8
5…18	0000	Not used

Response from controllerE to host in the normal case

Word no.	Value [hex.]	Meaning
1	0F09	Copy of the request Command processed, no error occurred
2…18	0000	Not changed

Response from controllerE to host in case of a fault

Word no.	Value [hex.]	Meaning
1	8F09	Copy of the request E=1: Error when executing the command
2	0000	Reserved
3	0007	07 = error code: slave does not support extended ID code
4…18	0000	Not changed

7.2.9 Module 19, commands 10...20$_{dec}$ (0A...14$_{hex}$): force analogue data transmission directly to / from 3 AS-i slaves in each case

With these commands, the analogue input or output data of 3 slaves can be overwritten. The commands are assigned to 3 slave addresses:

Command number		Slave addresses		
decimal	hexadecimal			
10	**0A**	1	2	3
11	**0B**	4	5	6
12	**0C**	7	8	9
13	**0D**	10	11	12
14	**0E**	13	14	15
15	**0F**	16	17	18
16	**10**	19	20	21
17	**11**	22	23	24
18	**12**	25	26	27
19	**13**	28	29	30
20	**14**	31	-	-

Table: Assignment command number - slave addresses

Example: In the command 14$_{dec}$ (0E$_{hex}$) the data of the slave addresses 13, 14 and 15 are transmitted.

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	res. = 0		M	UID					Command number = $0A…14_{hex}$ = $10…20_{dec}$							
2	Reserved = 0								Reserved = 0							
3	output data AS-i slave 1/4/7/10/13/16/19/22/25/28/31, channel 0															
4	output data AS-i slave 1/4/7/10/13/16/19/22/25/28/31, channel 1															
5	output data AS-i slave 1/4/7/10/13/16/19/22/25/28/31, channel 2															
6	output data AS-i slave 1/4/7/10/13/16/19/22/25/28/31, channel 3															
7	Reserved = 0								O3	V3	O2	V2	O1	V1	O0	V0
8	output data AS-i slave 2/5/8/11/14/17/20/23/26/29, channel 0															
9	output data AS-i slave 2/5/8/11/14/17/20/23/26/29, channel 1															
10	output data AS-i slave 2/5/8/11/14/17/20/23/26/29, channel 2															
11	output data AS-i slave 2/5/8/11/14/17/20/23/26/29, channel 3															
12	Reserved = 0								O3	V3	O2	V2	O1	V1	O0	V0
13	output data AS-i slave 3/6/9/12/15/18/21/24/27/30, channel 0															
14	output data AS-i slave 3/6/9/12/15/18/21/24/27/30, channel 1															
15	output data AS-i slave 3/6/9/12/15/18/21/24/27/30, channel 2															
16	output data AS-i slave 3/6/9/12/15/18/21/24/27/30, channel 3															
17	Reserved = 0								O3	V3	O2	V2	O1	V1	O0	V0
18	Not used								Not used							

Legend:

M	master no.	1 bit	0 = master 1 1 = master 2
UID	user ID	5 bits	$00…1F_{hex}$ = $0…31_{dec}$ (a change to the user ID starts the command call)
Vn	valid bit	1 bit	0 = values in channel n are invalid 1 = values in channel n are valid Output data must be valid (Vn = 1) to be enabled in the AS-i slave!
On	overflow bit	1 bit	0 = data are in the valid range, 1 = data are in the invalid range (especially in case of input modules when the measuring range is not reached or exceeded)

Response from controllerE to host

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3	input data or reflected output data AS-i slave 1/4/7/10/13/16/19/22/25/28/31, channel 0															
4	input data or reflected output data AS-i slave 1/4/7/10/13/16/19/22/25/28/31, channel 1															
5	input data or reflected output data AS-i slave 1/4/7/10/13/16/19/22/25/28/31, channel 2															
6	input data or reflected output data AS-i slave 1/4/7/10/13/16/19/22/25/28/31, channel 3															
7	Reserved = 0						TV	OV	O3	V3	O2	V2	O1	V1	O0	V0
8	input data or reflected output data AS-i slave 2/5/8/11/14/17/20/23/26/29, channel 0															
9	input data or reflected output data AS-i slave 2/5/8/11/14/17/20/23/26/29, channel 1															
10	input data or reflected output data AS-i slave 2/5/8/11/14/17/20/23/26/29, channel 2															
11	input data or reflected output data AS-i slave 2/5/8/11/14/17/20/23/26/29, channel 3															
12	Reserved = 0						TV	OV	O3	V3	O2	V2	O1	V1	O0	V0
13	input data or reflected output data AS-i slave 3/6/9/12/15/18/21/24/27/30, channel 0															
14	input data or reflected output data AS-i slave 3/6/9/12/15/18/21/24/27/30, channel 1															
15	input data or reflected output data AS-i slave 3/6/9/12/15/18/21/24/27/30, channel 2															
16	input data or reflected output data AS-i slave 3/6/9/12/15/18/21/24/27/30, channel 3															
17	Reserved = 0						TV	OV	O3	V3	O2	V2	O1	V1	O0	V0
18	Not changed								Not changed							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used
M	master no.	1 bit	0 = master 1 1 = master 2
OV	output valid (**O**utput **v**alid)	1 bit	0 = the AS-i slave has not received any valid output values for at least 3.5 seconds, or: it is an input slave 1 = AS-i slave has received valid data at least once in the last 3 seconds
TV	transmission valid	1 bit	0 = the last value transmission to the AS-i slave was faulty 1 = the last value transmission to the AS-i slave was carried out correctly
On	overflow bit	1 bit	0 = data are in the valid range, 1 = data are in the invalid range (especially in case of input modules when the measuring range is not reached or exceeded)
Vn	valid bit	1 bit	0 = values in channel n are invalid 1 = values in channel n are valid Output data must be valid (Vn = 1) to be enabled in the AS-i slave!

Example: force analogue data (4 channels) to slave 1 on master 1

Request of host to controllerE

Word no.	Value [hex.]	Meaning
1	090A	M = 0: AS-i master 1 UID = 09: user ID changes to 9 0A = command 10
2	0000	Reserved
3	0169	output data AS-i slave 1, channel 0
4	0202	output data AS-i slave 1, channel 1
5	0395	output data AS-i slave 1, channel 2
6	1033	output data AS-i slave 1, channel 3
7	0055	**o**verflow and **v**alid bits for AS-i slave 1: O3 = 0, V3 = 1, O2 = 0, V2 = 1, O1 = 0, V1 = 1, O0 = 0, V0 = 1
8	2009	output data AS-i slave 2, channel 0
9	2202	output data AS-i slave 2, channel 1
10	0195	output data AS-i slave 2, channel 2
11	1022	output data AS-i slave 2, channel 3
12	0055	**o**verflow and **v**alid bits for AS-i slave 2: O3 = 0, V3 = 1, O2 = 0, V2 = 1, O1 = 0, V1 = 1, O0 = 0, V0 = 1
13	3339	output data AS-i slave 3, channel 0
14	1102	output data AS-i slave 3, channel 1
15	1953	output data AS-i slave 3, channel 2
16	1234	output data AS-i slave 3, channel 3
17	0055	**o**verflow and **v**alid bits for AS-i slave 3: O3 = 0, V3 = 1, O2 = 0, V2 = 1, O1 = 0, V1 = 1, O0 = 0, V0 = 1
18	0000	Not used

Response from controllerE to host

Word no.	Value [hex.]	Meaning
1	090A	Copy of the request Command processed, no error occurred
2	0000	Reserved
3	3169	(slave 1 is a 4-channel input slave) input data AS-i slave 1, channel 0
4	2202	input data AS-i slave 1, channel 1
5	1395	input data AS-i slave 1, channel 2
6	0033	input data AS-i slave 1, channel 3
7	0055	**o**verflow and **v**alid bits for AS-i slave 1: TV = 1, OV = 0, O3 = 0, V3 = 1, O2 = 0, V2 = 1, O1 = 0, V1 = 1, O0 = 0, V0 = 1
8	2229	(slave 2 is a 2-channel input slave) input data AS-i slave 2, channel 0
9	2332	input data AS-i slave 2, channel 1
10	7FFF	for AS-i slave 2, channel 2 no valid value
11	7FFF	for AS-i slave 2, channel 3 no valid value
12	0055	**o**verflow and **v**alid bits for AS-i slave 2: TV = 1, OV = 0, O3 = 0, V3 = 1, O2 = 0, V2 = 1, O1 = 0, V1 = 1, O0 = 0, V0 = 1
13	3339	(slave 3 is a 4-channel output slave) output data AS-i slave 3, channel 0
14	1102	output data AS-i slave 3, channel 1
15	1953	output data AS-i slave 3, channel 2
16	1234	output data AS-i slave 3, channel 3
17	0055	**o**verflow and **v**alid bits for AS-i slave 3: TV = 1, OV = 1, O3 = 0, V3 = 1, O2 = 0, V2 = 1, O1 = 0, V1 = 1, O0 = 0, V0 = 1
18	0000	Not used

Since this flag "TV" evaluates the data transfer cycle which was last completed the response is delayed by up to 140 ms.

7.2.10 Module 19, command 21_{dec} (15_{hex}): read the ID string of an AS-i slave with the profile S-7.4

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	res. = 0		M	UID					Command number = 15_{hex} = 21_{dec}							
2	res. = 0		ST	SLA					res. = 0			DL				
3...18	Not used								Not used							

Legend:

M	master no.	1 bit	0 = master 1 1 = master 2
UID	user ID	5 bits	00…$1F_{hex}$ = 0…31_{dec} (a change to the user ID starts the command call)
ST	slave type	1 bit	0 = single slave or A slave 1 = B slave (addition of 20_{hex} or 32_{dec} to the slave address)
SLA	slave address	5 bits	00…$1F_{hex}$ = 0…31_{dec}
DL	Data length	5 bits	00…$1F_{hex}$ = 0…31_{dec}

Response from controllerE to host in the normal case

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	S	M	Reflected user ID					Reflected command number							
2	TG	res.	AS-i slave address					F	res. = 0			DL				
3	I/O	2D	DT-Start			DT-Count			Mux field			E type				
4	number of parameters to be read								EDT Read			res. = 0		diag	res. = 0	
5	EDT Write			Reserved = 0					number of parameters to be written							
6	device-specific information								manufacturer identification							
7...16	device-specific information								device-specific information							
17	Reserved = 0								number of bytes received							
18	Not changed								Not changed							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
S	Sequence bit	1 bit	A large data packet is transmitted in several partial sequences: 0 = data transmission completed. 1 = data transmission not yet completed, at least one more packet follows.
M	master no.	1 bit	0 = master 1 1 = master 2
TG	toggle bit	1 bit	value changes for each execution of the command
F	Error bit	1 bit	0 = no error detected 1 = error when executing the command
DL	Data length	5 bits	00…$1F_{hex}$ = 0…31_{dec}

I/O	Direction of data	1 bit	direction of data for the devices with E type ≠ 3 0 = input 1 = output
2D	double data transfer	1 bit	double data transfer (redundancy) possible 0 = simple data transfer 1 = double data transfer
DT-Start	start triple	3 bits	(information for the driver in the master)
DT-Count	number of data triples	3 bits	(information for the driver in the master)
Mux field	number of multiplexed data words	3 bits	0…3 number = value in "Mux field" +1
E type	slave function + data structure	5 bits	characterises the slave as regards functionality and data structure 00 = reserved 01 = transmitted values are measured values 02 = transmitted values are 16 digital bit values 03 = normal operation in 4-bit mode (4I/4O) $04...1F_{hex} = 04...31_{dec}$ = reserved
	number of parameters to be read	1 byte	number of bytes which can be read as parameter string 00 = no parameter string readable $01...DB_{hex} = 01...219_{dec}$ = number of bytes
EDT Read	reserved	3 bits	reserved for later profiles
diag	slave supports the 7.4 diagnosis string	1 bit	0 = diagnosis string is not supported 1 = diagnosis string is supported
EDT Write	reserved	3 bits	reserved for later profiles
	number of parameters to be written	1 byte	number of bytes which can be written as parameter string 0 = no parameter string readable $01...DB_{hex} = 01...219_{dec}$ = number of bytes
	device-specific information	1 byte	as an option more bytes for the manufacturer-specific device description
	manufacturer identification	1 byte	defined manufacturer number assigned by AS-International

Response from controllerE to host in case of a fault

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	S	M	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3	Reserved = 0								Error code							
4...18	Not changed								Not changed							

Possible error codes:

Value [hex.]	Meaning
0C	Faulty S-7.4 protocol sequence
0D	S-7.4 protocol aborted (timeout)
0E	Invalid AS-i slave address for the S-7.4 protocol (e.g. B slaves)
0F	AS-i slave has completed the S-7.4 string
10	AS-i S-7.4 no longer connected (no longer in LAS)
11	Another S-7.4 transfer to the addressed AS-i slave is already active
12	The previous segmented S-7.4 transfer was not yet completed
13	Invalid S-7.4 data length
14	Invalid S-7.4 command

*) here: master is not in the normal mode

Example: read ID string of AS-i slave 3(A) on AS-i master 1

Request of host to controllerE

Word no.	Value [hex.]	Meaning
1	0215	M = 0: AS-i master 1 UID = 02: user ID changes to 2 15 = command 21
2	0300	03_{hex} = 03_{dec} = slave address 3(A)
3…18	0000	Not used

Response from controllerE to host in the normal case

Word no.	Value [hex.]	Meaning
1	0215	Copy of the request Command processed, no error occurred
2	0608 8608	0x/8x = the toggle bit changes after each execution x6 = slave address 3(A), shifted 1 bit to the left 08 ⇒ 8 data bytes ID strings have been received
3	2D01	1st word of the ID string of slave 3: $2D01_{hex}$ = 0 0 101 101 000 00001_{bin}
4	0203	2nd word of the ID string of slave 3: 0203_{hex} = 00000010 000 00 0 11_{bin}
…	…	etc.
17	0008	08 = unit sends an ID string with 8 bytes length
18	0000	Not changed

Response from controllerE to host in case of a fault

Word no.	Value [hex.]	Meaning
1	8215	Copy of the request E=1: Error when executing the command
2	0000	Reserved
3	0014	14 = error code: invalid S-7.4 command / master not in the normal mode
4…18	0000	Not changed

7.2.11 Module 19, command 28_{dec} ($1C_{hex}$): no slave reset when changing to the protected mode

Normally, when changing from projection mode to protected mode, all slaves are briefly reset (reset or offline phase). This may lead to problems when the system is running. In such cases the "deactivation of the slave reset" prevents the short-term deactivation of the slave outputs during changing of the operating mode.

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	res. = 0		M	UID					Command number = $1C_{hex}$ = 28_{dec}							
2	Reserved = 0								Reserved = 0							
3	Reserved = 0								OLP							
4...18	Not used								Not used							

Legend:

M	master no.	1 bit	0 = master 1 1 = master 2
UID	user ID	5 bits	00…$1F_{hex}$ = 0…31_{dec} (a change to the user ID starts the command call)
OLP	offline phase = slave reset	1 byte	00 = offline phase when changing over to the protected mode 01 = no offline phase when changing over to the protected mode

Response from controllerE to host

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2...18	Not changed								Not changed							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used
M	master no.	1 bit	0 = master 1 1 = master 2

Example: master 1 – no slave reset when changing to the protected mode

Request of host to controllerE

Word no.	Value [hex.]	Meaning
1	041C	M = 0: AS-i master 1 UID = 04: user ID changes to 4 1C = command 28
2	0000	Reserved
3	0001	01 = no offline phase when changing to the protected mode
4…18	0000	Not used

Response from controllerE to host

Word no.	Value [hex.]	Meaning
1	041C	Copy of the request Command processed, no error occurred
2…18	0000	Not changed

7.2.12 Module 19, command 31_{dec} ($1F_{hex}$): one-time execution of the Extended safety monitor protocol in the Safety-at -Work monitor

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	res. = 0		M	UID					Command number = $1F_{hex}$ = 31_{dec}							
2	Reserved = 0								Reserved = 0							
3	sub command								res. = 0		ST	SLA				
4...16	Not used								Not used							
17	field number								data length							
18	Not used								Not used							

Legend:

M	master no.	1 bit	0 = master 1 1 = master 2
UID	user ID	5 bits	00…$1F_{hex}$ = 0…31_{dec} (a change to the user ID starts the command call)
ST	slave type	1 bit	0 = single slave or A slave 1 = B slave (addition of 20_{hex} or 32_{dec} to the slave address)
SLA	slave address	5 bits	00…$1F_{hex}$ = 0…31_{dec}

Response from controllerE to host in the normal case

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3	reflected sub command								Reflected slave address							
4	LEDs OSSD 1				LEDs OSSD 2				data call 1				data call 0			
5	OSSD2 not green								OSSD1 not green							
6	1st colour output circuit 1								1st module address output circuit 1							
7	2nd colour output circuit 1								2nd module address output circuit 1							
8	3rd colour output circuit 1								3rd module address output circuit 1							
9	4th colour output circuit 1								4th module address output circuit 1							
10	5th colour output circuit 1								5th module address output circuit 1							
11	6th colour output circuit 1								6th module address output circuit 1							
12	1st colour output circuit 2								1st module address output circuit 2							
13	2nd colour output circuit 2								2nd module address output circuit 2							
14	3rd colour output circuit 2								3rd module address output circuit 2							
15	4th colour output circuit 2								4th module address output circuit 2							
16	5th colour output circuit 2								5th module address output circuit 2							
17	6th colour output circuit 2								6th module address output circuit 2							
18	field number = 0/1								Reserved = 0							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used
M	master no.	1 bit	0 = master 1 1 = master 2

Description of the different fields in word no. 4 for [LEDs OSSD 1/2]

Value [hex.]	Value [bin.]	Meaning
0	0000	green = contacts of the output circuits closed
1	0001	yellow = startup / restart disable active
2	0010	yellow flashing or red: = contacts of the output circuits open
3	0011	Red flashing = error on the level of the monitored AS-i components
> 4	≥ 0100	reserved

Description of the different fields in word no. 4 for [data call 1/2]

data call 1		data call 0		Meaning
Value [hex.]	Value [bin.]	Value [hex.]	Value [bin.]	
8	1000	0	0000	protective operation, everything ok (also not available, not configured or depending output circuits are displayed as "ok")
9	1001	1	0001	protective operation, output circuit 1 off.
A	1010	2	0010	protective operation, output circuit 2 off.
B	1011	3	0011	protective operation, both output circuits off.
C	1100	4	0100	configuration operation: power on
D	1101	5	0101	configuration operation
E	1110	6	0110	reserved / not defined
F	1111	7	0111	configuration operation: fatal device error, RESET or device exchange required.
-	-	≥ 8	1xxx	no current diagnosis information available, please wait.

Description of the different fields in word no. 5 for [OSSD1/2 not green]

Value [hex.]	Value [bin.]	Meaning
0	0000	no module, responses of the data calls in the words 6...17 are not relevant
1	0001	the number of POUs in the output circuit is 1
…	…	…
6	0110	the number of POUs in the output circuit is 6
7	0111	the number of POUs in the output circuit is > 6
≥ 8	1xxx	reserved / not defined

Description of the different fields in word no. 6...17 for [colour output circuit]

Module address 1...6 in output circuit 1/2: Indicates the index of the module of the configuration. The module address which was defined in the program ASIMON is indicated.

