Fieldbus Appendix
Anybus-S Ethernet

Modbus/TCP & IT Functionality

HMS

Connecting Devices™

HALMSTAD • CHICAGO • KARLSRUHE • TOKYO • BEIJING • MILANO • MULHOUSE • COVENTRY • PUNE • COPENHAGEN

Table of Contents

Chapter 10   Fieldbus Specific Area

About This Document

How To Use This Document

This document is intended to be used in conjunction with the Anybus-S Parallel Design Guide. The reader of this document is expected to have basic knowledge in the Modbus/TCP networking system, and communication systems in general. Please consult the general Anybus-S Parallel Design Guide for general information about the Anybus-S platform.

Note: This document describes the functionality provided by the latest firmware release. Some features may be missing or working somewhat differently in older firmware releases. Please contact HMS to obtain the latest version.

Important User Information

The data and illustrations found in this document are not binding. We, HMS Industrial Networks AB, reserve the right to modify our products in line with our policy of continuous product development. The information in this document is subject to change without notice and should not be considered as a commitment by HMS Industrial Networks AB. HMS Industrial Networks AB assumes no responsibility for any errors that may appear in this document.

There are many applications of this product. Those responsible for the use of this device must ensure that all the necessary steps have been taken to verify that the application meets all performance and safety requirements including any applicable laws, regulations, codes, and standards.

Anybus® is a registered trademark of HMS Industrial Networks AB. All other trademarks are the property of their respective holders.

The examples and illustrations in this document are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular implementation, HMS cannot assume responsibility or liability for actual use based on these examples and illustrations.

Warning:	This is a class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.
ESD Note:	This product contains ESD (Electrostatic Discharge) sensitive parts that may be damaged if ESD control procedures are not followed. Static control precautions are required when handling the product. Failure to observe this may cause damage to the product.

Related Documents

Document name	Author
Open Modbus/TCP Specification, Release 1.0	Schneider Automation
RFC 821	Network Working Group
RFC 1918	
Anybus-S Parallel Design Guide	HMS

Document History

Summary of Recent Changes (1.44 .. 1.5)

Change	Page(s)
Corrected description of how to format the file system	A-2

Revision List

Revision	Date	Author(s)	Chapter(s)	Description
<1.20	-	-	-	(see previous releases)
1.20	2003-05-26	PeP	All	Major update
1.21	2003-09-24	PeP	7, 3	Minor update
1.22	2003-10-15	ToT	9	Minor corrections in the response messages
1.30	2005-01-13	PeP	All	Minor updates
1.40	2006-06-28	PeP	All	Major rewrite
1.41	2006-10-19	PeP	3, B, C	Minor update
1.42	2007-03-22	ToT	2, 3, 4, 5, 6, 10, A	Minor corrections and updates
1.43	2008-10-30	HeS	10, 11	Minor corrections and updates
1.44	2010-12-03	KeL	P, 2, 5, 9, B, E	Minor corrections and updates
1.5	2016-10-07	KeL	A,	Corrections

Conventions & Terminology

The following conventions are used throughout this document:

- Numbered lists provide sequential steps
- Bulleted lists provide information, not procedural steps
- Mailbox commands that must to be sent "during module initialization" must be sent between the "START_INIT" and "END_INIT" commands.
- The term 'module' is used when referring to the ABS-EIT-2.
- The term 'application' is used when referring to the hardware that is connected to the Anybus Application Connector.
- Hexadecimal values are written in the format NNNNh or 0xNNNN, where NNNN is the hexadecimal value.
- Binary values are written in the format NNNNb, where NNNN is the binary value.
- All pictures in this manual shows the standard version of this product. However, other connectors, leds and switches may be present depending on configuration.
- 16/32 bit values are written in big endian Motorola format
- Floating point values are in the IEEE Standard 754 format

Support

Please consult the support pages at www.anybus.com.

About the Anybus-S Ethernet

The Anybus-S Ethernet communication module provides instant integration in any ethernet based lan via SMTP, FTP, Telnet, HTTP as well as Modbus/TCP. Additional protocols can be implemented on top of TCP/IP or UDP using the transparent socket interface.

The data exchange can be monitored via the built in web server, Modbus/TCP, or using event triggered email messages. SSI (Sever Side Include) technology enables web pages and email messages to carry dynamic content such as I/O data, configuration settings, or even application specific data passed to the Anybus module through the mailbox interface.

As a member of the Anybus concept of interchangeable network products, the Anybus-S Ethernet is compatible with any product that supports the Anybus-S application interface with only little or no software adjustments.

Features

- Supports shielded (FTP) and unshielded (UTP) cables
- Flexible file system providing both volatile and non-volatile storage areas
- Security framework
- Integrated FTP server provides easy file management using standard FTP clients.
- Server Side Include (SSI) capability
- Web server
- Email client (Messages can be triggered by data events or directly by the application)
- Modbus/TCP server

Overview
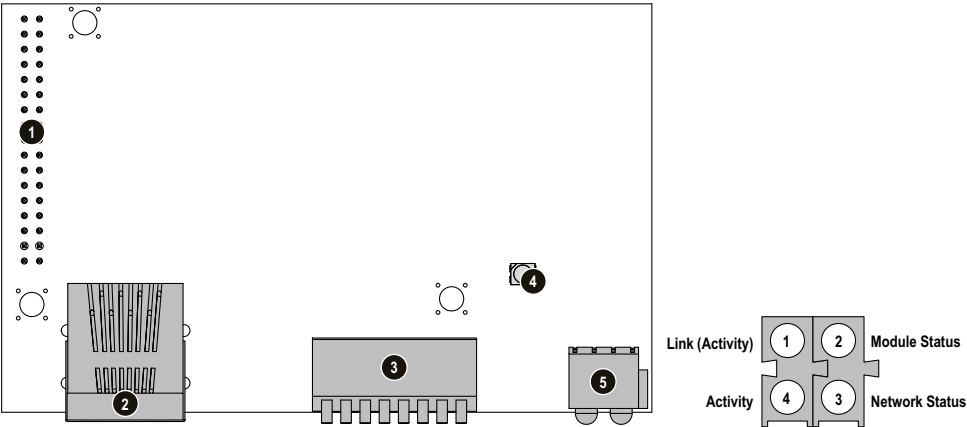
#	Description	Comment
1	Application Connector	See C-1 "Application Connector"
2	Ethernet Connector	See C-1 "Ethernet"
3	Configuration Switch	3-1 "TCP/IP Settings"
4	Anybus Watchdog	Consult the general Anybus-S Parallel Design Guide for further information.
5	Status Indicators	These LEDs indicate run time status and errors to the user, see below.

Status Indicators

Link (Activity)

The behaviour of this led can be altered, see 9-74 "Alter LNK and ACT LED Functionality (ALT_LNK_ACT_LEDS)".

- **Led configuration 1 (Default)**

State	Description
Off	Link not sensed
Green	Link sensed

- **Led configuration 2**

State	Indicates:
Off	Link not sensed
Green	Link sensed
Green, flashing	Activity; receiving or transmitting ethernet packets

Module Status

The behaviour of this led can be changed using the command ALT_MS_LED_FUNCTION, see 9-75 "Alter Module Status LED Functionality (ALT_MS_LED_FUNCTION)".

- **Led configuration 1 (Default)**

State	Description
Off	No power
Green (1Hz)	IP address not set using configuration switch.
Red (1Hz)	Invalid MAC address. (Internal Error)
Red (2Hz	Failed to load Ethernet configuration from FLASH
Red (4Hz	Internal error. (Fatal)
Red	Duplicate IP address detected

- **Led configuration 2**

State	Indicates:
Off	Operating at 10Mbit
Green	Operating at 100Mbit
Red telephone flash (2 flashes on, long off)	Invalid MAC address.
Red telephone flash (3 flashes on, long off)	Failed to load Ethernet configuration from FLASH
Red telephone flash (4 flashes on, long off)	Internal error. (Fatal)
Red telephone flash (5 flashes on, long off)	Duplicate IP address detected

Network Status

The behaviour of this led can be changed using the command ALT_NS_LED_FUNCTION, see 9-76 "Alter Network Status LED Functionality (ALT_NS_LED_FUNCTION)"

- **Led configuration 1 (Default)**

  In this configuration, this led indicates the number established Modbus/TCP connections to the module. The number of established connections is equal to the number of flashes on this led.

- **Led configuration 2**

  This configuration deactivates this led.

Activity

The behaviour of this led can be altered, see 9-76 "Alter Network Status LED Functionality (ALT_NS_LED_FUNCTION)".

- **Led configuration 1 (Default)**

  The Activity led flashes green each time a packet is received or transmitted.

- **Led configuration 2**

  In this configuration, this led is deactivated and the Activity and Link indications are merged.

Basic Operation

Modbus/TCP

General

The Modbus/TCP protocol is an implementation of the standard Modbus protocol running on top of TCP/IP. The built in Modbus/TCP server provides access to the Input- and Output Data areas via a subset of the functions defined in the Modbus/TCP specification.

All Modbus/TCP messages are exchanged through TCP port no. 502 and the Modbus/TCP server can handle a maximum of 8 simultaneous connections. For detailed information regarding the Modbus/TCP protocol, consult the Open Modbus Specification.

Addressing Modes

The module features two different modes of operation regarding the Modbus/TCP communication:

- **Anybus Addressing Mode (Default)**

   Compared to Modbus Addressing Mode, this mode allows data to be addressed in a more flexible manner. Note however that several function codes can be used to access the same data in the module.

   While this may appear confusing at first, it allows data to be manipulated in ways not possible in Modbus Addressing Mode (e.g. it is possible to manipulate individual bits of a register by accessing coils associated with the same memory location).

   See also...

   - 2-3 "Anybus Addressing Mode (Default)"

- **Modbus Addressing Mode**

   In this mode, the Input- and Output Data areas are mapped to different function codes.

   See also...

   - 2-4 "Modbus Addressing Mode"

See also...

- 9-48 "Set Modbus Coil Area Size (MB_SET_COIL_AREA_SIZE)"

Supported Modbus/TCP Commands

All commands according to Class 0 and Class 1 are implemented as well as some Class 2 commands.

Function code	Function name	Class	Addressing Method
1	Read coils	1	Bit
2	Read input discretes	1	Bit
3	Read multiple registers	0	Word
4	Read input registers	1	Word
5	Write coil	1	Bit
6	Write single register	1	Word
7	Read exception status[a]	1	-
15	Force multiple coils	2	Bit
16	Force multiple registers	0	Word
22	Mask write registers	2	Word
23	Read/Write registers	2	Word

a. This command is used to read the exception status reported by the command WR_EXP_STAT

Supported Exception Codes

Code	Name	Description
0x01	Illegal function	The function code in the query is not supported
0x02	Illegal data address	The data address received in the query is outside the initialized memory area
0x03	Illegal data value	The data in the request is illegal

See also...

- 9-48 "Write Exception Status (WR_EXP_STAT)"

Anybus Addressing Mode (Default)

Supported Function Codes

The following function codes can be used in this mode:

Modbus Function	Function Code	Associated with...	No. of I/Os or data points per command
Read Coils	1	Input- and Output Data	1 - 2000 bits
Read Input Discretes	2		1 - 2000 bits
Read Multiple Registers	3		1 - 125 registers
Read Input Registers	4		1 - 125 registers
Write Coil	5	Output Data	1 bit
Write Single Register	6		1 register
Force Multiple Coils	15		1 - 800 bits
Force Multiple Registers	16		1 - 800 registers
Mask Write Register	22		1 register
Read/Write Registers	23	Output Data	125 reg. read/ 100 reg write

Coil & Register Map

The data is mapped to Coils & Registers as follows:

Register #	Coil #	Area	Offset in Area	Comments
1	1... 16	Input Data	000h... 001h	-
2	17... 32		002h... 003h	
3	33... 48		004h... 005h	
4	49... 64		006h... 007h	
...	...		...	
1024	16369... 16384		7FEh... 7FFh	
1025	16385... 16400	Output Data	000h... 001h	-
1026	16401... 16416		002h... 003h	
1027	16417... 16432		004h... 005h	
1028	16433... 16448		006h... 007h	
...	...		...	
2048	32753... 32768		7FEh... 7FFh	

Note 1: The table above applies to all function codes.

Note 2: Coils are mapped MSB first, i.e. coil #1 corresponds to bit 15 of register #1.

Note 3: In this manual "coil" is used for both output and input bit accessible data, as opposed to using "input discrete" for bit accessible input data and "coil" for bit accessible output data.

Modbus Addressing Mode

To enter this mode, the mailbox MB_SET_COIL_AREA_SIZE shall be sent to the module during the initialization. The mailbox command sets up a part of the IN/OUT areas to be coil areas.

Note: In this manual "coil" is used for both output and input bit accessible data, as opposed to using "input discrete" for bit accessible input data and "coil" for bit accessible output data.

Supported Function Codes

The following function codes can be used in this mode:

Modbus Function	Function Code	Associated with...
Read Coils	1	Output Bit Data
Read Input Discretes	2	Input Bit Data
Read Multiple Registers	3	Output Word Data
Read Input Registers	4	Input Word Data
Write Coil	5	Output Bit Data
Write Single Register	6	Output Word Data
Force Multiple Coils	15	Output Bit Data
Force Multiple Registers	16	Output Word Data
Mask Write Register	22	
Read/Write Registers	23	

See also...

Input Data

The Input Data is mapped to Coils & Registers as follows:

Register #	Input Data Bytes	Comments
1	000h + Coil Size In	Each register corresponds to two bytes of Input Data.
2	002h + Coil Size In	Coil Size In is a parameter of the mailbox "MB_-SET_COIL_AREA_SIZE" which is used to create bit accessible data areas, see 9-48.
3	004h + Coil Size In	
5	...	
1024	7FEh + Coil Size In	

Coil #	Input Data Bytes	Comments
1... 8	000h	Each coil corresponds a single bit of Input Data.
9... 16	001h	
17... 24	002h	**Note:** Coils are mapped MSB first, i.e. coil #1 corresponds to the most significant bit of Input Data byte 000h
...	...	
16377... 16384	7FFh	

Output Data

The Output Data is mapped to Coils & Registers as follows:

Register #	Output Data Bytes	Comments
1	000h + Coil Size Out	Each register corresponds to two bytes of Output Data.
2	002h + Coil Size Out	Coil Size Out is a parameter of the mailbox "MB_-SET_COIL_AREA_SIZE" which is used to create bit accessible data areas, see 9-48.
3	004h + Coil Size Out	
5	...	
1024	7FEh + Coil Size Out	

Coil #	Output Data Byte	Comments
1... 8	000h	Each coil corresponds a single bit of Output Data
9... 16	001h	
17... 24	002h	**Note:** Coils are mapped MSB first, i.e. coil #1 corresponds to the most significant bit of Output Data byte 000h
...	...	
16377... 16384	7FFh	

Filesystem

General

The module features a built in filesystem, which is used to store information such as web files, network communication settings, email messages etc.

The filesystem can be accessed using FTP, HTTP, Telnet, and directly by the application via the mailbox interface.

Storage Areas

The filesystem consists of the different storage areas:

- **Non-volatile area (approx. 1,4 Mb)**

  This section is intended for static files such as web files, configuration files etc.

- **Volatile area (approx. 1 Mb)**

  This area is intended for temporary storage; data placed here will be lost in case of power loss or reset. Note that this area is not available by default, and must be mounted by the application during initialisation (see 9-40 "Create RAM disc (CREATE_RAM_DISC)")

Conventions

- '\' (backslash) is used as a path separator
- A 'path' originates from the system root and as such must begin with a '\'
- A 'path' must not end with a '\'
- Names may contain spaces (' ') but must not begin or end with one.
- Names must not contain one of the following characters: '\ / : * ? " < > |'
- Names cannot be longer than 48 characters (plus null termination)
- A path cannot be longer than 256 characters (filename included)
- The maximum number of simultaneously open files is 40
- The maximum number of simultaneously open directories is 40

Important Note:

The non-volatile storage is located in FLASH memory. Each FLASH segment can only be erased approximately 100000 times due to the nature of this type of memory.

The following operations will erase one or more FLASH segments:

- Deleting, moving or renaming a file or directory
- Writing or appending data to an existing file
- Formatting the filesystem

Filesystem Overview
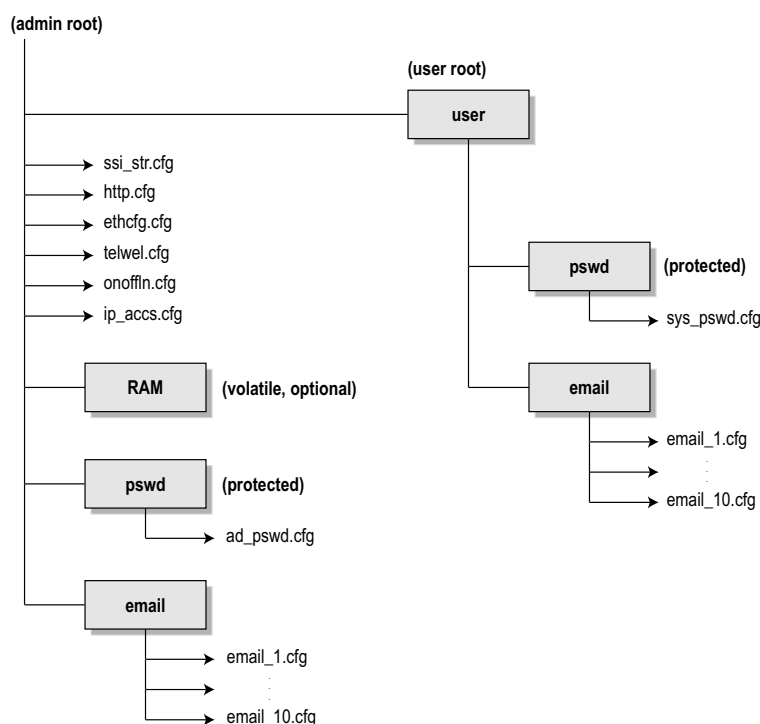
System Files

The filesystem contains a set of files used for system configuration. These files, known as "system files", are regular ASCII files which can be altered using a standard text editor (such as the Notepad in Microsoft Windows™). Note that some of these files may also be altered by the Anybus module itself, e.g. when using SSI (see 8-1 "Server Side Include (SSI)").

