Fieldbus Appendix

AnyBus-S Ethernet

ABS-EIT-2 - Modbus/TCP, EtherNet/IP & IT Functionality

Rev.1.32

HMS Industrial Networks AB

	☎	✉
Germany	+49- 721 - 96472 - 0	sales-ge@hms-networks.com
Japan	+81- 45 - 478 -5340	sales-jp@hms-networks.com
Sweden	+46- 35 - 17 29 20	sales@hms-networks.com
U.S.A	+1- 773 - 404 - 2271	sales-us@hms-networks.com

Table of Contents

Chapter 12 Fieldbus Specific Mailbox Commands

About This Manual

How To Use This Manual

This document is intended to be used as in conjunction with the AnyBus-S Parallel Design Guide. The reader of this document is expected to have basic knowledge in the Ethernet network 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.

Related Documentation

Document name	Author	Web
Open Modbus/TCP Specification	Schneider Automation	www.modbus.org
RFC 821	Network Working Group	-
RFC 1918	Network Working Group	-
ENIP Specifications	ControlNet International and ODVA	www.odva.org
AnyBus-S Parallel Design Guide	HMS	www.hms-networks.com

Revision List

Revision	Date	Author	Chapter	Description
<1.30	-	-	-	(See previous versions)
1.30	2003-05-23	PeP	All	Major update
1.31	2003-09-24	PeP	7, 3	Minor update
1.32	2003-10-15	ToT	Mailbox File System Interface	Minor corrections in the response messages

Conventions Used in This Manual

The following conventions are used throughout this manual:

- 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-EIP-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

Europe (Sweden)

E-mail:	support@hms-networks.com
Phone:	+46 (0) 35 - 17 29 20
Fax:	+46 (0) 35 - 17 29 09
Online:	www.hms-networks.com

HMS America

E-mail:	us-support@hms-networks.com
Phone:	+1-773-404-2271
Toll Free:	888-8-AnyBus
Fax:	+1-773-404-1797
Online:	www.hms-networks.com

HMS Germany

E-mail:	ge-support@hms-networks.com
Phone:	+49-721-96472-0
Fax:	+49-721-964-7210
Online:	www.hms-networks.com

HMS Japan

E-mail:	jp-support@hms-networks.com
Phone:	+81-45-478-5340
Fax:	+81-45-476-0315
Online:	www.hms-networks.com

About the AnyBus-S Ethernet

The AnyBus-S Ethernet v2 communication module provides a full Ethernet TCP/IP interface, supporting common protocols such as SMTP, FTP, HTTP as well as control protocols such as EtherNet/IP and Modbus/TCP. Other protocols can be implemented on top of TCP/IP or UDP using a transparent socket interface.

The module features a web server, allowing remote access of data and parameter settings via a customizable web interface.

The module is based on a high performance CPU operating at 66MHz, ensuring high troughput and trouble free operation.

Features

General

- 10 and 100mbit operation, full or half duplex
- Twisted-pair 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.
- Telnet server featuring a command line interface similar to the MS-DOS™ environment.
- Server Side Include (SSI) capability
- Web server
- Email client (Messages can be triggered by events or directly by the application)
- IP Access Control
- DNS support

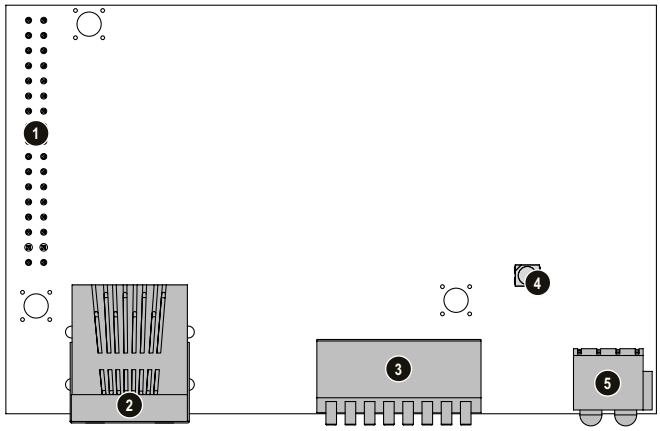
Control Protocols

- **Modbus/TCP**
  The module supports the Modbus/TCP protocol and is conform to the Modbus/TCP specification v1.0.
- **Ethernet/IP**
  The module can act as a group 2 and 3 server on an EtherNet/IP based network.
- **Transparent Socket Interface**
  Other protocols can be implemented on top of TCP/IP or UDP/IP using the transparent mailbox socket interface.

Compatible Products

This product is a member of the AnyBus concept of interchangeable fieldbus modules. This makes it fully interchangeable with any fieldbus system supported by the AnyBus-S platform with only little or no software adjustments required. This module is available in two versions, sharing some common features such as the Modbus/TCP protocol and the mailbox socket interface. These features are totally compatible between these modules.

Overview



#	Description
1	Application Connector (See general AnyBus-S Parallel Design Guide)
2	Ethernet Connector
3	Configuration Switch
4	AnyBus Watchdog (See general AnyBus-S Parallel Design Guide)
5	Status Indicators

Application Connector

The module features a standard AnyBus-S parallel interface. Consult the general AnyBus-S Parallel Design Guide for more information.

Ethernet Connector

The module supports RJ45 and board to board connector types. For more information about the connector types and their pin assignments, see Appendix D-1 “Connectors”

AnyBus Watchdog

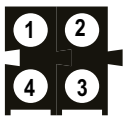
Consult the general AnyBus-S Parallel Design Guide for more information.

Configuration Switches

The configuration switches can be used to configure the IP address of the module. For more information, see 4-2 “Using the Configuration Switch”

Status Indicators

These leds indicate run time status and errors to the user. During power up, a led test sequence is performed according to the EtherNet/IP specification.



LED 1 - Link (Activity)

The behaviour of this led can be altered, see 12-92 "Alter LNK and ACT LED Functionality (ALT_LNK_ACT_LEDS)".

- **Led configuration 1 (Default)**

Colour	State	Indicates:
Green	On	The module has a link
	Off	The module does not sense a link

- **Led configuration 2 (Activity / Link)**

Colour	State	Indicates:
Green	On	The module has a link
	Off	The module does not sense a link
	Flashing	The module is receiving/transmitting on ethernet

LED 2 - Module Status

This led is required for EtherNet/IP conformance and must be labelled with one of the following: "MS", "Mod", "Mod Status" or "Module Status".

The behaviour of this led can be changed using the command ALT_MS_LED_FUNCTION, see 12-93 "Alter Module Status LED Functionality (ALT_MS_LED_FUNCTION)".

- **Led configuration 1**

Colour	Frequency	Indicates:
Green	1Hz	IP address not set using configuration switch.
Red	1Hz	Invalid MAC address. (Internal Error)
	2Hz	Failed to load Ethernet configuration from FLASH
	4Hz	Internal error. (Fatal)
	ON	Duplicate IP address detected

- **Led configuration 2**

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

- **Led configuration 3 (Default)**

State	Summary	Description
Steady Off	No power	No power applied to the module.
Steady Green	Device operational	The module is operating correctly.
Flashing Green	Standby	The module has not been configured
Flashing Red	Minor fault	A minor recoverable fault has been detected
Steady Red	Major fault	A major internal error has been detected
Flashing Green/Red	Self-test	The module is performing a power on self test

- **Led configuration 4**
  This configuration deactivates this led.

LED 3 - Network Status

This led is required for EtherNet/IP conformance and must be labelled with one of the following: "NS", "Net", "Net Status" or "Network Status".

The behaviour of this led can be changed using the command ALT_NS_LED_FUNCTION, see 12-94 "Alter Network Status LED Functionality (ALT_NS_LED_FUNCTION)"

- **Led configuration 1**
  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.

- **Led configuration 3 (Default)**

State	Summary	Description
Steady Off	No power or no IP address	The module has no power or no IP address has been assigned.
Steady Green	Connected	The module has at least one established EtherNet/IP connection.
Flashing Green	No connections	There are no EtherNet/IP connections established to the module.
Flashing Red	Connection timeout	One or more of the connections in which this module is the target has timed out. This state is only left if all timed out connections are re-established or if the module is reset.
Steady Red	Duplicate IP	The module has detected that its IP address is already in use.
Flashing Green/Red	Self-test	The module is performing a power on self test

LED 4 - Activity

The behaviour of this led can be altered, see 12-92 "Alter LNK and ACT LED Functionality (ALT_LNK_ACT_LEDS)".

- **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.

Quick Start

This chapter provides a quick example on how to get the module up and running. It also demonstrates how easy it is to create your own webpage and upload it to the module.

Requirements: An Ethernet network and a PC connected to that network, running Windows™ 98/ME/ 2000/XP and Microsoft Internet Explorer 5.5 or higher.

Step 1: Connect the module to your network

Make sure that the power is off. Connect the Ethernet network cable to the fieldbus connector of the module.