Value [hex.]	Value [bin.]	Meaning
0	0000	green, continuous
1	0001	green, flashing
2	0010	yellow, continuous
3	0011	yellow, flashing
4	0100	red, continuous
5	0101	red, flashing
6	0110	grey, off

Response from controllerE to host in case of a fault

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3	Reserved = 0								Error code							
4...18	Not changed								Not changed							

Possible error codes:

Value [hex.]	Meaning
00…02	General errors during command processing
0A…0C	Internal protocol error
10	Sub command invalid
11	No slave with the profile S-7.F.F on the slave address
16	The monitor with the address was changed in the protocol mode
20	It was not possible to process the command within the specified time
EE	Fatal error during command execution

Example: one-time execution of the extended safety monitor protocol on address 30

Request of host to controllerE

Word no.	Value [hex.]	Meaning
1	071F	M = 0: AS-i master 1 UID = 07: user ID changes to 7 1F = command 31
2	0000	Reserved
3	001E	00 = sub command 0 = one-time execution of the extended safety monitor protocol $1E_{hex} = 30_{dec}$ = Safety-at-Work monitor with the slave address 30
4…18	0000	Not used

Response from controllerE to host in the normal case: Safety-at-Work monitor has not triggered

Word no.	Value [hex.]	Meaning
1	071F	Copy of the request Command processed, no error occurred
2	0000	Reserved
3	001E	Reflected command data
4	0000	Green: contacts of the output circuits closed
5	0000	both output circuits green
6…17	xxxx	not relevant, because 5th word = 0000
18	0000	Not changed

Response from controllerE to host in the normal case: Safety-at-Work monitor has triggered

Word no.	Value [hex.]	Meaning
1	071F	Copy of the request Command processed, no error occurred
2	0000	Reserved
3	001E	Reflected command data
4	2211	2x = output circuit 1 red; x2 = invalid, → word 5; 11 = protective operation, output circuit 1 off
5	0003	00 = OSSD2 green 03 = OSSD1 not green, provides 3 modules which are not green
6	0421	04 = red permanently lit 21 = module 33
7	0422	04 = red permanently lit 22 = module 34
8	0423	04 = red permanently lit 23 = module 35
9…11	xxxx	not relevant, because low byte of 5th word = 03 ⇒ 3 modules relevant
12…17	xxxx	not relevant, because high byte of 5th word = 00 ⇒ green, no module relevant
18	0100	01 = field nubmer 1

Response from controllerE to host in case of a fault

Word no.	Value [hex.]	Meaning
1	871F	Copy of the request E=1: Error when executing the command
2	0000	Reserved
3	0011	11 = error code: no slave with the profile S-7.F.F on the slave address
4…18	0000	Not changed

7.2.13 Module 19, command 33_{dec} (21_{hex}): read the diagnosis string of an AS-i slave with profile S-7.4

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	res.	S	M	UID					Command number = 21_{hex} = 33_{dec}							
2	res. = 0		ST	SLA					res. = 0			DL				
3...18	Not used								Not used							

Legend:

S	Sequence bit	1 bit	A large data packet is transmitted in several partial sequences: 0 = data transmission completed. 1 = data transmission not yet completed, at least one more packet follows.
M	master no.	1 bit	0 = master 1 1 = master 2
UID	user ID	5 bits	00…$1F_{hex}$ = 0…31_{dec} (a change to the user ID starts the command call)
ST	slave type	1 bit	0 = single slave or A slave 1 = B slave (addition of 20_{hex} or 32_{dec} to the slave address)
SLA	slave address	5 bits	00…$1F_{hex}$ = 0…31_{dec}
DL	Data length	5 bits	00…$1F_{hex}$ = 0…31_{dec}

Response from controllerE to host

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	S	M	Reflected user ID					Reflected command number							
2	TG	res.	AS-i slave address					F	res. = 0			DL				
3	diagnosis string 1								diagnosis string 0							
4...16	diagnosis strings 2…27															
17	diagnosis string 29								diagnosis string 28							
18	Not changed								Not changed							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
S	Sequence bit	1 bit	A large data packet is transmitted in several partial sequences: 0 = data transmission completed. 1 = data transmission not yet completed, at least one more packet follows.
M	master no.	1 bit	0 = master 1 1 = master 2
TG	toggle bit	1 bit	value changes for each execution of the command
F	Error bit	1 bit	0 = no error detected 1 = error when executing the command
DL	Data length	5 bits	00…$1F_{hex}$ = 0…31_{dec}

Info

The control bytes defined in profile 7.4 with follow and valid bits are filtered out by the system.

Example: read diagnosis string of AS-i slave 3(A) on AS-i master 1

Request of host to controllerE

Word no.	Value [hex.]	Meaning
1	0721	S=0: sequence here always 0 M = 0: AS-i master 1 UID = 07: user ID changes to 7 21 = command 33
2	0300	03_{hex} = 03_{dec} = slave address 3(A)
3…18	0000	Not used

Response from controllerE to host

Word no.	Value [hex.]	Meaning
1	0721	S=0: last sequence Copy of the request Command processed, no error occurred
2	0608 8608	0x/8x = the toggle bit changes after each execution x6 = slave address 3(A), shifted 1 bit to the left 08 ⇒ 8 diagnosis strings received
3	2D01	01 = diagnosis string 0 of slave 3 2D = diagnosis string 1 of slave 3
4	0203	03 = diagnosis string 2 of slave 3 02 = diagnosis string 3 of slave 3
5	1122	22 = diagnosis string 4 of slave 3 11 = diagnosis string 5 of slave 3
6	3344	44 = diagnosis string 6 of slave 3 33 = diagnosis string 7 of slave 3
7…18	0000	Not changed

7.2.14 Module 19, command 34_{dec} (22_{hex}): read the parameter string of an AS-i slave with the profile S-7.4

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	res.	S	M	UID					Command number = 22_{hex} = 34_{dec}							
2	res. = 0		ST	SLA					res. = 0			DL				
3...18	Not used								Not used							

Legend:

S	Sequence bit	1 bit	A large data packet is transmitted in several partial sequences: 0 = data transmission completed. 1 = data transmission not yet completed, at least one more packet follows.
M	master no.	1 bit	0 = master 1 1 = master 2
UID	user ID	5 bits	00…$1F_{hex}$ = 0…31_{dec} (a change to the user ID starts the command call)
ST	slave type	1 bit	0 = single slave or A slave 1 = B slave (addition of 20_{hex} or 32_{dec} to the slave address)
SLA	slave address	5 bits	00…$1F_{hex}$ = 0…31_{dec}
DL	Data length	5 bits	00…$1F_{hex}$ = 0…31_{dec}

Response from controllerE to host

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	S	M	Reflected user ID					Reflected command number							
2	TG	res.	AS-i slave address					F	res. = 0			DL				
3	parameter string 1								parameter string 0							
4...16	parameter string 2…27															
17	parameter string 29								parameter string 28							
18	Not changed								Not changed							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
S	Sequence bit	1 bit	A large data packet is transmitted in several partial sequences: 0 = data transmission completed. 1 = data transmission not yet completed, at least one more packet follows.
M	master no.	1 bit	0 = master 1 1 = master 2
TG	toggle bit	1 bit	value changes for each execution of the command
F	Error bit	1 bit	0 = no error detected 1 = error when executing the command
DL	Data length	5 bits	00…$1F_{hex}$ = 0…31_{dec}

Info

The control bytes defined in profile 7.4 with follow and valid bits are filtered out by the system.

Example: read parameter string of AS-i slave 3(A) on AS-i master 1

Request of host to controllerE

Word no.	Value [hex.]	Meaning
1	0822	S=0: sequence here always 0 M = 0: AS-i master 1 UID = 08: user ID changes to 8 22 = command 34
2	0300	03_{hex} = 03_{dec} = slave address 3(A)
3…18	0000	Not used

Response from controllerE to host

Word no.	Value [hex.]	Meaning
1	0822	S=0: last sequence Copy of the request Command processed, no error occurred
2	0608 8608	0x/8x = the toggle bit changes after each execution x6 = slave address 3(A), shifted 1 bit to the left 08 ⇒ 8 parameter strings received
3	1234	34 = parameter string 0 of slave 3 12 = parameter string 1 of slave 3
4	5678	78 = parameter string 2 of slave 3 56 = parameter string 3 of slave 3
5	1234	34 = parameter string 4 of slave 3 12 = parameter string 5 of slave 3
6	5678	78 = parameter string 6 of slave 3 56 = parameter string 7 of slave 3
7…18	0000	Not changed

7.2.15 Module 19, command 35_{dec} (23_{hex}): write parameter string of an AS-i slave with the profile S-7.4

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	res.	S	M	UID					Command number = 23_{hex} = 35_{dec}							
2	res. = 0		ST	SLA					res. = 0			DL				
3	parameter string 1								parameter string 0							
4...11	parameter string 2…17															
12	parameter string 19								parameter string 18							
13...18	Not used								Not used							

Legend:

S	Sequence bit	1 bit	A large data packet is transmitted in several partial sequences: 0 = data transmission completed. 1 = data transmission not yet completed, at least one more packet follows.
M	master no.	1 bit	0 = master 1 1 = master 2
UID	user ID	5 bits	00…$1F_{hex}$ = 0…31_{dec} (a change to the user ID starts the command call)
ST	slave type	1 bit	0 = single slave or A slave 1 = B slave (addition of 20_{hex} or 32_{dec} to the slave address)
SLA	slave address	5 bits	00…$1F_{hex}$ = 0…31_{dec}
DL	Data length	5 bits	00…$1F_{hex}$ = 0…31_{dec}

Response from controllerE to host

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	S	M	Reflected user ID					Reflected command number							
2	TG	res.	AS-i slave address					F	res. = 0			DL				
3...18	Not changed								Not changed							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
S	Sequence bit	1 bit	A large data packet is transmitted in several partial sequences: 0 = data transmission completed. 1 = data transmission not yet completed, at least one more packet follows.
M	master no.	1 bit	0 = master 1 1 = master 2
TG	toggle bit	1 bit	value changes for each execution of the command
F	Error bit	1 bit	0 = no error detected 1 = error when executing the command
DL	Data length	5 bits	00…$1F_{hex}$ = 0…31_{dec}

NOTE

The number of the bytes to be sent must be divisible by 2 since the system always transmits only multiples of 2 bytes in the S7.4 protocol.

The control bytes defined in profile 7.4 with follow bit and valid bit are automatically added by the system. Therefore, without segmentation, this command is limited to 20 bytes of parameter data. Larger data volumes must be divided into several segments.

Example: write parameter string in AS-i slave 3(A) on AS-i master 1

Request of host to controllerE

Word no.	Value [hex.]	Meaning
1	0923	S=0: sequence here always 0 M = 0: AS-i master 1 UID = 09: user ID changes to 9 23 = command 35
2	0304	03_{hex} = 03_{dec} = slave address 3(A) 04 ⇒ send 4 parameter strings
3	1AF4	F4 = parameter string 0 for slave 3 2D = parameter string 1 for slave 3
4	5BB8	B8 = parameter string 2 for slave 3 5B = parameter string 3 for slave 3
5…18	0000	Not used

Response from controllerE to host

Word no.	Value [hex.]	Meaning
1	0923	S=0: last sequence Copy of the request Command processed, no error occurred
2	0604 8604	0x/8x = the toggle bit changes after each execution x6 = slave address 3(A), shifted 1 bit to the left 04 = number of bytes to be received
3…18	0000	Not changed

7.2.16 Module 19, command 50_{dec} (32_{hex}): read current configuration AS-i slaves 0(A)...15(A)

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	res. = 0		M	UID					Command number = 32_{hex} = 50_{dec}							
2	Reserved = 0								Reserved = 0							
3...18	Not used								Not used							

Legend:

M	master no.	1 bit	0 = master 1 1 = master 2
UID	user ID	5 bits	00…$1F_{hex}$ = 0…31_{dec} (a change to the user ID starts the command call)

Response from controllerE to host

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3	Slave 0(A), ID2				Slave 0(A), ID1				Slave 0(A), ID code				Slave 0(A), IO config.			
4	Slave 1(A), ID2				Slave 1(A), ID1				Slave 1(A), ID code				Slave 1(A), IO config.			
5	Slave 2(A), ID2				Slave 2(A), ID1				Slave 2(A), ID code				Slave 2(A), IO config.			
6	Slave 3(A), ID2				Slave 3(A), ID1				Slave 3(A), ID code				Slave 3(A), IO config.			
7	Slave 4(A), ID2				Slave 4(A), ID1				Slave 4(A), ID code				Slave 4(A), IO config.			
8	Slave 5(A), ID2				Slave 5(A), ID1				Slave 5(A), ID code				Slave 5(A), IO config.			
9	Slave 6(A), ID2				Slave 6(A), ID1				Slave 6(A), ID code				Slave 6(A), IO config.			
10	Slave 7(A), ID2				Slave 7(A), ID1				Slave 7(A), ID code				Slave 7(A), IO config.			
11	Slave 8(A), ID2				Slave 8(A), ID1				Slave 8(A), ID code				Slave 8(A), IO config.			
12	Slave 9(A), ID2				Slave 9(A), ID1				Slave 9(A), ID code				Slave 9(A), IO config.			
13	Slave 10(A), ID2				Slave 10(A), ID1				Slave 10(A), ID code				Slave 10(A), IO config.			
14	Slave 11(A), ID2				Slave 11(A), ID1				Slave 11(A), ID code				Slave 11(A), IO config.			
15	Slave 12(A), ID2				Slave 12(A), ID1				Slave 12(A), ID code				Slave 12(A), IO config.			
16	Slave 13(A), ID2				Slave 13(A), ID1				Slave 13(A), ID code				Slave 13(A), IO config.			
17	Slave 14(A), ID2				Slave 14(A), ID1				Slave 14(A), ID code				Slave 14(A), IO config.			
18	Slave 15(A), ID2				Slave 15(A), ID1				Slave 15(A), ID code				Slave 15(A), IO config.			

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used
M	master no.	1 bit	0 = master 1 1 = master 2

Example: read current configuration AS-i slaves 0(A)...15(A) on AS-i master 1

Request of host to controllerE

Word no.	Value [hex.]	Meaning
1	0232	M = 0: AS-i master 1 UID = 02: user ID changes to 2 32 = command 50
2…18	0000	Not used

Response from controllerE to host

Word no.	Value [hex.]	Meaning
1	0232	S=0: last sequence Copy of the request Command processed, no error occurred
2	00FF	Reserved
3	FFFF	current configuration slave 0: ID2 =F, ID1=F, ID=F and IO=F
4	EF03	current configuration slave 1(A): ID2 =E, ID1=F, ID=0 and IO=3
…	…	…
18	EF37	current configuration slave 15(A): ID2 =E, ID1=F, ID=3 and IO=7

7.2.17 Module 19, command 51_{dec} (33_{hex}): read current configuration AS-i slaves 16(A)...31(A)

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	res. = 0		M	UID					Command number = 33_{hex} = 51_{dec}							
2	Reserved = 0								Reserved = 0							
3...18	Not used								Not used							

Legend:

M	master no.	1 bit	0 = master 1 1 = master 2
UID	user ID	5 bits	00…$1F_{hex}$ = 0…31_{dec} (a change to the user ID starts the command call)

Response from controllerE to host

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3	Slave 16(A), ID2				Slave 16(A), ID1				Slave 16(A), ID code				Slave 16(A), IO config.			
4	Slave 17(A), ID2				Slave 17(A), ID1				Slave 17(A), ID code				Slave 17(A), IO config.			
5	Slave 18(A), ID2				Slave 18(A), ID1				Slave 18(A), ID code				Slave 18(A), IO config.			
6	Slave 19(A), ID2				Slave 19(A), ID1				Slave 19(A), ID code				Slave 19(A), IO config.			
7	Slave 20(A), ID2				Slave 20(A), ID1				Slave 20(A), ID code				Slave 20(A), IO config.			
8	Slave 21(A), ID2				Slave 21(A), ID1				Slave 21(A), ID code				Slave 21(A), IO config.			
9	Slave 22(A), ID2				Slave 22(A), ID1				Slave 22(A), ID code				Slave 22(A), IO config.			
10	Slave 23(A), ID2				Slave 23(A), ID1				Slave 23(A), ID code				Slave 23(A), IO config.			
11	Slave 24(A), ID2				Slave 24(A), ID1				Slave 24(A), ID code				Slave 24(A), IO config.			
12	Slave 25(A), ID2				Slave 25(A), ID1				Slave 25(A), ID code				Slave 25(A), IO config.			
13	Slave 26(A), ID2				Slave 26(A), ID1				Slave 26(A), ID code				Slave 26(A), IO config.			
14	Slave 27(A), ID2				Slave 27(A), ID1				Slave 27(A), ID code				Slave 27(A), IO config.			
15	Slave 28(A), ID2				Slave 28(A), ID1				Slave 28(A), ID code				Slave 28(A), IO config.			
16	Slave 29(A), ID2				Slave 29(A), ID1				Slave 29(A), ID code				Slave 29(A), IO config.			
17	Slave 30(A), ID2				Slave 30(A), ID1				Slave 30(A), ID code				Slave 30(A), IO config.			
18	Slave 31(A), ID2				Slave 31(A), ID1				Slave 31(A), ID code				Slave 31(A), IO config.			

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used
M	master no.	1 bit	0 = master 1 1 = master 2

7.2.18 Module 19, command 52_{dec} (34_{hex}): read current configuration AS-i slaves 0...15B

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	res. = 0		M	UID					Command number = 34_{hex} = 52_{dec}							
2	Reserved = 0								Reserved = 0							
3...18	Not used								Not used							

Legend:

M	master no.	1 bit	0 = master 1 1 = master 2
UID	user ID	5 bits	00…$1F_{hex}$ = 0…31_{dec} (a change to the user ID starts the command call)

Response from controllerE to host

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3	Slave 0B, ID2				Slave 0B, ID1				Slave 0B, ID code				Slave 0B, ID code			
4	Slave 1B, ID2				Slave 1B, ID1				Slave 1B, ID code				Slave 1B, IO config.			
5	Slave 2B, ID2				Slave 2B, ID1				Slave 2B, ID code				Slave 2B, IO config.			
6	Slave 3B, ID2				Slave 3B, ID1				Slave 3B, ID code				Slave 3B, IO config.			
7	Slave 4B, ID2				Slave 4B, ID1				Slave 4B, ID code				Slave 4B, IO config.			
8	Slave 5B, ID2				Slave 5B, ID1				Slave 5B, ID code				Slave 5B, IO config.			
9	Slave 6B, ID2				Slave 6B, ID1				Slave 6B, ID code				Slave 6B, IO config.			
10	Slave 7B, ID2				Slave 7B, ID1				Slave 7B, ID code				Slave 7B, IO config.			
11	Slave 8B, ID2				Slave 8B, ID1				Slave 8B, ID code				Slave 8B, IO config.			
12	Slave 9B, ID2				Slave 9B, ID1				Slave 9B, ID code				Slave 9B, IO config.			
13	Slave 10B, ID2				Slave 10B, ID1				Slave 10B, ID code				Slave 10B, IO config.			
14	Slave 11B, ID2				Slave 11B, ID1				Slave 11B, ID code				Slave 11B, IO config.			
15	Slave 12B, ID2				Slave 12B, ID1				Slave 12B, ID code				Slave 12B, IO config.			
16	Slave 13B, ID2				Slave 13B, ID1				Slave 13B, ID code				Slave 13B, IO config.			
17	Slave 14B, ID2				Slave 14B, ID1				Slave 14B, ID code				Slave 14B, IO config.			
18	Slave 15B, ID2				Slave 15B, ID1				Slave 15B, ID code				Slave 15B, IO config.			