The format of the system files are based on the concept of 'keys', where each 'key' can be assigned a value, see example below.

Example:

```
[Key1]
value of key1

[Key2]
value of key2
```

The format of each system file is described later in this document.

The contents of the above files can be redirected:

Example:

In this example, the contents will be loaded from the file 'here.cfg'.

```
[File path]
\i\put\it\over\here.cfg
```

Note: Any directory in the file system can be protected from web access by placing the file web_accs.cfg in the directory, see 6-2 "Authorization".

Basic Network Configuration

TCP/IP Settings

The module offers three modes of operation regarding the TCP/IP settings:

- **Settings specified via Mailbox Interface**

  The application can specify the IP-settings during startup via the mailbox interface. Note that this overrides any settings specified by the on-board Configuration Switches (above) or in the ethernet configuration file (below).

  See also...

    - 9-4 "Set Ethernet Configuration (SET_ETN_CONFIG)"

- **Settings specified in Ethernet Configuration File ('ethcfg.cfg')**

  If the on-board switches are set to 0 (zero), and no settings have been specified by the application during startup, the module will use the settings stored in the system file 'ethcfg.cfg'.

  If this file is missing, the module will attempt to retrieve the settings via DHCP or HICP for 30 seconds. If no configuration has been received within this period, the module will halt and indicate an error on the on-board LEDs.

  Note: This file does not exist by default, but will be created by the module when any settings in the module need to be saved or modified.

  See also...

    - 3-2 "Ethernet Configuration File ('ethcfg.cfg')"

- **Settings specified by Configuration Switches**

  If no settings have been downloaded via the mailbox interface, the IP settings can be specified using the on-board switches. If set to a non-zero value, the module will use the following settings:

IP Address:	192.168.0.x	(x = binary switch value)
Gateway:	255.255.255.0	
Subnet:	255.255.255.0	
DHCP:	OFF	

  The switches specify the binary value of the last byte of the IP address as illustrated by the following example, where the IP address is set to 192.168.0.42.
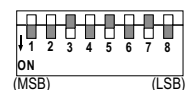
EtherNet/IP

The TCP/IP settings can be accessed from EtherNet/IP through the TCP/IP Interface Object.

See also...

- 9-1 "CIP Object Implementation"

DHCP/BootP

The module can retrieve the TCP/IP settings from a DHCP or BootP server. If no DHCP server is found, the module will fall back on it's current settings (i.e. the settings currently stored in '\ethcfg.cfg').

If no current settings are available (i.e. 'ethcfg.cfg' is missing, or contains invalid settings), the module will halt and indicate an error on the on-board status LEDs (the settings may however still be accessed using HICP or gleaning, see 3-5 "Anybus IPconfig (HICP)" and 3-5 "ARP Gleaning").

Ethernet Configuration File ('ethcfg.cfg')

General

Basic network settings are stored in the system file '\ethcfg.cfg'. Note that the IP-settings may be specified by other sources, see 3-1 "TCP/IP Settings".

Note: This file does not exist by default, but will be created by the module when any settings in the module need to be saved or modified.

Parameter[a]	Default	Description/Comment
IP address	0.0.0.0	
Subnet mask	0.0.0.0	
Gateway address	0.0.0.0	
DHCP/BOOTP	OFF	DHCP/BOOTP <u>Value:</u> <u>Meaning:</u> ON Enabled OFF Disabled
Speed	auto	Speed <u>Value:</u> <u>Meaning:</u> auto Auto negotiation will be used 100 Forces the module to operate only at 100 Mbit 10 Forces the module to operate only at 10 Mbit
Duplex	auto	Duplex <u>Value:</u> <u>Meaning:</u> auto Auto negotiation will be used full Forces the module to operate only at full duplex half Forces the module to operate only at half duplex
HICP Password	"" (empty string)	
SMTP address	"" (empty string)	SMTP login/server settings.
SMTP username	"" (empty string)	Username and password are only necessary if required by the server.
SMTP password	"" (empty string)	
DNS1 address	0.0.0.0	Primary and secondary DNS.
DNS2 address	0.0.0.0	Needed to be able to resolve host names.
Domain name	"" (empty string)	Default domain name for not fully qualified host names (Max. 48 char.)
Host name	"" (empty string)	Host name (Max. 64 char.)
Mcast TTL	1	Multicast Time-To-Live
Mcast Alloc Control	0	Multicast address allocation
Mcast Num Mcast	1	Number of IP multicast addresses
Mcast Start Addr	0	Starting multicast address
Select ACD	1	Enable Address Conflict Detection <u>Value:</u> <u>Meaning:</u> 1 Enabled 0 Disabled
ACD activity[b]	0	State of Address Conflict Detection activity.
Remote MAC[b]	00:00:00:00:00:00	MAC address of the remote node for which an address conflict is detected. Retrieved from the ARP PDU.
ARP PDU[b]	0	Address Resolution Protocol, Protocol Data Unit.

a. When used in the configuration file, a parameter has to be enclosed by brackets.
b. Read-only parameter. Used e.g. to detect why a restart was necessary.

The settings in this file may also be affected by...

- EtherNet/IP (See 3-1 "EtherNet/IP").
- HICP (See 3-5 "Anybus IPconfig (HICP)")
- SSI (See 8-1 "Server Side Include (SSI)")

See also...

- 3-1 "TCP/IP Settings"
- 4-1 "FTP Server"

IP Access Control

It is possible to specify which IP addresses that are permitted to connect to the module. This information is stored in the system file '\ip_accs.cfg'.

File Format:

```
[Web]
xxx.xxx.xxx.xxx
```
• Nodes listed here may access the web server

```
[FTP]
xxx.xxx.xxx.xxx
```
• Nodes listed here may access the FTP server

```
[Modbus/TCP]
xxx.xxx.xxx.xxx
```
• Nodes listed here may access the module via Modbus/TCP

```
[EtherNet/IP]
xxx.xxx.xxx.xxx
```
• Nodes listed here may access the module via EtherNet/IP

```
[Telnet]
xxx.xxx.xxx.xxx
```
• Nodes listed here may access the module via Telnet

```
[All]
xxx.xxx.xxx.xxx
```
• Fallback setting, used by the module when one or several of the keys above are omitted. If this value is not specified, the module will accept all connections regardless of the other

Note: '*' may be used as a wildcard to select IP series.

On/Off Line Configuration

By default, the On/Off Line indication (Event Notification) is triggered by the Link Status. Other triggering options can however be specified in the system file '\onoffln.cfg', which looks as follows:

File Format:
```
[ON/OFF-line trigger]
Modbus

[Timeout]
10

[Commands]
3, 16, 23

[ON-line method]
1
```

- **ON/OFF-line trigger source**

  Values: 'Link' (default), 'EIP', 'Modbus' or a combination

- **Timeout Value**

  Range: 1... 65535 (default = 1).

  A value of 10 equals 1000ms.

- **Commands (Optional)**

  Selects what Modbus commands that must be received during the timeout period.

  If the keyword 'ALL' is given (default), the ON/OFF line functionality will trigger on all Modbus commands.

- **ON-line method (Optional)**

  Defines how to handle data in the OUT I/O area when going from OFF-line to ON-line.

  If "1" (default) , "old data" is restored
  If "2", "OFF-line" data is kept until overwritten by master.

The contents of this file can be redirected by placing the line '[File path]' on the first row, and a file path on the second.

File example:

```
[File path]
\my_settings\on-off-line_configuration.cfg
```

In this example , the settings described above will be loaded from the file '\my_settings\on-off-line_-configuration.cfg'.

Note 1: The keys '[Timeout]' and '[Commands]' shall only be given if the ON/OFF-line Trigger value is set to 'Modbus'.

Note 2: The settings in this file will be ignored if the application has issued MB_ON_OFF_LINE_-CONFIG.

See also...

- 9-47 "Modbus ON/OFF Line Configuration (MB_ON_OFF_LINE_CONFIG)"
- A-1 "Event Notification Cause/Source Registers"

Anybus IPconfig (HICP)

The module supports the HICP protocol used by the Anybus IPconfig utility from HMS, which can be downloaded free of charge from the HMS website. This utility may be used to configure the network settings of any Anybus product connected to the network. Note that if successful, this will replace the settings currently stored in the configuration file ('ethcfg.cfg').

Upon starting the program, the network is scanned for Anybus products. The network can be rescanned at any time by clicking 'Scan'. In the list of detected devices, the module will appear as 'ABS-EIP'. To alter its network settings, double-click on its entry in the list.

A window will appear, containing the IP configuration and password settings. Validate the new settings by clicking 'Set', or click 'Cancel' to abort.

Optionally, the configuration may be protected from unauthorized access by a password. To enter a password, click on the 'Change password' checkbox, and enter the password under 'New password'. When protected, any changes in the configuration requires that the user supplies a valid password.

When done, click 'Set'. The adopted configuration will be stored in the ethernet configuration file.

Note: The HICP protocol communicates over UDP port 3250.

ARP Gleaning

The module supports the Address Resolution Protocol (ARP), allowing the IP settings to be altered using the ARP-command on a PC.

> *Syntax:*
>
> ```
> arp -s <IP address> <MAC address>
> ping <IP address>
> arp -d <IP address>
> ```

The 'arp -s' command stores the IP and MAC address in the PCs ARP-table. When the 'ping'-command is issued, the PC will address the module with the new IP address; the module recognizes that it was addressed with the correct MAC address and adopts the new IP address from the 'ping' message.

If successful, new settings will be stored in the ethernet configuration file as follows:

IP Address:	xxx.xxx.xxx.xxx	(value supplied in ARP command)
Gateway:	0.0.0.0	(no gateway)
Subnet:	255.255.255.0	
DHCP:	OFF	

Note: This functionality may cause problems if multiple devices continuously issue 'ping'-messages towards the module. The reason for this lies in the very nature of this functionality; since the module adopts the IP address from all 'ping'-messages, any additional 'ping'-messages may cause the module to change back and forth between old and new settings.

See also...

- 9-10 "Disable / Enable ARP Config (ARP_CFG_CONTROL)"

FTP Server

General

The built in FTP server provides a way to access the file system using a standard FTP client.

The following port numbers are used for FTP communication:

- TCP, port 20 (FTP data port)
- TCP, port 21 (FTP command port)

The FTP server can handle a maximum of 16 simultaneous connections.

See also...

- 9-20 "Disable FTP server (DISABLE_FTP_SERVER)"

Security Levels

The FTP-server features two security levels; admin and normal.

- **Normal-level users**

  The root directory will be '\user'.

- **Admin-level users**

  The root directory will be '\', i.e. the user has unrestricted access to the file system.

User Accounts

The user accounts are stored in two files, which are protected from web access:

- **'\user\pswd\sys_pswd.cfg'**

  This file holds the user accounts for normal-level users.

- **'\pswd\ad_pswd.cfg'**

  This file holds the user accounts for admin-level users.

  File Format:

  The format of these files are as follows:

  ```
  Username1:Password1
  Username2:Password2
  Username3:Password3
  ```

Note 1: If no valid user accounts have been defined, or if the mailbox command 'GLOBAL_ADMIN-_MODE' has been issued during startup, the module will grant Admin-level access to all users. In such case, the FTP accepts any username/password combination, and the root directory will be '\'.

Note 2: The FTP server shares user accounts with the Telnet server.

FTP Connection Example (Windows Explorer)

The built in FTP client in Windows Explorer can easily be used to access the file system as follows:

1. Open the Windows Explorer by right-clicking on the 'Start' button and selecting 'Explore'.

2. In the address field, type FTP://<user>:<password>@<address>
   - Substitute <address> with the IP address of the Anybus module
   - Substitute <user> with the username
   - Substitute <password> with the password

3. Press enter. The Explorer will now attempt to connect to the module using the specified settings. If successful, the built in file system is displayed in the Explorer window.
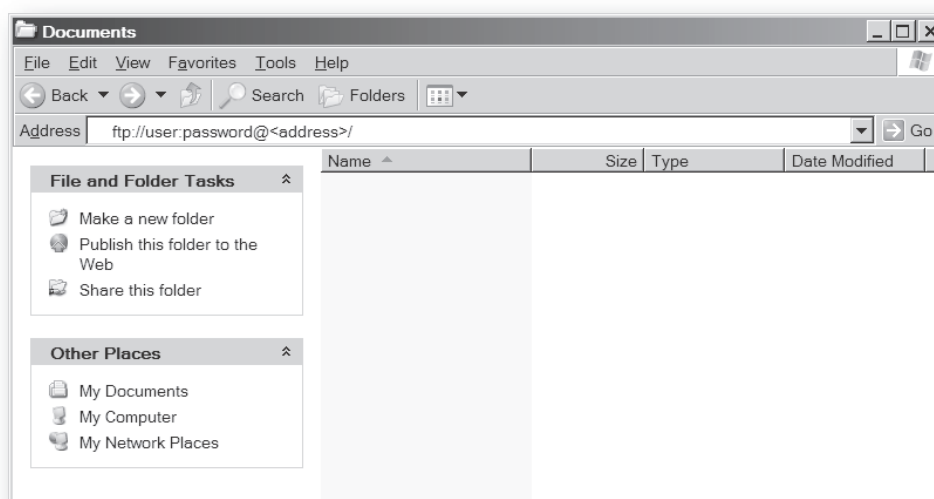
Telnet Server

General

The built in Telnet server provides a way to access the filesystem using a standard Telnet client. The server communicates through TCP port 23. The Telnet server can handle a maximum of 8 simultaneous connections.

Information for the configuration of the Telnet connection is available in the file telwel.cfg. This file is stored in the root directory of the built-infile system and the contenst is sent when a Telnet connection is opened.

See also...

- 9-21 "Disable Telnet Server (DISABLE_TELNET_SERVER)"
- 2-8 "Filesystem Overview"

Security Levels

Just like the FTP server, the Telnet server features two security levels; admin and normal.

- **Normal-level users**

  The root directory will be '\user'.

- **Admin-level users**

  The root directory will be '\', i.e. the user has unrestricted access to the filesystem.

User Accounts

The Telnet server shares user accounts with the FTP server. If no valid user accounts have been defined, or if the mailbox command 'GLOBAL_ADMIN_MODE' has been issued during startup, the module will grant Admin-level access to all users. In such case, no login is required, and the root directory will be '\'.

For more information, see 4-1 "User Accounts"

General Commands

admin

- **Syntax**

  ```
  admin
  ```

- **Description**

  Provided that the user can supply a valid admin username/password combination, this command provides Admin access rights to Normal-level users.

exit

- **Syntax**

  ```
  exit
  ```

- **Description**

  This command closes the Telnet session.

help

- **Syntax**

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

- **Description**

  If no argument is specified, the following menu will be displayed.

  ```
  General commands:

     help      - Help with menus
     version   - Display version information
     exit      - Exit station program

  Also try 'help [general|diagnostic|filesystem]'
  ```

version

- **Syntax**

  ```
  version
  ```

- **Description**

  This command will display version information, serial number and MAC ID of the module.

Diagnostic Commands

arps

- **Syntax**

  `arps`
- **Description**

  Display ARP stats and table

iface

- **Syntax**

  `iface`
- **Description**

  Display net interface stats

routes

- **Syntax**

  `routes`
- **Description**

  Display IP route table

sockets

- **Syntax**

  `sockets`
- **Description**

  Display socket list

Filesystem Operations

For commands where filenames, directory names or paths shall be given as an argument the names can be written directly or within quotes. For names including spaces the filenames must be surrounded by quotes. It is also possible to use relative pathnames using '.', '\' and '..'

append

- **Syntax**

  `append [file] ["The line to append"]`
- **Description**

  Appends a line to a file.

cd

- **Syntax**

  `cd [path]`

- **Description**

  Changes current directory.

copy

- **Syntax**

  ```
  copy [source] [destination]
  ```

- **Description**

  This command creates a copy of the source file at a specified location.

del

- **Syntax**

  ```
  del [file]
  ```

- **Description**

  Deletes a file.

dir

- **Syntax**

  ```
  dir [path]
  ```

- **Description**

  Lists the contents of a directory. If no path is given, the contents of the current directory is listed.

df

- **Syntax**

  ```
  df
  ```

- **Description**

  Displays filesystem info.

format

- **Syntax**

  ```
  format
  ```

- **Description**

  Formats the filesystem. This command is only valid for admin level users.

md

- **Syntax**

  md [directory]

- **Description**

  Creates a directory. If no path is given, the directory is created in the current directory.

mkfile

- **Syntax**

  mkfile [filename]

- **Description**

  Creates an empty file.

move

- **Syntax**

  move [source] [destination]

- **Description**

  This command moves a file or directory from the source location to a specified destination.

rd

- **Syntax**

  rd [directory]

- **Description**

  Removes a directory. The directory can only be removed if it is empty.

ren

- **Syntax**

  ren [old name] [new name]]

- **Description**

  Renames a file or directory.

type

- **Syntax**

  type [filename]

- **Description**

  Types the contents of a file.