Step 2: Configure the IP address

1. Set the configuration switch to 00000000. Turn the power ON.
2. Initialize the module using the following mailbox commands START_INIT - AnyBus_INIT - END_INIT: (See AnyBus-S Design Guide)
3. Open a DOS prompt on the PC.
4. Type 'arp -s <IP address> <MAC address>'

   Substitute <MAC address> with the MAC address of your module. The MAC address is printed on a label on the bottom side of the module

   Substitute <IP address> with the IP number you wish to use for the module. Ask your network administrator for a free IP number.
5. Type 'ping <IP address>'
6. Type 'arp -d <IP address>'

   The module will now adopt the IP address specified in the 'arp -s' command.

   To read more about configuring the network settings, see 4-1 "Network Configuration".

Step 3: Browse the File System

7. Open a Windows Explorer™ window.
8. Type 'FTP://<IP address>' in the address field. (Substitute <IP address> with the IP address you are using for the module).

   You can now browse the file system.

   However, there is nothing to browse yet (except for the '\ehtcfg.cfg' file), as we haven't created any files. Close the Explorer window again.

Step 4: Create some files

9. Click on the 'Start' menu and select 'Run...'
10. Type 'telnet <IP address>'

    Substitute <IP address> with the IP address you are using for the module.

The Telnet client will be opened, and connected to the module. You can now browse the filesystem using a command line interface. Type 'help' for a quick explanation on the available commands.

11. Type 'md mydirectory'. You have now created a directory called 'mydirectory'. Type 'dir' to view it.

To move inside the directory, type 'cd mydirectory'. To move back again, type 'cd..'

12. Type 'mkfile myfile'. You have now created an empty file called 'myfile'. Let's put something in it.

13. Type 'append myfile "Easy file handling!" '. You have now added the line 'Easy file handling!' to your new file. View it's contents by typing 'type myfile'

14. Open a Windows Explorer™ window. Type 'FTP://<IP address>' in the address field. (Substitute <IP address> with the IP address you are using for the module). Watch the files you created earlier using the Telnet application!

(If your files are not present, press 'F5' to update the window content)

Don't close this window yet. If you are not in the root directory, make sure you are.

Step 5: Upload a Web page

15. Open the Windows™ Notepad and type the following:

```
<html>
<head>
<title>AnyBus-S Ethernet</title>
</head>
<body>
<center><h3>Hello world!</h3>Amazing.</center>
</body>
</html>
```

16. Save the file using the filename 'index.htm'.

17. To upload the webpage to the module, simply drag it to the Explorer window you opened earlier.

Note: This example requires Internet Explorer 5.5 or higher, but it is possible to use any FTP client. However, the procedure may not be similar to this example.

You have now uploaded your webpage to the module.

Step 6: View the Web page

18. Open a web browser.

19. Type 'HTTP://<IP address>' in the address field. (Substitute <IP address> with the IP address you are using for the module).

You are now viewing the webpage you created earlier!

Filesystem

The filesystem is a fixed-size storage area with a hierarchical directory structure. Any user- or application data can be stored in files within the filesystem. Files can be grouped in directories for increased readability.

The file system provides both non volatile (FLASH) and volatile (RAM) storage. The FLASH disc is intended for static data such as user HTML files, configuration files etc. The RAM disc area is intended for frequently accessed files such as log files etc. Note that the RAM disc is disabled by default and has to be enabled by the application using the mailbox command CREATE_RAM_DISC (See 12-90 "Create RAM disc (CREATE_RAM_DISC)").

The filesystem features two security levels. Depending on security level, different users can have access to different files and directories. The filesystem can be accessed via FTP, Telnet, HTTP, and from the application using mailbox commands. The application has unrestricted access to the filesystem regardless of security settings.

- **Case Sensitivity**

  The file system is case sensitive. This means that the file 'AnyBus.txt' is not identical to the file 'AnyBus.TXT'.

- **Filename / Pathname length**

  Filenames can be a maximum of 48 characters long. Pathnames can be 256 characters in total, filename included.

- **File size**

  File size is not restricted. However, the size cannot exceed the space available in the file system.

- **Free space**

  Approximately 1.4MB non-volatile (FLASH) / 1.0MB volatile (RAM). By default, the RAM disc is disabled. It can be enabled using the mailbox command CREATE_RAM_DISC, see 12-90 "Create RAM disc (CREATE_RAM_DISC)".

Important Note

The non volatile storage area of the filesystem is located in FLASH memory. Each FLASH segment can only be erased approximately 1000000 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

Security

The file system features two security levels; Admin and Normal. Security level is set at a per user basis, or globally using the mailbox command GLOBAL_ADMIN_MODE. (See 12-97 “Global Administration Mode (GLOBAL_ADMIN_MODE)”)

- **Admin Mode**

  Admin users has full access to the filesystem through FTP and Telnet. This enables the user to access areas of the filesystem, that is restricted or inaccessible in Normal mode.

  The Admin user accounts are defined in the file ‘ad_pswd.cfg’.

- **Normal Mode**

  This mode is recommended for normal operation, so that web pages and other settings are protected from FTP and Telnet access.

  The accounts for normal users are defined in the file ‘sys_pswd.cfg’.

Files within the file system can be protected from web access through username/password authorization, see 3-7 “Password Files” and 3-8 “web_accs.cfg”. It is also possible to configure which IP addresses and what protocols that are allowed to connect to the module, see 3-5 “ip_accs.cfg”.

Normal Mode

In this mode, the FTP and Telnet servers are enabled only if there is a subdirectory called “\user”. When a normal user connects via FTP or Telnet, this directory will be their root directory. The user will not be able to access files outside this directory and it’s subdirectories.

If user/password protection for FTP and Telnet is required in normal mode, a file called “sys_pswd.cfg” must be placed in the directory “\user\pswd\”. Files in this directory cannot be accessed from a web browser.

If the mailbox command GLOBAL_ADMIN_MODE isn’t sent to the module during initialization and a valid admin password file (See 3-7 “Password Files”) is found, the module will operate in this mode.

Note: The application has unrestricted access to the filesystem regardless of security settings.

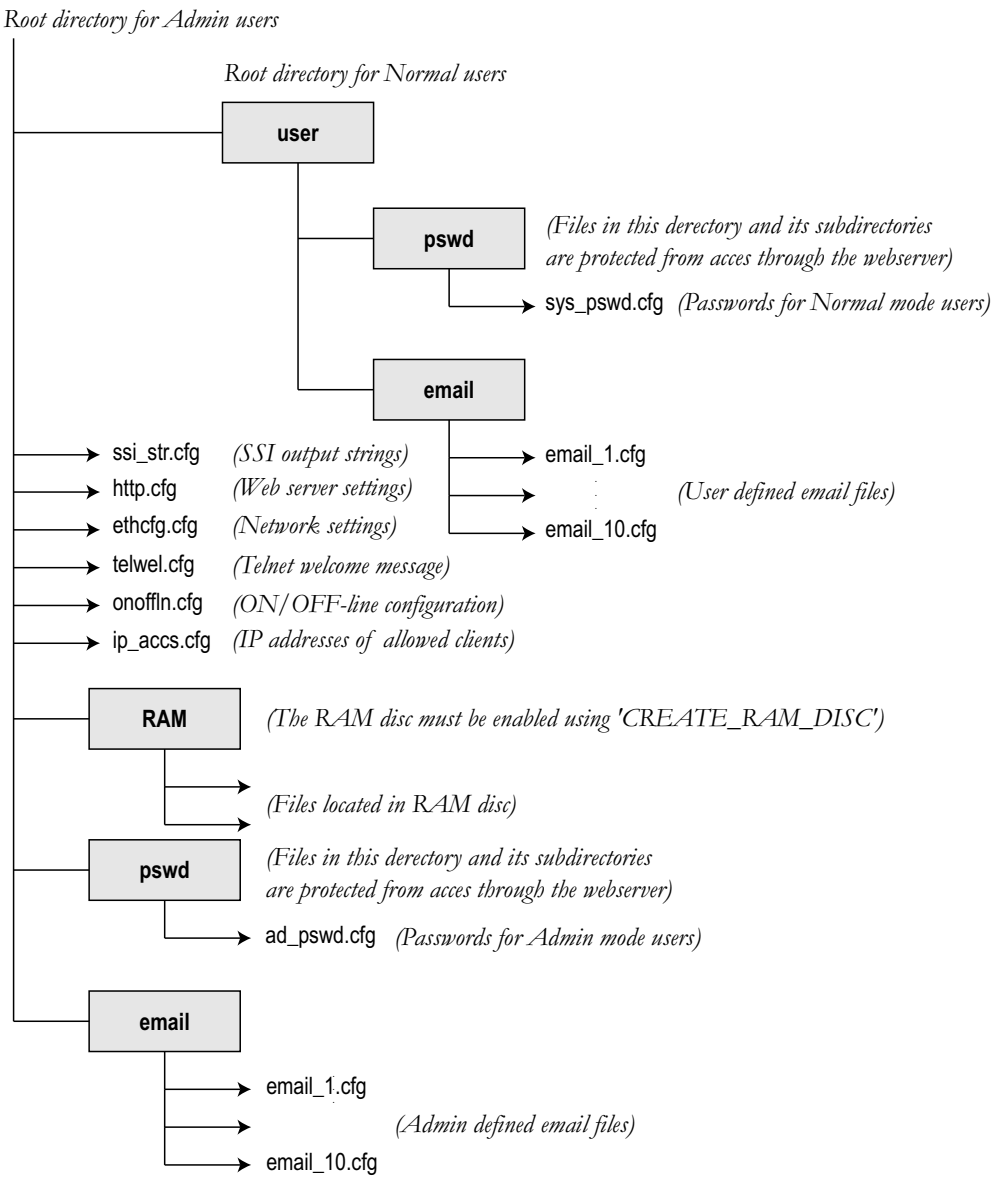
Global Admin Mode

If no admin password file (See 3-7 “Password Files”) is found or the mailbox command GLOBAL_ADMIN_MODE is sent to the module during initialization, the module will run in Global Admin Mode Mode; i.e. all users will have Admin access rights. No login is needed for Telnet, and the FTP server accepts any username/password combination.