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used
M	master no.	1 bit	0 = master 1 1 = master 2

7.2.19 Module 19, command 54_{dec} (36_{hex}): read current parameters of the AS-i slaves

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	res. = 0		M	UID					Command number = 36_{hex} = 54_{dec}							
2...18	Not used								Not used							

Legend:

M	master no.	1 bit	0 = master 1 1 = master 2
UID	user ID	5 bits	00…$1F_{hex}$ = 0…31_{dec} (a change to the user ID starts the command call)

Response from controllerE to host

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3	Parameter Slave 4(A)				Parameter Slave 3(A)				Parameter Slave 2(A)				Parameter Slave 1(A)			
4	Parameter Slave 8(A)				Parameter Slave 7(A)				Parameter Slave 6(A)				Parameter Slave 5(A)			
5	Parameter Slave 12(A)				Parameter Slave 11(A)				Parameter Slave 10(A)				Parameter Slave 9(A)			
6	Parameter Slave 16(A)				Parameter Slave 15(A)				Parameter Slave 14(A)				Parameter Slave 13(A)			
7	Parameter Slave 20(A)				Parameter Slave 19(A)				Parameter Slave 18(A)				Parameter Slave 17(A)			
8	Parameter Slave 24(A)				Parameter Slave 23(A)				Parameter Slave 22(A)				Parameter Slave 21(A)			
9	Parameter Slave 28(A)				Parameter Slave 27(A)				Parameter Slave 26(A)				Parameter Slave 25(A)			
10	Parameter Slave 1B				Parameter Slave 31(A)				Parameter Slave 30(A)				Parameter Slave 29(A)			
11	Parameter Slave 5B				Parameter Slave 4B				Parameter Slave 3B				Parameter Slave 2B			
12	Parameter Slave 9B				Parameter Slave 8B				Parameter Slave 7B				Parameter Slave 6B			
13	Parameter Slave 13B				Parameter Slave 12B				Parameter Slave 11B				Parameter Slave 10B			
14	Parameter Slave 17B				Parameter Slave 16B				Parameter Slave 15B				Parameter Slave 14B			
15	Parameter Slave 21B				Parameter Slave 20B				Parameter Slave 19B				Parameter Slave 18B			
16	Parameter Slave 25B				Parameter Slave 24B				Parameter Slave 23B				Parameter Slave 22B			
17	Parameter Slave 29B				Parameter Slave 28B				Parameter Slave 27B				Parameter Slave 26B			
18	Not changed				Not changed				Parameter Slave 31B				Parameter Slave 30B			

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used
M	master no.	1 bit	0 = master 1 1 = master 2

Example: read current parameters of the AS-i slaves on AS-i master 1

Request of host to controllerE

Word no.	Value [hex.]	Meaning
1	0636	M = 0: AS-i master 1 UID = 06: user ID changes to 6 36 = command 54
2…18	0000	Not used

Response from controllerE to host

Word no.	Value [hex.]	Meaning
1	0636	S=0: last sequence Copy of the request Command processed, no error occurred
2	00FF	Reserved
3	4321	1 = parameter of slave 1(A) 2 = parameter of slave 2(A) 3 = parameter of slave 3(A) 4 = parameter of slave 4(A)
4	8765	5 = parameter of slave 5(A) 6 = parameter of slave 6(A) 7 = parameter of slave 7(A) 8 = parameter of slave 8(A)
…	…	…
18	0098	8 = parameter of slave 30(A) 9 = parameter of slave 31B

7.2.20 Module 19, command 55_{dec} (37_{hex}): read current AS-i slave lists

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	res. = 0		M	UID					Command number = 37_{hex} = 55_{dec}							
2...18	Not used								Not used							

Legend:

M	master no.	1 bit	0 = master 1 1 = master 2
UID	user ID	5 bits	00…$1F_{hex}$ = 0…31_{dec} (a change to the user ID starts the command call)

Response from controllerE to host

Word no.		bit (AS-i slave address)															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1		E	B	M	Reflected slave address					Reflected command number							
2		Reserved = 0								Reserved = 0							
3	LAS	15(A)	14(A)	13(A)	12(A)	11(A)	10(A)	9(A)	8(A)	7(A)	6(A)	5(A)	4(A)	3(A)	2(A)	1(A)	res
4		31(A)	30(A)	29(A)	28(A)	27(A)	26(A)	25(A)	24(A)	23(A)	22(A)	21(A)	20(A)	19(A)	18(A)	17(A)	16(A)
5		15B	14B	13B	12B	11B	10B	9B	8B	7B	6B	5B	4B	3B	2B	1B	res
6		31B	30B	29B	28B	27B	26B	25B	24B	23B	22B	21B	20B	19B	18B	17B	16B
7	LDS	15(A)	14(A)	13(A)	12(A)	11(A)	10(A)	9(A)	8(A)	7(A)	6(A)	5(A)	4(A)	3(A)	2(A)	1(A)	0
8		31(A)	30(A)	29(A)	28(A)	27(A)	26(A)	25(A)	24(A)	23(A)	22(A)	21(A)	20(A)	19(A)	18(A)	17(A)	16(A)
9		15B	14B	13B	12B	11B	10B	9B	8B	7B	6B	5B	4B	3B	2B	1B	res
10		31B	30B	29B	28B	27B	26B	25B	24B	23B	22B	21B	20B	19B	18B	17B	16B
11	LPF	15(A)	14(A)	13(A)	12(A)	11(A)	10(A)	9(A)	8(A)	7(A)	6(A)	5(A)	4(A)	3(A)	2(A)	1(A)	res
12		31(A)	30(A)	29(A)	28(A)	27(A)	26(A)	25(A)	24(A)	23(A)	22(A)	21(A)	20(A)	19(A)	18(A)	17(A)	16(A)
13		15B	14B	13B	12B	11B	10B	9B	8B	7B	6B	5B	4B	3B	2B	1B	res
14		31B	30B	29B	28B	27B	26B	25B	24B	23B	22B	21B	20B	19B	18B	17B	16B
15	LPS	15(A)	14(A)	13(A)	12(A)	11(A)	10(A)	9(A)	8(A)	7(A)	6(A)	5(A)	4(A)	3(A)	2(A)	1(A)	res
16		31(A)	30(A)	29(A)	28(A)	27(A)	26(A)	25(A)	24(A)	23(A)	22(A)	21(A)	20(A)	19(A)	18(A)	17(A)	16(A)
17		15B	14B	13B	12B	11B	10B	9B	8B	7B	6B	5B	4B	3B	2B	1B	res
18		31B	30B	29B	28B	27B	26B	25B	24B	23B	22B	21B	20B	19B	18B	17B	16B

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used
M	master no.	1 bit	0 = master 1 1 = master 2

Example: read current AS-i slave lists

Request of host to controllerE

Word no.	Value [hex.]	Meaning
1	0737	M = 0: AS-i master 1 UID = 07: user ID changes to 7 37 = command 55
2…18	0000	Not used

Response from controllerE to host

Word no.	Value [hex.]	Meaning
1	0737	S=0: last sequence Copy of the request Command processed, no error occurred
2	00FF	Reserved
3	0102	LAS of slaves (0) to 15(A) 0102_{hex} = 0000 0001 0000 0010_{bin} ⇒ slaves 1(A) and 8(A) are active
4	8001	LAS of slaves 16(A) to 31(A) 8001_{hex} = 1000 0000 0000 0001_{bin} ⇒ slaves 16(A) and 31(A) are active
…	…	…
18	8001	LPS of slaves 16B to 31B 8001_{hex} = 1000 0000 0000 0001_{bin} ⇒ slaves 16B and 31B are projected

7.2.21 Module 19, command 56_{dec} (38_{hex}): read projected configuration AS-i slaves 1(A)...15(A)

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	res. = 0		M	UID					Command number = 38_{hex} = 56_{dec}							
2...18	Not used								Not used							

Legend:

M	master no.	1 bit	0 = master 1 1 = master 2
UID	user ID	5 bits	00…$1F_{hex}$ = 0…31_{dec} (a change to the user ID starts the command call)

Response from controllerE to host

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3	Slave 0, ID2				Slave 0, ID1				Slave 0, ID code				Slave 0, IO config.			
4	Slave 1(A), ID2				Slave 1(A), ID1				Slave 1(A), ID code				Slave 1(A), IO config.			
5	Slave 2(A), ID2				Slave 2(A), ID1				Slave 2(A), ID code				Slave 2(A), IO config.			
6	Slave 3(A), ID2				Slave 3(A), ID1				Slave 3(A), ID code				Slave 3(A), IO config.			
7	Slave 4(A), ID2				Slave 4(A), ID1				Slave 4(A), ID code				Slave 4(A), IO config.			
8	Slave 5(A), ID2				Slave 5(A), ID1				Slave 5(A), ID code				Slave 5(A), IO config.			
9	Slave 6(A), ID2				Slave 6(A), ID1				Slave 6(A), ID code				Slave 6(A), IO config.			
10	Slave 7(A), ID2				Slave 7(A), ID1				Slave 7(A), ID code				Slave 7(A), IO config.			
11	Slave 8(A), ID2				Slave 8(A), ID1				Slave 8(A), ID code				Slave 8(A), IO config.			
12	Slave 9(A), ID2				Slave 9(A), ID1				Slave 9(A), ID code				Slave 9(A), IO config.			
13	Slave 10(A), ID2				Slave 10(A), ID1				Slave 10(A), ID code				Slave 10(A), IO config.			
14	Slave 11(A), ID2				Slave 11(A), ID1				Slave 11(A), ID code				Slave 11(A), IO config.			
15	Slave 12(A), ID2				Slave 12(A), ID1				Slave 12(A), ID code				Slave 12(A), IO config.			
16	Slave 13(A), ID2				Slave 13(A), ID1				Slave 13(A), ID code				Slave 13(A), IO config.			
17	Slave 14(A), ID2				Slave 14(A), ID1				Slave 14(A), ID code				Slave 14(A), IO config.			
18	Slave 15(A), ID2				Slave 15(A), ID1				Slave 15(A), ID code				Slave 15(A), IO config.			

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used
M	master no.	1 bit	0 = master 1 1 = master 2

Example: read projected configuration AS-i slaves 0(A)...15(A) on AS-i master 1

Request of host to controllerE

Word no.	Value [hex.]	Meaning
1	0238	M = 0: AS-i master 1 UID = 02: user ID changes to 2 38 = command 56
2…18	0000	Not used

Response from controllerE to host

Word no.	Value [hex.]	Meaning
1	0238	Copy of the request Command processed, no error occurred
2	00FF	Reserved
3	FFFF	here not used since slave 0 cannot be projected
4	EF03	projected configuration slave 1(A): ID2 =E, ID1=F, ID=0 and IO=3
…	…	…
18	EF37	projected configuration slave 15(A): ID2 =E, ID1=F, ID=3 and IO=7

7.2.22 Module 19, command 57_{dec} (39_{hex}): read projected configuration AS-i slaves 16(A)...31(A)

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	res. = 0		M	UID					Command number = 39_{hex} = 57_{dec}							
2...18	Not used								Not used							

Legend:

M	master no.	1 bit	0 = master 1 1 = master 2
UID	user ID	5 bits	00…$1F_{hex}$ = 0…31_{dec} (a change to the user ID starts the command call)

Response from controllerE to host

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3	Slave 16(A), ID2				Slave 16(A), ID1				Slave 16(A), ID code				Slave 16(A), IO config.			
4	Slave 17(A), ID2				Slave 17(A), ID1				Slave 17(A), ID code				Slave 17(A), IO config.			
5	Slave 18(A), ID2				Slave 18(A), ID1				Slave 18(A), ID code				Slave 18(A), IO config.			
6	Slave 19(A), ID2				Slave 19(A), ID1				Slave 19(A), ID code				Slave 19(A), IO config.			
7	Slave 20(A), ID2				Slave 20(A), ID1				Slave 20(A), ID code				Slave 20(A), IO config.			
8	Slave 21(A), ID2				Slave 21(A), ID1				Slave 21(A), ID code				Slave 21(A), IO config.			
9	Slave 22(A), ID2				Slave 22(A), ID1				Slave 22(A), ID code				Slave 22(A), IO config.			
10	Slave 23(A), ID2				Slave 23(A), ID1				Slave 23(A), ID code				Slave 23(A), IO config.			
11	Slave 24(A), ID2				Slave 24(A), ID1				Slave 24(A), ID code				Slave 24(A), IO config.			
12	Slave 25(A), ID2				Slave 25(A), ID1				Slave 25(A), ID code				Slave 25(A), IO config.			
13	Slave 26(A), ID2				Slave 26(A), ID1				Slave 26(A), ID code				Slave 26(A), IO config.			
14	Slave 27(A), ID2				Slave 27(A), ID1				Slave 27(A), ID code				Slave 27(A), IO config.			
15	Slave 28(A), ID2				Slave 28(A), ID1				Slave 28(A), ID code				Slave 28(A), IO config.			
16	Slave 29(A), ID2				Slave 29(A), ID1				Slave 29(A), ID code				Slave 29(A), IO config.			
17	Slave 30(A), ID2				Slave 30(A), ID1				Slave 30(A), ID code				Slave 30(A), IO config.			
18	Slave 31(A), ID2				Slave 31(A), ID1				Slave 31(A), ID code				Slave 31(A), IO config.			

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used
M	master no.	1 bit	0 = master 1 1 = master 2

7.2.23 Module 19, command 58_{dec} ($3A_{hex}$): read projected configuration AS-i slaves 1B...15B

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	res. = 0		M	UID					Command number = $3A_{hex}$ = 58_{dec}							
2...18	Not used								Not used							

Legend:

M	master no.	1 bit	0 = master 1 1 = master 2
UID	user ID	5 bits	00…$1F_{hex}$ = 0…31_{dec} (a change to the user ID starts the command call)

Response from controllerE to host

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3	Reserved = 0					Reserved = 0			Reserved = 0				Reserved = 0			
4	Slave 1B, ID2					Slave 1B, ID1			Slave 1B, ID code				Slave 1B, IO config.			
5	Slave 2B, ID2					Slave 2B, ID1			Slave 2B, ID code				Slave 2B, IO config.			
6	Slave 3B, ID2					Slave 3B, ID1			Slave 3B, ID code				Slave 3B, IO config.			
7	Slave 4B, ID2					Slave 4B, ID1			Slave 4B, ID code				Slave 4B, IO config.			
8	Slave 5B, ID2					Slave 5B, ID1			Slave 5B, ID code				Slave 5B, IO config.			
9	Slave 6B, ID2					Slave 6B, ID1			Slave 6B, ID code				Slave 6B, IO config.			
10	Slave 7B, ID2					Slave 7B, ID1			Slave 7B, ID code				Slave 7B, IO config.			
11	Slave 8B, ID2					Slave 8B, ID1			Slave 8B, ID code				Slave 8B, IO config.			
12	Slave 9B, ID2					Slave 9B, ID1			Slave 9B, ID code				Slave 9B, IO config.			
13	Slave 10B, ID2					Slave 10B, ID1			Slave 10B, ID code				Slave 10B, IO config.			
14	Slave 11B, ID2					Slave 11B, ID1			Slave 11B, ID code				Slave 11B, IO config.			
15	Slave 12B, ID2					Slave 12B, ID1			Slave 12B, ID code				Slave 12B, IO config.			
16	Slave 13B, ID2					Slave 13B, ID1			Slave 13B, ID code				Slave 13B, IO config.			
17	Slave 14B, ID2					Slave 14B, ID1			Slave 14B, ID code				Slave 14B, IO config.			
18	Slave 15B, ID2					Slave 15B, ID1			Slave 15B, ID code				Slave 15B, IO config.			

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used
M	master no.	1 bit	0 = master 1 1 = master 2

7.2.24 Module 19, command 59_{dec} ($3B_{hex}$): read projected configuration AS-i slaves 16B...31B

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	res. = 0		M	UID					Command number = $3B_{hex}$ = 59_{dec}							
2...18	Not used								Not used							

Legend:

M	master no.	1 bit	0 = master 1 1 = master 2
UID	user ID	5 bits	00…$1F_{hex}$ = 0…31_{dec} (a change to the user ID starts the command call)

Response from controllerE to host

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	M	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3	Slave 16B, ID2				Slave 16B, ID1				Slave 16B, ID code				Slave 16B, IO config.			
4	Slave 17B, ID2				Slave 17B, ID1				Slave 17B, ID code				Slave 17B, IO config.			
5	Slave 18B, ID2				Slave 18B, ID1				Slave 18B, ID code				Slave 18B, IO config.			
6	Slave 19B, ID2				Slave 19B, ID1				Slave 19B, ID code				Slave 19B, IO config.			
7	Slave 20B, ID2				Slave 20B, ID1				Slave 20B, ID code				Slave 20B, IO config.			
8	Slave 21B, ID2				Slave 21B, ID1				Slave 21B, ID code				Slave 21B, IO config.			
9	Slave 22B, ID2				Slave 22B, ID1				Slave 22B, ID code				Slave 22B, IO config.			
10	Slave 23B, ID2				Slave 23B, ID1				Slave 23B, ID code				Slave 23B, IO config.			
11	Slave 24B, ID2				Slave 24B, ID1				Slave 24B, ID code				Slave 24B, IO config.			
12	Slave 25B, ID2				Slave 25B, ID1				Slave 25B, ID code				Slave 25B, IO config.			
13	Slave 26B, ID2				Slave 26B, ID1				Slave 26B, ID code				Slave 26B, IO config.			
14	Slave 27B, ID2				Slave 27B, ID1				Slave 27B, ID code				Slave 27B, IO config.			
15	Slave 28B, ID2				Slave 28B, ID1				Slave 28B, ID code				Slave 28B, IO config.			
16	Slave 29B, ID2				Slave 29B, ID1				Slave 29B, ID code				Slave 29B, IO config.			
17	Slave 30B, ID2				Slave 30B, ID1				Slave 30B, ID code				Slave 30B, IO config.			
18	Slave 31B, ID2				Slave 31B, ID1				Slave 31B, ID code				Slave 31B, IO config.			