Web Server

General

The Anybus module features a flexible web server with SSI capabilities. The built in web pages can be customized to fit a particular application and allow access to I/O data and configuration settings.

The web server communicates through port 80 and can handle a maximum of 48 simultaneous connections.

See also...

- 9-18 "Disable Web Server (DISABLE_WEB_SERVER)"

Protected Files

For security reasons, the following files are protected from web access:

- Files located in '\user\pswd'
- Files located in '\pswd'
- Files located in a directory which contains a file named 'web_accs.cfg'

Default Web Pages

The Anybus module contains a set of virtual files that can be used when building a web page for configuration of network parameters. These virtual files can be overwritten (not erased) by placing files with the same name in the root of disc 0.

This makes it possible to, for example, replace the HMS logo by uploading a new logo named '\logo.jpg'. It is also possible to make links from a web page to the virtual configuration page. In that case the link shall point to '\config.htm'.

These virtual files are:

```
\index.htm      - Points to the contents of config.htm
\config.htm     - Configuration frame page
\configform.htm- Configuration form page
\configform2.htm- Configuration form page
\store.htm      - Configuration store page
\logo.jpg       - HMS logo
\configuration.gif- Configuration picture
\boarder.bg.gif- picture
\boarder_m_bg.gif- picture
```

Authorization

Directories can be protected from web access by placing a file called 'web_accs.cfg' in the directory to protect. This file shall contain a list of users that are allowed to access the directory and its subdirectories.

File Format:

```
Username1:Password1
Username2:Password2
...
UsernameN:PasswordN
```
• List of approved users.

```
[AuthName]
(message goes here)
```
Optionally, a login message can be specified by including the key [AuthName]. This message will be displayed by the web browser upon accessing the pro-

The list of approved users can optionally be redirected to one or several other files.

Example:

In this example, the list of approved users will be loaded from the files 'here.cfg' and 'too.cfg'.

```
[File path]
\i\put\it\over\here.cfg
\i\actually\put\some\of\it\over\here\too.cfg

[AuthName]
Yeah. Whatsda passwoid?
```

Note that when using this feature, make sure to put the user/password files in a directory that is protected from web access, see 6-1 "Protected Files".

Content Types

By default, the following content types are recognized by their file extension:

Content Type	File Extension
text/html	*.htm, *.html, *.shtm
image/gif	*.gif
image/jpeg	*.jpeg, *.jpg, *.jpe
image/x-png	*.png
application/x-javascript	*.js
text/plain	*.bat, *.txt, *.c, *.h, *.cpp, *.hpp
application/x-zip-compressed	*.zip
application/octet-stream	*.exe, *.com
text/vnd.wap.wml	*.wml
application/vnd.wap.wmlc	*.wmlc
image/vnd.wap.wbmp	*.wbmp
text/vnd.wap.wmlscript	*.wmls
application/vnd.wap.wmlscriptc	*.wmlsc
text/xml	*.xml
application/pdf	*.pdf

It is possible to configure/reconfigure the reported content types, and which files that shall be scanned for SSI. This is done in the system file '\http.cfg'.

File Format:

```
[FileTypes]
FileType1:ContentType1
FileType2:ContentType2
...
FileTypeN:ContentTypeN

[SSIFileTypes]
FileType1
FileType2
...
FileTypeN
```

Note: Up to 50 content types and 50 SSI file types may be specified in this file.

SMTP Client

General

The built in email client can send predefined email messages based on trigger-events in the dual port memory (DPRAM). The application can also use the client directly via the mailbox interface.

The client supports SSI, however note that some SSI functions cannot be used in email messages (specified separately for each SSI function).

See also...

- 8-1 "Server Side Include (SSI)"
- 9-79 "Send Email (SEND_EMAIL)"

Server Settings

The module needs a valid SMTP server configuration in order to be able to send email messages. These settings are stored in the system file '\ethcfg.cfg'. This file is read once during startup, i.e. any changes requires that the module is restarted in order to have effect. See also...

- 9-15 "Set SMTP Server (SET_SMTP_SERVER)"
- 9-79 "Send Email (SEND_EMAIL)"

Event-Triggered Messages

As mentioned previously, the email client can send predefined message based on events in the DPRAM. In operation, this works as follows:

1. The trigger source is fetched from the dual port memory

2. A logical AND is performed between the trigger source and a mask value

3. The result is compared to a reference value according to a specified operand

4. If the end result is true, the email is sent to the specified recipient(s).

Which events that shall cause a particular message to be sent, is specified separately for each message. For more information, see 7-2 "Email Definitions".

Note that the DPRAM is scanned once every 0.5 second, i.e. a trigger-event must be present longer than 0.5 seconds to ensure that it is detected by the Anybus module.

Email Definitions

The email definitions are stored in the following two directories:

- **'\user\email'**
  This directory holds up to 10 messages which can be altered by normal-level FTP-users.
- **'\email'**
  This directory holds up to 10 messages which can be altered by admin-level FTP-users.

Email definition files must be named 'email_1.cfg', 'email_2.cfg'... 'email_10.cfg' in order to be properly recognized by the module.

File Format:

```
[Register]
Area, Offset, Type

[Register Match]
Value, Mask, Operand

[To]
recipient

[From]
sender

[Subject]
subject line

[Headers]
Optional extra headers

[Message]
message body
```

Key	Value	Scanned for SSI
Area	Source area in DPRAM. Possible values are 'IN' or 'OUT'	No
Offset	Source offset, written in decimal or hexadecimal.	
Type	Source data type. Possible values are 'byte', 'word', and 'long'	
Value	Used as a reference value for comparison.	
Mask	Mask value, applied on the trigger source prior to comparison (logical AND).	
Operand	Possible values are '<', '=' or '>'	
To	Email recipient	Yes
From	Sender email address	
Subject	Email subject. One line only.	
Headers	Optional; may be used to provide additional headers.	
Message	The actual message.	

Note: Hexadecimal values must be written with the prefix '0x' in order to be recognized by the module.

Server Side Include (SSI)

General

Server Side Include (from now on referred to as SSI) functionality enables dynamic content to be used on web pages and in email messages.

SSI are special commands embedded in the source document. When the Anybus module encounters such a command, it will execute it, and replace it with the result (when applicable).

Syntax

The 'X's below represents a command opcode and parameters associated with the command.

```
<?--#exec cmd_argument='XXXXXXXXXXXXXXXXXXXXXX'-->
```

Example

The following example causes a web page to display the Ethernet Mac ID of the module:

```
<HTML>
<HEAD><TITLE>SSI Test</TITLE></HEAD>
<BODY>
The Ethernet Mac ID of the Anybus module is:
<?--#exec cmd_argument='DisplayMacID'-->
</BODY>
</HTML>
```

Resulting webpage:
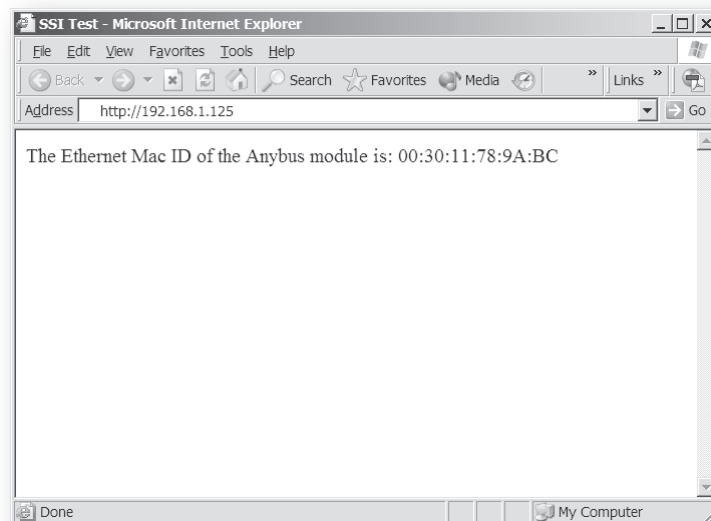
Functions

DisplayMacID

This function returns the MAC ID in format xx:xx:xx:xx:xx:xx.

Syntax:

```
<?--#exec cmd_argument='DisplayMacId'-->
```

DisplaySerial

This function returns the serial number of the Anybus module.

Syntax:

```
<?--#exec cmd_argument='DisplaySerial'-->
```

DisplayFWVersion

This function returns the main firmware revision of the Anybus module.

Syntax:

```
<?--#exec cmd_argument='DisplayFWVersion'-->
```

DisplayBLVersion

This function returns the bootloader firmware revision of the Anybus module.

Syntax:

```
<?--#exec cmd_argument='DisplayBLVersion'-->
```

DisplayIP

This function returns the currently used IP address.

Syntax:

```
<?--#exec cmd_argument='DisplayIP'-->
```

DisplaySubnet

This function returns the currently used Subnet mask.

Syntax:

```
<?--#exec cmd_argument='DisplaySubnet'-->
```

DisplayGateway

This function returns the currently used Gateway address.

Syntax:

```
<?--#exec cmd_argument='DisplayGateway'-->
```

DisplayDNS1

This function returns the address of the primary DNS server.

Syntax:

```
<?--#exec cmd_argument='DisplayDNS1'-->
```

DisplayDNS2

This function returns the address of the secondary DNS server.

Syntax:

```
<?--#exec cmd_argument='DisplayDNS2'-->
```

DisplayHostName

This function returns the hostname.

Syntax:

```
<?--#exec cmd_argument='DisplayHostName'-->
```

DisplayDomainName

This function returns the default domain name.

Syntax:

```
<?--#exec cmd_argument='DisplayDomainName'-->
```

DisplayDhcpState

This function returns whether DHCP/BootP is enabled or disabled.

Syntax:

```
<?--#exec cmd_argument='DisplayDhcpState( "Output when ON", "Output when OFF"
)'-->
```

DisplayDhcpSupport

DHCP support can be disabled using the mailbox command DISABLE_DHCP, see 9-12 "Disable DHCP (DISABLE_DHCP)". This function returns 'Arg1' if it's enabled and 'Arg2' if it's disabled.

Syntax:

```
<?--#exec cmd_argument='DisplayDhcpSupport( "Arg1", "Arg2" )'-->
```

DisplayEmailServer

This function returns the currently used SMTP server address.

Syntax:

```
<?--#exec cmd_argument='DisplayEmailServer'-->
```

DisplaySMTPUser

This function returns the username used for SMTP authentication.

Syntax:

```
<?--#exec cmd_argument='DisplaySMTPUser'-->
```

DisplaySMTPPswd

This function returns the password used for SMTP authentication.

> *Syntax:*
>
> ```
> <?--#exec cmd_argument='DisplaySMTPPswd'-->
> ```

StoreEtnConfig

Note: This function cannot be used in email messages.

This SSI function stores a passed IP configuration in the configuration file 'ethcfg.cfg'.

> *Syntax:*
>
> ```
> <?--#exec cmd_argument='StoreEtnConfig'-->
> ```
>
> Include this line in a HTML page and pass a form with new IP settings to it.
>
> *Accepted fields in form:*
>
> ```
> SetIp
> SetSubnet
> SetGateway
> SetEmailServer
> SetDhcpState - value "on" or "off"
> SetDNS1
> SetDNS2
> SetHostName
> SetDomainName
> SetSMTPUser
> SetSMTPPswd
> ```
>
> *Default output:*
>
> ```
> Invalid IP address!
> Invalid Subnet mask!
> Invalid Gateway address!
> Invalid IP address or Subnet mask!
> Invalid Email Server IP address!
> Invalid DHCP state!
> Invalid DNS1!
> Invalid DNS2!
> Configuration stored correctly.
> Failed to store configuration.
> ```

GetText

Note: This function cannot be used in email messages.

This SSI function gets the text from an object and stores it in the OUT area.

> *Syntax:*
>
> ```
> <?--#exec cmd_argument='GetText( "ObjName", OutWriteString ( offset ), n )'-->
> ```
>
> ObjName   - Name of object.
> offset   - Specifies the offset from the beginning of the OUT area.
> n   - Specifies maximum number of characters to read (Optional)
>
> *Default output:*
>
> ```
> Success     - Write succeeded
> Failure     - Write failed
> ```

printf

This SSI function includes a formatted string, which may contain data from the Anybus IN/OUT area, on a web page. The formatting of the string is equal to the standard C function printf().

Syntax:

```
<?--#exec cmd_argument='printf("String to write", Arg1, Arg2, ..., ArgN)'-->
```

Like the standard C function printf() the "String to write" for this SSI function contains two types of objects: Ordinary characters, which are copied to the output stream, and conversion specifications, each of which causes conversion and printing of the next successive argument to printf. Each conversion specification begins with the character % and ends with a conversion character. Between the % and the conversion character there may be, in order:

- Flags (in any order), which modify the specification:
  - which specifies left adjustment of the converted argument in its field.
  + which specifies that the number will always be printed with a sign
  (space) if the first character is not a sign, a space will be prefixed.
  0 for numeric conversions, specifies padding to the field with leading zeroes.
  # which specifies an alternate output form. For o, the first digit will be zero. For x or X, 0x or 0X will be prefixed to a non-zero result. For e, E,f, g and G, the output will always have a decimal point; for g and G, trailing zeros will not be removed.

- A number specifying a minimum field width. The converted argument will be printed in a field at least this wide, and wider if necessary. If the converted argument has fewer characters than the field width it will be padded on the left (or right, if left adjustment has been requested) to make up the field width. The padding character is normally space, but can be 0 if the zero padding flag is present.

- A period, which separates the field width from the precision.

- A number, the precision, that specifies the maximum number of characters to be printed from a string, or the number of digits to be printed after the decimal point for e, E, or F conversions, or the number of significant digits for g or G conversion, or the minimum number of digits to be printed for an integer (leading 0s will be added to make up the necessary width)

- A length modifier h, l (letter ell), or L. "h" Indicates that the corresponding argument is to be printed as a short or unsigned short; "l" indicates that the argument is along or unsigned long.

The conversion characters and their meanings are shown below. If the character after the % is not a conversion character, the behaviour is undefined.

Character	Argument type, Converted to
d, i	byte, short; decimal notation (For signed representation. Use signed argument)
o	byte, short; octal notation (without a leading zero).
x, X	byte, short; hexadecimal notation (without a leading 0x or 0X), using abcdef for 0x or ABCDEF for 0X.
u	byte, short; decimal notation.
c	byte, short; single character, after conversion to unsigned char.
s	char*; characters from the string are printed until a "\0" is reached or until the number of characters indicated by the precision have been printed
f	float; decimal notation of the form [-]mmm.ddd, where the number of d's is specified by the precision. The default precision is 6; a precision of 0 suppresses the decimal point.
e, E	float; decimal notation of the form [-]m.dddddd e+-xx or[-]m.ddddddE+-xx, where the number of d's specified by the precision. The default precision is 6; a precision of 0 suppresses the decimal point.
g, G	float; %e or %E is used if the exponent is less than -4 or greater than or equal to the precision; otherwise %f is used. Trailing zeros and trailing decimal point are not printed.
%	no argument is converted; print a %

The arguments that can be passed to the SSI function *printf* are:

Argument	Description
InReadSByte(*offset*)	Read a signed byte from position *offset* in the IN area
InReadUByte(*offset*)	Read an unsigned byte from position *offset* in the IN area
InReadSWord(*offset*)	Read a signed word from position *offset* in the IN area
InReadUWord(*offset*)	Read an unsigned word from position *offset* in the IN area
InReadSLong(*offset*)	Read a signed longword from position *offset* in the IN area
InReadULong(*offset*)	Read an unsigned longword from position *offset* in the IN area
InReadString(*offset*)	Read a string (char*) from position *offset* in the IN area
InReadFloat(*offset*)	Read a floating point (float) value from position *offset* in the IN area
OutReadSByte(*offset*)	Read a signed byte from position *offset* in the OUT area
OutReadUByte(*offset*)	Read an unsigned byte from position *offset* in the OUT area
OutReadSWord(*offset*)	Read a signed word (short) from position *offset* in the OUT area
OutReadUWord(*offset*)	Read an unsigned word (short) from position *offset* in the OUT area
OutReadSLong(*offset*)	Read a signed longword (long) from position *offset* in the OUT area
OutReadULong(*offset*)	Read an unsigned longword (long) from position *offset* in the OUT area
OutReadString(*offset*)	Read a null-terminated string from position *offset* in the OUT area
OutReadFloat(*offset*)	Read a floating point (float) value from position *offset* in the OUT area
MbReadSByte(*id*)	Read a signed byte (short) from the application via the mailbox interface
MbReadUByte(*id*)	Read an unsigned byte (short) from the application via the mailbox interface
MbReadSWord(*id*)	Read a signed word from the application via the mailbox interface
MbReadUWord(*id*)	Read an unsigned word from the application via the mailbox interface
MbReadSLong(*id*)	Read a signed longword from the application via the mailbox interface
MbReadULong(*id*)	Read an unsigned longword from the application via the mailbox interface
MbReadString(*id*)	Read a null-terminated string from the application via the mailbox interface
MbReadFloat(*id*)	Read a floating point (float) value from the application via the mailbox interface

scanf

Note: This function cannot be used in email messages.