Global Admin Mode is primarily intended for product configuration and development.

Structure

The figure below illustrates the structure of the file system, where the system files are located, and which areas that can be accessed by Normal/Admin users.



Virtual File System

The module also contains a virtual file system containing a set of files used to build the default configuration webpage. The virtual file system can be overwritten or disabled, but not erased; A file with the same name in the file system replaces the file in the virtual file system until it is removed.

The entire virtual file system can be disabled using the mailbox command DISABLE_VFS, see 12-89 “Disable Virtual File System (DISABLE_VFS)”.

For more information about the virtual files and their contents, see 8-1 “Default Web Pages”.

System Files

The module uses these files for configuration purposes. The system files are ASCII files and can be edited with any text editor. Depending on security settings, the files may be inaccessible for normal users. Generally, the module has to be restarted in order for any changes in these files to have effect.

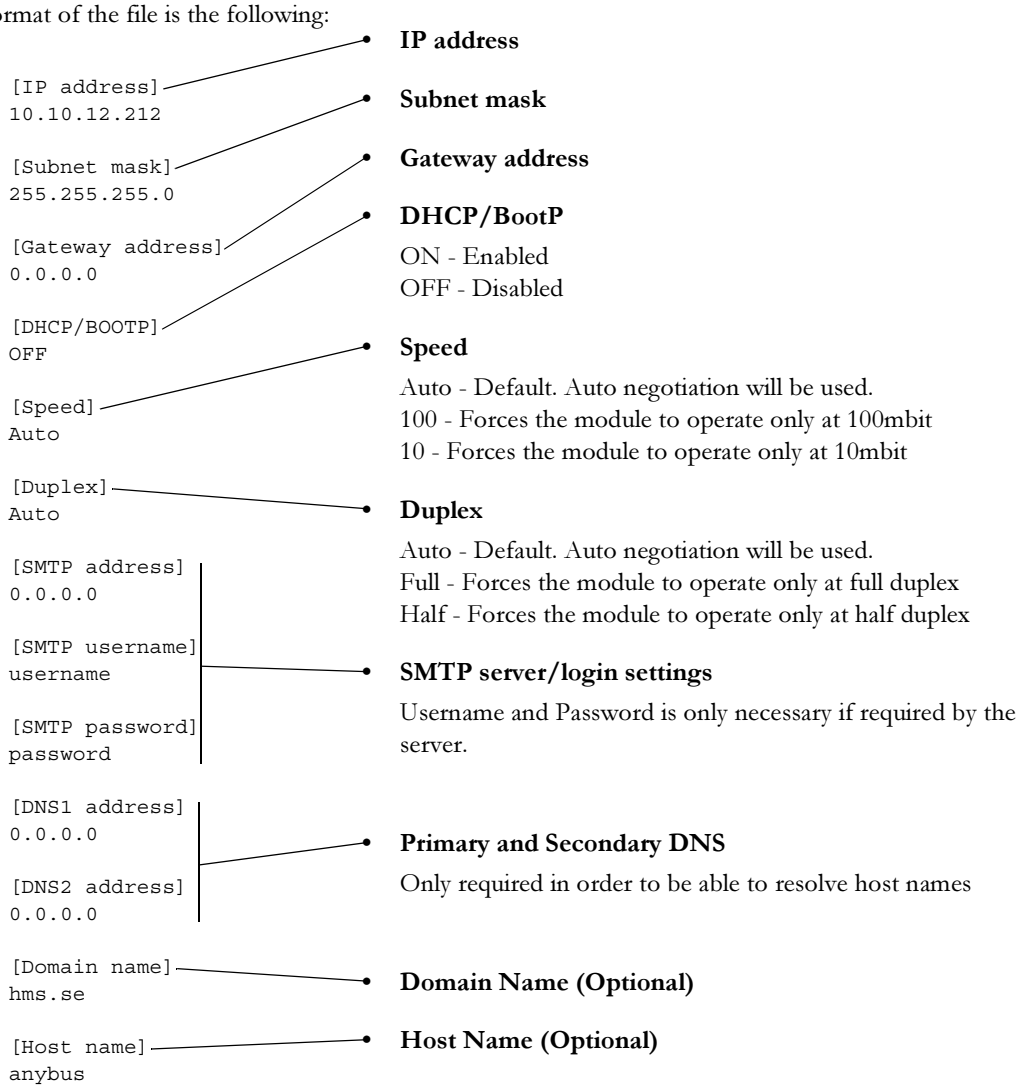
Note: It is very important to follow the exact syntax specifications for each configuration file, otherwise the module might have problems interpreting it, which can result in a faulty or non-expected behaviour.

Configuration files

'ethcfg.cfg'

This file contains the network configuration and is read by the module at start up. The settings in this file may be affected by several mailbox- and SSI commands. For more information about network configuration see 4-1 "Network Configuration".

The format of the file is the following:



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\eth_settings.cfg
```

In this example, the settings described above will be loaded from the file 'user\eth_settings.cfg'. This permits normal users to access the network configuration settings.

ip_accs.cfg

It is possible to configure which IP addresses and what protocols that are allowed to connect to the module. This information is stored in the file '\ip_accs.cfg'. The file contains one or several of the headers below.

```
[Web]
[FTP]
[Telnet]
[Modbus/TCP]
[Ethernet/IP]
[All]
```

Under each header the allowed IP addresses are written. The wildcard '*' can be used to allow series of IP addresses. If a protocol header is not given, the system will use the configuration set below the header 'All'. If the 'All' header is not given, the protocol will not accept any connections.

Example:

```
[Web]
10.10.12.*
10.10.13.*
[FTP]
10.10.12.*
[Telnet]
10.10.12.*
[All]
*.*.*.*
```

The above example will allow all IP addresses beginning with 10.10.12 to access all protocols in the module. Adresses beginning with 10.10.13 will be able to access the web server, but not the FTP and Telnet servers. The Modbus/TCP and Ethernet/IP servers will accept connections from any IP address.

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]
\my_settings\ip_access_rights.cfg
```

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

onoffln.cfg

The ON/OFF line functionality is by default configured to be triggered by the Link Status. It can however be configured to be triggered by for example a Modbus command. This is done by creating the file '\onoffln.cfg'.

Example:

```
[ON/OFF-line trigger]
Modbus

[Timeout]
10

[Commands]
3, 16, 23
```

ON/OFF-line trigger

Values: 'Link', 'EIP' and 'Modbus'

Timeout

Value: Timeout value. 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, the ON/OFF line functionality will trigger on all Modbus commands.

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

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]
\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: The information in this file will be ignored if the mailbox command 'MB_ON_OFF_LINE_CONFIG' has been sent to the module. (See 12-25 "Modbus ON/OFF Line Configuration (MB_ON_OFF_LINE_CONFIG)")

http.cfg

This file holds web server configuration data. For more information about the contents of this file, see 8-1 "Configuration".

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\config\http_configuration.cfg
```

Password Files

ad_pswd.cfg & sys_pswd.cfg

User/password information for FTP and Telnet is stored in the files 'sys_pswd.cfg' (Normal users) and 'ad_pswd.cfg' (Admin users). These files must be placed in '\user\pswd' and '\pswd\ respectively. These directories are protected from web browser access.

The file format is the following:

```
User1:password1
User2:password2
...
User3:password3
```

Example:

```
User:Password
```

In this example, the username is 'User', and the password is 'Password'.

If no ':' is present, the password will be equal to the username.

Example:

```
Username
```

In this example, both username and password will be 'Username'.

web_accs.cfg

To protect a directory from web access, a file called 'web_accs.cfg' must be placed in the directory to protect. This file shall contain a list of users that are allowed to browse the protected directory and its subdirectories. Multiple of these password files may be present in the system, allowing different users to access different files and directories.

The file format is the same as for the 'ad_pswd.cfg' and 'sys_pswd.cfg' files, except that the optional parameter 'AuthName' can be added. The value of this parameter will be presented in the login window. If it is not given, the requested file/pathname will be presented instead.

File format:

```
User:Password

[AuthName]
(Message goes here)
```

The contents of this file can be redirected by placing the line '[File path]' on the first row, followed by a list of password files.

Example:

```
[File path]
\user\pswd\my_passwords\web_pswd.cfg
\user\pswd\my_passwords\more_pswd.cfg

[AuthName]
(Message goes here)
```

In this example, the accepted user/passwords will be loaded from the files '\user\pswd\my_passwords\web_pswd.cfg' and '\user\pswd\my_passwords\more_pswd.cfg'

If any errors in the format of these files is detected the user/password protection will be ignored.

Other

telwel.cfg

The default Telnet welcome message can be changed by creating this file. It shall contain the new welcome message in ASCII form.

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]
\my_settings\telnet_welcome_message.txt
```

In this example, the welcome message will be loaded from the file '\my_settings\telnet_welcome_message.txt'.