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used
M	master no.	1 bit	0 = master 1 1 = master 2

7.2.25 Module 19, command 96_{dec} (60_{hex}): save data non-volatilely in the flash memory of the unit

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Reserved = 0			UID					Command number = 60_{hex} = 96_{dec}							
2	Reserved = 0								Reserved = 0							
3	Reserved = 0								area number							
4...18	Not used								Not used							

Legend:

UID	user ID	5 bits	00…1F_{hex} = 0…31_{dec} (a change to the user ID starts the command call)

Response from controllerE to host

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	res.	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3	Reserved = 0								reflected area number							
4...18	Not changed								Not changed							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used

Example: save configuration of AS-i master 1

Request of host to controllerE

Word no.	Value [hex.]	Meaning
1	0960	UID = 09: user ID changes to 9 60 = command 96
2	0000	Reserved
3	0002	area number = 2 ⇒ non-volatiley saves the configuration of AS-i master 1
4…18	0000	Not used

Response from controllerE to host

Word no.	Value [hex.]	Meaning
1	0960	Copy of the request Command processed, no error occurred
2	0000	Reserved
3	001E	Reflected command data
4…18	0000	Not changed

7.2.26 Module 19, command 97_{dec} (61_{hex}): carry out various settings in the controllerE

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Reserved = 0			UID					Command number = 61_{hex} = 97_{dec}							
2	Reserved = 0								Reserved = 0							
3	Reserved = 0								command number							
4	Parameter(s) (according to command number)															
5...18	Not used								Not used							

Legend:

UID	user ID	5 bits	00…$1F_{hex}$ = 0…31_{dec} (a change to the user ID starts the command call)
	command number	1 byte	10_{hex} = changes the operating mode of the PLC (without function in the gateway), (according parameters → word 4). 12_{hex} = reset all slave error counters 13_{hex} = reset configuration error counter 14_{hex} = reset AS-i cycle error counter
	Parameter for command number	1 word	Parameters; here for command number 10: 0000 = activates the gateway mode 0001 = stops the PLC 0002 = sets the operating mode of the PLC to RUN

Response from controllerE to host

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	res.	Reflected user ID					Reflected command number							
2...18	Not changed								Not changed							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used

Example: carry out settings in the controllerE

Request of host to controllerE

Word no.	Value [hex.]	Meaning
1	0961	UID = 09: user ID changes to 9 61 = command 97
2	0000	Reserved
3	0002	command number = 2 ⇒ sets the operation mode of the PLC to RUN
4…18	0000	Not used

Response from controllerE to host

Word no.	Value [hex.]	Meaning
1	0961	Copy of the request Command processed, no error occurred
2…18	0000	Not changed

7.2.27 Module 19, command 102_{dec} (66_{hex}): user menu

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Reserved = 0			UID					Command number = 66_{hex} = 102_{dec}							
2	Reserved = 0								Reserved = 0							
3	Reserved = 0								command number							
4...18	Parameters 1...n for the command number or: Not used															

Legend:

UID	user ID	5 bits	00…1F_{hex} = 0…31_{dec} (a change to the user ID starts the command call)
	command number	1 byte	01 = reads the current menu status 02 = jumps to the start menu screen no. 0 03 = jumps to the user menu screen no. 161 04 = deletes the → empty screen, only for user menus 05 = writes a defined string to a defined position in the display, only for user menus: parameter 1 = X position (1…128 pixels) parameter 2 = Y position(1…8 lines per 8 pixels) parameter 3 = character set and representation (values can be combined): 00x1 = "Small" 00x2 = "Big" 00x3 = "Fix" 00x4 = "Bitmap" 00x5 = "Big underlined" 001x = delete line invertedly (→ black bar) 002x = do not delete points 1 to X 004x = do not delete from string end to point 128 008x = shows the string invertedly Parameter 4…5 = pointer towards string (string with "0000" completed) 06 = writes a defined "byte matrix" at a defined position in the display, only for user menus: Parameter 1 = X1 position upper left (1…128 pixels) Parameter 2 = Y1 position upper left (1…8 lines 8 pixels each) Parameter 3 = X2 position bottom right (1…128 pixels) Parameter 4 = Y2 position bottom right (1…8 lines 8 pixels each) Parameter 5…6 = pointer towards byte matrix (1 byte corresponds to a vertical field of 8 pixels height, bit 0 = top … Bit 7 = bottom) 07 = shows a defined arrow in the display next to the image number, only for user menus: Parameter = 0001 → ▲ Parameter = 0002 → ▼ Parameter = 0003 → ▲ + ▼ -- continued on the following page --

	command number	1 byte	-- continued -- 08 = defines the texts allocated to the outer keys, only for user menus: parameter 1 = key index (0…13) parameter 2 = definition of the key index, e.g.:

Index	left key	right key
0000	OK	ESC
0001	==>	ESC
0002	MORE	ESC
0003	NEXT	ESC
0004	OK	
0005		ESC
0006	MORE	MENU
0007	OK	MENU
0008	MENU	USER
0009	<==	ESC
000A	INFO	ESC
000B	CLEAR	ESC
000C		
000D	-WAIT-	-WAIT-

Response from controllerE to host (command number = 01)

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	res.	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3	buttons pressed															
4	activated menu area															
5	process error occurred															
6	currently displayed menu window															
7	activated system language															
8...18	Not changed								Not changed							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command	
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used	
	keys pressed	1 word	0001 = bit 0: left key 0002 = bit 1: key [▲] 0004 = bit 2: key [▼] 0008 = bit 3: right key	Combinations possible by adding the values
	activated menu area	1 word	00A0 = system menu active 00A1 = user menu active 00AE = process error display active (E10...E30) 00AF = system error display active (acknowledgement required)	
	process error occurred	1 word	0000 = no process errors given 0001 = one or more process errors given	

	Currently displayed menu window	1 word	number of the menu screen
	activated system language	1 word	0000 = menu display in English 0001 = menu display in the second system language

Response from controllerE to host (command number = 02)

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	res.	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3...18	Not changed								Not changed							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used

Response from controllerE to host (command number = 03)

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	res.	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3...18	Not changed								Not changed							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used

Response from controllerE to host in the normal case (command number = 04...08)

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E=0	B	res.	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3...18	Not changed								Not changed							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used

Response from controllerE to host in case of a fault (command number = 04...08)

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E=1	B	res.	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3	Not changed								Error code							
4...18	Not changed								Not changed							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used

Possible error codes:

Value [hex.]	Meaning
AD	Access denied. The user menu must be active!
E0	Parameter invalid.

Example: retrieve the status of the controllerE display

Request of host to controllerE

Word no.	Value [hex.]	Meaning
1	0766	UID = 07: user ID changes to 7 66 = command 102
2	0000	Reserved
3	0001	command number = 0001 ⇒ reads the current menu status
4…18	0000	not used here

Response from controllerE to host

Word no.	Value [hex.]	Meaning
1	0766	Copy of the request Command processed, no error occurred
2	0000	Reserved
3	0008	0008 = bit 3 ⇒ right button is actuated
4	00A0	00A0 = system menu active
5	0001	0001 = one or more process errors given
6	001B	$001B_{hex} = 0027_{dec}$ ⇒ menu screen "Quick Setup" is currently displayed
7	0000	0000 = the English menus are displayed
8…18	0000	Not changed

7.2.28 Module 19, command 105_{dec} (69_{hex}): read the device properties of the controllerE

Request of host to controllerE

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	Reserved = 0			UID					Command number = 69_{hex} = 105_{dec}							
2...18	Not used								Not used							

Legend:

UID	user ID	5 bits	00…$1F_{hex}$ = 0…31_{dec} (a change to the user ID starts the command call)

Response from controllerE to host

Word no.	Bit															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	E	B	res.	Reflected user ID					Reflected command number							
2	Reserved = 0								Reserved = 0							
3	2M	DP	EN	Reserved = 0					Mod							
4	Reserved = 0								AT							
5	Reserved = 0								flash memory type							
6	hardware version															
7	RTS firmware version number															
8	RTS firmware release number															
9	AS-i master 1 firmware version number															
10	AS-i master 1 firmware release number															
11	AS-i master 2 firmware version number															
12	AS-i master 2 firmware release number															
13	Linux kernel version															
14	Linux ramdisk version															
15...18	Not changed								Not changed							

Legend:

E	Error bit	1 bit	0 = no error detected 1 = error when executing the command
B	Busy	1 bit	0 = command processed, buffer response valid 1 = command in process, channel used
2M	2 AS-i master	1 bit	0 = unit has 1 AS-i master 1 = unit has 2 AS-i masters
DP	Profibus DP	1 bit	0 = fieldbus interface Profibus DP not available 1 = fieldbus interface Profibus DP available
EN	Ethernet	1 bit	0 = Ethernet programming interface not available 1 = Ethernet programming interface available
Mod	PLC mode	1 byte	01 = PLC in RUN 02 = PLC in STOP 04 = PLC stops at the breakpoint 08 = gateway mode

AT	Anybus type	1 byte	01 = Anybus Profibus DP 04 = Anybus CANopen 05 = Anybus DeviceNet 09 = Anybus Ethernet IT 0A = Anybus Ethernet/IP 0B = ifm Profibus DP 0C = no fieldbus module detected

Example: read the device properties of the controllerE

Request of host to controllerE

Word no.	Value [hex.]	Meaning
1	0669	UID = 06: user ID changes to 6 69 = command 105
2…18	0000	Not used

Response from controllerE to host

Word no.	Value [hex.]	Meaning
1	0669	Copy of the request Command processed, no error occurred
2	0000	Reserved
3	0008	0008_{hex} = $0000\ 0000\ 0000\ 1000_{bin}$ 2M = 0 ⇒ unit contains 1 AS-i master DP = 0 ⇒ fieldbus interface Profibus DP not available EN = 0 ⇒ Ethernet programming interface not available PLC mode = 08 ⇒ gateway mode
4	000B	Anybus type = 000B ⇒ ifm Profibus DP
5	0002	flash memory type
6	1000	hardware version
7	0002	1st part of the RTS firmware version (here: 02.218B): version number 02.xxxx
8	218B	2nd part of the RTS firmware version (here: 02.218B) : release number xx.218B
9	0000	1st part of the AS-i master 1 firmware version (here: 0.238A): version number 0.xxxx
10	238A	2nd part of the AS-i master 1 firmware version (here: 0.238A): version number x.238A
11	0000	1st part of the AS-i master 2 firmware version (here: 0.238A): version number 0.xxxx
12	238A	2nd part of the AS-i master 2 firmware version (here: 0.238A): version number x.238A
13	0196	Linux kernel version: 406
14	0A6E	Linux ramdisk version 10.110
15…18	0000	Not changed

8 Operating and display elements

Contents

Diagnostic LEDs Key functions Display basic functions	→ separate basic device manual

8.1 Status LEDs on the network connection

4 status LEDs on the controllerE inform about the status of the Ethernet interface and the systems connected to it:

Module State	O	O	Net State
Link to Ethernet	O	O	Transmission Activity

Graphics: status LEDs on the network connection

8.1.1 LED [module status]

LED status	Description
Off	No supply voltage
Permanently green	Active connection to an Ethernet/IP network.
Green flashing	No active connection to an Ethernet/IP network.
Red flashing	Insignificant, reversible error detected.
Permanently red	Significant, non reversible error detected.
Red / green alternatively	Active self-test.

8.1.2 LED [network status]

LED status	Description
Off	No supply voltage or no IP address
Permanently green	The connection to at least one Ethernet IP participant is active
Green flashing	The Ethernet network is connected, but there is no connection to an Ethernet IP participant
Permanently red	Double IP address assignment
Red flashing	Timeout of the connection monitoring
Red / green alternatively	Active self-test

8.1.3 LED [connection with Ethernet]

LED status	Description
Off	No connection to an Ethernet network.
Permanently green	The controllerE is connected to an Ethernet network.

8.1.4 LED [transmission activity]

LED status	Description
Off	There is no data traffic.
Green flashing	Flashes for each transmitted or received data packet.

9 Menu

Contents

Info

All menu texts in this manual are in English.

Basic functions → separate basic instructions of the device manual

9.1 Main menu [Quick Setup]

Setting and reading of the fieldbus parameters (password level 1 required).

Details → chapter Setting and reading of the fieldbus parameters (→ page 175)

Menu tree	Explanation
Quick Setup Fieldbus Setup	> Display of the current IP address ► Change the fieldbus address using the keys [▲] / [▼] / [➔] ► After pressing [OK]: > Display subnet mask ► Change the subnet mask using the keys [▲] / [▼] / [➔] ► After pressing [OK]: > Display MAC ID ► After pressing [MORE] > Display of the fieldbus module 1 ► Change fieldbus module 1 using the keys [▲] / [▼] ► After pressing [OK]: > Display of the fieldbus module 2 > … ► Change fieldbus module 19 using the keys [▲] / [▼] ► After pressing [OK]: > Display of the fieldbus module 1 > … ► After pressing [ESC] twice: > Return to the start screen

9.2 Main menu [Fieldbus Setup]

Setting and reading of the fieldbus parameters (password level 1 required).
Details → chapter Setting and reading of the fieldbus parameters (→ page 175)

Menu tree	Explanation
Fieldbus Setup	> Display of the current IP address ► Change the fieldbus address using the keys [▲] / [▼] / [➔] ► After pressing [OK]: > Display subnet mask ► Change the subnet mask using the keys [▲] / [▼] / [➔] ► After pressing [OK]: > Display MAC ID ► After pressing [MORE] > Display of the fieldbus module 1 ► Change fieldbus module 1 using the keys [▲] / [▼] ► After pressing [OK]: > Display of the fieldbus module 2 > … ► Change fieldbus module 19 using the keys [▲] / [▼] ► After pressing [OK]: > Display of the fieldbus module 1 > … ► After pressing [ESC] twice: > Return to the start screen

10 Set-up

Contents

This chapter shows you how to get the Ethernet interface started quickly.

10.1 Basic settings of the fieldbus interface

NOTE

In the Ethernet network every IP address MUST be unique. The following IP addresses, however, are reserved for network-internal purposes and are therefore not allowed as addresses for participants: xxx.xxx.xxx.0 and xxx.xxx.xxx.255.

Only network participants whose subnet mask is identical and whose IP addresses are identical with respect to the subnet mask can communicate with each other.

Rule:
If part of the subnet mask = 255, the corresponding IP address parts must be identical.
If part of the subnet mask = 0, the corresponding IP address parts must be different.

Info

If the subnet mask = 255.255.255.0, 254 participants communicating with each other are possible in the network.
If the subnet mask = 255.255.0.0, 256x254=65 024 participants communicating with each other are possible in the network.

In the same physical network different subnet masks of the participants are allowed. They form different groups of participants which cannot communicate with groups of participants having other subnet masks.

Examples:

Participant A IP address	Participant A Subnet mask	Participant B IP address	Participant B Subnet mask	Communication of participants possible?
192.168.82.247	255.255.255.0	192.168.82.10	255.255.255.0	Yes, 254 participants possible
192.168.82.**247**	255.255.255.0	192.168.82.**247**	255.255.255.0	No (same IP address)
192.168.82.247	255.255.**255**.0	192.168.82.10	255.255.**0**.0	No (different subnet mask)
192.168.**82**.247	255.255.255.0	192.168.**116**.10	255.255.255.0	No (different IP address range: 82 ≠ 116)
192.168.222.213	255.255.0.0	192.168.222.123	255.255.0.0	Yes, 65 024 participants possible
192.168.111.213	255.255.0.0	192.168.222.123	255.255.0.0	Yes, 65 024 participants possible

Participant A IP address	Participant A Subnet mask	Participant B IP address	Participant B Subnet mask	Communication of participants possible?
192.168.82.247	255.255.255.0	192.168.82.**0**	255.255.255.0	No, the whole network is disturbed because the IP address xxx.xxx.xxx.0 is not allowed

The necessary settings of the fieldbus interface of the Ethernet controllerE can be made by means of the integrated display and the four operating keys. In the menu [Fieldbus Setup] the user can make all the necessary basic settings or take a look at the given configuration.

[Menu] > [Fieldbus Setup] or:
[Menu] > [Quick Setup] > [Fieldbus Setup]

Every controllerE with Ethernet interface has an individual physical address, the so-called MAC address (MAC = Media Access Control), which serves for the unambiguous identification of the unit in the network. The MAC address is only displayed and cannot be modified.

In any case the following parameters must be set on the unit for use on the intranet/internet.

- the IP address (IP = Internet Protocol) and
- the subnet mask.

If further settings are required for the integration of the unit in an Ethernet network (e.g. gateway addresses, DNS or SMTP settings), these can be made via the integrated web server of the unit via the html page `configform.htm`. The html page `configform.htm` can be opened with a standard browser (→ chapter The Ethernet configuration file ethcfg.cfg and its backup file ethcfg_old.cfg, → page 62).

The use of symbolic names (DNS = **D**omain **N**ame **S**ystem) instead of the IP addresses is possible. To do so, the IP address of at least one name server must be indicated (DNS1 address, DNS2 address). The name of the controllerE (host name) and the group name (domain name) can also be indicated. On an intranet, these names can be freely selected. On the internet the names of the top level domains (e.g. .de, .net or .com) are assigned and the second level domain names (e.g. ifm-electronic.com) are managed by authorised institutions.

The SMTP (**S**imple **M**ail **T**ransfer **P**rotocol) controls the sending of e-mails. The controllerE sends its e-mails to the unit with the IP address indicated as SMTP server (SMTP address). This server places the e-mail in the recipient's inbox until it is retrieved. If the SMTP server requires password authentication, this can also be defined (SMTP username / SMTP password).

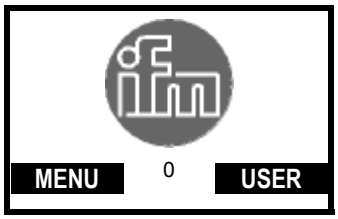
10.2 Parameter setting of the controllerE

10.2.1 Parameter setting of slaves in the controllerE

Set the parameters of the slaves in the AS-i controllerE as described in the basic device manual.

10.2.2 Parameter setting of fieldbus interface in the controllerE

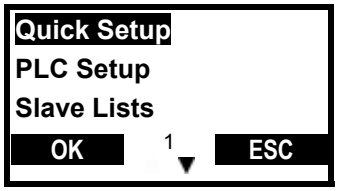
[Menu] > [Fieldbus Setup] > Set IP address > Set subnet mask > [OK]



⇑

Step 1:

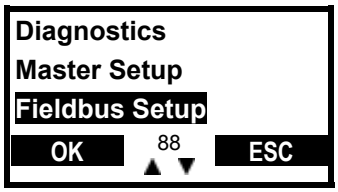
► Press [Menu].



⇑

Step 2:

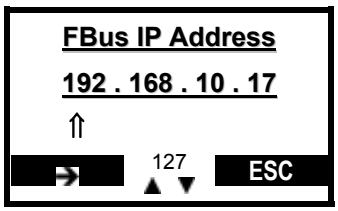
► Press [▼] to scroll to [Fieldbus Setup].



⇑

Step 3:

► Press [OK] to select [Fieldbus Setup].



⇑

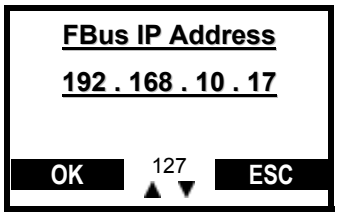
Step 4:

> Display of the current IP address.
Arrow points to editable address group (here: 192).