This SSI function reads a string passed from an object in a HTML form, interprets the string according to the specification in format, and stores the result in the OUT area according to the passed arguments. The formatting of the string is equal to the standard C function call scanf()

Syntax:

```
<?--#exec cmd_argument='scanf( "ObjName", "format", Arg1, ..., ArgN), ErrVal1,
..., ErrvalN'-->
```

ObjName	- The name of the object with the passed data string
format	- Specifies how the passed string shall be formatted
Arg1 - ArgN	- Specifies where to write the data
ErrVal1 -ErrValN	- Optional; specifies the value/string to write in case of an error.

Character	Input, Argument Type
d	Decimal number; byte, short
i	Number, byte, short. The number may be in octal (leading 0(zero)) or hexadecimal (leading 0x or 0X)
o	Octal number (with or without leading zero); byte, short
u	Unsinged decimal number; unsigned byte, unsigned short
x	Hexadecimal number (with or without leading 0x or 0X); byte, short
c	Characters; char*. The next input characters (default 1) are placed at the indicated spot. The normal skip over white space is suppressed; to read the next non-white space character, use %1s.
s	Character string (not quoted); char*, pointing to an array of characters large enough for the string and a terminating "\0" that will be added.
e, f, g	Floating-point number with optional sign, optional decimal point and optional exponent; float*
%	Liteal %; no assignment is made.

The conversion characters d, i, o, u and x may be preceded by l (letter ell) to indicate that a pointer to 'long' appears in the argument list rather than a 'byte' or a 'short'

The arguments that can be passed to the SSI function scanf are:

Argument	Description
OutWriteByte(*offset*)	Write a byte to position *offset* in the OUT area
OutWriteWord(*offset*)	Write a word to position *offset* in the OUT area
OutWriteLong(*offset*)	Write a long to position *offset* in the OUT area
OutWriteString(*offset*)	Write a string to position *offset* in the OUT area
OutWriteFloat(offset)	Write a floating point value to position *offset* in the OUT area
MbWriteByte(*id*)	Write a byte to the application via the mailbox interface
MbWriteWord(*id*)	Write a word to the application via the mailbox interface
MbWriteLong(*id*)	Write a longword to the application via the mailbox interface
MbWriteString(*id*)	Write a string to the application via the mailbox interface
MbWriteFloat(*id*)	Write a floating point value to the application via the mailbox interface

Default output:

```
Write succeeded
Write failed
```

IncludeFile

This SSI function includes the contents of a file on a web page.

Syntax:

```
<?--#exec cmd_argument='IncludeFile( "File name" )'-->
```

Default output:

```
Success       - <File content>
Failure       - Failed to open <filename>
```

SaveToFile

Note: This function cannot be used in email messages.

This SSI function saves the contents of a passed form to a file. The passed name/value pair will be written to the file "File name" separated by the "Separator" string. The [Append|Overwrite] parameter determines if the specified file shall be overwritten, or if the data in the file shall be appended.

Syntax:

```
<?--#exec cmd_argument='SaveToFile( "File name", "Separator",[Append|Over-
write] )'-->
```

Default output:

```
Success       - Form saved to file
Failure       - Failed to save form
```

SaveDataToFile

Note: This function cannot be used in email messages.

This SSI function saves the data of a passed form to a file. The "Object name" parameter is optional, if specified, only the data from that object will be stored. If not, the data from all objects in the form will be stored.

The [Append|Overwrite] parameter determines if the specified file shall be overwritten, or if the data in the file shall be appended.

> *Syntax:*
>
> ```
> <?--#exec cmd_argument='SaveDataToFile( "File name", "Object name",[Ap-
> pend|Overwrite] )'-->
> ```
>
> *Default output:*
>
> ```
> Success        - Form saved to file
> Failure        - Failed to save form
> ```

DisplayRemoteUser

Note: This function cannot be used in email messages.

This SSI function returns the user name on an authentication session.

> *Syntax:*
>
> ```
> <?--#exec cmd_argument='DisplayRemoteUser'-->
> ```

Changing SSI output

There is two methods of changing the output strings from SSI functions:

1. Changing SSI output defaults by creating a file called "\ssi_str.cfg" containing the output strings for all SSI functions in the system

2. Temporary changing the SSI output by calling the SSI function "SsiOutput()".

SSI Output String File

If the file "\ssi_str.cfg" is found in the filesystem and the file is correctly according to the specification below, the SSI functions will use the output strings specified in this file instead of the default strings.

The files shall have the following format:

```
[StoreEtnConfig]
Success: "String to use on success"
Invalid IP: "String to use when the IP address is invalid"
Invalid Subnet: "String to use when the Subnet mask is invalid"
Invalid Gateway: "String to use when the Gateway address is invalid"
Invalid Email server: "String to use when the SMTP address is invalid"
Invalid IP or Subnet: "String to use when the IP address and Subnet mask does
not match"
Invalid DNS1: "String to use when the primary DNS cannot be found"
Invalid DNS2: "String to use when the secondary DNS cannot be found"
Save Error: "String to use when storage fails"
Invalid DHCP state: "String to use when the DHCP state is invalid"

[scanf]
Success: "String to use on success"
Failure: "String to use on failure"

[IncludeFile]
Failure: "String to use when failure"1

[SaveToFile]
Success: "String to use on success"
Failure: "String to use on failure"1

[SaveDataToFile]
Success: "String to use on success"
Failure: "String to use on failure"1

[GetText]
Success: "String to use on success"
Failure: "String to use on failure"
```

The contents of this file can be redirected by placing the line '[File path]' on the first row, and a file path on the second.

Example:

```
[File path]
\user\ssi_strings.cfg
```

In this example, the settings described above will be loaded from the file 'user\ssi_strings.cfg'.

---

1. '%s' includes the filename in the string

Temporary SSI Output change

The SSI output for the next called SSI function can be changed with the SSI function "SsiOutput()" The next called SSI function will use the output according to this call. Thereafter the SSI functions will use the default outputs or the outputs defined in the file '\ssi_str.cfg'. The maximum size of a string is 128 bytes.

Syntax:

```
<?--#exec cmd_argument='SsiOutput( "Success string", "Failure string" )'-->
```

Example:

This example shows how to change the output strings for a scanf SSI call.

```
<?--#exec cmd_argument='SsiOutput ( "Parameter1 updated", "Error" )'-->
<?--#exec cmd_argument="scanf( "Parameter1", "%d", OutWriteByte(0) )'-->
```

Fieldbus Specific Mailbox Commands

Fault Information

When a mailbox command cannot be processed, the Message Information register in the header of the response will indicate that an error occurred (Consult the Anybus-S Parallel Design Guide for more in-

formation). If the error code is 'Invalid Other' (Fh), extended error information is available in the Fault Information register (Extended word 8).

The fault codes in the Fault Information register are:

Register Value	Description
0001h	Invalid IP-address or Subnet mask
0002h	Invalid socket type
0003h	No free socket
0004h	Invalid socket
0005h	Not connected
0006h	Command failed
0007h	Invalid data size
0008h	Invalid fragment type
0009h	Fragment error
000Ah	Invalid timeout time
000Bh	Can't send more
000Ch	Failed to open file or file not found
000Dh	Invalid file descriptor
000Eh	Invalid open method
000Fh	No email server configured
0010h	Command aborted
0011h	Too many registered objects
0012h	Object already registered
0013h	Deregistering invalid object
0015h	Unsupported Command
0016h	Failed to send UCMM command
0017h	No timeout
0018h	Invalid port number
0019h	Duplicate port number
001Ah	EPATH too big
001Bh	Mapping Failed
001Ch	Reset notification unsupported
001Dh	Too many open files
001Eh	Failed to create directory
001Fh	Failed to delete directory
0020h	Failed to rename file
0021h	Failed to move file
0022h	Failed to copy file
0023h	Too many open directories
0024h	Failed to open directory or directory not found
0025h	Failed to resolve hostname with DNS
0026h	Timed out resolving hostname with DNS

General Configuration Commands

Commands in this category:

Mailbox Commands	Description	Page
Set Ethernet Configuration (SET_ET-N_CONFIG)	Set network settings	9-4
Read Ethernet Configuration (READ_ETN_CONFIG)	Retrieve the currently used network settings	9-5
Get MAC Address (GET_MAC_ADDR)	Retrieve the ethernet MAC address from the module	9-6
Connection Timeout (CONNECT_-TIMEOUT)	Set the connection timeout value	9-7
Set Host and Domain (SET_HOST_DOMAIN)	Set host and default domain names	9-8
Get Host and Domain (GET_HOST_DOMAIN)	Returns the currently used host and domain	9-9
Disable / Enable ARP Config (ARP_CFG_CONTROL)	Disable/enable ARP support	9-10
Disable / Enable HICP (HICP_CFG_-CONTROL)	Disable/enable HICP support	9-11
Disable DHCP (DISABLE_DHCP)	Disable DHCP support	9-12
Set DNS Server Settings (SET_DNS_SERVERS)	Configures the DNS server settings	9-13
Get DNS Server Settings (GET_DNS_SERVERS)	Returns the currently used DNS server settings	9-14
Set SMTP Server (SET_SMTP_-SERVER)	Configures the SMTP server IP address	9-15
Set SMTP Server by Name (SET_SMTP_SERVER_NAME)	Configures the SMTP server address	9-16
Get SMTP Server (GET_SMTP_-SERVER)	Returns the currently used SMTP server IP address	9-17
Disable Web Server (DISA-BLE_WEB_SERVER)	This command disables the built in web server	9-18
Enable Web Server (ENA-BLE_WEB_SERVER)	This command enables the built in web server	9-19
Disable FTP server (DISABLE_FT-P_SERVER)	This command disables the built in FTP server	9-20
Disable Telnet Server (DISA-BLE_TELNET_SERVER)	This command disables the built in telnet server	9-21
Global Admin Mode (GLOBAL_AD-MIN_MODE)	This command instruct the module to run in global admin mode	9-22
Disable Virtual File System (DISA-BLE_VFS)	Disable the virtual file system	9-23
Set Serial Number (SET_SERIAL_-NUMBER)	Set the serial number of the module	9-24
Set HTTP Server Name (SET_HTTP_SERVER)	Set the web server name reported in the http header	9-25
Set SMTP Port Number (SET_SMT-P_PORT)	Change the port number used for SMTP	9-26

Set Ethernet Configuration (SET_ETN_CONFIG)

Description

This command can be used to set the IP address, Gateway address, and Subnet mask. The settings in this command overrides the settings in 'ethcfg.cfg'.

Note: This command may only be issued during initialisation.

Initiated by	Application
Command no.	0001h
Extended Header	Fault information
Message data	Network settings.
Response data	(the response holds a copy of the command data)

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0001h	0001h	*SET_ETN_CONFIG*
Data size	000Ch	000Ch	*12 bytes of data (6 words)*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message dataword 1	IP address (high)	IP address (high)	
Message dataword 2	IP address (low)	IP address (low)	
Message dataword 3	Subnet mask (high)	Subnet mask (high)	
Message dataword 4	Subnet mask (low)	Subnet mask (low)	
Message dataword 5	Gateway address (high)	Gateway address (high)	
Message dataword 6	Gateway address (low)	Gateway address (low)	

Read Ethernet Configuration (READ_ETN_CONFIG)

Description

This command returns the currently used network settings.

Note: This command may only be issued during runtime.

Initiated by	Application
Command no.	0002h
Extended Header	-
Message data	-
Response data	Currently used network settings.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0002h	0002h	*READ_ETN_CONFIG*
Data size	0000h	000Ch	*12 bytes of data (6 words)*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	-	
		IP address (high)	**Response dataword 1**
		IP address (low)	**Response dataword 2**
		Subnet mask (high)	**Response dataword 3**
		Subnet mask (low)	**Response dataword 4**
		Gateway address (high)	**Response dataword 5**
		Gateway address (low)	**Response dataword 6**

Get MAC Address (GET_MAC_ADDR)

Description

This command returns the MAC address of the module.

Initiated by	Application
Command no.	0010h
Extended Header	-
Message data	-
Response data	MAC Address, 6 bytes

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0010h	0010h	*GET_MAC_ADDR*
Data size	0000h	0006h	*6 bytes of data (3 words)*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	-	
		MAC Address (high)	**Response dataword 1**
		MAC Address (mid)	**Response dataword 2**
		MAC Address (low)	**Response dataword 3**

Connection Timeout (CONNECT_TIMEOUT)

Description

This mailbox command configures the timeout value used when trying to establish a connection to a host.

Note: This command may only be issued during initialization.

Initiated by	Application
Command no.	0003h
Extended Header	Fault information
Message data	Timeout Value
Response data	(the response holds a copy of the command data)

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0003h	0003h	*CONNECT_TIMEOUT*
Data size	0001h	0001h	*1 data byte*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message databyte 1	Timeout Value	Timeout Value	

- **Timeout Value**

  Valid timeout values range from 1 to 255 seconds (Default = 75).

Set Host and Domain (SET_HOST_DOMAIN)

Description

This command is used to set the host and the default domain name.

Max host name length is 64 charachters and max domain dame length is 48 characters.

Note: This command may only be issued during initialisation.

Initiated by	Application
Command no.	0032h
Extended Header	Fault information
Message data	Host and Domain settings
Response data	(the response holds a copy of the command data)

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0032h	0032h	*SET_HOST_DOMAIN*
Data size	(size)	(size)	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message data	Host name (string, null-terminated)	Host name (string, null-terminated)	*Max 64 characters*
	Domain name (string, null-terminated)	Domain name (string, null-terminated)	*Max 48 characters*

Get Host and Domain (GET_HOST_DOMAIN)

Description

This command returns the configured host and default domain name settings.

Initiated by	Application
Command no.	0034h
Extended Header	-
Message data	-
Response data	Currently used Host and Domain settings

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0034h	0034h	*GET_HOST_DOMAIN*
Data size	0000h	(size)	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	-	
		Host name (string, null-terminated)	Response data
		Domain name (string, null-terminated)	

Disable / Enable ARP Config (ARP_CFG_CONTROL)

Description

This command is used to enable / disable the ARP functionality.

Initiated by	Application
Command no.	0014h
Extended Header	Fault information
Message data	ARP state
Response data	(the response holds a copy of the command data)

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0014h	0014h	*ARP_CFG_CONTROL*
Data size	0001h	0001h	*1 data byte*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message databyte	Config Value	Config Value	

- **Config Value**

00h:	Disable
01h:	Enable

Disable / Enable HICP (HICP_CFG_CONTROL)

Description

This command is used to enable / disable support for HICP (Anybus IP Config).

Initiated by	Application
Command no.	0013h
Extended Header	-
Message data	HICP state
Response data	(the response holds a copy of the command data)

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0013h	0013h	*HICP_CFG_CONTROL*
Data size	0001h	0001h	*1 data byte*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message databyte	Config Value	Config Value	

- **Config Value**

00h:	Disable
01h:	Enable

Disable DHCP (DISABLE_DHCP)

Description

This command disables support for DHCP, and overrides the DHCP setting in the configuration file 'ethcfg.cfg'. Furthermore, the DHCP configuration checkbox on the default web page will be excluded, and the Anybus module will not report its DHCP status via HICP.

Note: This command may only be issued during initialisation.

Initiated by	Application
Command no.	0015h
Extended Header	-
Message data	-
Response data	The response indicates if the command was accepted.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0015h	0015h	*DISABLE_DHCP*
Data size	0000h	0000h	*1 data byte*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	

Set DNS Server Settings (SET_DNS_SERVERS)

Description

This command configures the DNS server settings.

Note: This command may only be issued to during initialisation.

Initiated by	Application
Command no.	0031h
Extended Header	-
Message data	DNS server settings
Response data	The response indicates if the command was accepted.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0031h	0031h	*SET_DNS_SERVERS*
Data size	0008h	0008h	*8 bytes (4 words) of data*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message data word 1	Primary DNS (msb)	Primary DNS (msb)	
Message data word 2	Primary DNS (lsb)	Primary DNS (lsb)	
Message data word 3	Secondary DNS (msb)	Secondary DNS (msb)	
Message data word 4	Secondary DNS (lsb)	Secondary DNS (lsb)	

- **Primary DNS**

    IP address to primary DNS server.

- **Secondary DNS**

    IP address to secondary DNS server, or 0.0.0.0 when using primary DNS only.

Get DNS Server Settings (GET_DNS_SERVERS)

Description

This command returns the currently used DNS server settings.

Initiated by	Application
Command no.	0033h
Extended Header	-
Message data	-
Response data	DNS server settings.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0033h	0033h	*GET_DNS_SERVERS*
Data size	0000h	0008h	*(size of data)*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
		Primary DNS (msb)	
		Primary DNS (lsb)	
		Secondary DNS (msb)	
		Secondary DNS (lsb)	

- **Primary DNS**

  IP address to primary DNS server, or 0.0.0.0 if unused.

- **Secondary DNS**

  IP address to secondary DNS server, or 0.0.0.0 if unused.

Set SMTP Server (SET_SMTP_SERVER)

Description

This mailbox command may be used to specify the IP address of the SMTP server.

Note 1: This command overrides the settings stored in the configuration file 'ethcfg.cfg'.

Note 2: This command may only be issued during initialisation.