Email files (email_1.cfg, email_2.cfg ... email_10.cfg)

These files contain predefined email messages and information on how and when to send them. It is possible to have a maximum of 10 admin defined email files and 10 user defined email files. The files must be named 'email_1.cfg'... 'email_10.cfg', and placed in the folders '\email' and '\user\email' respectively. For a specification of the file format, see 9-1 "Sending a predefined email on data event".

Network Configuration

Before you can use the module on your network, you must configure the IP address, the Subnet mask, and optionally, the Gateway address.

For a quick example on how to configure the module, see 2-1 "Quick Start".

Introduction

IP address

The IP address is used to identify each node on the TCP/IP network. Therefore, each node on the network must have a unique IP address. IP addresses are written as four decimal integers (0-255) separated by periods, where each integer represents the binary value of one byte in the IP address. This is called dotted-decimal notation.

Example:

```
Address 10000000 00001010 00000010 00011110 is written as 128.10.2.30
```

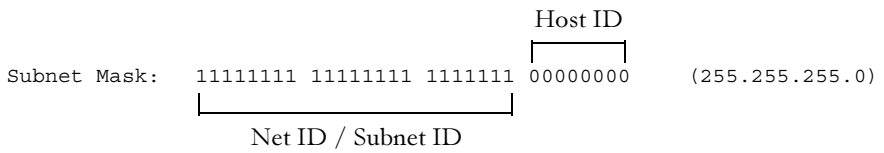
Subnet Mask

The IP address is divided into three parts - *net ID, subnet ID* and *host ID*. To separate the *net ID* and the *subnet ID* from the *host ID*, a *subnet mask* is used.

The subnet mask is a 32-bit binary pattern, where a set bit allocates a bit for network/subnet ID, and a cleared bit allocates a bit for the host ID. Like the IP address, the subnet mask is commonly written in dotted-decimal notation.

Example:

```
To make the IP address 128.10.2.30 belong to subnet 128.10.2, the subnet mask
shall be set to 255.255.255.0.
```



Note: To be able to establish communication between two devices both devices must belong to the same subnet. If not, the communication must be done through a gateway. It is therefore recommended to configure the module to the same subnet as your PC.

Special case IP addresses

Devices on an Ethernet network are not allowed to be configured to the following IP addresses; therefore do not configure the module to use any of them.

0.x.x.x	- IP address where the first byte is zero
127.x.x.x	- IP address where the first byte is 127
x.x.x.0	- IP address where the last byte is zero
x.x.x.255	- IP address where the last byte is 255

Configuring the IP Address

The module offers several ways to configure the IP address:

- Using the mailbox command SET_ETN_CONFIG
- Configuration Switch
- DHCP/BootP
- Using a predefined IP address stored in the file 'ethcfg.cfg'.
- Using SSI (see 7-1 "Server Side Include (SSI) Functionality")
- ARP
- EtherNet/IP (see 11-11 "TCP/IP Interface Object, Class F5h")

Note: Some of these configuration methods may be overridden by others. See A-1 "IP Configuration Sequence"

Using the Mailbox Interface

This mailbox command can be used during module initialization to configure the IP settings. This overrides the configuration switch setting (see below).

To read more about this method, see 12-3 "Set Ethernet Configuration (SET_ETN_CONFIG)"

Using the Configuration Switch

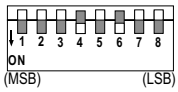
The configuration switch provides an easy way to configure the module for intranet use. The switch represents the binary value of the last byte in the IP address.

If the switch is set to a value between 1-255, and the mailbox message SET_ETN_CONFIG isn't sent by the application during module initialization, the module will use the settings described below.

```
IP address:       192.168.0.n
Subnet mask:      255.255.255.0
Gateway address:  0.0.0.0 (No gateway set)
```

The last byte (n) represents the binary value of the switches. Subnet mask and Gateway address settings are fixed to the above values when using the configuration switches.

Example:



The switches are set to 00010100 (20 decimal)

The IP address of the module will be set to 192.168.0.20

Note: The mailbox command SET_ETN_CONFIG overrides the switch settings.

Note: These settings can only be used on an intranet. This is because the IP address that is being set belongs to the private address set, see RFC 1918.

Using DHCP/BootP

If the mailbox command SET_ETN_CONFIG hasn't been sent to the module during initialization and the configuration switch is set to 0, the module will use the configuration stored in the file 'etccfg.cfg'.

If DHCP/BootP is enabled in the configuration file 'ethcfg.cfg' or if the configuration file is missing, the module will attempt to retreive the following information via DHCP/BootP:

- IP address
- Subnet mask
- Gateway address
- SMTP server address
- Default Domain Name
- Host Name
- DNS server (Primary and seconary)

The retreived information will be stored in the configuration file 'ethcfg.cfg¨.

Note: The module supports DHCP Reboot, i.e. it will ask the DHCP/BootP for the IP address stored in the configuration file. If the address is free to use, it will be assigned to the module. If not, the module will be assigned a new address.

Using a Predefined Configuration

If the mailbox command SET_ETN_CONFIG hasn't been sent to the module during initialization and the configuration switch is set to 0, the module will use the configuration stored in the file 'etccfg.cfg'.

If DHCP/BootP is disabled or a DHCP/BootP server cannot be found, the module will try to use the configuration stored in the file 'etccfg.cfg'. If this file is missing, the module will indicate an error on the Network Status LED. In this state, the module will only run the ARP protocol,. see below.

Using the Address Resolution Protocol (ARP)

The IP address can be changed during runtime using the ARP command from a PC. The new IP address will be stored in the configuration file 'ethcfg.cfg'.

Below is an example on how to change the IP address from a MS DOS™ window:

```
arp -s <IP address> <MAC address>
ping <IP address>
arp -d <IP address>
```

The arp -s command will store the IP and MAC addresses in the PC's ARP table. When the ping command is executed, the PC sends this information to the module using the MAC address. The module detects that it was addressed with the correct MAC address and adopts the IP address sent by the PC. (The arp -d command is optional, but it removes the static route from the PC ARP table). The new IP address will be stored in the configuration file 'ethcfg.cfg'.

This method can be used to reconfigure modules that already has been configured, or even to reconfigure modules outside the host's subnet.

The MAC address is printed on a label on the bottom side of the module.

Note: As the Arp command automatically configures the subnet mask to 255.255.255.0, the first three bytes of the IP address must be the same as for the PC executing the command.

Example:

```
PC - 10.10.12.67
Module- 10.10.12.x (Where x is a value between 1 and 254)
```

DNS Support

Introduction

DNS is a service that translates host names into IP addresses. Because domain names are alphabetic, they're much easier to remember. Every time a host name is used, a DNS service must translate the name into the corresponding IP address.

If a DNS server doesn't know how to translate a particular domain name, it will ask another one until the corresponding IP address is found.

The DNS configuration is specified in the configuration file 'ethcfg.cfg', see 3-4 "'ethcfg.cfg"'.

Primary and Secondary DNS (a.k.a DNS1 and DNS2)

It is possible to define two DNS servers in the module, a primary and a secondary. When a domain name should be translated, the module will ask the primary DNS server. If it for some reason fails to process the request, the module will retry using the secondary DNS.

Host / Domain Names Conventions

Before a request is sent to a DNS server, the module will process the host name as follows:

- If the hostname does not contain a dot, the module will automatically append the default domain
- If the hostname ends with a dot, that dot will automatically be removed.
- If the hostname contains a dot but isn't ending with one, the module will...
  - send a request to the DNS server for the hostname
  - if that request fails, append the default domain name and try again

Example:

(In the examples below, the default domain name is 'hms.com')

- 'test' becomes 'test.hms.com'
- 'test.hms.com.' becomes 'test.hms.com'
- 'test.a' will if not found become 'test.a.hms.com'

FTP Server

It is possible to upload/download files to/from the file system using a standard FTP client. Depending on security settings, different parts of the filesystem can be accessed by the user:

- **Normal users**

  The root directory will be '\user' unless the user has Admin access rights, see below.

- **Admin users**

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

- **Global Admin Mode**

  Any username/password combination will be accepted. All users has unrestricted access to the file system, i.e. the root directory will be '\'.

The FTP server can be disabled using the mailbox command DISABLE_FTP_SERVER, see 12-20 "Disable FTP server (DISABLE_FTP_SERVER)".

For more information about the security framework in the module, see 3-2 "Security".