► Press [▲] / [▼] to scroll to the requested address.

► Press [➔] to go to the next address group.

After the last address group:



⇑

Step 5:

> Display of newly set IP address.

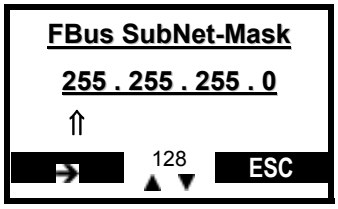
► Press [ESC] to quit without changes.

> The current IP address remains valid.

OR:

► Confirm the new IP address with [OK].

> The new IP address becomes valid.



⇑

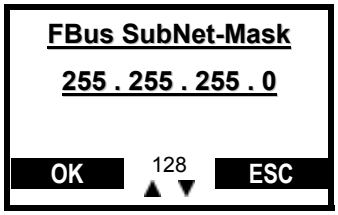
Step 6:

> Display current subnet mask.
Arrow points to editable address group (here: 255).

► Press [▲] / [▼] to scroll to the requested address.

► Press [➔] to go to the next address group.

After the last address group:



⇑

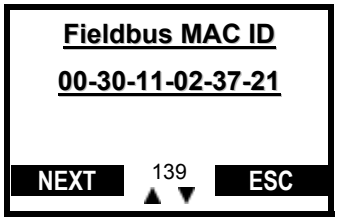
Step 7:

> Display current subnet mask.

► Confirm the new subnet mask with [OK].

OR:

► Press [ESC] to quit without changes.



⇑

Step 8:

> Display MAC ID in the fieldbus interface.

► Press [NEXT] to go to the next entry
→ chapter Setting and reading of the fieldbus parameters
(→ page 175).

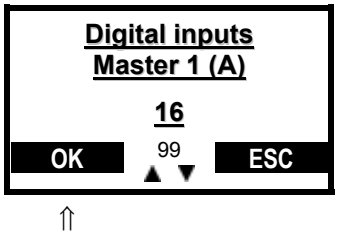
OR:

► Press [ESC] twice to return to the start screen.

Continued in the next chapter.

10.3 Setting and reading of the fieldbus parameters

Continued from the preceding chapter. Details → chapter The fieldbus modules (→ page 37)



⇑

Step 9:

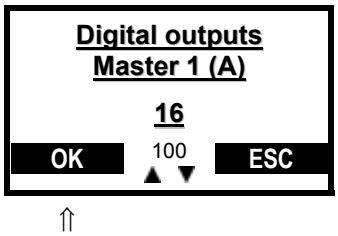
> Display that 16 bytes in the fieldbus master were configured for digital inputs in the fieldbus master of single or A slaves on AS-i master 1.

► Press [▲] / [▼] to set the requested number of bytes.

► Press [OK] to save the settings and scroll to the next display.

OR:

► Press [ESC] to return to screen 127 [Fieldbus IP address] (→ step 4)



⇑

Step 10:

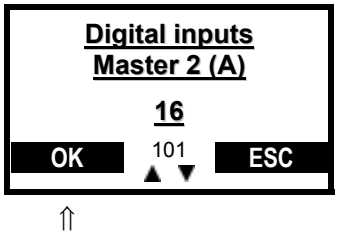
> Display that 16 bytes in the fieldbus master were configured for digital outputs in the fieldbus master of single or A slaves on AS-i master 1

► Press [▲] / [▼] to set the requested number of bytes.

► Press [OK] to save the settings and scroll to the next display.

OR:

► Press [ESC] to return to screen 127 [Fieldbus IP address] (→ step 4)



⇑

Step 11:

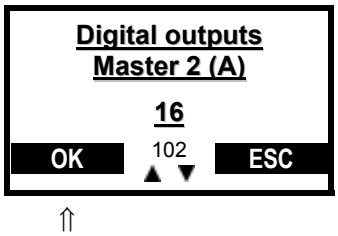
> Display that 16 bytes in the fieldbus master were configured for digital inputs in the fieldbus master of single or A slaves on AS-i master 2.

► Press [▲] / [▼] to set the requested number of bytes.

► Press [OK] to save the settings and scroll to the next display.

OR:

► Press [ESC] to return to screen 127 [Fieldbus IP address] (→ step 4)



⇑

Step 12:

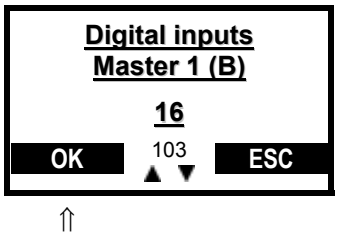
> Display that 16 bytes in the fieldbus master were configured for digital outputs in the fieldbus master of single or A slaves on AS-i master 2.

► Press [▲] / [▼] to set the requested number of bytes.

► Press [OK] to save the settings and scroll to the next display.

OR:

► Press [ESC] to return to screen 127 [Fieldbus IP address] (→ step 4)



⇑

Step 13:

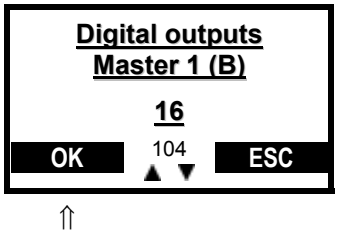
> Display that 16 bytes in the fieldbus master were configured for digital inputs in the fieldbus master of B slaves on AS-i master 1.

► Press [▲] / [▼] to set the requested number of bytes.

► Press [OK] to save the settings and scroll to the next display.

OR:

► Press [ESC] to return to screen 127 [Fieldbus IP address] (→ step 4)



⇑

Step 14:

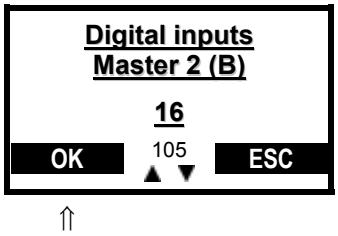
> Display that 16 bytes in the fieldbus master were configured for digital outputs in the fieldbus master of B slaves on AS-i master 1.

► Press [▲] / [▼] to set the requested number of bytes.

► Press [OK] to save the settings and scroll to the next display.

OR:

► Press [ESC] to return to screen 127 [Fieldbus IP address] (→ step 4)



⇑

Step 15:

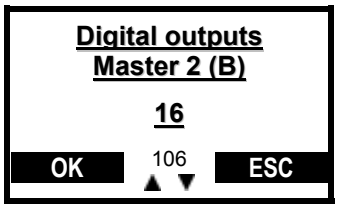
> Display that 16 bytes in the fieldbus master were configured for digital inputs in the fieldbus master of B slaves on AS-i master 2.

► Press [▲] / [▼] to set the requested number of bytes.

► Press [OK] to save the settings and scroll to the next display.

OR:

► Press [ESC] to return to screen 127 [Fieldbus IP address] (→ step 4)



⇑

Step 16:

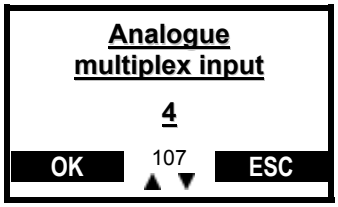
> Display that 16 bytes in the fieldbus master were configured for digital outputs in the fieldbus master of B slaves on AS-i master 2.

► Press [▲] / [▼] to set the requested number of bytes.

► Press [OK] to save the settings and scroll to the next display.

OR:

► Press [ESC] to return to screen 127 [Fieldbus IP address] (→ step 4)



⇑

Step 17:

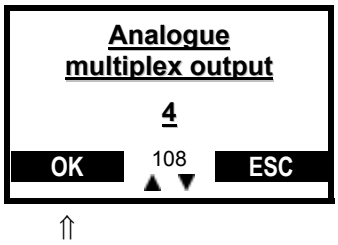
> Display that 4 bytes in the fieldbus master were configured for analogue multiplex inputs in the fieldbus master.

► Press [▲] / [▼] to set the requested number of bytes.

► Press [OK] to save the settings and scroll to the next display.

OR:

► Press [ESC] to return to screen 127 [Fieldbus IP address] (→ step 4)



⇑

Step 18:

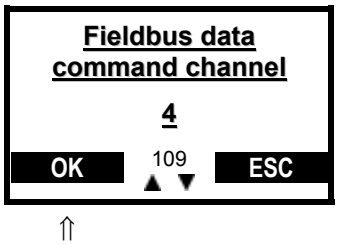
> Display that 4 bytes in the fieldbus master were configured for analogue multiplex outputs in the fieldbus master.

► Press [▲] / [▼] to set the requested number of bytes.

► Press [OK] to save the settings and scroll to the next display.

OR:

► Press [ESC] to return to screen 127 [Fieldbus IP address] (→ step 4)



⇑

Step 19:

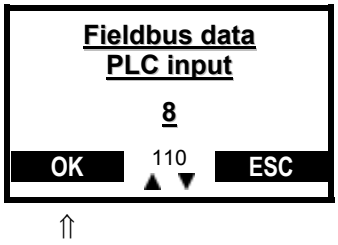
> Display that 4 bytes in the fieldbus master were configured for the fieldbus data command channel.

► Press [▲] / [▼] to set the requested number of bytes.

► Press [OK] to save the settings and scroll to the next display.

OR:

► Press [ESC] to return to screen 127 [Fieldbus IP address] (→ step 4)



⇑

Step 20:

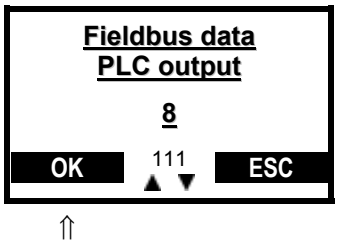
> Display that 8 bytes in the fieldbus master were configured for fieldbus data PLC inputs in the fieldbus master.

► Press [▲] / [▼] to set the requested number of bytes.

► Press [OK] to save the settings and scroll to the next display.

OR:

► Press [ESC] to return to screen 127 [Fieldbus IP address] (→ step 4)



⇑

Step 21:

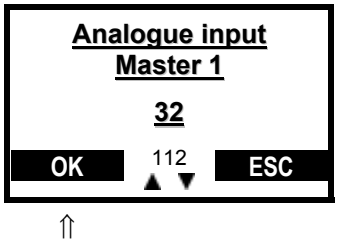
> Display that 8 bytes in the fieldbus master were configured for fieldbus data PLC outputs in the fieldbus master.

► Press [▲] / [▼] to set the requested number of bytes.

► Press [OK] to save the settings and scroll to the next display.

OR:

► Press [ESC] to return to screen 127 [Fieldbus IP address] (→ step 4)



⇑

Step 22:

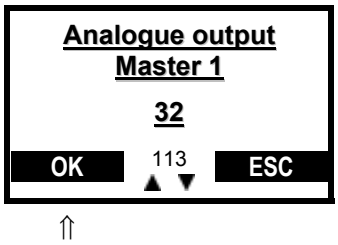
> Display that 32 bytes in the fieldbus master were configured for analogue inputs in the fieldbus master on AS-i master 1.

► Press [▲] / [▼] to set the requested number of bytes.

► Press [OK] to save the settings and scroll to the next display.

OR:

► Press [ESC] to return to screen 127 [Fieldbus IP address] (→ step 4)



⇑

Step 23:

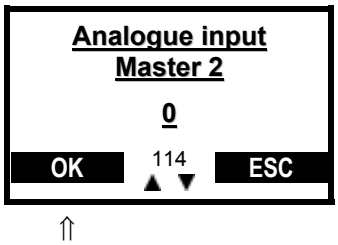
> Display that 32 bytes in the fieldbus master were configured for analogue outputs in the fieldbus master on AS-i master 1.

► Press [▲] / [▼] to set the requested number of bytes.

► Press [OK] to save the settings and scroll to the next display.

OR:

► Press [ESC] to return to screen 127 [Fieldbus IP address] (→ step 4)



⇑

Step 24:

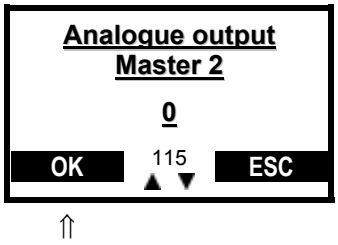
> Display that 0 bytes in the fieldbus master were configured for analogue inputs in the fieldbus master on AS-i master 2.

► Press [▲] / [▼] to set the requested number of bytes.

► Press [OK] to save the settings and scroll to the next display.

OR:

► Press [ESC] to return to screen 127 [Fieldbus IP address] (→ step 4)



⇑

Step 25:

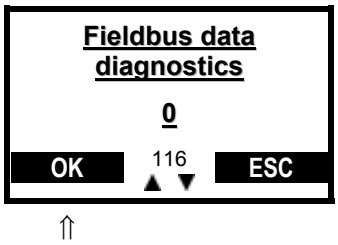
> Display that 0 bytes in the fieldbus master were configured for analogue outputs in the fieldbus master on AS-i master 2.

► Press [▲] / [▼] to set the requested number of bytes.

► Press [OK] to save the settings and scroll to the next display.

OR:

► Press [ESC] to return to screen 127 [Fieldbus IP address] (→ step 4)



⇑

Step 26:

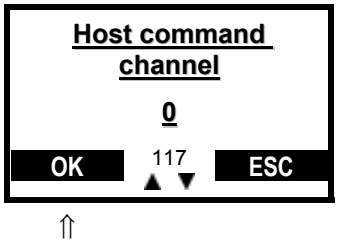
> Display that 0 bytes in the fieldbus master were configured for diagnostic data in the fieldbus master.

► Press [▲] / [▼] to set the requested number of bytes.

► Press [OK] to save the settings and scroll to the next display.

OR:

► Press [ESC] to return to screen 127 [Fieldbus IP address] (→ step 4)



⇑

Step 27:

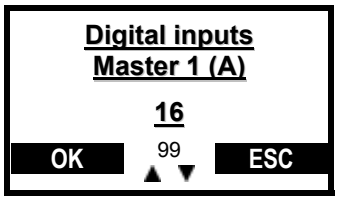
> Display number of data for module 19
(→ The host command channel, → page 105)

► Press [▲] / [▼] to set the requested number of bytes.

► Press [OK] to save the settings and scroll to the next display.

OR:

► Press [ESC] to return to screen 127 [Fieldbus IP address] (→ step 4)



Step 28:

> Repetition of the display series (→ step 9)

► Press [OK] to scroll to the next display.

OR:

⇑ ► Press [ESC] to return to screen 127 [Fieldbus IP address] (→ step 4)

10.4 Store system parameters

→ Basic device manual

11 Data exchange between controllerE and HTML pages

Contents

11.1 General

This chapter is supposed to illustrate how the integrated web server of the unit can be used in order to carry out a data exchange between the AS-interface system and user-defined HTML pages (HTML = **H**yper **T**ext **M**arkup **L**anguage). The goal is to provide the user with the necessary knowledge for such a data transfer.

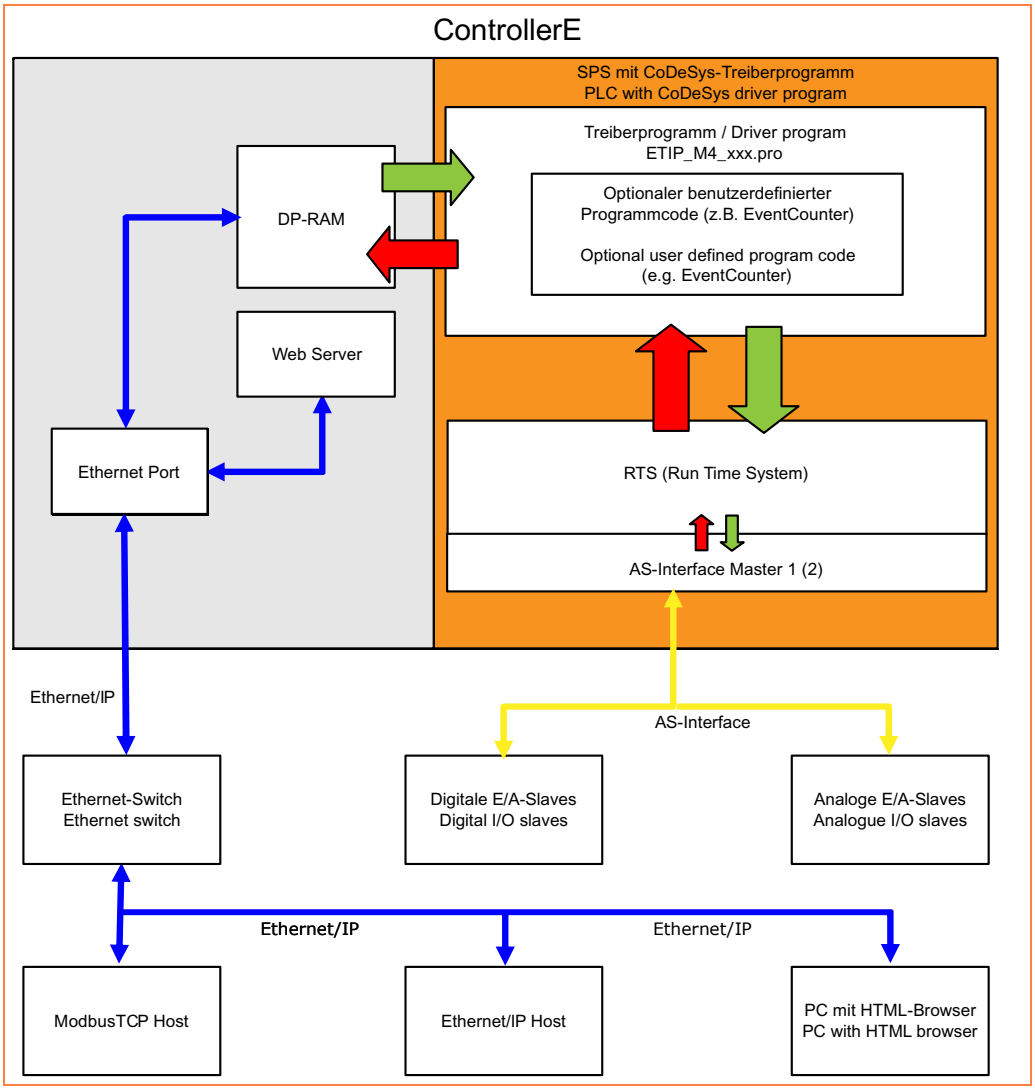
Info

This document is not intended as programming manual for HTML pages!

Detailed information about programming of HTML pages → special publications regarding this topic.

11.2 System description

In its general structure, the system described here corresponds to the block diagram below.



11.3 The HTML address

In order to display a signal of the AS-i system (e.g. a digital input) on an HTML page, you need to know first of all at which location of the memory this information can be found. Such a location will be called HTML address in the following.

The memory is the so-called dual-port RAM (→ page 36) or, short, DP-RAM. It is the interface between the AS-i master with the PLC and the fieldbus interface, and so also with the web server. A so-called driver (→ block diagram in System description, → page 181) ensures a defined data exchange between the PLC and the fieldbus interface via the DP-RAM.