Initiated by	Application
Command no.	000Eh
Extended Header	Fault information
Message data	SMTP server address
Response data	(the response holds a copy of the command data)

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	000Eh	000Eh	*SET_SMTP_SERVER*
Data size	0004h	0004h	*4 bytes of data (2 words)*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message dataword 1	SMTP IP address (high)	SMTP IP address (high)	
Message dataword 2	SMTP IP address (low)	SMTP IP address (low)	

Set SMTP Server by Name (SET_SMTP_SERVER_NAME)

Description

This command may be used to specify the address to the SMTP server in ASCII form.

Note 1: This command overrides the settings stored in the configuration file 'ethcfg.cfg'.

Note 2: This command may only be issued during initialisation.

Initiated by	Application
Command no.	0016h
Extended Header	-
Message data	SMTP server (string)
Response data	(the response holds a copy of the command data)

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0016h	0016h	*SET_SMTP_SERVER_NAME*
Data size	(size)	(size)	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message Data	SMTP Server (String, null-terminated)	SMTP Server (String, null-terminated)	

- **SMTP Server**

  ASCII string, null terminated (e.g. "192.168.1.42" or "smtp.server.com")

Get SMTP Server (GET_SMTP_SERVER)

Description

This function returns the currently used SMTP server IP address.

Initiated by	Application
Command no.	000Fh
Extended Header	-
Message data	-
Response data	SMTP server IP address

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	000Fh	000Fh	*GET_SMTP_SERVER*
Data size	0000h	0004h	*4 bytes of data (2 words)*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
		SMTP IP address (high)	**Response dataword 1**
		SMTP IP address (low)	**Response dataword 2**

Disable Web Server (DISABLE_WEB_SERVER)

Description

This command disables the onboard web server. The web server is enabled by default.

Initiated by	Application
Command no.	0004h
Extended Header	-
Message data	-
Response data	-

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0004h	0004h	*DISABLE_WEB_SERVER*
Data size	0000h	0000h	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	-	

Enable Web Server (ENABLE_WEB_SERVER)

Description

This command enables the onboard web server. The web server is enabled by default.

Initiated by	Application
Command no.	0005h
Extended Header	-
Message data	-
Response data	-

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0005h	0005h	*ENABLE_WEB_SERVER*
Data size	0000h	0000h	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	-	

Disable FTP server (DISABLE_FTP_SERVER)

Description

This command disables the FTP server.

Initiated by	Application
Command no.	0006h
Extended Header	-
Message data	-
Response data	-

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0006h	0006h	*DISABLE_FTP_SERVER*
Data size	0000h	0000h	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	-	

Disable Telnet Server (DISABLE_TELNET_SERVER)

Description

This command disables the onboard Telnet server.

Initiated by	Application
Command no.	0008h
Extended Header	-
Message data	-
Response data	-

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0008h	0008h	*DISABLE_TELNET_SERVER*
Data size	0000h	0000h	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	-	

Global Admin Mode (GLOBAL_ADMIN_MODE)

Description

This command instructs the module to run in Global Admin Mode. For more information, see 3-2 "Security Framework".

Note: This command may only be issued during initialization.

Initiated by	Application
Command no.	000Bh
Extended Header	-
Message data	-
Response data	-

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	000Bh	000Bh	*GLOBAL_ADMIN_MODE*
Data size	0000h	0000h	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	-	

Disable Virtual File System (DISABLE_VFS)

Description

This command disables the virtual files in the file system.

Note: This command may only be issued during initialization.

Initiated by	Application
Command no.	0011h
Extended Header	-
Message data	-
Response data	-

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0011h	0011h	*DISABLE_VFS*
Data size	0000h	0000h	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	-	

Set Serial Number (SET_SERIAL_NUMBER)

Description

This command is used to change the serial number of the module.

Initiated by	Application
Command no.	00F0h
Extended Header	-
Message data	Serial number
Response data	(the response holds a copy of the command data)

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	00F0h	00F0h	*SET_SERIAL_NUMBER*
Data size	0004h	0004h	*4 bytes of data*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	-	
Message data byte 1	Serial number	Serial number	*Serial number, high byte*
Message data byte 2	Serial number	Serial number	
Message data byte 3	Serial number	Serial number	
Message data byte 4	Serial number	Serial number	*Serial number, low byte*

Set HTTP Server Name (SET_HTTP_SERVER)

Description

This command is used to set the web server name that is reported in the http header.

Initiated by	Application
Command no.	00F1h
Extended Header	-
Message data	Web server name to be used in thehttp header, max 64 bytes including line breaks and NULL characters
Response data	(the response holds a copy of the command data)

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	00F1h	00F1h	*SET_HTTP_SERVER*
Data size	(size)	(size)	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	-	
Message data	Web server name (string, null-terminated)	Web server name (string, null-terminated)	*Max 64 characters*

Set SMTP Port Number (SET_SMTP_PORT)

Description

This command is used to set the destination TCP port number for an SMTP connection.

Note: This command may only be issued during initialization.

Initiated by	Application
Command no.	00F2h
Extended Header	-
Message data	The port number to be used.
Response data	(the response holds a copy of the command data)

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	00F2h	00F2h	*SET_SMTP_PORT*
Data size	(size)	(size)	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	-	
Message data byte 1	SMTP Port	SMTP Port	*Port number, high byte*
Message data byte 2	SMTP Port	SMTP Port	*Port number, low byte*

Mailbox File System Interface

The filesystem is available to the application through the mailbox interface. Note that the application always has unrestricted access to the filesystem, regardless of security mode.

Commands in this category:

Mailbox Command	Description	Page
Open File (FILE_OPEN)	Open a file for reading, writing, or appending.	9-28
Close File (FILE_CLOSE)	Close a file previously opened using FILE_OPEN	9-29
Read File (FILE_READ)	Read data from a file	9-30
Write File (FILE_WRITE)	Write data to a file.	9-31
Delete File (FILE_DELETE)	Delete a file	9-32
Move File (FILE_MOVE)	Moves a file	9-32
Rename File (FILE_RENAME)	Rename a file	9-33
Copy File (FILE_COPY)	Copy a file	9-35
Create Directory (DIR_CREATE)	Create a new directory	9-35
Delete Directory (DIR_DELETE)	Delete an empty directory	9-36
Open Directory (DIR_OPEN)	Open a directory	9-37
Read Directory (DIR_READ)	Read contents of a directory previously opened using DIR_OPEN	9-38
Close Directory (DIR_CLOSE)	Close a directory previously opened using DIR_OPEN	9-40
Create RAM disc (CREATE_RAM_-DISC)	Mounts the RAM disc into a specified directory in the file system.	9-40
Format file system (FORMAT_FS)	Formats the filesystem.	9-41
File system CRC (CRC_FS)	Calculate and return the CRC for the Used Sector Field and the File System	9-43

Open File (FILE_OPEN)

Description

This command opens a file for reading, writing, or appending. If the specified file does not exist, it will be created.

Initiated by	Application
Command no.	0060h
Extended Header	Mode, Filesize & Fault information
Message data	Name and path to the file to open (NULL terminated)
Response data	File Handle

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0060h	0060h	*FILE_OPEN*
Data size	(size)	0004h	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	Mode	Filesize (high)	
Extended word 2	-	Filesize (low)	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message data	Path + filename (String, null-terminated)	File Handle (high)	Response data word 1
		File Handle (low)	Response data word 2

- **Mode**

Value	Mode
0000h	Open a file in read mode
0001h	Open a file in write mode. If the specified file does not exist, it will be created. If the specified file already exists, it will be overwritten.
0002h	Open a file in append mode. If the specified file does not exist, it will be created. If the specified file exists, any data written to the file will be appended at end-of-file.

- **Filesize**

  Current filesize (if applicable).

- **File Handle**

  Unique identifier which must be used on all further operations associated with the file.

Close File (FILE_CLOSE)

Description

This command closes a file previously opened using FILE_OPEN.

Initiated by	Application
Command no.	0061h
Extended Header	File Handle, Filesize & Fault information.
Message data	-
Response data	-

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0061h	0061h	*FILE_CLOSE*
Data size	0000h	0000h	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	File Handle (high)	File Handle (high)	
Extended word 2	File Handle (low)	File Handle (low)	
Extended word 3	-	Filesize (high)	
Extended word 4	-	Filesize (low)	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	

- **File Handle**

  Handle of the file to close. See also 9-28 "File Handle".

- **Filesize**

  Size of the file.

Read File (FILE_READ)

Description

This command reads data from a file previously opened in read mode using FILE_OPEN.

Initiated by	Application
Command no.	0062h
Extended Header	File Handle, no. of bytes to read & Fault information
Message data	-
Response data	The read data is returned in the response data field.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0062h	0062h	*FILE_READ*
Data size	0000h	(size)	*Bytes read*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	File Handle (high)	File Handle (high)	
Extended word 2	File Handle (low)	File Handle (low)	
Extended word 3	No. of bytes	No. of bytes	*Maximum 256 bytes.*
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
		Data	**Response data**

- **File Handle**

  File handle of the file to read data from. See also 9-28 "File Handle".

- **No. of bytes**

  Number of bytes to read minus 1 (i.e. a value of 42 will read 43 bytes).

- **Data**

  The actual data read from the file (if applicable).

Write File (FILE_WRITE)

Description

This mailbox command writes data to a file previously opened in write or append mode using FILE_OPEN.

Initiated by	Application
Command no.	0063h
Extended Header	File Handle & Fault information
Message data	Data to write
Response data	A 'Data size' value of 0 (zero) indicates that the command was unsuccessful, possibly due to a faulty handle, or that the module has run out of storage.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0063h	0063h	*FILE_WRITE*
Data size	(number of bytes to write)	(number of written bytes)	*Max. 256 bytes*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	File Handle (high)	File Handle (high)	
Extended word 2	File Handle (low)	File Handle (low)	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message data	Data	Data	

- **File Handle**

  File handle of the file to write data to. See also 9-28 "File Handle".

- **No. of bytes**

  Number of bytes to write minus 1 (i.e. a value of 42 will read 43 bytes).

- **Data**

  The actual data that shall be written.

Delete File (FILE_DELETE)

Description

This mailbox command deletes a file from the file system.

Initiated by	Application
Command no.	0064h
Extended Header	Fault information
Message data	Name and path to the file to delete (NULL terminated)
Response data	The response data is a copy of the command data.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0064h	0064h	*FILE_DELETE*
Data size	(size)	(size)	*Maximum 256 bytes*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message data	Path + filename (String, null-terminated)	Path + filename (String, null-terminated)	

Move File (FILE_MOVE)

Description

This command renames a file in the filesystem.

Note: Although the filesystem supports path lengths of up to 256 characters, the total length of the source and destination paths summed together must be less than 256 characters when using this command due to limitations in the mailbox command structure.

Initiated by	Application
Command no.	0065h
Extended Header	Fault information
Message data	Name + Path of source and destination, both NULL terminated
Response data	The response data is a copy of the command data.

Command and response layout

	Command		Expected response	
Message ID	(ID)		(ID)	
Message information	4002h		0002h	*Fieldbus Specific Message*
Command	0065h		0065h	*FILE_MOVE*
Data size	(size)		(size)	*Size of path strings*
Frame count	0001h		0001h	
Frame number	0001h		0001h	
Offset high	0000h		0000h	
Offset low	0000h		0000h	
Extended word 1	-		-	
Extended word 2	-		-	
Extended word 3	-		-	
Extended word 4	-		-	
Extended word 5	-		-	
Extended word 6	-		-	
Extended word 7	-		-	
Extended word 8	-		Fault information	
Message data	Source: Path + filename (String, null-terminated)		Source: Path + filename (String, null-terminated)	
	Destination: Path + filename (String, null-terminated)		Destination: Path + filename (String, null-terminated)	

Rename File (FILE_RENAME)

Description

This command renames a file in the filesystem.

Note: Although the filesystem supports path lengths of up to 256 characters, the total length of the two pathnames summed together must be less than 256 characters when using this command due to limitations in the mailbox command structure.

Initiated by	Application
Command no.	0066h
Extended Header	Fault information
Message data	Name + Path of source and destination, both NULL terminated
Response data	The response data is a copy of the command data.

Command and response layout

	Command		Expected response	
Message ID	(ID)		(ID)	
Message information	4002h		0002h	*Fieldbus Specific Message*
Command	0066h		0066h	*FILE_RENAME*
Data size	(size)		(size)	*Size of path strings*
Frame count	0001h		0001h	
Frame number	0001h		0001h	
Offset high	0000h		0000h	
Offset low	0000h		0000h	
Extended word 1	-		-	

Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message data	Old: Path + filename (String, null-terminated)	Old: Path + filename (String, null-terminated)	
	New: Path + filename (String, null-terminated)	New: Path + filename (String, null-terminated)	

Copy File (FILE_COPY)

Description

This command copies a file in the filesystem to a specified location.

Note: Although the filesystem supports path lengths of up to 256 characters, the total length of the source and destination paths summed together must be less than 256 characters when using this command due to limitations in the mailbox command structure.

Initiated by	Application
Command no.	0067h
Extended Header	Fault information
Message data	Name + Path of source and destination, both NULL terminated
Response data	The response data is a copy of the command data.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0067h	0067h	*FILE_COPY*
Data size	(size)	(size)	*Size of path strings*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message data	Source: Path + filename (String, null-terminated)	Source: Path + filename (String, null-terminated)	
	Destination: Path + filename (String, null-terminated)	Destination: Path + filename (String, null-terminated)	

Create Directory (DIR_CREATE)

Description

This command creates a directory in the file system.

Initiated by	Application
Command no.	0068h
Extended Header	Fault information
Message data	Path and name of the new directory, null terminated.
Response data	The response data is a copy of the command data.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0068h	0068h	*DIR_CREATE*
Data size	(size)	(size)	*Size of path string*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message data	Path + name (String, null-terminated)	Path + name (String, null-terminated)	

Delete Directory (DIR_DELETE)

Description

This command deletes an empty directory from the file system.

Initiated by	Application
Command no.	0069h
Extended Header	-
Message data	Path and name of the directory, null terminated.
Response data	The response data is a copy of the command data.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0069h	0069h	*DIR_DELETE*
Data size	(size)	(size)	*Size of path string*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	

Extended word 7	-		-
Extended word 8	-		Fault information
Message data	Path + name (String, null-terminated)		Path + name (String, null-terminated)

Open Directory (DIR_OPEN)

Description

This command opens a directory and returns a descriptor that should be used on all further operations on the directory.

See also 9-39 "Reading the Contents of a Directory".

Initiated by	Application
Command no.	006Ah
Extended Header	-
Message data	Path and name of the directory, null terminated.
Response data	Directory handle & Fault information

Command and response layout

	Command		Expected response	
Message ID	(ID)		(ID)	
Message information	4002h		0002h	*Fieldbus Specific Message*
Command	006Ah		006Ah	*DIR_OPEN*
Data size	(size)		0004h	
Frame count	0001h		0001h	
Frame number	0001h		0001h	
Offset high	0000h		0000h	
Offset low	0000h		0000h	
Extended word 1	-		-	
Extended word 2	-		-	
Extended word 3	-		-	
Extended word 4	-		-	
Extended word 5	-		-	
Extended word 6	-		-	
Extended word 7	-		-	
Extended word 8	-		Fault information	
Message data	Path + name (String, null-terminated)		Directory Handle (high)	**Response data word 1**
			Directory Handle (low)	**Response data word 2**

- **Directory Handle**

    Unique identifier which must be used on all further operations associated with the directory.

Read Directory (DIR_READ)

Description

This command reads the contents of a directory previously opened using DIR_OPEN. This must be repeated until the response to the command is empty (i.e. until the response data size equals zero).

See also 9-39 "Reading the Contents of a Directory".

Initiated by	Application
Command no.	006Bh
Extended Header	Directory Handle & Fault information
Message data	-
Response data	Details about one object in the directory.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	006Bh	006Bh	*DIR_READ*
Data size	0000h	(size)	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	Directory Handle (high)	Directory Handle (high)	*(See DIR_OPEN)*
Extended word 2	Directory Handle (low)	Directory Handle (low)	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
		Object Size (long)	**Response data byte 1**
			Response data byte 2
			Response data byte 3
			Response data byte 4
		Object Flags	**Response data byte 5**
		Object Name (string, null-terminated)	**Response data...**

- **Directory Handle**

  Unique identifier which must be used on all further operations associated with the directory.

- **Object Size**

  Size of object (i.e. filesize).

- **Object Flags**

  Various flags specifying the nature of the object:

b7	b6	b5	b4	b3	b2	b1	b0
(reserved)				SYS	H	RO	DIR

Bit	Description	
DIR	Directory flag	0: Object is a file 1: Object is a directory
RO	Read only	0: Object can be read or written 1: Object is read-only
H	Hidden	0: Object is visible 1: Object is hidden
SYS	System	0: User object 1: System object

- **Object Name**

  Name of object, null-terminated (e.g. filename or directory name).

Reading the Contents of a Directory

The following flowchart illustrates the process of reading the contents of a directory:
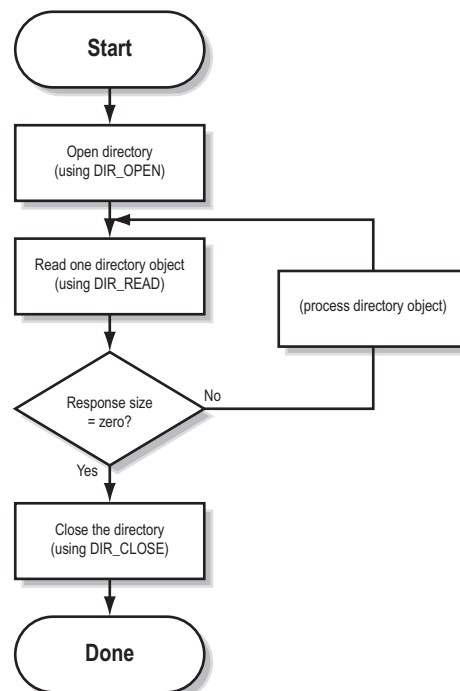
Close Directory (DIR_CLOSE)

Description

This command closes a directory previously opened using DIR_OPEN.