Telnet Server

Through a Telnet client, the user can access the filesystem using a command line interface similar to MS-DOS™. Depending on security settings, different parts of the filesystem can be accessed by the user:

- **Normal users**

  The root directory will be '\user' unless the user has Admin access rights, see below.

- **Admin users**

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

- **Global Admin Mode**

  No login is required in this mode. All users have unrestricted access to the file system, i.e. the root directory will be '\'.

The telnet server can be disabled using the mailbox command DISABLE_TELNET_SERVER, see 12-21 "Disable Telnet Server (DISABLE_TELNET_SERVER)".

General Commands

admin

Syntax:

```
admin
```

Provided that the user can supply a valid admin username/password combination, this command enables Admin access in normal mode. Note that this command has no effect in Global Admin Mode.

help

Syntax:

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

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
```

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

exit

Syntax:

```
exit
```

This command closes the Telnet session.

Diagnostic Commands

The following commands can be viewed by the command 'help diagnostic'

arps

Syntax:

```
arps
```

Display ARP stats and table

iface

Syntax:

```
iface
```

Display net interface stats

sockets

Syntax:

```
sockets
```

Display socket list

routes

Syntax:

```
routes
```

Display IP route table

File System 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 '..'

dir

Syntax:

```
dir [path]
```

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

md

Syntax:

```
md [[path][directory name]]
```

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

rd

Syntax:

```
md [[path][directory name]]
```

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

cd

Syntax:

```
cd [path]
```

Changes current directory.

format

Syntax:

```
format
```

Formats the filesystem. This is a privileged command and can only be called in administration mode.

del

Syntax:

```
del [[path][filename]]
```

Deletes a file.

ren

Syntax:

```
ren [[path][old name]] [[path][new name]]
```

Renames a file or directory.

move

Syntax:

```
move [[source path][source file]] [[destination path]]
```

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

copy

Syntax:

```
copy [[source path][source file]] [[destination path][destination file]]
```

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

type

Syntax:

```
type [[path][filename]]
```

Types the contents of a file.

mkfile

Syntax:

```
mkfile [[path][filename]]
```

Creates an empty file.

append

Syntax:

```
append [[path][filename]] ["The line to append"]
```

Appends a line to a file.

df

Syntax:

```
df
```

Displays filesystem info.

Server Side Include (SSI) Functionality

The SSI functionality makes it possible to display or alter I/O data and configuration settings on a web page. It is also possible to use SSI functions in email messages (See 9-2 "SSI in Email Messages") however due to natural reasons some SSI functions cannot be used in email messages.

Functions

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 12-11 "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[1]

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.
```

For information about how to change the SSI output, please see chapter 7-7 "Changing SSI output"

GetText[2]

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

Syntax:

```
<?--#exec cmd arbgument='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
```

For information about changing the default SSI output, see 7-7 "Changing SSI output".

1. This function cannot be used within email messages
2. This function cannot be used within email messages

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*)	Reads a signed byte from position *offset* in the IN area
InReadUByte(*offset*)	Reads an unsigned byte from position *offset* in the IN area
InReadSWord(*offset*)	Reads a signed word (short) from position *offset* in the IN area
InReadUWord(*offset*)	Reads an unsigned word (short) from position *offset* in the IN area
InReadSLong(*offset*)	Reads a signed longword (long) from position *offset* in the IN area
InReadULong(*offset*)	Reads an unsigned longword (long) from position *offset* in the IN area
InReadString(*offset*)	Reads a string (char*) from position *offset* in the IN area
InReadFloat(*offset*)	Reads a floating point (float) value from position *offset* in the IN area
OutReadSByte(*offset*)	Reads a signed byte from position *offset* in the OUT area
OutReadUByte(*offset*)	Reads an unsigned byte from position *offset* in the OUT area
OutReadSWord(*offset*)	Reads a signed word (short) from position *offset* in the OUT area
OutReadUWord(*offset*)	Reads an unsigned word (short) from position *offset* in the OUT area
OutReadSLong(*offset*)	Reads a signed longword (long) from position *offset* in the OUT area
OutReadULong(*offset*)	Reads an unsigned longword (long) from position *offset* in the OUT area
OutReadString(*offset*)	Reads a NULL terminated string (char*) from position *offset* in the OUT area
OutReadFloat(*offset*)	Reads a floating point (float) value from position *offset* in the OUT area

scanf[1]

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 data, 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*)	Writes a byte to position *offset* in the OUT area
OutWriteWord(*offset*)	Writes a word (short) to position *offset* in the OUT area
OutWriteLong(*offset*)	Writes a long to position *offset* in the OUT area
OutWriteString(*offset*)	Writes a string to position *offset* in the OUT area
OutWriteFloat(offset)	Writes a floating point (float) value to position *offset* in the OUT area

Default output:

```
Write succeeded
Write failed
```

For information about changing the default SSI output, see 7-7 "Changing SSI output".

1. This function cannot be used within email messages

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>
```

For information about changing the default SSI output, see 7-7 "Changing SSI output".

SaveToFile[2]

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 contents can either be Appended to the file or overwrite the current content of the file.

Syntax:

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

Default output:

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

For information about changing the default SSI output, see 7-7 "Changing SSI output".

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 file system 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" To include filename %s can be included
to the string once

[SaveToFile]
Success: "String to use on success"
Failure: "String to use on failure" To include filename %s can be included to
the string once

[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'.

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) )'-->
```

Web Server

The module features a complete web server with SSI functionality. It is possible to upload web pages to the module, giving access to parameters in the DPRAM using a customizable interface.

By default the HTTP server is enabled, but it can be enabled/disabled during runtime using mailbox commands, see 12-18 "Disable Web Server (DISABLE_WEB_SERVER)" and 12-19 "Enable Web Server (ENABLE_WEB_SERVER)".

Default Web Pages

The 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 the file system.

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'.

The virtual file system contains the following files:

```
\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
```

The virtual file system can be disabled using the mailbox command 'DISABLE_VFS', see 12-89 "Disable Virtual File System (DISABLE_VFS)".

Configuration

It is possible to configure/reconfigure the reported content-type for different file types and which files that should be scanned for SSI. This is done in the configuration file '\http.cfg'. See next page for a specification of the file format.

Settings contained in '\http.cfg':

- **Content Types**

  There are a number of file types that by default will return predefined content types when requested through the web server (see 8-2 "Default Content Types"). When a file is requested through the web server it will first search for the file types specified in this file. If it's not found in this file it will search for it in its predefined content types. This means that adding file type in this file will replace it's predefined type. File types shall be added under the heading [FileTypes], see 8-2 "File format". Up to 50 additional file types can be defined.

- **SSI File Types**

  By default, all files with the extensions '.htm', '.html', '.wml' and '.wmls' are scanned for SSI. It is possible to add additional file types to be scanned under the heading [SSIFileTypes], see 8-2 "File format". Up to 50 additional SSI file types can be defined.

Default Content Types

By default, the following content types are recognized by on their filename 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
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.wmlscript	*.wmlsc
text/xml	*.xml
application/pdf	*.pdf

If the extension of a file is not recognized the content type is set to binary data "/".

File format

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

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

Example

```
[FileTypes]
tif:image/tiff
tiff:image/tiff
doc:application/msword
avi:video/x-msvideo

[SSIFileTypes]
htm
html
xml
```

Security

All files except files in the directories '\user\pswd\', '\pswd\' and files named 'web_accs.cfg' can be viewed by default. Other directories can be protected by placing a file called 'web_accs.cfg' in the directory to protect. The file should contain a list of users that are allowed to browse the directory. See 3-8 "web_accs.cfg" for a specification of the file format. Also, it is possible to configure which IP addresses that are allowed to connect to the HTTP sever, see 3-5 "ip_accs.cfg".

Email Client

It is possible to send emails from the module. To send an email, the SMTP server address must be configured. Without a valid SMTP address the module will not be able to send any email messages.

Sending an email using mailbox commands

The application can send email messages through the module using the mailbox command SEND_EMAIL, see 12-98 “Send Email (SEND_EMAIL)”

Sending a predefined email on data event

It is possible to send predefined email messages, triggered by an event in the IN/OUT area of the DPRAM. The IN and OUT area is scanned once every 0.5 second. This means that an event must be present longer than 0.5 seconds to ensure that it is detected by the module. It is possible to have up to 10 user defined, and 10 admin defined emails, triggered on different events. These shall be placed in the directories “\user\email\” for user configurable emails and “\email” for non-user configurable emails. The files must be named ‘email_1.cfg’, ‘email_2.cfg’ ... ‘email_10.cfg’.