The exact HTML address, i.e. the exact memory location at which a requested piece of information can be found, is defined by the fieldbus module settings in the menu [Fieldbus Setup] of the controllerE. Given that the module settings may vary, the HTML address ranges of the respective modules must be determined according to the effected fieldbus module settings (→ The fieldbus modules, → page 37).

When determining an HTML address the types of signals always have to be distinguished. So, data from the controllerE to the fieldbus interface (controllerE output data) are displayed in a different area than data from the fieldbus interface to the controllerE (controllerE input data). To determine the HTML address, the different module lengths must be added in rising order of the module numbers, as from HTML address 0 for the controllerE output data / as from HTML address 512 for the controllerE input data (→ Example: Determination of the HTML address ranges, → page 183).

The following table gives an overview of the possible module settings and the corresponding HTML address range lengths:

Direction of data	Module name → The fieldbus modules (→ page 37)	Possible number of bytes in the HTML address range...	
		0...511 (controllerE output data)	512...1023 (controllerE input data)
C ⇒ F	Module 1: digital input master 1(A) (→ page 38)	0…16	
C ⇐ F	Module 2: digital output master 1(A) (→ page 40)		0…16
C ⇒ F	Module 3: digital input master 2(A) (→ page 41)	0…16	
C ⇐ F	Module 4: digital output master 2(A) (→ page 41)		0…16
C ⇒ F	Module 5: digital input master 1(B) (→ page 42)	0…16	
C ⇐ F	Module 6: digital output master 1(B) (→ page 43)		0…16
C ⇒ F	Module 7: digital input master 2(B) (→ page 44)	0…16	
C ⇐ F	Module 8: digital output master 2(B) (→ page 44)		0…16
C ⇔ F	Module 9: analogue multiplexed input (→ page 46)	0 or 4	0 or 4
C ⇔ F	Module 10: analogue multiplexed output (→ page 48)	0 or 4	0 or 4
C ⇔ F	Module 11: fieldbus data command channel (→ page 50)	0 or 4	0 or 4
C ⇐ F	Module 12: fieldbus data PLC input (→ page 51)		0…128
C ⇒ F	Module 13: fieldbus data PLC output (→ page 51)	0…128	
C ⇒ F	Module 14: analogue input master 1 (→ page 52)	0...120	
C ⇐ F	Module 15: analogue output master 1 (→ page 54)		0...120
C ⇒ F	Module 16: analogue input master 2 (→ page 56)	0...120	
C ⇐ F	Module 17: analogue output master 2 (→ page 56)		0...120
C ⇒ F	Module 18: fieldbus data diagnosis (→ page 57)	0 / 26 / 52	
C ⇔ F	Module 19: Host command channel (→ page 59)	0 / 10 / 36	0 / 10 / 36

C ⇒ F	Data from controllerE to fieldbus interface (controllerE output data)
C ⇐ F	Data from fieldbus interface to controllerE (controllerE input data)
C ⇔ F	Bidirectional data (controllerE output data as well as controllerE input data)

11.3.1 Example: Determination of the HTML address ranges

Task:

The module lengths of the controllerE were set in the menu [Fieldbus Setup] as shown in the below table. The task now is to determine the corresponding HTML address ranges of the different modules.

Solution:

In the two right-hand columns of the table the corresponding module lengths for the controllerE output and input data were continuously added in the order of the activated module numbers (module setting ≠ 0). From the accordingly determined HTML address ranges the resulting total lengths of the input and output data can be easily determined.

In our example these are 432 bytes transmitted from the controllerE to the fieldbus interface and 312 bytes transmitted from the fieldbus interface to the controllerE :

Module number	Direction of data	Module name	Module length set in the module	Possible number of bytes in the HTML address range...	
				0...511 (ControllerE output data)	512...1023 (ControllerE input data)
1	C ⇒ F	digital input master 1(A)	16	0...15	
2	C ⇐ F	digital output master 1(A)	16		512...527
3	C ⇒ F	digital input master 2(A)	16	16...31	
4	C ⇐ F	digital output master 2(A)	16		528...543
5	C ⇒ F	digital input master 1(B)	16	32...47	
6	C ⇐ F	digital output master 1(B)	16		544...559
7	C ⇒ F	digital input master 2(B)	16	48...63	
8	C ⇐ F	digital output master 2(B)	16		560...575
9	C ⇔ F	analogue multiplexed input	0	-	-
10	C ⇔ F	analogue multiplexed output	0	-	-
11	C ⇔ F	fieldbus data command channel	0	-	-
12	C ⇐ F	fieldbus data PLC input	8		576...583
13	C ⇒ F	fieldbus data PLC output	128	64...191	
14	C ⇒ F	analogue input master 1	15	192 -311	
15	C ⇐ F	analogue output master 1	15		584...703
16	C ⇒ F	analogue input master 2	15	312...431	
17	C ⇐ F	analogue output master 2	15		704...823
18	C ⇒ F	fieldbus data diagnosis	0	-	
19	C ⇔ F	host command channel	0	-	-
				⇓ **432 bytes controllerE output data**	⇓ **312 bytes controllerE input data**

C ⇒ F	Data from controllerE to fieldbus interface (controllerE output data)
C ⇐ F	Data from fieldbus interface to controllerE (controllerE input data)
C ⇔ F	Bidirectional data (controllerE output data as well as controllerE input data)

11.4 The exact HTML addresses in the HTML address ranges

Having described in detail the correlations between fieldbus module settings and HTML address ranges, the question of the exact HTML address allocation to the digital and analogue AS-i signals in these ranges now arises. Therefore, the allocations for the digital and analogue AS-i signals will be described in the following.

11.4.1 The digital inputs and outputs

The addresses of the digital AS-interface signals depend on the corresponding AS-i slave address. This type of addressing can in principle be compared to a location-dependent addressing as given in different PLC systems.

The following table shows the allocation of the different slave data bits in the respective HTML address range.

Byte offset from the start address of the HTML address range	Bit							
	7	6	5	4	3	2	1	0
+0	reserved				slave 1			
					D3	D2	D1	D0
+1	slave 2				slave 3			
	D3	D2	D1	D0	D3	D2	D1	D0
+2	slave 4				slave 5			
	D3	D2	D1	D0	D3	D2	D1	D0
+3	slave 6				slave 7			
	D3	D2	D1	D0	D3	D2	D1	D0
+4	slave 8				slave 9			
	D3	D2	D1	D0	D3	D2	D1	D0
+5	slave 10				slave 11			
	D3	D2	D1	D0	D3	D2	D1	D0
+6	slave 12				slave 13			
	D3	D2	D1	D0	D3	D2	D1	D0
+7	slave 14				slave 15			
	D3	D2	D1	D0	D3	D2	D1	D0
+8	slave 16				slave 17			
	D3	D2	D1	D0	D3	D2	D1	D0
+9	slave 18				slave 19			
	D3	D2	D1	D0	D3	D2	D1	D0
+10	slave 20				slave 21			
	D3	D2	D1	D0	D3	D2	D1	D0
+11	slave 22				slave 23			
	D3	D2	D1	D0	D3	D2	D1	D0
+12	slave 24				slave 25			
	D3	D2	D1	D0	D3	D2	D1	D0

Byte offset from the start address of the HTML address range	Bit							
	7	6	5	4	3	2	1	0
+13	slave 26				slave 27			
	D3	D2	D1	D0	D3	D2	D1	D0
+14	slave 28				slave 29			
	D3	D2	D1	D0	D3	D2	D1	D0
+15	slave 30				slave 31			
	D3	D2	D1	D0	D3	D2	D1	D0

Examples for the determination of the exact HTML address for digital signals

The following example is supposed to illustrate the use of the above table. For the example we adopt the module settings from the table in the example: Determination of the HTML address ranges (→ page 183).

1st task: The exact HTML addresses of the inputs of the digital input module slave 17B on master 2 are to be determined.

Solution for 1: The start address of the HTML address range for the B slaves (inputs) of master 2 is 48. The offset from this start address and the corresponding data bits can be read from the table.

Byte offset from the start address of the HTML address range	Bit							
	7	6	5	4	3	2	1	0
+0	reserved				slave 1			
					D3	D2	D1	D0
+1	slave 2				slave 3			
	D3	D2	D1	D0	D3	D2	D1	D0
...	...				...			
+8	slave 16				**slave 17**			
	D3	D2	D1	D0	**D3**	**D2**	**D1**	**D0**
...	...				...			
+15	slave 30				slave 31			
	D3	D2	D1	D0	D3	D2	D1	D0

⇒ The exact HTML addresses of the inputs of the digital input module slave 17B on master 2 are bits 0...3 of byte 56 (start address of the HTML address range (48) + byte offset (8)).

2nd task: The exact HTML addresses of the outputs of the digital output module slave 3 on master 1 are to be determined.

Solution for 2: The start address of the HTML address range for the single / A slaves (outputs) of master 1 is 512. The offset from this start address and the corresponding data bits can be read from the table.

Byte offset from the start address of the HTML address range	Bit							
	7	6	5	4	3	2	1	0
+0	reserved				slave 1			
					D3	D2	D1	D0
+1	slave 2				**slave 3**			
	D3	D2	D1	D0	**D3**	**D2**	**D1**	**D0**
...	...				...			
+15	slave 30				slave 31			
	D3	D2	D1	D0	D3	D2	D1	D0

⇒ The exact HTML addresses of the outputs of the digital output module slave 3 on master 1 are bits 0...3 of byte 513 (start address of the HTML address range + byte offset).

11.4.2 The analogue inputs and outputs

As for the digital inputs and outputs, the exact HTML addresses in the HTML address ranges of the analogue signals also depend on the AS-i addresses of the slaves. Each analogue slave can provide data of up to 4 analogue channels.

The following tables show the allocation of the AS-i slave address to the byte offset in the corresponding HTML address ranges. The number of channels transmitted by each analogue slave is set in the configuration file 'ifm.cfg' (→ page 64).

4 analogue channels per slave (standard)					
Possible AS-i slave address for...		Byte offset for...			
analogue INput modules	analogue OUTput modules	channel 1 for single or A slaves	channel 2 for single or A slaves	channel 3 for single slaves OR: channel 1 for B slaves	channel 4 for single slaves OR: channel 2 for B slaves
1	16	+0	+2	+4	+6
2	17	+8	+10	+12	+14
3	18	+16	+18	+20	+22
4	19	+24	+26	+28	+30
5	20	+32	+34	+36	+38
6	21	+40	+42	+44	+46
7	22	+48	+50	+52	+54
8	23	+56	+58	+60	+62
9	24	+64	+66	+68	+70
10	25	+72	+74	+76	+78
11	26	+80	+82	+84	+86
12	27	+88	+90	+92	+94
13	28	+96	+98	+100	+102
14	29	+104	+106	+108	+110
15	30	+112	+114	+116	+118

2 analogue channels per slave (can be set in the file `ifm.cfg`)					
Possible AS-i slave address for...		Byte offset for...			
analogue INput modules	analogue OUTput modules	channel 1	channel 2	channel 3	channel 4
1(A)	1(A)	+0	+2	-	-
2(A)	2(A)	+4	+6	-	-
3(A)	3(A)	+8	+10	-	-
4(A)	4(A)	+12	+14	-	-
5(A)	5(A)	+16	+18	-	-
6(A)	6(A)	+20	+22	-	-
7(A)	7(A)	+24	+26	-	-
8(A)	8(A)	+28	+30	-	-
9(A)	9(A)	+32	+34	-	-
10(A)	10(A)	+36	+38	-	-
11(A)	11(A)	+40	+42	-	-
12(A)	12(A)	+44	+46	-	-

2 analogue channels per slave (can be set in the file `ifm.cfg`)					
Possible AS-i slave address for...		Byte offset for...			
analogue INput modules	analogue OUTput modules	channel 1	channel 2	channel 3	channel 4
13(A)	13(A)	+48	+50	-	-
14(A)	14(A)	+52	+54	-	-
15(A)	15(A)	+56	+58	-	-
16(A)	16(A)	+60	+62	-	-
17(A)	17(A)	+64	+66	-	-
18(A)	18(A)	+68	+70	-	-
19(A)	19(A)	+72	+74	-	-
20(A)	20(A)	+76	+78	-	-
21(A)	21(A)	+80	+82	-	-
22(A)	22(A)	+84	+86	-	-
23(A)	23(A)	+88	+90	-	-
24(A)	24(A)	+92	+94	-	-
25(A)	25(A)	+96	+98	-	-
26(A)	26(A)	+100	+102	-	-
27(A)	27(A)	+104	+106	-	-
28(A)	28(A)	+108	+110	-	-
29(A)	29(A)	+112	+114	-	-
30(A)	30(A)	+116	+118	-	-

1 analogue channel per slave (can be set in the file `ifm.cfg`)					
Possible AS-i slave address for...		Byte offset for...			
analogue INput modules	analogue OUTput modules	channel 1	channel 2	channel 3	channel 4
1(A)	1(A)	+0	-	-	-
2(A)	2(A)	+2	-	-	-
3(A)	3(A)	+4	-	-	-
4(A)	4(A)	+6	-	-	-
5(A)	5(A)	+8	-	-	-
6(A)	6(A)	+10	-	-	-
7(A)	7(A)	+12	-	-	-
8(A)	8(A)	+14	-	-	-
9(A)	9(A)	+16	-	-	-
10(A)	10(A)	+18	-	-	-
11(A)	11(A)	+20	-	-	-
12(A)	12(A)	+22	-	-	-
13(A)	13(A)	+24	-	-	-
14(A)	14(A)	+26	-	-	-
15(A)	15(A)	+28	-	-	-
16(A)	16(A)	+30	-	-	-
17(A)	17(A)	+32	-	-	-

1 analogue channel per slave (can be set in the file `ifm.cfg`)					
Possible AS-i slave address for...		Byte offset for...			
analogue INput modules	analogue OUTput modules	channel 1	channel 2	channel 3	channel 4
18(A)	18(A)	+34	-	-	-
19(A)	19(A)	+36	-	-	-
20(A)	20(A)	+38	-	-	-
21(A)	21(A)	+40	-	-	-
22(A)	22(A)	+42	-	-	-
23(A)	23(A)	+44	-	-	-
24(A)	24(A)	+46	-	-	-
25(A)	25(A)	+48	-	-	-
26(A)	26(A)	+50	-	-	-
27(A)	27(A)	+52	-	-	-
28(A)	28(A)	+54	-	-	-
29(A)	29(A)	+56	-	-	-
30(A)	30(A)	+58	-	-	-
31(A)	31(A)	+60	-	-	-

Examples for the determination of the exact HTML address for analogue signals

The following examples are supposed to illustrate the use of the above tables. For our example we take the module settings from the example: Determination of the HTML address ranges (→ page 183). Further assumption: 4 channels per analogue slave set in the configuration file `ifm.cfg`.

1st task: The exact HTML address of the second analogue input channel of slave 7 on master 1 is to be determined.

Solution for 1: The start address of the HTML address range for the analogue inputs of master 1 is 192. For the offset from this start address please refer to the table.

Possible AS-i slave address for...		Byte offset for...			
analogue INput modules	analogue OUTput modules	channel 1	channel 2	channel 3	channel 4
1	16	+0	+2	+4	+6
2	17	+8	+10	+12	+14
...	...	...	...	...	...
7	22	+48	**+50**	+52	+54
...	...	...	...	...	...
15	30	+112	+114	+116	+118

⇒ The exact HTML address of the second analogue input channel of slave 7 on master 1 is byte 242 (start address of the HTML address range (192) + byte offset (50)).

2nd task: The exact HTML address of the fourth analogue output channel of slave 29 on master 2 is to be determined.

Solution: The start address of the HTML address range for the analogue outputs of master 2 is 704. For the offset from this start address please refer to the table.

Possible AS-i slave address for...		Byte offset for...			
analogue INput modules	analogue OUTput modules	channel 1	channel 2	channel 3	channel 4
1	16	+0	+2	+4	+6
2	17	+8	+10	+12	+14
...	...	...	...	...	...
14	**29**	+104	+106	+108	**+110**
15	30	+112	+114	+116	+118

⇒ The exact HTML address of the 4th analogue output channel of slave 29 on master 2 is byte 814 (start address of the HTML address range (704) + byte offset (110)).

11.5 HTML programming and applets

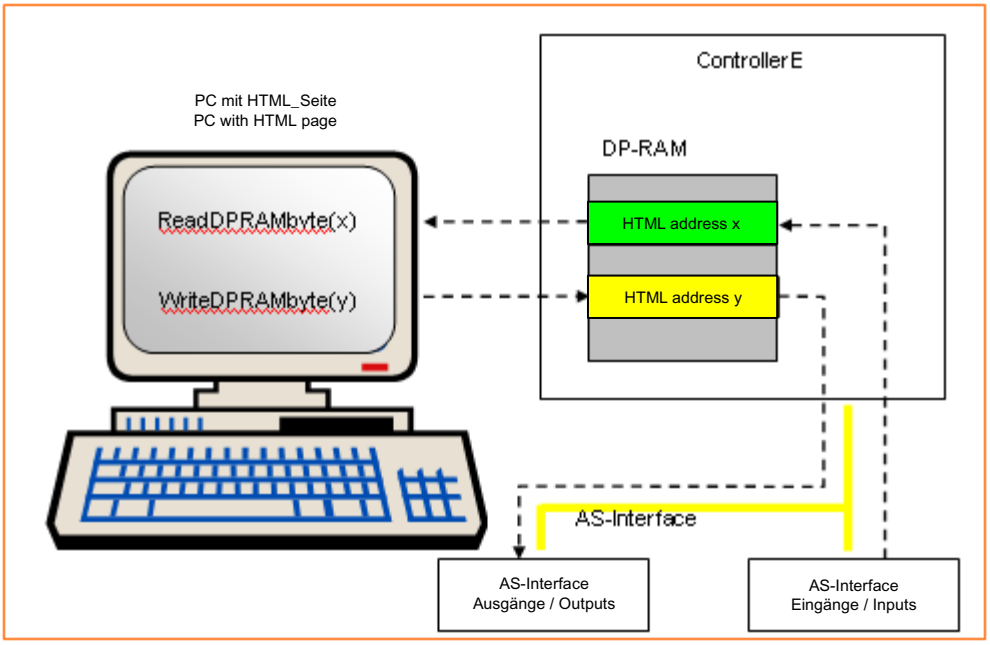
The software interfaces between objects on an HTML page and the HTML addresses are so-called applets (abbreviation of the English 'application program'). The controllerE units AC1327 / AC1337 use the applet `ModbusTCPClient.class`. This is a Java class library (compiled Java source code, so-called byte code) in which the different possible data access functions are defined.

Two different types are distinguished:

Functions for **reading** data from a certain HTML address of the DP-RAM:
ReadDPRAM<type> (HTML address)
where <type> indicates the corresponding data length (BYTE or WORD).

Functions for **writing** data to a certain HTML address of the DP-RAM:
WriteDPRAM<type> (HTML address)
where <type> indicates the corresponding data length (BYTE or WORD).

The figure below illustrates the data flow between the inputs and outputs of the AS-i system and an HTML page. The operating principle of the applet functions described above can also be seen here.