See also 9-39 "Reading the Contents of a Directory".

Initiated by	Application
Command no.	006Ch
Extended Header	Directory Handle & Fault information
Message data	-
Response data	-

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	006Ch	006Ch	*DIR_CLOSE*
Data size	0000h	0000h	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	Directory Handle (high)	Directory Handle (high)	*(See DIR_OPEN)*
Extended word 2	Directory Handle (low)	Directory Handle (low)	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	

Create RAM disc (CREATE_RAM_DISC)

Description

This command mounts the RAM disc into a specified directory in the file system.

- The directory must be empty

- If the directory doesn't exist, it will be created

Initiated by	Application
Command no.	0018h
Extended Header	-
Message data	String containing RAM disc location, null terminated
Response data	The response data is a copy of the command data

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0018h	0018h	*FORMAT_FS*
Data size	(size)	(size)	*Size of data in bytes*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	-	
Message data	Path (String, null terminated)	Path (String, null terminated)	

Format file system (FORMAT_FS)

Description

This command formats the file system.

- The directory must be empty

• If the directory doesn't exist it will be created.

Initiated by	Application
Command no.	006Dh
Extended Header	-
Message data	-
Response data	The response indicates if the command was accepted

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	006Dh	006Dh	*FORMAT_FS*
Data size	(size)	(size)	*Size of data in bytes*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	-	

File system Checksum (CRC_FS)

Description

This command calculates the checksum for the used sector field and the file system.

Initiated by	Application
Command no.	006Eh
Extended Header	-
Message data	String containing RAM disc location, null terminated
Response data	The response indicates if the command was accepted

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	006Eh	006Eh	*CRC_FS*
Data size	0001h	0004h	*Size of data in bytes*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	-	
Message data 1	Disc ID	Used sector CRC	
Message data 2		File system CRC	

Modbus/TCP Specific Commands

Commands in this category:

Mailbox Commands	Description	Page
Modbus Connection Timeout (MB_CONN_TIMEOUT)	Set the timeout value for Modbus/TCP connections	9-44
Disable Modbus/TCP server (DISA-BLE_MB_TCP)	Disables the Modbus/TCP server	9-45
Modbus ON/OFF Line Configuration (MB_ON_OFF_LINE_CONFIG)	Configures the Modbus ON/OFF line setting	9-47
Write Exception Status (WR_EX-P_STAT)	Defines the response to the Modbus command 'Read Exception Status'	9-48
Set Modbus Coil Area Size (MB_-SET_COIL_AREA_SIZE)	This command affects the Modbus addressing scheme and defines part of the I/O data as coil-based	9-48

Modbus Connection Timeout (MB_CONN_TIMEOUT)

Description

This mailbox command configures the timeout value used for Modbus/TCP connections. If an established Modbus/TCP connection does not receive a request within the specified time, the connection will be shut down by the Modbus/TCP server.

Valid timeout values are:

- 0   - No timeout is used for Modbus connections

- 10 - 65535   - Timeout value in seconds.

The default value is 60 seconds.

Note: This command may only be issued during initialization.

Initiated by	Application
Command no.	0020h
Extended Header	-
Message data	Timeout value
Response data	(the response holds a copy of the command data)

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0020h	0020h	*MB_CONN_TIMEOUT*
Data size	0002h	0002h	*2 bytes (1 word)*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	-	
Message data word 1	Timeout value	Timeout value	

Disable Modbus/TCP server (DISABLE_MB_TCP)

Description

This mailbox command disables the Modbus/TCP server.

Note: This may only be issued during initialization.

Initiated by	Application
Command no.	0021h
Extended Header	-
Message data	-
Response data	-

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0021h	0021h	*DISABLE_MB_TCP*

Data size	0000h		0000h
Frame count	0001h		0001h
Frame number	0001h		0001h
Offset high	0000h		0000h
Offset low	0000h		0000h
Extended word 1	-		-
Extended word 2	-		-
Extended word 3	-		-
Extended word 4	-		-
Extended word 5	-		-
Extended word 6	-		-
Extended word 7	-		-
Extended word 8	-		-

Modbus ON/OFF Line Configuration (MB_ON_OFF_LINE_CONFIG)

Description

This command configures the Modbus ON/OFF line setting[1]. The message data contain the Modbus function codes that will trigger the on line setting. If a trigger command arrives within the defined time-out, the module is set on line, else it is set off line.

Note: This command overrides the configuration stored in the configuration file 'onoffln.cfg'.

Initiated by	Application
Command no.	0023h
Extended Header	Time out value.
Message data	List of triggering modbus commands (up to 11 entries). Data size = 0 = trigger for all commands.
Response data	(the response holds a copy of the command data)

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0023h	0023h	*MB_ON_OFF_LINE_CONFIG*
Data size	(size)	(size)	*Up to 11 bytes*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	Timeout value	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message databyte 1	Trigger command 1	Trigger command 1	
Message databyte 2	Trigger command 2	Trigger command 2	
...	...	...	
Message databyte <=11	Trigger command <=11	Trigger command <=11	

- **Timeout Value**

  A value of 1 equals 100ms.

  A value of 0 configures the ON/OFF line functionality to be triggered by the Link Status.

- **Trigger Commands**

  Trigger Command = Modbus function code to trigger from. If zero, any Modbus command will trigger on line functionality.

---

1. Bit 0 (FBRS) in the module status register is set or reset depending on the outcome of this command. See "Anybus-S Parallel Design Guide".

Write Exception Status (WR_EXP_STAT)

Description

This command is used to define the response to the Modbus command 'Read exception status' (FC7). This means that the exception status value is set by the application, not by the Anybus module.

Initiated by	Application
Command no.	0022h
Extended Header	-
Message data	Exception status byte
Response data	(the response holds a copy of the command data)

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0022h	0022h	*WR_EXP_STAT*
Data size	0001h	0001h	*1 data byte*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	-	
Message data byte 1	Exception status	Exception status	

Set Modbus Coil Area Size (MB_SET_COIL_AREA_SIZE)

Description

This command causes the module to operate in Modbus Address Mode, and defines part of the Input/Output Areas as coil based.

See also...

• 2-2 "Modbus/TCP"

Note: This command may only be issued during initializtion.

Initiated by	Application
Command no.	0024h
Extended Header	-
Message data	Coil Size In, Coil Size Out
Response data	(the response holds a copy of the command data)

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0024h	0024h	*Set Modbus Coil Area Size*
Data size	0004h	0004h	*4 bytes of data*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	-	
Message data word 1	Coil Size In	Coil Size In	
Message data word 2	Coil Size Out	Coil Size Out	

• **Coil Size In**

   Number of bytes in the Input Area that shall be treated as coils.

• **Coil Size Out**

   Number of bytes in the Output Area that shall be treated as coils.

Mailbox Socket Interface

The Anybus module features a transparent socket interface, allowing the application to send and receive transparent data via TCP/IP or UDP/IP. The mailbox socket interface can be used in two modes:

- **Non-blocking**

    All mailbox operations on these sockets will respond directly - not block until the command is performed. Up to 16 simultaneous non-blocking sockets are supported.

    Note: Status information for all non-blocking sockets are available in the fieldbus specific area, see 10-1 "Fieldbus Specific Area".

- **Blocking**

    Blocking sockets means that the Anybus will not respond to further socket commands until the previous one has been completed (However, non-socket related commands can still be processed as normal). Up to 32 simultaneous blocking sockets are supported.

    Note: Blocking sockets do *not* have any status information in the fieldbus specific area.

Commands in this category:

Mailbox Command	Description	Page
Socket non-blocking (SOCKET_NB)	Creates a socket in non-blocking mode.	9-51
Socket blocking (SOCKET_B)	Creates a socket in blocking mode.	9-52
Listen (LISTEN)	Starts listen on a socket for incoming connections.	9-52
Accept (ACCEPT)	Accepts connections for sockets in blocking mode.	9-54
Connect (CONNECT)	Tries to connect a socket to a client.	9-55
Send (SEND)	Sends a message to a connected socket.	9-56
Receive (RECEIVE)	Receives a message form a connected socket.	9-57
Send To (SEND_TO)	Sends a message to an unconnected UDP socket to a specified host.	9-58
Receive From (RECV_FROM)	Receives a message from an unconnected UDP socket.	9-59
Close (CLOSE)	Closes a socket (and connection).	9-60
Send Fragment (SEND_FRAG)	Sends a fragment of a message with a maximum total size of 4096 bytes.	9-61
Receive Fragment (RECV_FRAG)	Receives a fragment of a message with a total maximum size of 4096 bytes.	9-63
Send Fragment To (SEND_FRAG_TO)	Sends a fragment of a message with a total maximum size of 4096 bytes to an unconnected UDP socket.	9-64
Receive Fragment From (RECV_FRAG_FROM)	Receives a fragment of a message with a total maximum size of 4096 bytes from an unconnected UDP socket.	9-66
Get Socket Option (GET_SOCKET_OPTION)	Read options from a socket.	9-68
Set Socket Option (SET_SOCKET_OPTION)	Sets options to a socket	9-69

Socket Non-Blocking (SOCKET_NB)

Description

This mailbox command creates a socket in non-blocking mode and associates it to a specific port number. If the specified port number is 0, the Anybus module selects a free port.

The response message contains a socket descriptor and the port number. The socket descriptor shall be used on all following operations on the socket.

Initiated by	Application
Command no.	0040h
Extended Header	-
Message data	The socket type (TCP or UDP) and the port number to bind the socket to.
Response data	The response indicates if the command was accepted. The response indicates which socket descriptor that is used and the port number the socket is associated to.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0040h	0040h	*SOCKET_NB*
Data size	0004h	0004h	*4 bytes of data (2 words)*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message dataword 1	Socket type	Socket descriptor	
Message dataword 2	Port number	Port number	

• **Socket Type**

Value	Socket type
0001h	TCP socket
0002h	UDP socket

Socket Blocking (SOCKET_B)

Description

This mailbox command creates a socket in blocking mode and associates it to a specific port number. If the specified port number is 0, the Anybus module selects a free port.

The response message contains a socket descriptor and the port number. This descriptor shall be used on all following operations on this socket.

Initiated by	Application
Command no.	003Fh
Extended Header	-
Message data	The socket type (TCP or UDP) and the port number to bind the socket to.
Response data	The response indicates if the command was accepted. The response indicates which socket descriptor that is used and the port number the socket is associated to.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	003Fh	003Fh	*SOCKET_B*
Data size	0004h	0004h	*4 bytes of data (2 words)*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message dataword 1	Socket type	Socket descriptor	
Message dataword 2	Port number	Port number	

- **Socket Type**

Value	Socket type
0001h	TCP socket
0002h	UDP socket

Listen (LISTEN)

Description

This mailbox command makes a socket listen for new connections. If the Anybus module detects a connection request on the specified socket, a new connected socket will be created, and the current socket

will continue listening for new connections. This means that multiple hosts can connect to one listening socket simultaneously.

Note: This command can only be used on a TCP socket.

- **Non-blocking sockets**

  Information about active connections on this socket can be read in the fieldbus specific area, see 10-1 "Memory Map" and 10-2 "Socket Status Structure".

- **Blocking sockets**

Socket descriptors for new connections connected to this socket can be received by the mailbox command ACCEPT, see 9-54 "Accept (ACCEPT)".

Initiated by	Application
Command no.	0041h
Extended Header	Socket Descriptor, Fault Information
Message data	-
Response data	-

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0041h	0041h	*LISTEN*
Data size	0000h	0000h	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	Socket descriptor	Socket descriptor	
Extended word 2	(reserved, set to 0000h)	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	

Accept (ACCEPT)

Description

When a connection request to a listening socket in blocking mode is received, this command receives the socket descriptor of the newly created connected socket.

This command is blocking and will not respond until a connection request is received.

Initiated by	Application
Command no.	0050h
Extended Header	Socket Descriptor, Fault Information, Local Port no, Host Port no, Host IP
Message data	-
Response data	New socket descriptor

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0050h	0050h	*ACCEPT*
Data size	0000h	0002h	

Frame count	0001h		0001h	
Frame number	0001h		0001h	
Offset high	0000h		0000h	
Offset low	0000h		0000h	
Extended word 1	Socket descriptor		Socket Descriptor	
Extended word 2	(reserved, set to 0000h)		Local Port No.	
Extended word 3	-		Host Port No.	
Extended word 4	-		Host IP-address word 1	
Extended word 5	-		Host IP-address word 2	
Extended word 6	-		-	
Extended word 7	-		-	
Extended word 8	-		Fault information	
			New socket descriptor	Response dataword

Connect (CONNECT)

Description

This mailbox command tries to establish a connection to a specified IP address and port number.

If the socket is of UDP type this command specifies the peer with which the socket is to be associated, the address is to which datagrams are sent and the only address from which datagrams are received.

If the socket is of TCP type this command attempts to make a connection to another socket. TCP sockets may CONNECT only once, while UDP sockets may use CONNECT multiple times to change their association.

- **Non-blocking sockets**

   If this command is correctly sent, it will be accepted regardless it's possible to establish a connection or not. The result of the operation is available in the fieldbus specific area, see 10-1 "Fieldbus Specific Area".

- **Blocking sockets**

    This command will block until a connection is established or the connection request is cancelled due to timeout or connection error.

Initiated by	Application
Command no.	0042h
Extended Header	Socket Descriptor, Fault Information, Connection Result
Message data	IP address, Port number
Response data	(the response holds a copy of the command data)

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0042h	0042h	*CONNECT*
Data size	0006h	0006h	*6 bytes of data (3 words)*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	Socket descriptor	New Socket Descriptor	
Extended word 2	(reserved, set to 0000h)	Connection result	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message data word 1	IP address (high)	IP address (high)	
Message data word 2	IP address (low)	IP address (low)	
Message data word 3	Port number	Port number	

- **Connection Result Code (Only for blocking sockets)**

Code	Status
0003h	Connected
0004h	Connection Refused
0005h	Connection Timeout
0006h	Connection Failed

Send (SEND)

Description

This mailbox command writes data to a connected socket. A maximum of 256 bytes of data can be sent using this command.

- **Non-blocking sockets**

    If there isn't enough space available for the data in the output buffers, the response will indicate that the amount of data actually sent was less than requested.

• **Blocking sockets**

If there isn't buffer space available for the data in the output buffers this command will block until there is.

Initiated by	Application
Command no.	0043h
Extended Header	Socket Descriptor, Fault Information
Message data	Data to send
Response data	(the response holds a copy of the command data)

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0043h	0043h	*SEND*
Data size	(size)	(size)	*Max. 256 bytes*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	Socket descriptor	Socket Descriptor	
Extended word 2	(reserved, set to 0000h)	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message data	Data to send	Sent data	

Receive (RECV)

Description

This mailbox command receives data from a connected socket.

If the specified socket is of TCP type this command will return the requested number of bytes from the received data stream. If the available data is less than requested, all available data will be returned.

If the specified socket is of UDP type this command will return the requested amount of data from the next received datagram. If the datagram is smaller than requested, the entire datagram will be returned in the response message. If the datagram is larger than requested, the excess bytes will be discarded.

A maximum of 256 bytes of data can be received using this command.

• **Non-blocking sockets**

If no data is available on the socket the response will indicate that 0 bytes of data was received.

- **Blocking sockets**

  If this command is called and no data is available the command will block until there is. If the response indicates that 0 bytes of data was received the connection has been closed by the host. The socket however is still valid and must be closed using the mailbox command CLOSE.

Initiated by	Application
Command no.	0044h
Extended Header	Socket Descriptor, Bytes to receive, Fault Information
Message data	-
Response data	Received data

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0044h	0044h	*RECV*
Data size	0000h	(size)	*Maximum 256 bytes*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	Socket descriptor	Socket Descriptor	
Extended word 2	Bytes to receive (in bytes)	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
		Received data	**Response data**

Send To (SEND_TO)

Description

This mailbox command sends a UDP datagram to a specified IP address and port number. A maximum of 256 bytes of data can be sent using this command. (Unconnected UDP sockets only)

Initiated by	Application
Command no.	0045h
Extended Header	Socket Descriptor, IP-address, Port number, Fault Information
Message data	Data to send
Response data	Sent data

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0045h	0045h	*SEND_TO*

Data size	(size)	(size)	*Maximum 256 bytes*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	Socket descriptor	Socket descriptor	
Extended word 2	IP-address (high)	IP-address (high)	*Destination IP address*
Extended word 3	IP-address (low)	IP-address (low)	
Extended word 4	Port number	Port number	*Port number*
Extended word 5	(reserved, set to 0000h)	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message data	Data to send	Sent data	

Receive From (RECV_FROM)

Description

This mailbox command reads the next received datagram from a UDP type socket. The response message contains the IP address and port number of the sender.

If the received datagram is smaller than requested, the entire datagram will be returned in the response message. If the received datagram is larger than requested, the excess bytes will be discarded.

A maximum of 256 bytes of data can be received using this command.

- **Non-blocking sockets**

  If the amount of data available on the socket is less than requested, this is reflected in the data size of the response.