The files shall have the following format:

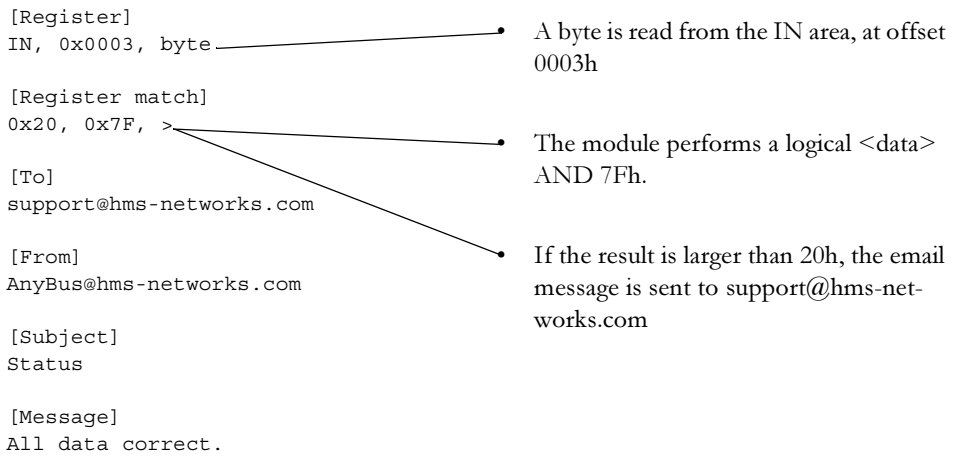
```
[Register]
Area, Offset, Type

[Register match]
Match Value, Mask, Match operand

[To]
Recipient(s)

[From]
Sender

[Subject]
Subject line

[Headers]
Extra Headers

[Message]
Message body
```

Parameter	Description
Area	Source area in DPRAM. Possible values are ‘IN’ or ‘OUT’
Offset	Source offset in DPRAM, shall be written in decimal or hexadecimal.
Type	Source data type. Possible values are ‘byte’, ‘word’, and ‘long’
Match Value	Value to compare with the source data. Shall be written in decimal or hexadecimal.
Mask	The module performs a logical ‘and’ on the source data and this Mask before the value is compared with the Match Value. The value shall be written in decimal or hexadecimal.
Match Operand	Specifies how the data shall be compared with the Match Value. Possible values: ‘<‘, ‘=’, ‘>’
Recipient(s)	Destination email addresses, semicolon separated
Sender	Sender email address
Subject line	Email subject (One line only)
Extra Headers	Optional. May be useful for advanced users when for example sending HTML emails etc.
Message Body	The actual email message.

The data is read in the DPRAM from the area and offset specified by the parameters Area, and Offset. The datasize to read is specified by the Type parameter. The module performs a logical 'AND' between the read data and the parameter Mask. The result is compared with the parameter Match Value. How the data shall be compared is specified by the Match Operand.

Note: If the [Register] or [Register match] information is changed, a reset is required for changes to take effect. Other changes will take effect directly without a reset.

Example:



Note: Hexadecimal values must be written in the format 0xN where 'N' is the hexadecimal value.

SSI in Email Messages

For predefined emails it is possible to include data in the mails. This is done in a similar way as data is added to web pages with SSI includes. Due to natural reasons, some SSI functions cannot be used in email messages.

The supported SSI commands for emails are:

- DisplayIP
- DisplaySubnet
- DisplayGateway
- DisplayDNS1
- DisplayDNS2
- DisplayHostName
- DisplayDomainName
- DisplayEmailServer
- DisplaySMTPUser
- DisplaySMTPPswd
- DisplayDhcpState
- DisplayDhcpSupport
- printf
- IncludeFile
- SsiOutput

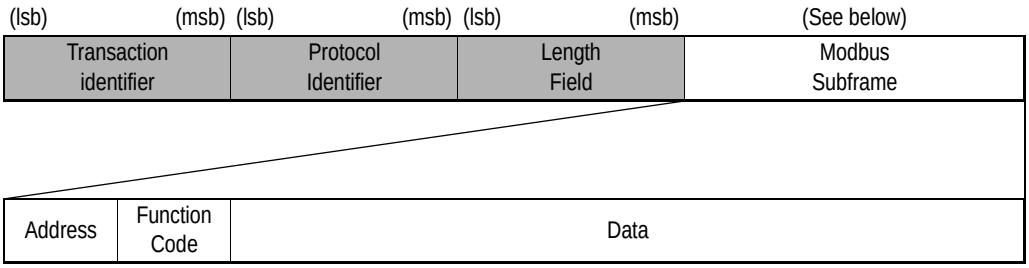
Modbus/TCP

The implementation of the Modbus/TCP server is done according to the Modbus/TCP specification v1.0. All commands according to class 0 and class 1 is implemented, as well as some of the class 2 commands. The module can handle 8 simultaneous connections.

The Modbus/TCP server can be disabled using a mailbox command, see 12-24 "Disable Modbus/TCP server (DISABLE_MB_TCP)".

Message frame format

The Modbus/TCP protocol is an implementation of the standard Modbus protocol running on top of TCP/IP. The same function codes and addressing model are used.



Note that the Modbus/TCP message frame does not include a CRC field as Modbus does, since the TCP/IP frame format already features sophisticated error checking.

For detailed information regarding the Modbus/TCP protocol, consult the Open Modbus Specification v1.0.

Port

All Modbus/TCP messages are received/transmitted on TCP port no. 502.

Supported Functions

Function Code	Function Name	Class	Affects Area	Addressing method
1	Read coils	1	IN/OUT	Bit
2	Read Input discretes	1	IN/OUT	Bit
3	Read multiple registers	0	IN/OUT	Word
4	Read input registers	1	IN/OUT	Word
5	Write coil	1	OUT	Bit
6	Write single register	1	OUT	Word
7	Read exception status	1	-	-
15	Force multiple coils	2	OUT	Bit
16	Force multiple registers	0	OUT	Word
22	Mask write register	2	OUT	Word
23	Read/Write registers	2	IN/OUT	Word

Modbus/TCP Addressing

The AnyBus-S IN and OUT areas can be configured to a maximum size of 2048 bytes each. When accessing these areas, with Modbus commands, the addressing is done according to the following tables.

IN area

Addr.	Bit address															
000h	0000h	0001h	0002h	0003h	0004h	0005h	0006h	0007h	0008h	0009h	000Ah	000Bh	000Ch	000Dh	000Eh	000Fh
001h	0010h	0011h	0012h	0013h	0014h	0015h	0016h	0017h	0018h	0019h	001Ah	001Bh	001Ch	001Dh	001Eh	001Fh
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3FFh	3FF0h	3FF1h	3FF2h	3FF3h	3FF4h	3FF5h	3FF6h	3FF7h	3FF8h	3FF9h	3FFAh	3FFBh	3FFCh	3FFDh	3FFEh	3FFFh

OUT area

Addr.	Bit address															
400h	4000h	4001h	4002h	4003h	4004h	4005h	4006h	4007h	4008h	4009h	400Ah	400Bh	400Ch	400Dh	400Eh	400Fh
401h	4010h	4011h	4012h	4013h	4014h	4015h	4016h	4017h	4018h	4019h	401Ah	401Bh	401Ch	401Dh	401Eh	401Fh
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
7FFh	7FF0h	7FF1h	7FF2h	7FF3h	7FF4h	7FF5h	7FF6h	7FF7h	7FF8h	7FF9h	7FFAh	7FFBh	7FFCh	7FFDh	7FFEh	7FFFh

Supported Exception codes

Code	Name	Description
01	Illegal function	The module does not support the function code in the query
02	Illegal data address	The data address received in the query is outside the initialized memory area
03	Illegal data value	The data in the request is illegal

Ethernet/IP

The module can act as a Group 2 and 3 server on an EtherNet/IP network. EtherNet/IP is based on the Common Industrial Protocol (CIP) which is also the application layer used by DeviceNet and ControlNet to exchange data between nodes.

The EtherNet/IP protocol support can be disabled using the mailbox command DISABLE_ETHERNET_IP. (See 12-54 "Disable EtherNet/IP (DISABLE_ETHERNET_IP)")

Implementation Notes

Rockwell Automation PLCs uses the first four bytes consumed by a device defined for status information. This status information is not defined in the EtherNet/IP specification, but all currently available PLCs have this implementation.

By default, these four I/O bytes are parsed off and ignored. However, this behaviour can be changed using the COPY_IO_STATUS mailbox command (see 12-53 "Copy I/O Status (COPY_IO_STATUS)".

Implemented Objects

EtherNet/IP requires some mandatory objects; these are implemented as well as some vendor specific objects. The mandatory objects are the ones in the specification from ODVA.

The following vendor specific objects are implemented:

- Identity Object, Class 01h
- Message Router, Class 02h
- Assembly Object, Class 04h
- Connection Manager, Class 06h
- I/O data input mapping object, Class A0h
- I/O data output mapping object, Class A1h
- Diagnostic object, Class AAh
- Parameter data input mapping object, Class B0h
- Parameter data output mapping object, Class B1h
- Port Object, Class F4h
- TCP/IP Interface Object, Class F5h
- Ethernet Link Object, Class F6h

Identity Object, Class 01h

Services

Class services: Get Attribute All
Get Attribute Single

Instance services: Get Attribute All
Get Attribute Single
Reset

Class Attributes

#	Access	Name	Type	Value	Description
1	Get	Revision	UINT	0001h	Revision 1

Instance Attributes

#	Access	Name	Type	Value	Description
1	Get	Vendor ID	UINT	Default: 005Ah	HMS Industrial Networks AB[a]
2	Get	Device Type	UINT	Default: 000Ch	Communication Adapter[a]
3	Get	Product Code	UINT	Default: 000Eh	AnyBus-S Ethernet[a]
4	Get	Revision	Struct of:		-
			USINT		Major fieldbus version[a]
			USINT		Minor fieldbus version[a]
5	Get	Status	WORD	-	Device status, see table below
6	Get	Serial Number	UDINT	Module serial number	Serial number of the module
7	Get	Product Name	SHORT_STRING	AnyBus-S EtherNet/IP	Name of product[a]

a. Can be customized using mailbox commands, see 12-27 “EtherNet/IP Specific Commands”.