HTML programming example (digital inputs)

Task: A website on which the digital input signals of the AS-i slaves 1 and 2 of master 1 are displayed is to be created. The HTML page shall look as follows:

Display Bits Slaves 1 and 2

(1 =TRUE, 0 = FALSE)

Bit Number:	D0	D1	D2	D3
Slave 1:	0	0	0	0
Slave 2:	0	0	0	0

Solution: The required HTML code for this website is as follows:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"-->
<html>
<head>
        <title>Counter Test</title>
</HEAD>

<body Onload="reader()">

<!-- Introduce an applet and assign the name AnyBus -->
<applet code="ModbusTCPClient" width="0" height="0" name="AnyBus"></applet>

<SCRIPT LANGUAGE="JavaScript">

// Define a Function to:
// read the BYTE containing the data BITS from slaves 1 and 2 master 1 (see table)
// mask out the data BITS of each slave
// (for example the mask for slave 2 data BITS D3,D2,D1 and D0 is 128,64,32 and 16 respectively)
// transfer the values for display - either 1 or 0 dependent upon the value of the respective data BIT

function reader(){

        inByte1 = AnyBus.ReadDPRAMbyte(0);

        if ((inByte1 & 1) >0){document.form3.abyte11_1.value = 1}
                else{document.form3.abyte11_1.value = 0}
        if ((inByte1 & 2) >0){document.form3.abyte11_2.value = 1}
                else{document.form3.abyte11_2.value = 0}
        if ((inByte1 & 4) >0){document.form3.abyte11_3.value = 1}
                else{document.form3.abyte11_3.value = 0}
        if ((inByte1 & 8) >0){document.form3.abyte11_4.value = 1}
                else{document.form3.abyte11_4.value = 0}

        inByte2 = AnyBus.ReadDPRAMbyte(1);

        if ((inByte2 & 16) >0){document.form3.abyte12_1.value = 1}
                else{document.form3.abyte12_1.value = 0}
        if ((inByte2 & 32) >0){document.form3.abyte12_2.value = 1}
                else{document.form3.abyte12_2.value = 0}
        if ((inByte2 & 64) >0){document.form3.abyte12_3.value = 1}
                else{document.form3.abyte12_3.value = 0}
        if ((inByte2 & 128)>0){document.form3.abyte12_4.value = 1}
                else{document.form3.abyte12_4.value = 0}
        setTimeout("reader()",450);
}
</SCRIPT>

<center>
<h2> Display Bits Slaves 1 and 2 </h2>
</center>
```

```
<center>
<h2> ( 1 =TRUE, 0 = FALSE) </h2>
</center>

<FORM NAME="form3">
        <TABLE BORDER="1" align="center">
                <tr>
                        <td>Bit Number:</td>
                        <td>D0</td>
                        <td>D1</td>
                        <td>D2</td>
                        <td>D3</td>
                </tr>

                <tr>
                        <td>Slave 1:</td>
                        <td><INPUT TYPE="text" NAME="abyte11_1" value="0" SIZE=12></td>
                        <td><INPUT TYPE="text" NAME="abyte11_2" value="0" SIZE=12></td>
                        <td><INPUT TYPE="text" NAME="abyte11_3" value="0" SIZE=12></td>
                        <td><INPUT TYPE="text" NAME="abyte11_4" value="0" SIZE=12></td>
                </tr>

                <tr>
                        <td>Slave 2:</td>
                        <td><INPUT TYPE="text" NAME="abyte12_1" value="0" SIZE=12></td>
                        <td><INPUT TYPE="text" NAME="abyte12_2" value="0" SIZE=12></td>
                        <td><INPUT TYPE="text" NAME="abyte12_3" value="0" SIZE=12></td>
                        <td><INPUT TYPE="text" NAME="abyte12_4" value="0" SIZE=12></td>
                </tr>
        </TABLE>
</FORM>
</BODY>
</HTML>
```

11.6 Example for the display of counter values on an HTML page

A condition for this example is that the program module "EventCounter" is called cyclically in the PLC_PRG of the controllerE driver program `ETIP_M4_xxx.pro`. This program is not included in the scope of delivery of the controllerE , but it can be obtained free of charge from **ifm electronic gmbh**. This program is a small program routine, by means of which the rising edges of all four inputs of each digital input slave in the AS-i address range 16...31 are counted separately.

Before we have a look at the HTML programming example, further details about the program module "EventCounter" and the used HTML addresses are to be provided in the following two sections.

11.6.1 The counter values

Each of the 64 counter values (16 slaves with 4 inputs each) has a counting range from 0...65 535. This corresponds to a used data length of one word (16 bits without consideration of the leading sign) in the memory for each individual counter value. The counter values are provided to the DP-RAM via module 13: fieldbus data PLC output (On page 51) of the fieldbus setup.

The following table shows the offset values for each counter value in the used HTML address range (module 13 = fieldbus data PLC output). For our example we take the module settings from the Example: Determination of the HTML address ranges (On page 183). The start address of the HTML address range consequently is 64.

AS-i slave address	Byte offset for...			
	counter value D0	counter value D1	counter value D2	counter value D3
16	+0	+2	+4	+6
17	+8	+10	+12	+14
18	+16	+18	+20	+22
19	+24	+26	+28	+30
20	+32	+34	+36	+38
21	+40	+42	+44	+46
22	+48	+50	+52	+54
23	+56	+58	+60	+62
24	+64	+66	+68	+70
25	+72	+74	+76	+78
26	+80	+82	+84	+86
27	+88	+90	+92	+94
28	+96	+98	+100	+102
29	+104	+106	+108	+110
30	+112	+114	+116	+118
31	+120	+122	+124	+126

Examples: The HTML address for the counter value of the first digital input of slave 16 is 64 (start address of the HTML address range + byte offset)

The HTML address for the counter value of the fourth digital input of slave 26 is 150 (start address of the HTML address range + byte offset)

11.6.2 Resetting of counter values

The program module "EventCounter" enables, in addition to the output of counter values, also the reset of the different counter values. To do so, 64 bits are required (one bit for each counter value), which must be transmitted from the web server to the controllerE.

The following table shows the allocation of the byte offset values of the reset bits in the HTML address ranges used (Module 12: fieldbus data PLC input, → page 51). For our example we take the module settings from the example: Determination of the HTML address ranges (→ page 183). The start address of the HTML address range consequently is 576.

Byte offset from the start address of the HTML address range	Bit							
	7	6	5	4	3	2	1	0
+0	slave 16				slave 17			
	D3	D2	D1	D0	D3	D2	D1	D0
+1	slave 18				slave 19			
	D3	D2	D1	D0	D3	D2	D1	D0
+2	slave 20				slave 21			
	D3	D2	D1	D0	D3	D2	D1	D0
+3	slave 22				slave 23			
	D3	D2	D1	D0	D3	D2	D1	D0
+4	slave 24				slave 25			
	D3	D2	D1	D0	D3	D2	D1	D0
+5	slave 26				slave 27			
	D3	D2	D1	D0	D3	D2	D1	D0
+6	slave 28				slave 29			
	D3	D2	D1	D0	D3	D2	D1	D0
+7	slave 30				slave 31			
	D3	D2	D1	D0	D3	D2	D1	D0

Examples:

To reset the counter value D1 of slave 16, bit 5 of the byte 576 is to be set to "1"
⇒ byte 576 = $0010\ 0000_{bin}$

To reset the counter value D3 of slave 21, bit 3 of byte 578 is to be set to "1"
⇒ byte 578 = $0000\ 1000_{bin}$

To reset the counter value D0...D3 of slave 26, bits 4...7 of byte 581 are to be set to "1"
⇒ byte 581 = $1111\ 0000_{bin}$

To reset the counter values D0...D3 of slaves 28 and 29, bits 0...7 of byte 582 are to be set to "1"
⇒ byte 582 = $1111\ 1111_{bin}$

11.6.3 HTML programming example (counter values and reset)

Task: A website on which the counter values D0...D3 of AS-i slave 26 are displayed is to be created. On this page it shall be possible to reset the displayed counter values. The HTML page shall look as follows:

Display Slave 26 Counter

Bit Number:	D0	D1	D2	D3
Slave 26:	0	0	0	0
	Reset 0	Reset 1	Reset 2	Reset 3

Solution: The required HTML code for this website is as follows:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"-->
<html>
<head>
        <title>Counter Test</title>
</HEAD>

<body Onload="reader()">

<!-- Introduce an applet and assign the name AnyBus -->
<applet code="ModbusTCPClient" width="0" height="0" name="AnyBus"></applet>

<SCRIPT LANGUAGE="JavaScript">

// Define a Function to:
// reset slave counters
// Slave pair is given by o1 (example slave addr. = 26/26 -> o1 = 581 see table)
// Counter to be reset given by xval (example reset slave 26 counter D0 -> xval = 16 ,see table)

function writeCMD(xval,o1){
AnyBus.WriteDPRAMByte( o1, xval);
}

// Define a Function to:
// deactivate counter reset of slave 26 /27 (adr = 581, see table)
// read counter values from DPRAM
// Here the counter values are form slave 26 counters D0,D1,D2,D3 (addr. = 144, 146, 148 and 150, see table)

function reader(){
        o1=581;
        AnyBus.WriteDPRAMByte( o1, 0);

        inByte1 = AnyBus.ReadDPRAMword(144);
        inByte2 = AnyBus.ReadDPRAMword(146);
        inByte3 = AnyBus.ReadDPRAMword(148);
```

```
        inByte4 = AnyBus.ReadDPRAMword(150);
        document.form3.abyte11_1.value = inByte1;
        document.form3.abyte11_2.value = inByte2;
        document.form3.abyte11_3.value = inByte3;
        document.form3.abyte11_4.value = inByte4;
        setTimeout("reader()",450);
}

</SCRIPT>

<center>
<h2> Display Slave 26 Counter </h2>
</center>

<FORM NAME="form3">
        <TABLE BORDER="1" align="center">
                <tr>
                        <td>Bit Number:</td>
                        <td>D0</td>
                        <td>D1</td>
                        <td>D2</td>
                        <td>D3</td>
                </tr>

                <tr>
                        <td>Slave 26:</td>
                        <td><INPUT TYPE="text" NAME="abyte11_1" value="0" SIZE=12></td>
                        <td><INPUT TYPE="text" NAME="abyte11_2" value="0" SIZE=12></td>
                        <td><INPUT TYPE="text" NAME="abyte11_3" value="0" SIZE=12></td>
                        <td><INPUT TYPE="text" NAME="abyte11_4" value="0" SIZE=12></td>
                </tr>

                <tr>
                        <td> </td>
                        <td><INPUT TYPE="button" VALUE="Reset 0" onClick="writeCMD(16,581)"></td>
                        <td><INPUT TYPE="button" VALUE="Reset 1" onClick="writeCMD(32,581)"></td>
                        <td><INPUT TYPE="button" VALUE="Reset 2" onClick="writeCMD(64,581)"></td>
                        <td><INPUT TYPE="button" VALUE="Reset 3" onClick="writeCMD(128,581)"></td>
                </tr>

        </TABLE>

</FORM>
</BODY>
</HTML>
```

12 SSI functions

Contents

The integrated HMS interface module enables the dynamic creation of contents for HTML pages and e-mails. These SSI functions (SSI = **S**erver **S**ide **I**ncludes) automatically add the contents of system variables to the contents to be displayed before sending an HTML page or an e-mail.

12.1 Available SSI functions

Command	DisplayMacID
Syntax	`<?--#exec cmd_argument='DisplayMacID'-->`
Description	This function returns the MAC address of the HMS Anybus S module.
Used in	HTML, e-mails

Command	DisplaySerial
Syntax	`<?--#exec cmd_argument='DisplaySerial'-->`
Description	This function returns the serial number of the HMS Anybus S module.
Used in	HTML, e-mails

Command	DisplayFWVersion
Syntax	`<?--#exec cmd_argument='DisplayFWVersion'-->`
Description	This function returns the firmware version number of the HMS Anybus S module.
Used in	HTML, e-mails

Command	DisplayBLVersion
Syntax	`<?--#exec cmd_argument='DisplayBLVersion'-->`
Description	This function returns the bootloader version number of the HMS Anybus S module.
Used in	HTML, e-mails

Command	DisplayIP
Syntax	`<?--#exec cmd_argument='DisplayIP'-->`
Description	This function returns the currently used IP address.
Used in	HTML, e-mails

Command	DisplaySubnet
Syntax	`<?--#exec cmd:argument='DisplaySubnet'-->`
Description	This function returns the currently used subnet mask.
Used in	HTML, e-mails

Command	DisplayGateway
Syntax	`<?--#exec cmd_argument='DisplayGateway'-->`
Description	This function returns the currently used gateway address.
Used in	HTML, e-mails

Command	DisplayDNS1
Syntax	`<?--#exec cmd_argument='DisplayDNS1'-->`
Description	This function returns the IP address of the primary DNS server.
Used in	HTML, e-mails

Command	DisplayDNS2
Syntax	`<?--#exec cmd_argument='DisplayDNS2'-->`
Description	This function returns the IP address of the secondary DNS server.
Used in	HTML, e-mails

Command	DisplayHostName
Syntax	`<?--#exec cmd_argument='DisplayHostName'-->`
Description	This function returns the used host name.
Used in	HTML, e-mails

Command	DisplayDomainName
Syntax	`<?--#exec cmd_argument='DisplayDomainName'-->`
Description	This function returns the used domain name.
Used in	HTML, e-mails

Command	DisplayDhcpState
Syntax	`<?--#exec cmd_argument='DisplayDhcpState( "Output when ON", "Output when OFF")'-->`
Description	This function returns whether DHCP/BootP is enabled or blocked.
Used in	HTML, e-mails

Command	DisplayEmailServer
Syntax	`<?--#exec cmd:argument='DisplayEmailServer'-->`
Description	This function returns the currently used SMTP server address.
Used in	HTML, e-mails

Command	DisplaySMTPUser
Syntax	`<?--#exec cmd:argument='DisplaySMTPUser'-->`
Description	This function returns the user name for SMTP authentication.
Used in	HTML, e-mails

Command	DisplaySMTPPswd
Syntax	`<?--#exec cmd:argument='DisplaySMTPPswd'-->`
Description	This function returns the password the SMTP authentication.
Used in	HTML, e-mails

Command	StoreEtnConfig
Syntax	`<?--#exec cmd_argument='StoreEtnConfig'-->`
Description	This function stores an entered IP configuration in the file ethcfg.cfg (→ page 62). Insert this line into an HTML page and transfer this page to a form with new IP settings. In this form the following fields are accepted: - SetIp - SetSubnet - SetGateway - SetEmailServer - SetDhcpState - value "on" or "off" - SetDNS1 - SetDNS2 - SetHostName - SetDomainName - SetSMTPUser - SetSMTPPswd Standard outputs: - Invalid IP address! - Invalid Subnet mask! - Invalid Gateway address! - Invalid IP address or Subnet mask! - Invalid Email Server IP address! - Configuration stored correctly. - Invalid DHCP state! - Invalid DNS1! - Invalid DNS2! - Configuration stored correctly. - Failed to store the configuration. More information about SSI outputs → chapter Redirecting SSI output (→ page 206).
Used in	HTML only

Command	printf
Syntax	`<?--#exec cmd_argument='printf("Text:", Arg1, Arg2, ..., ArgN)'-->`
Description	This SSI function inserts a formatted text into a website or e-mail. The text can contain data from the PLC project in the controllerE. Text formatting corresponds to that of the standard C function printf(). As in the standard C function printf() this text can contain 2 object types: • Simple characters which are copied to the website and • Placeholders with format indications for the next argument. Each placeholder starts with % and ends with a formatting letter. The following can be inserted between the placeholders: • Control character to influence formatting:

-	(Minus) defines a left-aligned output.
+	(Plus) forces the output of the value with sign.
	(Space) inserts a space for positive values instead of the sign.
0	Defines the leading zeros for the output of values.
#	defines an alternative output format. For "o" the output is octal. For "x" or "X", "0x" or "0X" the representation is hexadecimal. The following formats always contain a decimal point: "e", "E" (exponential notation), "f" (floating point without exponent), "g" and "G" (floating point, in exponential notation if required); attached zeroes are not removed.

- A number which defines the minimum field length. The character inserted is normally the space.
- A decimal point to separate the field length and the digits after the decimal point.
- A number for the quantity of the digits after the decimal point for "e", "E" or "F" representations or the number of the significant digits for "g" or "G" representations or the minimum represented digits for integer numbers (leading zeros are inserted if necessary).
- A length indication "h" (short or unsigned short), "l" (small L) or "L" (long or unsigned long).

The formatting characters and their meaning are described in the following table. "%" must always be followed by a formatting character.

Formatting characters	Type of argument; converted into...
d, i	byte short; decimal notation
o	byte short; octal notation (without leading zero)
x, X	byte short; hexadecimal notation (without leading zero "0x" or "0X": output "abcdef" in case of "x", output "ABCDEF" in case of "X".
u	byte short; decimal notation
c	byte short; single character, after conversion to unsigned char.
s	char*; characters of a string are returned until "\0" is reached or until the amount of the defined decimal places has been reached.

f	float; decimal notation of the format "[-]mmm.ddd", where the number of "d" is defined by the decimal places. The presetting of the decimal places is 6; a zero as number of decimal places suppresses the decimal point.
e, E	float; decimal notation of the format "[-]m.dddddd e+-xx" or "[-]m.ddddddE+-xx", where the number of "d" is defined by the number of decimal places. The presetting of the decimal places is 6; a zero as number of decimal places suppresses the decimal point.
g, G	float; %e or %E is used if the exponent is between -4 and the number of decimal places; otherwise %f is used. Subsequent zeros as well as a subsequent decimal point are not displayed.
%	If no argument is indicated, this is shown by %.