- **Blocking sockets**

  If this command is called and no data is available the command will block until there is.

Initiated by	Application
Command no.	0045h
Extended Header	Socket Descriptor, Bytes to receive, IP-address, Port number, Fault Information
Message data	-
Response data	Received data

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0046h	0046h	*RECV_FROM*
Data size	0000h	(size)	*Maximum 256 bytes*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	Socket descriptor	Socket descriptor	
Extended word 2	Receive data size	IP address (high)	*Senders IP-address*
Extended word 3	(reserved, set to 0000h)	IP address (low)	
Extended word 4	-	Port number	*Sender port number*
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
		Received data	**Response data**

Close (CLOSE)

Description

This mailbox command causes a connected socket to shut down and release its socket descriptor.

- **Blocking sockets**

    Commands still blocking on the socket when it is closed will be aborted and return indicating 0010h (Command aborted)

Note: If a host closes a TCP connection while there is still data available to read on the socket in the client, the client socket will be indicated as connected until all data is read. In this case, if the client tries to send data the mailbox response will report "Can't send more".

Initiated by	Application
Command no.	0047h
Extended Header	Socket Descriptor, Fault Information
Message data	-
Response data	-

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0047h	0047h	*CLOSE*
Data size	0000h	0000h	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	Socket descriptor	Socket descriptor	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	

Send Fragment (SEND_FRAG)

Description

This mailbox command is used when sending messages larger than 256 bytes. Internally the fragments are stored in a buffer until the last fragment is received. The message is then sent to the socket. The maximum size of a fragmented message is 4096 bytes.

It is not possible to send multiple fragmented messages simultaneously. A fragmented message must be completely sent before another fragmented message can be sent on the same or another socket.

- **Non-blocking sockets**

    If there isn't enough space available for the data in the output buffers, the response will indicate that the amount of data actually sent was less than requested.

- **Blocking sockets**

  If there isn't buffer space available for the data in the output buffers this command will block until there is.

Initiated by	Application
Command no.	005Eh
Extended Header	Socket descriptor, Fragment Type
Message data	Data to send
Response data	Sent Data

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	005Eh	005Eh	*SEND_FRAG*
Data size	(size)	(size)	*Max. 256 bytes/fragment*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	Socket descriptor	Socket descriptor	
Extended word 2	Fragment type	Fragment type	*See below*
Extended word 3	(reserved, set to 0000h)	No. of sent bytes	*(Only in last fragment)*
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message data	Data to send	Sent data	

- **Fragment Type Value**

Value	Description
0000h	First fragment of a new message
0001h	Subsequent fragment of the message
0002h	Last fragment of the message. When this fragment is sent the entire message will be sent to the socket.

Receive Fragment (RECV_FRAG)

Description

This mailbox command is used to receive fragmented messages larger than 256 bytes from a connected socket. Internally the entire message will be read from the socket to a buffer. The fragments of the message can then be read from the buffer using this command.

If the specified socket is of TCP type this command will return the requested number of bytes from the received data stream. If the available data is less than requested, all available data will be returned.

If the specified socket is of UDP type this command will return the requested amount of data from the next received datagram. If the datagram is smaller than requested, the entire datagram will be returned in the response message. If the datagram is larger than requested, the excess bytes will be discarded.

The maximum size of a fragmented message is 4096 bytes.

- **Non-blocking sockets**

  If no data is available on the socket the response will indicate that 0 bytes of data was received.

- **Blocking sockets**

  If no data is available the command will block until there is. If the response indicates that 0 bytes of data was received the connection has been closed by the host. The socket however is still valid and must be closed using the mailbox command CLOSE.

Initiated by	Application
Command no.	005Fh
Extended Header	Socket descriptor, Fragment Type, Receive Data Size, Bytes Remaining, Fault information
Message data	-
Response data	Received Data

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	005Fh	005Fh	*RECV_FRAG*
Data size	0000h	(size)	*Max. 256 bytes/fragment*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	Socket descriptor	Socket Descriptor	
Extended word 2	Fragment type	Fragment type	*See below*
Extended word 3	Receive data size[a]	Bytes remaining	
Extended word 4	(reserved, set to 0000h)	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
		Received data	**Response data**

a. The receive data size is only used if the Fragment type = 0000h

- **Fragment Type Value**

Value	Description
0000h	Receive first fragment of a new message. This receives a new message from the network. Any unread fragments from earlier received datagrams will be overwritten.
0001h	Receive the next fragment of the message.

Send Fragment To (SEND_FRAG_TO)

Description

This mailbox command sends a UDP datagram to a specified IP address and port number. This command is used when sending a fragment of a message larger than 256 byte. Internally the fragments are

stored in a buffer until the last fragment is received. The message is then sent to the socket. The maximum size of a fragmented message is 4096 bytes.

Initiated by	Application
Command no.	005Ch
Extended Header	Socket descriptor, Fragment Type, IP-address, Port number, No. of sent bytes, Fault information
Message data	Data to send
Response data	Sent data

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	005Ch	005Ch	*SEND_FRAG_TO*
Data size	(size)	(size)	*Max. 256 bytes/fragment*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	Socket descriptor	Socket Descriptor	
Extended word 2	Fragment type	Fragment type	*See below*
Extended word 3	IP-address (high)[a]	IP-address (high)[a]	*Destination IP address*
Extended word 4	IP-address (low)[a]	IP-address (low)[a]	
Extended word 5	Port number[a]	Port number[a]	*Destination Port number*
Extended word 6	(reserved, set to 0000h)	No. of sent bytes	*(Only in last fragment)*
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message data	Data to send	Sent data	

a. IP-address and Port Number shall only be given in the first fragment.

- **Fragment Type Value**

Value	Description
0000h	First fragment of a new message.
0001h	Subsequent fragment of the message
0002h	Last fragment of the message. When this fragment is sent the entire message will be sent to the socket.

Receive Fragment From (RECV_FRAG_FROM)

Description

This mailbox command reads the next received datagram from a UDP type socket. The response message contains the IP address and port number of the sender.

This command is used to receive a fragment of a message larger than 256 bytes. The maximum total size of a fragmented message is 4096 bytes. The maximal size of each fragment is 256 bytes.

If the received datagram is smaller than requested, the entire datagram will be returned in the response message. If the received datagram is larger than requested, the excess bytes will be discarded.

For blocking sockets, the first fragment will block until there is data available on the socket.

Internally the entire message is read from the socket to a buffer. The fragments can then be read from the buffer using this command.

- **Non-blocking sockets**

  If no data is available on the socket the response will indicate that 0 bytes of data was received.

- **Blocking sockets**

  If this command is called but there is no data available on the socket the command will block and not return until there is data available.

Initiated by	Application
Command no.	005Dh
Extended Header	Socket descriptor, Fragment Type, Received data size, Bytes remaining, IP-address, port number, Fault information
Message data	-
Response data	Received data

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	005Dh	005Dh	*RECV_FRAG_FROM*
Data size	0000h	(size)	*Max. 256 bytes/fragment*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	Socket descriptor	Socket Descriptor	
Extended word 2	Fragment type	Fragment type	
Extended word 3	Receive data size	Bytes remaining	
Extended word 4	(reserved, set to 0000h)	IP-address (high)	*The senders IP address*
Extended word 5	-	IP-address (low)	
Extended word 6	-	Port number	*The senders port number*
Extended word 7	-	-	
Extended word 8	-	Fault information	
		Received data	**Response data**

- **Fragment Type Value**

Value	Description
0000h	Receive first fragment of a new message. This receives a new message from the network. Any unread fragments from earlier received datagrams will be overwritten.
0001h	Receive the next fragment of the message.

Get Socket Option (GET_SOCKET_OPTION)

Description

This command reads options from a socket.

Initiated by	Application
Command no.	0051h
Extended Header	Socket descriptor, Socket Option
Message data	-
Response data	Option Data

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0051h	0051h	*GET_SOCKET_OPTION*
Data size	0000h	Option data size	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	Socket Descriptor	Socket Descriptor	
Extended word 2	Socket Option HI	Socket Option HI	
Extended word 3	Socket Option LO	Socket Option LO	
Extended word 4			
Extended word 5			
Extended word 6			
Extended word 7			
Extended word 8			
		Option Data	Response data

Socket Options

The following options are used to Get settings from a socket:

SO_LINGER
SO_KEEPALIVE
SO_REUSEADDR
IP_MULTICAST_TTL
IP_MULTICAST_LOOP

For more information see section Socket Options page9-70.

Set Socket Option (SET_SOCKET_OPTION)

Description

This command changes the settings for a specified socket.

Initiated by	Application
Command no.	0052h
Extended Header	Socket descriptor, Socket Option
Message data	Option Data
Response data	-

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0052h	0052h	*SET_SOCKET_OPTION*
Data size	Option data size	Option data size	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	Socket Descriptor	Socket Descriptor	
Extended word 2	Socket Option HI	Socket Option HI	
Extended word 3	Socket Option LO	Socket Option LO	
Extended word 4			
Extended word 5			
Extended word 6			
Extended word 7			
Extended word 8		Fault Information	
Message data	Option Data	Option data	

Socket Options

The following options are used to Set settings on a socket:

SO_LINGER
SO_KEEPALIVE
SO_REUSEADDR
IP_MULTICAST_TTL
IP_MULTICAST_LOOP
IP_ADD_MEMBERSHIP
IP_DROP_MEMBERSHIP
TCP_NODELAY

For more information see section Socket Options page9-70.

Socket Options

Name	Option Value	DataType	Description

SO_LINGER	0x00000080	Struct of:
		UINT32 l_onoff
		UINT32 l_linger
			Controls the action taken when unsent data is queued on a socket that is being closed. This option is only valid for TCP sockets.
			
			l_onoff 0:Linger OFF (default)
			            Other:Linger ON
			
			l_linger Normally defines the linger timeout.
			            NOT SUPPORTED, ALWAYS SET TO 0.
			
			If SO_LINGER is disabled, Socket Close returns immediately and the connection is gracefully closed in the background.
			
			If SO_LINGER is enabled with a zero timeout, Socket Close returns immediately and the connection is reset.

SO_KEEPALIVE	0x00000008	UINT32 l_keepalive	Enables/disables keep alive probes on a socket. This option is only valid for TCP sockets. l_keepalive 0:Keep alive OFF (default) Other:Keep alive ON Keep alive can be used to detect if the host is still active, and if not close down the connection. If keep alive is enabled a keep alive probe will be sent to the host after 2 hours with no data being sent or received on a connection. This packet is designed to provoke an ACK response from the host. If no ACK is received another 8 keep alive probes will be sent with 75 seconds interval, and if non of them is ACKed the connection will be reset.
SO_REUSEADDR	0x00000004	UINT32 l_reuseaddr	Enables/disables reuse address option on a socket. This option is only valid for TCP sockets. l_reuseaddr 0:Reuse address OFF (default) Other:Reuse address ON When reuse address option is enabled it is possible to reuse a TCP port even if the port is busy in TIME_WAIT state. If the port is busy in other states an error will still be generated. This can be useful for a server implementation that is shut down and directly restarted while sockets are still active on its port.

IP_MULTICAST_TTL	0x0000000A	UINT8 b_ttl	Sets the TTL value for multicast packets. This option is only valid for UDP sockets. b_ttl 1-255 (Default 1) The TTL value is part of the IP packet header and specifies the number of routers a packet is allowed to pass before it shall be deleted. The default value of 1 prevents multicast packets from being forwarded beyond the local network.
IP_MULTICAST_LOOP	0x0000000B	UINT8 b_multicastloop	Enables/disables multicast packet loopback. This option is only valid for UDP sockets. l_reuseaddr 0:Multicast loopback OFF 1:Multicast loopback ON (default)
IP_ADD_MEMBERSHIP	0x0000000C	Struct of: UINT32 l_multiaddr UINT32 l_interface	Adds membership to a multicast group. This option is only valid for UDP sockets. l_multiaddr IP address of multicast group to join. l_interface IP address of interface to join (own IP address) By joining a multicast group the local multicast router will be notified about the multicast membership (using IGMP) and the local interface network driver will enable reception of multicast datagrams destined for this multicast address.
IP_DROP_MEMBERSHIP	0x0000000D	Struct of: UINT32 l_multiaddr UINT32 l_interface	Drops membership from a multicast group. This option is only valid for UDP sockets. l_multiaddr IP address of multicast group to leave. l_interface IP address of interface (own IP address) By leaving a multicast group the local multicast router will be notified and the local interface network driver will disable reception of multicast datagrams destined for this multicast address.

TCP_NODELAY	0x00002002	UINT32 l_nodelay	Enables/disables the Nagle algorithm on a socket. This option is only valid on TCP sockets.
			l_nodelay 0:Nagle algorithm ON (default)
			l_nodelay Other:Nagle algorithm OFF
			For some applications, especially request/response applications, the performance over a TCP connection may be poor due to the interaction between the Nagle algorithm and the delayed acknowledgment functionality. Then the TCP_NODELAY option can be used to disable the Nagle algorithm to increase performance.
			For more information about Nagle algorithm see RFC 896.

Other Commands

Commands in this category:

Mailbox Command	Description	Page
Alter LNK and ACT Led Functionality (ALT_LNK_ACT_LEDS)	Alter the behaviour of the Link and Activity LEDs.	9-74
Alter Module Status LED Functionality (ALT_MS_LED_FUNCTION)	Alter the behaviour of the Module Status LED	9-75
Alter Network Status LED Functionality (ALT_NS_LED_FUNCTION)	Alter the behaviour of the Network Status LED	9-76
Get DIP Switch GET_DIP_SWITCH	Returns the setting of the onboard DIP switch	9-77
DNS Request (DNS_REQUEST)	Asks the configured DNS server for the IP address of a specified host	9-78
Send Email (SEND_EMAIL)	Sends an email message to a specified recipient	9-79
Request SSI Data (REQUEST_SSI_DATA)	Requests SSI data from the application (issued by the Anybus module)	9-81
Write SSI Data (WRITE_SSI_DATA)	Writes SSI data to the application (issued by the Anybus module)	9-82
Write Output Area (WRITE_OUT-PUT_AREA)	Writes data to the Output Area	9-83

Alter LNK and ACT LED Functionality (ALT_LNK_ACT_LEDS)

Description

This command alters the behaviour of the Link and Activity LEDs.

Note: This command may only be issued during initialisation.

Initiated by	Application
Command no.	0017h
Extended Header	-
Message data	LED function
Response data	(the response holds a copy of the command data)

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0017h	0017h	*ALT_LNK_ACT_LEDS*
Data size	0001h	0001h	*1 data byte*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message databyte 1	LED configuration	LED configuration	

- **LED Configuration Value**

Value	Description
01h	Led configuration 1 (Led 1 = Link, Led 4 = Activity)
02h	Led configuration 2 (Led 1 = Link/Activity, Led 4 = Disabled)

Alter Module Status LED Functionality (ALT_MS_LED_FUNCTION)

Description

This function alters the behaviour of the Module Status LED.

Note: This command may only be issued during initialisation.

Initiated by	Application
Command no.	000Ch
Extended Header	-
Message data	LED function
Response data	(the response holds a copy of the command data)

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	000Ch	000Ch	*ALT_MS_LED_FUNCTION*
Data size	0001h	0001h	*1 data byte*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message databyte 1	LED configuration	LED configuration	

- **LED Configuration byte value**

Value	Description
01h	Led configuration 1
02h	Led configuration 2
03h	Led configuration 3
04h	Led configuration 4 (Disable)

Alter Network Status LED Functionality (ALT_NS_LED_FUNCTION)

Description

This function alters the behaviour of the Network Status LED.

Note: This command may only be issued during initialisation.

Initiated by	Application
Command no.	000Dh
Extended Header	-
Message data	LED function
Response data	(the response holds a copy of the command data)

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	000Dh	000Dh	*ALT_NS_LED_FUNCTION*
Data size	0001h	0001h	*1 data byte*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message databyte 1	LED configuration	LED configuration	

- **LED Configuration byte value**

Value	Description
01h	Led configuration 1
02h	Led configuration 2
03h	Led configuration 3

Get DIP Switch (GET_DIP_SWITCH)

Description

This command returns the setting of the onboard switch.

Initiated by	Application
Command no.	0012h
Extended Header	-
Message data	-
Response data	Switch value

Command and response layout

	Command		Expected response	
Message ID	(ID)		(ID)	
Message information	4002h		0002h	*Fieldbus Specific Message*
Command	0012h		0012h	*GET_DIP_SWITCH*
Data size	0000h		0001h	*1 data byte*
Frame count	0001h		0001h	
Frame number	0001h		0001h	
Offset high	0000h		0000h	
Offset low	0000h		0000h	
Extended word 1	-		-	
Extended word 2	-		-	
Extended word 3	-		-	
Extended word 4	-		-	
Extended word 5	-		-	
Extended word 6	-		-	
Extended word 7	-		-	
Extended word 8	-		Fault information	
			Switch Value	**Response databyte**

- **Switch Value**

b7	b6	b5	b4	b3	b2	b1	b0
Switch 1	Switch 2	Switch 3	Switch 4	Switch 5	Switch 6	Switch 7	Switch 8

A set bit indicates that the switch is in ON position.

DNS Request (DNS_REQUEST)

Description

This command sends a request to the configured DNS server for the IP address of a specified host.