Status Attribute

bit(s)	Name	Description
0	Module Owned	-
1	(reserved)	-
2	Configured	-
3	(reserved)	-
4 - 7	Extended Device Status	(See table on the right)
8	Minor recoverable fault	-
9	Minor recoverable fault	-
10	Major recoverable fault	-
11	Major unrecoverable fault	-
12 - 15	(reserved)	-

Extended Device Status

Value	Meaning
0000b	Unknown
0010b	Faulted I/O Connection
0011b	No I/O connection established
0100b	Non volatile configuration bad
0110b	Connection in Run mode
0111b	Connection in Idle mode

Reset Service

The Identity object provides a reset service. There are two different types of reset requests:

- **Type 0: 'Power Cycling Reset'**

  This service emulates a power cycling of the module.
- **Type 1: 'Out of box reset'**

  This service sets a "out of box" configuration and performs a reset.

The default behaviour when the module receives a reset command is to reset the module. If the reset type is a "out of box reset", the configuration file "ethcfg.cfg" is erased.

If the application should be notified about the reset, i.e. for resetting itself, there are two options:

1. Send ENABLE_ID_RESET_NOTIFY to the module during initialisation. (See 12-50 "Enable Reset Notification (ENABLE_ID_RESET_NOTIFY)").

   When a reset service is received by the module, it will generate the mailbox message ID_RESET_NOTIFY (see 12-51 "Reset Notification (ID_RESET_NOTIFY)"), and if necessary, delete the configuration file 'ethcfg.cfg'. The mailbox message indicates the type of reset.

   It is then up to the application to reset itself and the module.

2. Set the RST bit in the Module Status and Interrupt Notification registers when initialising the module using ANYBUS_INIT.

   When a reset service is received by the module, it will generate an interrupt. To get the type of reset, use the GET_ID_RESET_PARAM mailbox command (see 12-52 "Get Reset Parameter (GET_ID_RESET_PARAM)"). The mailbox response indicates the type of reset.

   It is then up to the application to reset itself and the module.

Message Router, Class 02h

Services

Class services: -
Instance services: -

Assembly Object, Class 04h

Services

Class services: Get Attribute Single

Instance services: Get Attribute Single
Set Attribute Single

Description

The Assembly Object binds all mapped I/O data. This data is used for I/O connections.

Default I/O Instances used are 64h and 96h. It is possible to map additional instances using the mailbox commands I/O_INPUT_MAP and I/O_OUTPUT_MAP. (See 12-35 “I/O Data Input Area Mapping (I/O_INPUT_MAP)” and 12-37 “I/O Data Output Area Mapping (I/O_OUTPUT_MAP)”)

Class Attributes

#	Access	Name	Type	Value	Description
1	Get	Revision	UINT	0002h	Revision 2
2	Get	Max Instance	UINT	-	The highest initiated instance no.

Instance Attributes - Instance/Connection Point 64h (IN)

Note: If the I/O input data size is set to 0 this instance will NOT be initialised.

#	Access	Name	Type	Value	Description
3	Get	Data	Array of BYTE	-	Data produced by AnyBus

Instance Attributes - Instance/Connection Point 96h (OUT)

Note: If the I/O output data size is set to 0 this instance will NOT be initialised.

#	Access	Name	Type	Value	Description
3	Set	Data	Array of BYTE	-	Data consumed by AnyBus[a]

a. Rockwell Automation PLCs have the first four bytes consumed by a device defined as status information. This status information is not defined in the EtherNet/IP specification, it is a Rockwell Automation implementation. Since all known PLCs have this implementation, the module strips off the first four bytes in the consumed data by default. However, this behaviour can be changed using the Copy I/O Status mailbox command, see 12-53 “Copy I/O Status (COPY_IO_STATUS)”.

I/O Data Input Mapping Object, Class A0h

Services

Class services: Get Attribute All

Instance services: Get Attribute Single

Description

This object is setup dynamically depending on how the module is initialized. The mapped attribute id:s are also available as vendor specific instance attributes (64h to 69h) in the assembly object (See 11-4 "Assembly Object, Class 04h").

The mapped data is the data that is sent *from* the module via scheduled connected messages; it is also possible to get the data with unconnected and connected message.

Note: To be compatible with AnyBus-S DeviceNet only 6 attributes are supported.

Class Attributes

#	Access	Name	Type	Value	Description
1	Get	Data	UINT	0001h	Revision 1

Instance Attributes, Instance 01h

#	Access	Name	Type	Value	Description
1	Get	Data	Array of USINT	-	Mapped I/O data
2	Get	Data	Array of USINT	-	Mapped I/O data
3	Get	Data	Array of USINT	-	Mapped I/O data
4	Get	Data	Array of USINT	-	Mapped I/O data
5	Get	Data	Array of USINT	-	Mapped I/O data
6	Get	Data	Array of USINT	-	Mapped I/O data

Attribute 1 is mapped to Assembly Instance 0x64	Attribute created automatically if input area size is zero
Attribute 2 is mapped to Assembly instance 0x65	Application creates attribute by sending IO_INPUT_MAP mailbox command
Attribute 3 is mapped to Assembly instance 0x66	Application creates attribute by sending IO_INPUT_MAP mailbox command
Attribute 4 is mapped to Assembly instance 0x67	Application creates attribute by sending IO_INPUT_MAP mailbox command
Attribute 5 is mapped to Assembly instance 0x68	Application creates attribute by sending IO_INPUT_MAP mailbox command
Attribute 6 is mapped to Assembly instance 0x69	Application creates attribute by sending IO_INPUT_MAP mailbox command

I/O Data Output Mapping Object, Class A1h

Services

Class services: Get Attribute All

Instance services: Get Attribute Single
Set Attribute Single

Description

This vendor specific object is setup dynamically depending on how the module is initialized. The mapped attribute id:s are also available as vendor specific instance attributes (96h to 9Bh) in the assembly object (See 11-4 "Assembly Object, Class 04h").

The mapped data is the data that is sent *from* the scanner via scheduled connected messages; it is also possible to get the data with unconnected and connected message.

Note: To be compatible with AnyBus-S DeviceNet only 6 attributes are supported.

Class Attributes

#	Access	Name	Type	Value	Description
1	Get	Data	UINT	0001h	Revision 1

Instance Attributes, Instance 01h

#	Access	Name	Type	Value	Description
1	Get/Set	Data	Array of USINT	-	Mapped I/O data
2	Get/Set	Data	Array of USINT	-	Mapped I/O data
3	Get/Set	Data	Array of USINT	-	Mapped I/O data
4	Get/Set	Data	Array of USINT	-	Mapped I/O data
5	Get/Set	Data	Array of USINT	-	Mapped I/O data
6	Get/Set	Data	Array of USINT	-	Mapped I/O data

Attribute 1 is mapped to Assembly Instance 0x96	Attribute created automatically if output area size is zero
Attribute 2 is mapped to Assembly instance 0x97	Application creates attribute by sending IO_OUTPUT_MAP mailbox command
Attribute 3 is mapped to Assembly instance 0x98	Application creates attribute by sending IO_OUTPUT_MAP mailbox command
Attribute 4 is mapped to Assembly instance 0x99	Application creates attribute by sending IO_OUTPUT_MAP mailbox command
Attribute 5 is mapped to Assembly instance 0x9A	Application creates attribute by sending IO_OUTPUT_MAP mailbox command
Attribute 6 is mapped to Assembly instance 0x9B	Application creates attribute by sending IO_OUTPUT_MAP mailbox command

Diagnostic Object, Class AAh

Services

Class services: Get Attribute All

Instance services: Get Attribute Single

Description

This vendor specific object provides diagnostic information from the module.