Argument	Description
InReadSByte(offset)	Reads a signed byte from the input range with offset
InReadUByte(offset)	Reads an unsigned byte from the input range with offset
InReadSWord(offset)	Reads a signed word (short) from the input range with offset
InReadUWord(offset)	Reads an unsigned word (short) from the input range with offset
InReadSLong(offset)	Reads a signed longword (long) from the input range with offset
InReadULong(offset)	Reads an unsigned longword (long) from the input range with offset
InReadString(offset)	Reads a string (char*) from the input range with offset
InReadFloat(offset)	Reads a floating point (float) value from the input range with offset
OutReadSByte(offset)	Reads a signed byte from the output range with offset
OutReadUByte(offset)	Reads an unsigned byte from the output range with offset
OutReadSWord(offset)	Reads a signed word (short) from the output range with offset
OutReadUWord(offset)	Reads an unsigned word (short) from the output range with offset
OutReadSLong(offset)	Reads a signed longword (long) from the output range with offset
OutReadULong(offset)	Reads an unsigned longword (long) from the output range with offset
OutReadString(offset)	Reads a ZERO terminated string (Char*) from the output range with offset
OutReadFloat(offset)	Reads a floating point (float) value from the output range with offset

Used in	HTML, e-mails

Command	scanf
Syntax	`<?--#exec cmd_argument='scanf( "ObjName", "format", Arg1, ..., ArgN), ErrVal1, ..., ErrvalN'-->`
Description	This SSI function reads a string generated by an HTML form, interprets it according to the format indication and stores the result in the data for the PLC project in the controllerE. The string format corresponds to the standard C function call scanf(). ObjName – Name of the object with the forwarded data string. format – Defines the format of the data string. Arg1…Argn – defines the targets of the data. ErrVal1…ErrValn – option; defines a value or string which is to be generated in case of a fault. The formatting characters "d", "i", "o", "u" and "x" can be supplemented by a small "L" so that the system generates a long value instead of byte or short. Standard output: - Write succeeded - Write failed More information about SSI outputs → chapter Redirecting SSI output (→ page 206).
Used in	HTML only

Formatting characters	Input data, type of argument
d	Decimal number; byte, short
i	Number, byte, short. Format octal (leading zero) or hexadecimal (leading "0x" or "0X")
o	Octal number (with or without leading zero); byte, short
u	Decimal number without sign; unsigned byte, unsigned short
x	Hexadecimal number (with or without leading "0x" or "0X"); byte, short
c	Character; char*. The following input characters (default=1) are stored at the indicated point. The normal suppression of space characters is not active. "%1s" reads the next character (≠ space characters).
s	Character string (string); char*. shows a sequence (array) of characters followed by "\0".
e, f, g	Floating point number with optional sign, decimal point and exponent; float*
%	Character "%"; not assigned.

Argument	Description
OutWriteByte(offset)	Writes a byte in the output range to the address offset
OutWriteWord(offset)	Writes a word (short) in the output range to the address offset
OutWriteLong(offset)	Writes a long value in the output range to the address offset
OutWriteString(offset)	Writes a string in the output range to the address offset
OutWriteFloat(offset)	Writes a float value in the output range to the address offset

Command	GetText
Syntax	`<?--#exec cmd argument='GetText( "ObjName", OutWriteString ( offset ),n)'-->`
Description	This SSI function gets a text from an object and stores it in the output area. "offset" defines the offset to the beginning of the output area. "n" (optional) defines the maximum number of characters to be read. Standard outputs: - Write succeeded - Write failed More information about SSI outputs → chapter Redirecting SSI output (→ page 206).
Used in	HTML only

Command	IncludeFile
Syntax	<?--#exec cmd_argument='IncludeFile ("file name")'-->
Description	This SSI function integrates the contents of a file into a website or e-mail. Standard outputs: - <file content> - Failed to open < file name> More information about SSI outputs → chapter Redirecting SSI output (→ page 206).
Used in	HTML, e-mails

Command	SaveToFile
Syntax	`<?--#exec cmd_argument='SaveToFile( "file name", "Separator", [Append|Overwrite] )'-->`
Description	This SSI function saves the contents of an assigned form in a file. The transferred name/value pair separated by the specified separator is added to the file "filename". These data are either appended or overwrite the current contents of the file. Standard output: - Form saved to file - Failed to save form More information about SSI outputs → chapter Redirecting SSI output (→ page 206).
Used in	HTML only

Command	SaveDataToFile
Syntax	`<?--#exec cmd_argument='SaveDataToFile'("File name", "Object name", [Append|Overwrite])'-->`
Description	This SSI function saves the contents of an assigned form in a file. The parameter "object name" is optional When using "object name" only the data of the indicated object are stored in the file as a result of the function. If this parameter is not used, the data of all objects are stored in this form. These data are either appended or overwrite the current contents of the file. Standard output: - Form saved to file - Failed to save form More information about SSI outputs → chapter Redirecting SSI output (→ page 206).
Used in	HTML only

Command	DisplayRemoteUser
Syntax	`<?--#exec cmd:argument='DisplayRemoteUser'-->`
Description	This function returns the user name of the currently logged on user.
Used in	HTML only

12.2 Redirecting SSI output

There are two options for redirecting the outputs of SSI functions:

- General SSI output redirection by creating a file named `ssi_str.cfg` as well as the option to generate own error messages.
- Temporary redirection by calling the SSI function "SSIOutput".

12.2.1 The general redirection of SSI outputs by means of the file 'ssi_str.cfg'.

If the file `'ssi_str.cfg'` is available in the file system of the Ethernet interface and conforms to the specification below, the output strings of the file `ssi_str.cfg` are used instead of the predefined output strings.

To do so, the file format must be as follows:

```
[StoreEtnConfig]
Success: "Text to be displayed if execution is successful"
Invalid IP: "Text to be displayed if IP address invalid"
Invalid Subnet: "Text to be displayed if subnet mask invalid"
Invalid Gateway: "Text to be displayed if gateway address invalid"
Invalid Email server: "Text to be displayed if SMTP address invalid"
Invalid IP or Subnet: "Text to be displayed if the IP address and subnet mask are not correct"
Invalid DNS1: "Text to be displayed if the primary DNS server cannot be found"
Invalid DNS2: "Text to be displayed if the secondary DNS server cannot be found"
Save Error: "Text to be displayed if saving operation has failed"
Invalid DHCP state: "Text to be displayed if DHCP status invalid"

[scanf]
Success: "Text to be displayed if execution is successful"
Failure: "Text to be displayed if execution has failed"

[IncludeFile]
Failure: "Text to be displayed if execution has failed"
NOTE: the file name can be integrated into the text by inserting the placeholder "%s".

[SaveToFile]
Success: "Text to be displayed if execution is successful"
Failure: "Text to be displayed if execution has failed"
NOTE: the file name can be integrated into the text by inserting the placeholder "%s".

[SaveDataToFile]
Success: "Text to be displayed if execution is successful"
Failure: "Text to be displayed if execution has failed"
NOTE: the file name can be integrated into the text by inserting the placeholder "%s".

[GetText]
Success: "Text to be displayed if execution is successful"
Failure: "Text to be displayed if execution has failed"
```

By inserting the entry [File path] in the first line of the configuration file `ssi_str.cfg` and an additional path/file indication in the second line, the contents for the SSI outputs can also be taken from an additional file.

Example:

Contents of the configuration file `ssi_str.cfg`:

[File path]

\user\ssi_strings.cfg

The file `ssi_strings.cfg` must then have the same structure as described above for the configuration file `ssi_str.cfg`.

12.2.2 The temporary redirection of SSI outputs

The output of the following SSI function can be redirected by calling the function SsiOutput. Only the outputs of the next following function are redirected! The allowed length of each of the two character strings is max. 128 bytes.

Command	SsiOutput
Syntax	`<?--#exec cmd_argument='SsiOutput( "Success string", "Failure string" )'-->`
Used in	HTML, e-mails

13 Glossary of Terms

A

A/B slave

AS-i slave with an A or B being appended to its address number and which may therefore be present twice on the →master.

Address

This is the "name" of the bus participant. All participants need a unique address so that the signals can be exchanged without problem.

Application software

Software specific to the application, implemented by the machine manufacturer, generally containing logic sequences, limits and expressions that control the appropriate inputs, outputs, calculations and decisions

Necessary to meet the specific (→SRP/CS) requirements.
→ Programming language, safety-related

Architecture

Specific configuration of hardware and software elements in a system.

B

Baud

Baud, abbrev.: Bd = unit for the data transmission speed. Do not confuse baud with "bits per second" (bps, bits/s). Baud indicates the number of changes of state (steps, cycles) per second over a transmission length. But it is not defined how many bits per step are transmitted. The name baud can be traced back to the French inventor J. M. Baudot whose code was used for telex machines.

1 MBd = 1024 x 1024 Bd = 1 048 576 Bd

Bus

Serial data transmission of several participants on the same cable.

C

CAN

CAN = **C**ontroller **A**rea **N**etwork

CAN is a priority controlled fieldbus system for larger data volumes. It is available in different variants, e.g. "CANopen" or "CAN in Automation" (CiA).

CoDeSys

CoDeSys® is a registered trademark of 3S – Smart Software Solutions GmbH, Germany.

"CoDeSys for Automation Alliance" associates companies of the automation industry whose hardware devices are all programmed with the widely used IEC 61131-3 development tool CoDeSys®.

Homepage → http://www.3s-software.com

ControllerE

Master in the AS-i bus system of the generation E.

Cycle time

This is the time for one cycle. The following happens:

- PLC cycle: The PLC program performs one complete run.
- AS-i cycle: all AS-i slaves are updated (5...10 ms).

Cyclical polling

AS-i master cyclically polls the data of all →slaves in the bus (see above). The data is updated in the →master after max. 5 ms. If A/B slaves are used, the →cycle time can be extended to 10 ms.

D

DeviceNet

Fieldbus system for larger data volumes based on →CAN technology, requires special cables, complex connection technology. Can be used e.g. as a supplier for AS-i over longer distances. Corresponding →gateways are available.

DHCP

DHCP = **D**ynamic **H**ost **C**onfiguration **P**rotocol = protocol for the dynamic configuration by the →host

DHCP is a protocol that provides dynamic configuration of IP addresses and associated information. The protocol supports use of IP addresses which are only available in limited number by a centralised management of the address assignment.

The participant logs on to a server with this service when it is switched on in a network for the first time. The server assigns a local free →IP address to the participant.

E

EMV

EMC = **E**lectro **M**agnetic **C**ompatibility

According to the EC directive (2004/108/EEC) concerning electromagnetic compatibility (in short EMC directive) requirements are made for electrical and electronic apparatus, equipment, systems or components to operate satisfactorily in the existing electromagnetic environment. The devices must not interfere with their environment and must not be adversely influenced by external electromagnetic interference.

Ethernet

Ethernet is a widely used, manufacturer-independent technology which enables data transmission in the network at a speed of 10 or 100 million bits per second (Mbps). Ethernet belongs to the family of so-called "optimum data transmission" on a non exclusive transmission medium. The concept was developed in 1972 and specified as IEEE 802.3 in 1985.

F

FE – functional earth

Functional **e**arth is a reference potential which is not connected to protective earth or only connected when special measures are taken. The functional earth serves as equalisation of potential for an ungrounded installation (e.g. →SELV).

Fieldbus

A →bus for industrial applications: mechanically extremely robust and excellent data protection.

Firmware

System software, basic program in the device, virtually the operating system.

The firmware establishes the connection between the hardware of the device and the user software. This software is provided by the manufacturer of the controller as a part of the system and cannot be changed by the user.

G

Gateway

access, coupler

Gateways enable connection of completely different systems. Gateways are used when two incompatible network types are to be connected by converting the protocol of one system to the protocol of the other system.

Here: connection between AS-i and higher-level fieldbus systems such as →Ethernet DP, →DeviceNet, Interbus-S or other interfaces, e.g. RS-485. The device includes an AS-i master which is directly coupled to the →host interface (e.g. →Ethernet DP slave).

GSD

Geräte-**S**tamm-**D**atei (deutsch) = Device Master File

Describes the interface to the device to be connected to the fieldbus. The file is provided on the ifm CD (→ folder gateway).

H

Host

The controller in the hierarchy above the AS-i master, e.g. a PLC or a processor.

I

ID

ID = **Id**entifier

Name to differentiate the devices / participants connected to a system or the message packets transmitted between the participants.

Instructions

Superordinate word for one of the following terms:
installation instructions, data sheet, user information, operating instructions, device manual installation information, online help, system manual, programming manual, etc.

Intended use

Use of a product in accordance with the information provided in the instructions for use.

IP address

IP = **I**nternet **P**rotocol

The IP address is a number which is necessary to clearly identify an internet participant. For the sake of clarity the number is written in 4 decimal values, e.g. 127.215.205.156.

J

Jitter

Jitter means a slight fluctuation in accuracy in the transmission cycle when transmitting digital signals. More generally, jitter in transmission technology means an abrupt and undesired change of the signal characteristics.

L

LAS

List of **A**ctive **S**laves

In this slave list the controllerE enters the slaves detected as active for this AS-i master.

LDS

List of **D**etected **S**laves

In this slave list the controller enters the slaves detected as present for this AS-i master.

LED

LED = **L**ight **E**mitting **D**iode

Light emitting diode, also called luminescent diode, an electronic element of high coloured luminosity at small volume with negligible power loss.

LFS

List of **F**ailed **S**laves = list of slaves with configuration errors

In this slave list the controller enters the slaves with a projection error on this AS-i master.

Link

A link is a cross-reference to another part in the document or to an external document.

LPS

List of **P**rojected **S**laves

In this slave list the controller enters the slaves projected for this AS-i master.

M

MAC-ID

MAC = **M**anufacturer‘s **A**ddress **C**ode = manufacturer's serial number
→ID = **Id**entifier

Every network card has a MAC address, a clearly defined worldwide unique numerical

code, more or less a kind of serial number. Such a MAC address is a sequence of 6 hexadecimal numbers, e.g. "00-0C-6E-D0-02-3F".

Master

Handles the complete organisation on the bus. The master decides on the bus access time and polls the →slaves cyclically.

Master-slave communication

AS-i strictly operates to the master-slave principle. The master polls all slaves one after the other in always the same order. Only one master per network line is allowed (→cyclical polling).

MBd

MegaBaud

Baud, abbrev.: Bd = unit for the data transmission speed. Do not confuse baud with "bits per second" (bps, bits/s). Baud indicates the number of changes of state (steps, cycles) per second over a transmission length. But it is not defined how many bits per step are transmitted. The name baud can be traced back to the French inventor J. M. Baudot whose code was used for telex machines.

1 MBd = 1024 x 1024 Bd = 1 048 576 Bd

Modbus

The Modbus protocol is a communication protocol based on a →master/slave architecture and was generated by Modicon in 1979 for communication with its PLCs. In the industry, Modbus has become a de facto standard.

Modbus/TCP is based on →Ethernet TCP/IP. Modbus/TCP ports the protocol defined for the serial interface to TCP. The →IP address clearly identifies each device in a network. Therefore the slave address was used to identify one of several logical units (unit IDs) in a physical device. To do so, the extended IP addressing is used.

Example: 192.168.83.28.1 means unit ID 1 on IP address 192.168.83.28.

*) Modicon passed from AEG to the group Schneider in 1994.

O

Operating system

Basic program in the device, establishes the connection between the hardware of the device and the user software.

OSSD

OSSD = **O**utput **S**ignal **S**witching **D**evice

= output signal of a switching device. Here: output signal of an AS-i safety monitor.

P

Password

In the menu [System Setup], menu item [Password] the handling can be restricted or enabled. When delivered, the device is in the user mode. By entering an invalid password (e.g. 1000) all menu items which can change settings are blocked.

PELV

PELV = **P**rotective **E**xtra **L**ow **V**oltage

Functional extra low voltage with safe separation, grounded variant of SELV.

Extra low voltage with safe separation (grounded variant of SELV). The specification as PELV system to IEC 364-4-41 covers a measure to protect against direct and indirect contact with dangerous voltages by a "safe separation" between primary and secondary side in the device (e.g. power supply to PELV specification).

For this reason no separate PE conductor is required in a PELV system. It is allowed to ground circuits and / or bodies in a PELV system.

Pictogram

Pictograms are figurative symbols which convey information by a simplified graphic representation.

→ Chapter What do the symbols and formats stand for? (→ page 7)

Polling

to poll = to count votes

The controller master fetches the data from every participant in the system successively:

1. Master calls participant 1.
2. Participant 1 replies with its current data (actual values).
3. Master transfers more data (target values) to participant 1, if needed.
4. Participant 1 acknowledges reception of the data.

etc. the same procedure for each further participant.

Cyclical polling: AS-i master cyclically polls the data of all →slaves in the bus (see above). The data is updated in the →master after max. 5 ms. If A/B slaves are used, the →cycle time can be extended to 10 ms.

R

redundant

Redundancy is the presence of more than the necessary means so that a function unit performs a requested function or that data can represent information.

Several kinds of redundancy are distinguished:

- Functional redundancy aims at designing safety-related systems in multiple ways in parallel so that in the event of a failure of one component the others ensure the task.
- In addition it is tried to separate redundant systems from each other with regard to space. Thus the risk that they are affected by a common interference is minimised.
- Finally, components from different manufacturers are sometimes used to avoid that a systematic fault causes all redundant systems to fail (diverse redundancy).

The software of redundant systems should differ in the following aspects:

- specification (different teams),
- specification language,
- programming (different teams),
- programming language,
- compiler.

remanent

Remanent data is protected against data loss in case of power failure.

The operating system for example automatically copies the remanent data to a flash memory as soon as the voltage supply falls below a critical value. If the voltage supply is available again, the operating system loads the remanent data back to the RAM memory.

The data in the RAM memory of a controller, however, is volatile and normally lost in case of power failure.

RTS

RTS = **R**un **T**ime **S**ystem

Runtime systems are basic versions of applications. These minimum versions are supplied with certain products to meet the prerequisites for the execution of the actual product or to be able to look at or use results generated by this product on other processors: making available all routines required to execute a program in a programming language, e.g. interactions with the →operating system, memory requirements, error routines, inputs and outputs.

S

SELV

SELV = **S**afety **E**xtra **L**ow **V**oltage

Active parts of safety extra low voltage circuits must neither be connected to ground nor to protective wires of other circuits. They must be safely separated from active parts with higher voltage.

SELV circuit = secondary circuit (output voltage) which is rated and protected so that its voltages do not exceed a safe value in case of correct operation (of the power supply) or in case of a single fault (of the power supply).

SELV circuits are separated from the input voltage (mains voltage) by double or enhanced insulation. The voltage value must not exceed 60 V DC (or 42.4 V AC).

Single slave

→Slave whose address number may only occur once on the →master.

Slave

Passive participant on the bus, only replies on request of the →master. Slaves have a clearly defined and unique →address in the bus.

Symbols

Pictograms are figurative symbols which convey information by a simplified graphic representation.

→ Chapter What do the symbols and formats stand for? (→ page 7)

T

Target

The target indicates the target system where the PLC program is to run. The target contains the files (drivers and if available specific help files) required for programming and parameter setting.

TCP

The **T**ransmission **C**ontrol **P**rotocol is part of the TCP/IP protocol family. Each TCP/IP data connection has a transmitter and a receiver. This principle is a connection-oriented data transmission. In the TCP/IP protocol family the TCP as the connection-oriented protocol assumes the task of data protection, data flow control and takes measures in the event of data loss.

(compare: →UDP)

U

UDP

UDP (**U**ser **D**atagram **P**rotocol) is a minimal connectionless network protocol which belongs to the transport layer of the internet protocol family. The task of UDP is to ensure that data which is transmitted via the internet is passed to the right application.

At present network variables based on CAN and UDP are implemented. The values of the variables are automatically exchanged on the basis of broadcast messages. In UDP they are implemented as broadcast messages, in CAN as PDOs. These services are not confirmed by the protocol, i.e. it is not checked whether the message is received. Exchange of network variables corresponds to a "1 to n connection" (1 transmitter to n receivers).

Unit ID

→Modbus

Use, intended

Use of a product in accordance with the information provided in the instructions for use.

W

Watchdog

In general the term watchdog is used for a component of a system which watches the function of other components. If a possible malfunction is detected, this is either signalled or suitable program branchings are activated. The signal or branchings serve as a trigger for other co-operating system components to solve the problem.

14 Index