Initiated by	Application
Command no.	0030h
Extended Header	-
Message data	Host (string, null-terminated)
Response data	IP address of host, or 0.0.0.0 if not found.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0030h	0030h	*DNS_REQUEST*
Data size	(size)	0004h	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	-	
Message data	Host (string, null-terminated)	IP address (high)	**Response data word 1**
		IP address (low)	**Response data word 2**

Send Email (SEND_EMAIL)

Description

This command sends an email to a specified recipient. The message data is sent as several fragments, with a total maximum size of 1024 bytes. The maximum size of each fragment is 256 bytes.

Initiated by	Application
Command no.	0070h
Extended Header	Fault information
Message data	Email message specification, fragmented.
Response data	The response data is a copy of the command data.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0070h	0070h	*SEND_EMAIL*
Data size	(fragment size)	(fragment size)	*Max. 256 bytes / fragment*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	Fragment Type	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	SMTP Error	*(Last fragment only)*
Extended word 8	-	Fault information	
Message data	Fragment Data	Fragment Data	

- **Fragment Type**

  This value must match the sequence of the fragments as follows:

Value	Description
0000h	This is the first fragment
0001h	This is a subsequent fragment
0002h	This is the last fragment

- **SMTP Error**

  If an SMTP error occurred, the 'SMTP Error' word contains the error code from the SMTP server, see RFC 821 "Simple Mail Transfer Protocol" for more information.

- **Fragment Data**

  The different parts of the email message shall be sent in the following order:

Fragment no.	Fragment Type	Description
1st	0000h	Recipient(s), separated by semicolon (string, null-terminated)
2nd	0001h	Sender address (string, null-terminated)
3rd		Subject line (string, null-terminated)
4th		Message body
...		
...		
...		
...		
(last fragment)	0002h	

Request SSI Data (REQUEST_SSI_DATA)

Description

This message is issued by the Anybus module when a SSI has requested data from the application.

Example:

The following SSI...

```
<?--#exec cmd_argument='printf( "Data: %u", MbReadWord( 42 ) )'-->
```

... will cause the module to issues a REQUEST_SSI_DATA message. The value '42' will be passed to the application.

See also 8-5 "printf".

Initiated by	Anybus
Command no.	00A0h
Extended Header	SSI Identifier
Message data	SSI Data
Response data	-

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	00A0h	00A0h	*REQUEST_SSI_DATA*
Data size	0000h	(data size)	*(size of data)*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	SSI Identifier	SSI Identifier	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	-	
		SSI Data	Response Data

- **SSI Identifier**

  Identifier which can be used as desired by the application to address a specific block of data.

- **SSI Data**

  Data associated with the specified SSI Identifier.

Write SSI Data (WRITE_SSI_DATA)

Description

This message is issued by the Anybus module when a SSI writes data to the application.

Example:

The following SSI...

```
<?--#exec cmd_argument='scanf( "Input", "%i", MbWriteWord( 24 ) )'-->
```

... will cause the module to issues a WRITE_SSI_DATA message each time a form with an object named "Input" is sent to the web server. The value '24' will be passed to the application.

See also 8-6 "scanf".

Initiated by	Anybus
Command no.	00A0h
Extended Header	SSI Identifier
Message data	-
Response data	SSI Data

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	00A1h	00A1h	*WRITE_SSI_DATA*
Data size	(data size)	0000h	*(size of data)*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	SSI Identifier	SSI Identifier	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	-	
Message Data	SSI Data		

- **SSI Identifier**

    Identifier which can be used as desired by the application to address a specific block of data.

- **SSI Data**

    Data associated with the specified SSI Identifier.

Write Output Area (WRITE_OUTPUT_AREA)

Description

This command writes data to the Output Area.

Initiated by	Application
Command no.	00C0h
Extended Header	Offset, No. of bytes
Message data	Data
Response data	The response data field holds a copy of the command data.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	00C0h	00C0h	*WRITE_OUTPUT_AREA*
Data size	(data size)	0000h	*(size of data)*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	Offset	-	
Extended word 2	No. of Bytes	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	-	
Message Data	Data	Data	

- **Offset**

  Destination in Output Area.

- **No. of Bytes**

  Number of bytes to write.

- **Data**

  Data that shall be written.

Fieldbus Specific Area

Memory Map

The Anybus module can handle 16 non-blocking sockets simultaneously. These can be accessed using the mailbox socket interface to send and receive transparent data over the network. Information about these 16 sockets can be read in the fieldbus specific area, see memory map below.

Address	Contents	Access
640h - 64Bh	Socket Status (Descriptor 0)	RO
64Ch - 657h	Socket Status (Descriptor 1)	RO
658h - 663h	Socket Status (Descriptor 2)	RO
664h - 66Fh	Socket Status (Descriptor 3)	RO
670h - 67Bh	Socket Status (Descriptor 4)	RO
67Ch - 677h	Socket Status (Descriptor 5)	RO
688h - 693h	Socket Status (Descriptor 6)	RO
694h - 69Fh	Socket Status (Descriptor 7)	RO
6A0h - 6ABh	Socket Status (Descriptor 8)	RO
6ACh - 6B7h	Socket Status (Descriptor 9)	RO
6B8h - 6C3h	Socket Status (Descriptor 10)	RO
6C4h - 6CFh	Socket Status (Descriptor 11)	RO
6D0h - 6DBh	Socket Status (Descriptor 12)	RO
6DCh - 6E7h	Socket Status (Descriptor 13)	RO
6E8h - 6F3h	Socket Status (Descriptor 14)	RO
6F4h - 6FFh	Socket Status (Descriptor 15)	RO
700h - 701h	Network Status	RO
702h	Num Modbus connections	RO
703h	Num active Class 1 connections	RO
704h	Num active Class 3 connections	RO
705h	Num timed out Class 1 connections	RO
706h	Num timed out Class 3 connections	RO
707h - 7AFh	Reserved	-
7B0h - 7B1h	EtherNet/IP Status	RO
7B2h - 7BFh	Reserved	-

Socket Status Structure

Offset	Register	Type
000h	Socket Type	Byte
001h	Socket Status	Byte
002h - 003h	Socket Information	Word
004h - 005h	Local Port Number	Word
006h - 007h	Host Port Number	Word
008h - 00Bh	Host IP Address	Long

Socket Type

Value	Description
00h	No active socket (free to use)
01h	TCP socket
02h	UDP socket
03h - FFh	(reserved)

Socket Status

Value	Description
00h	Not active
01h	Listening
02h	Connecting
03h	Connected
04h	Connection refused
05h	Connection timed out
06h	Connection failed
07h - FFh	(reserved)

Socket Information

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
(reserved)															DA

Bit	Description
DA	0: Data Not Available 1: Data Available

Local Port Number

This is the local port number that the socket is associated with.

Host Port Number

This is the host port number that the socket is associated with or connected to.

Host IP-address

This is the host IP-address that the socket is associated with or connected to.

Network Status

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
(reserved)													COLL	USE	LINK

Bit	Description
LINK	0: Ethernet hardware link is not established 1: Ethernet hardware link is established
USE	0: IP address collision probing has not yet been attempted 1: IP address collision probing has passed, the configured IP address is used by the module **Note:** This bit is only relevant when the 'COLL'-bit is zero (below)
COLL	0: No IP address collision detected 1: IP address collision detected, another host uses the configured IP address

EtherNet/IP Status

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Connection Status								Run/Idle Status							

7B0h (MSB)   7B1h (LSB)

Connection Status

Value	Description
00h	No connection
01h	Connected
02h	Connection time out

Run/Idle Status

Value	Description
00h	Idle
01h	Run

Num Modbus connections

Number of established ModbusTCP connections.

Num active Class1 connections

Indicates number of error free Class1 connections.

Num active Class3 connections

Indicates number of error free Class3 connections.

Num timed out Class1 connections

Indicates number of Class1 connections that's in timeout state.

Num timed out Class3 connections

Indicates number of Class3 connections that's in timeout state.

Miscellaneous

Control Register Area

Fieldbus Type

The fieldbus type value for this product is 0083h.

Module Type

The module type value for this product is 0101h (Anybus-S).

Watchdog Counter Input (7D2h... 7D3h)

If the application has enabled the Watchdog Counter Input and doesn't update it properly, the module will seize all network participation (the MAC controller will be held in reset).

Event Notification Cause/Source Registers

- **ON/OFF Line Indication (FBON/FBOF)**

  By default, these bits are triggered by the Link Status.

  For other options, see...

  - 3-4 "On/Off Line Configuration"
  - 9-47 "Modbus ON/OFF Line Configuration (MB_ON_OFF_LINE_CONFIG)"

- **Network Reset Functionality (RST)**

  The reset functionality is triggered by the reset service in the Identity Object. Additionally, the application can be notified of reset requests through the mailbox interface.

  See also...

  - 9-3 "Reset Service"
  - 10-80 "Reset Notification (ID_RESET_NOTIFY)"
  - 10-81 "Get Reset Parameter (GET_ID_RESET_PARAM)"
 - 10-84 "Reset on IP Change (RST_ON_IP_CHANGE)"

Firmware Upgrade

The Anybus module supports firmware updates via FTP. Follow the steps below:

1. As a precaution, make a backup copy of the filesystem contents before proceeding.

2. Upload the new firmware file(s) to the system root ("\"), or to the 'user\'-directory.

3. Reset the module and wait until the watchdog LED flashes 2Hz green (may take up to 1 minute).

4. Reset the module again. The new firmware is now operational.

Formatting the File System

In case of major file system damage, it is possible to reinitialize the file system as follows:

1. Attach a short jumper to the PCB as shown in figure. Apply power.

2. Wait until the watchdog LED turns red.

3. Disconnect power.

4. Remove jumper.

5. Apply power.

6. Wait approx. 1 minute while the filesystem is being formatted.
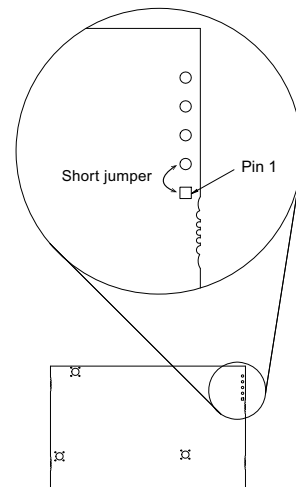
IP-related implementation details

In order to conserve memory the module has a limit on how many inbound TCP connections that can be open at a given time. In the present implementation this is set to 300 and the on-board servers as well as the application-accessible socket interface allocates their 'listening' TCP sockets from this pool.

Technical Specification

Electrical Specification

Protective Earth (PE) Requirements

All Anybus-S/M modules feature cable shield filters designed in accordance with each network standard. To be able to support this, the application *must* provide a connection to PE (Protective Earth) as described in the general Anybus-S Parallel Design Guide. HMS cannot guarantee proper EMC behaviour unless this requirement is fulfilled.

Isolation

Isolation between the application, the network, and protective earth (PE):

Isolation Barrier	Working Voltage		Distance	
	Creepage	Clearance	External	Internal
Application to PE	200V	2500V	2.0mm	0.4mm
Application to Network	250V	2500V	2.5mm	0.4mm
Network to PE	100V	1500V	1.4mm	0.4mm

(Tests performed according to EN 60950-1)

Power Supply

Supply Voltage

The module requires a regulated 5V power supply as specified in the Anybus-S Parallel Design Guide.

Power Consumption

The maximum power consumption is 450mA.

Environmental Specification

Temperature

Tests performed according to IEC-60068-2-1, IEC-60068-2-2 and IEC 60068-2-14.

Operating:	0 to 70°C	(32 to 158°F)
Storage:	-25 to 85°C	(-13 to 185°F)

Humidity

The product is designed for a relative humidity of 5 to 95% non-condensing.

Tests performed according to EN 60068.

EMC (CE) Pre-compliance

EMC pre-compliance testing has been conducted according to the Electromagnetic Compatibility Directive 2004/108/EC. For more information please consult the EMC pre-compliance document, see product/support pages for Anybus-S Ethernet at www.anybus.com.
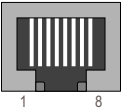
Connectors

Application Connector

(Consult the general Anybus-S Parallel Design Guide for more information)

Ethernet

RJ45 (Standard Connector)

| Pin	Signal	Notes
1	TD+	-
2	TD-	-
3	RD+	-
4	-	Normally left unused; to ensure signal integrity, these pins are tied together and terminated to PE via a filter circuit in the module.
5	-	
6	RD-	-
7	-	Normally left unused; to ensure signal integrity, these pins are tied together and terminated to PE via a filter circuit in the module.
8	-	

Board to Board

Pin	Signal	Connect to RJ45 pin...	Notes
1	Shield	Housing	-
2	-	4	(See notes for pins 4 and 5 in RJ45 connector)
3	-	5	
4	-	-	(not used)
5	TD+	1	-
6	TD-	2	-
7	RD+	3	-
8	-	7	(See notes for pins 7 and 8 in RJ45 connector)
9	RD-	6	-
10	-	8	(See notes for pins 7 and 8 in RJ45 connector)

Mechanical Specification

Measurements, Connectors & LEDs
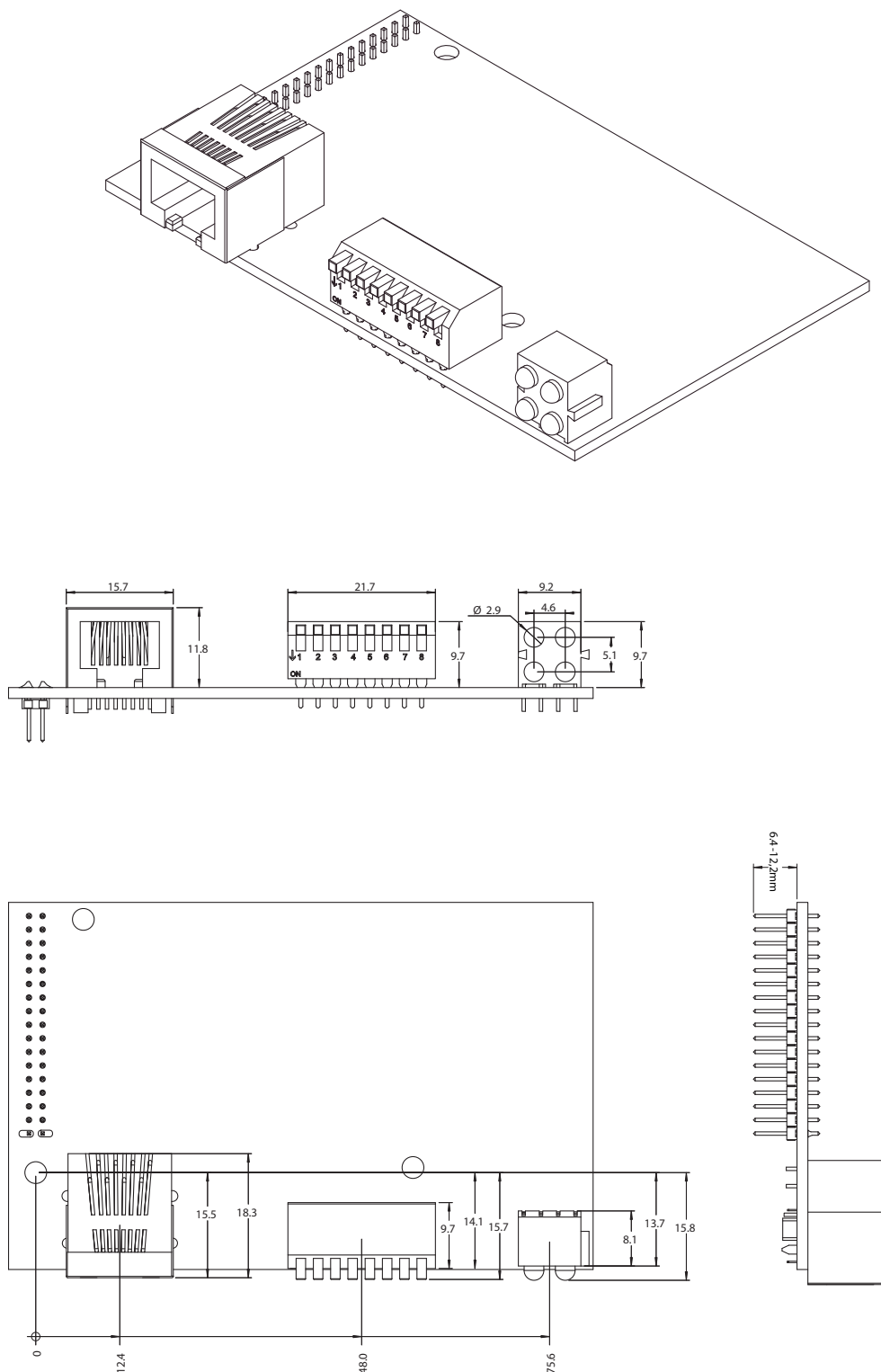
Copyright Notices

This product includes software developed by Carnegie Mellon, the Massachusetts Institute of Technology, the University of California, and RSA Data Security:

****

Copyright 1986 by Carnegie Mellon.

****

Copyright 1983,1984,1985 by the Massachusetts Institute of Technology

****

Copyright (c) 1988 Stephen Deering.

Copyright (c) 1982, 1985, 1986, 1992, 1993

The Regents of the University of California. All rights reserved.

This code is derived from software contributed to Berkeley by Stephen Deering of Stanford University.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- • Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- • Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- • Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' ANDANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

****

Copyright (C) 1990-2, RSA Data Security, Inc. All rights reserved.

License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD4 Message-Digest Algorithm" in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD4 Message-Digest Algorithm" in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

****

Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All rights reserved.

License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.