Class Attributes

#	Access	Name	Type	Value	Description
1	Get	Revision	UINT	0001h	Revision 1

Instance Attributes, Instance 01h

#	Access	Name	Type	Description
01h	Get	Module serial number	UDINT	Serial number
02h	Get	Vendor ID	UINT	Manufacturer Vendor ID
03h	Get	Fieldbus Type	UINT	Fieldbus Type
04h	Get	Module Software version	UINT	Module software version
05h	Get	Interrupt Count	UINT	Counter incremented each handshake interrupt
06h	Get	Watchdog counter in	UINT	(reserved, not implemented)
07h	Get	Watchdog counter out	UINT	Counter incremented every 1ms
09h	Get	LED	Struct of:	LED Indication:
			USINT	Top left
			USINT	Top right
			USINT	Bottom left
			USINT	Bottom right
			USINT	(reserved)
			USINT	(reserved)
0Ah	Get	Module Type	UINT	Module Type
0Bh	Get	Module Status	WORD	Bit information (freeze, clear etc.)
0Ch	Get	New data field	LWORD	Array of new data flags for 8 bytes area
0Dh	Get	Interrupt Cause	WORD	Interrupt Cause register
0Eh	Get	Interrupt Notification	WORD	Interrupt Notification register
0Fh	Get	IN cyclic I/O length	UINT	Size of I/O Input area (in bytes)
10h	Get	IN DPRAM length	UINT	Number of valid IN bytes in DPRAM
11h	Get	IN total length	UINT	Total number of IN bytes supported
12h	Get	OUT cyclic I/O length	UINT	Size of I/O Output area (in bytes)
13h	Get	OUT DPRAM length	UINT	Number of valid OUT bytes in DPRAM
14h	Get	OUT total length	UINT	Total number of OUT bytes supported
18h	Get	MAC ID	Array of USINT	Ethernet MAC ID of the module (6 bytes)
19h	Get	IP Address	UDINT	Currently configured IP address
1Ah	Get	Subnet mask	UDINT	Currently configured subnet mask
1Bh	Get	Gateway address	UDINT	Currently configured gateway address
1Ch	Get	SMTP server address	UDINT	SMTP server address
1Dh	Get	DHCP state	UDINT	0=No DHCP, 1=DHCP available
1Eh	Get	Bootloader version	UDINT	Bootloader firmware version

#	Access	Name	Type	Description
1Fh	Get	Application interface version	UINT	Application interface software version
20h	Get	Fieldbus software version	UINT	Fieldbus interface software version

Parameter Data Input Mapping Object, Class B0h

Services

Class services: Get Attribute All

Instance services: Get Attribute Single

Description

This vendor specific object is setup dynamically depending on how the module is initialized. The mapped attribute id:s corresponds to input parameter data. The mapped data is the data that is being sent *from* the module via connected- and unconnected messages.

Note: To be compatible with AnyBus-S DeviceNet only 50 attributes are supported.

Class Attributes

#	Access	Name	Type	Value	Description
1	Get	Revision	UINT	0001h	Revision 1

Instance Attributes, Instance 01h

#	Access	Name	Type	Description
01h	Get	Data	Array of USINT	Mapped data
02h	Get	Data	Array of USINT	Mapped data
...	...	...	...	...
32h	Get	Data	Array of USINT	Mapped data

Attributes are created by the application when sending the mailbox command PARAMETER_INPUT_MAP (See 12-31 “Parameter Data Input Mapping (PARAMETER_INPUT_MAP)”)

Parameter Data Output Mapping Object, Class B1h

Services

Class services: Get Attribute All

Instance services: Get Attribute Single
Set Attribute Single

Description

This vendor specific object is setup dynamically depending on how the module is initialized. The mapped attribute id:s corresponds to output parameter data in the DPRAM. The mapped data is the data that is being *received* by the module via connected and unconnected messages.

Note: To be compatible with AnyBus-S DeviceNet only 50 attributes are supported.

Class Attributes

#	Access	Name	Type	Value	Description
1	Get	Revision	UINT	0001h	Revision 1

Instance Attributes, Instance 01h

#	Access	Name	Type	Description
01h	Get/Set	Data	Array of USINT	Mapped data
02h	Get/Set	Data	Array of USINT	Mapped data
...	...	...	...	...
32h	Get/Set	Data	Array of USINT	Mapped data

Attributes are created by the application when sending the mailbox command PARAMETER_OUTPUT_MAP (See 12-33 “Parameter Data Out Area Mapping (PARAMETER_OUTPUT_MAP)”)

Port Object, Class F4h

Services

Class services: Get Attribute All
Get Attribute Single

Instance services: Get Attribute All
Get Attribute Single

Description

Instances in this class (New ports) can be registered using the REGISTER_PORT mailbox command, see 12-45 “Register Port (REGISTER_PORT)”.

Class Attributes

#	Access	Name	Type	Value	Description
1	Get	Revision	UINT	0001h	Revision 1
2	Get	Max Instance	UINT	0002h	2 is the highest instance number (Default)
3	Get	No. of instances	UINT	0001h	1 instance is implemented (Default)
8	Get	Entry Port	UINT	0002h	Returns the instance of the Port object that describes the port.
9	Get	All Ports	Array of STRUCT {UINT; UINT;}	0000h 0000h 0000h 0000h 0004h 0002h	Array of structure containing attributes 1 and 2 from each instance. Instance 1 is at byte offset 4. Instance 2 is at byte offset 8, etc. The 4 bytes at offset 0 shall be 0. (Default)

Instance Attributes, Instance 02h

#	Access	Name	Type	Value	Description
1	Get	Port Type	UINT	0004h	4 = TCP/IP
2	Get	Port Number	UINT	0002h	Port 2
3	Get	Port Object	Struct of:		-
		Path Size	UINT	0002h	Path Size
		Path	Padded EPATH	20 F5 24 01h	TCP class, Instance 1
4	Get	Port Name	SHORT_STIRNG	‘TCP/IP’	Name of port
8	Get	Node Address	Padded EPATH	-	EPATH describing our TCP/IP address

TCP/IP Interface Object, Class F5h

Services

Class services: Get Attribute All
Get Attribute Single

Instance services: Get Attribute All
Get Attribute Single
Set Attribute Single

Description

This object provides the a mechanism to configure the TCP/IP settings via EtherNet/IP. Note that writing to this object will affect the settings stored in the configuration file 'ethcfg.cfg'.

Class Attributes

#	Access	Name	Type	Value	Description
1	Get	Revision	UINT	0001h	Revision 1
2	Get	Max Instance	UINT	0001h	1 is the highest instance number
3	Get	No. of instances	UINT	0001h	1 instance is implemented

Instance Attributes

#	Access	Name	Type	Value	Description
1	Get	Status	DWORD	00000001h	1 = The interface configuration attribute contains a valid configuration
2	Get	Configuration Capability	DWORD	00000014h	Interface configuration attribute is settable. Capable of obtaining network configuration via DHCP.
3	Get/Set	Configuration Control	DWORD	-	0 - Configuration from non-volatile memory 2 - Configuration from DHCP
4	Get	Port Object	Struct of:		Physical link -> Ethernet object
		Path Size	UINT	0002h	2 words
		Path	Padded EPATH	20 F6 24 01h	Ethernet Class, Instance 1
5	Get/Set	Interface Configuration	Struct of:		-
		IP Address	UDINT	-	Currently used IP address
		Subnet Mask	UDINT	-	Currently used Subnet mask
		Gateway Address	UDINT	-	Currently used Gateway Address
		Name Server 1	UDINT	-	Primary DNS server
		Name Server 2	UDINT	-	Secondary DNS server
		Domain Name	STRING	-	Default domain name
6	Get/Set	Host Name	STRING	-	Host name

Ethernet Link Object, Class F6h

Services

Class services: Get Attribute All
Get Attribute Single

Instance services: Get Attribute All
Get Attribute Single

Description

This object maintains link specific counters and status information for the Ethernet communications interface.

Class Attributes

#	Access	Name	Type	Value	Description
1	Get	Revision	UINT	0001h	Revision 1
2	Get	Max Instance	UINT	0001h	1 is the highest instance number
3	Get	No. of instances	UINT	0001h	1 instance is implemented

Instance Attributes

#	Access	Name	Type	Value	Description
1	Get	Interface Speed	UDINT	10 or 100 (MBPS)	The actual speed of the module
2	Get	Interface Flags	DWORD	-	-
3	Get	Physical Address	Array of 6 USINTS	MAC address	Ethernet MAC address of the module

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 Design Guide for more information.)

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 type 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

Network Configuration Commands

Commands in this category:

Mailbox Commands	Description	Page
Set Ethernet Configuration (SET_ETN_CONFIG)	Set network settings	12-3
Read Ethernet Configuration (READ_ENT_CONFIG)	Get the currently used network settings	12-4
Get MAC Address (GET_MAC_ADDR)	Get the ethernet MAC address of the module	12-5
Connection Timeout (CONNECTION_TIMEOUT)	Set the connection timeout value	12-6
Set Host and Domain (SET_HOST_DOMAIN)	Set host and default domain names	12-7
Get Host and Domain (GET_HOST_DOMAIN)	Returns the currently used host and domain	12-8
Disable / Enable ARP Config (ARP_CFG_CONTROL)	Disables/enables ARP support	12-9
Disable / Enable HICP (HICP_CFG_CONTROL)	Disables/enables HICP support	12-10
Disable DHCP (DISABLE_DHCP)	Disable DHCP support	12-11

Set Ethernet Configuration (SET_ETN_CONFIG)

Description

To be able to communicate on the network the IP address, the Gateway address, and Subnet mask, must be configured. This mailbox command can be used during module initialization to configure these settings.

If this command is not sent during initialization, the module will try to get the settings some other way, see 4-1 "Network Configuration".

Note: These settings will not be stored in the configuration file 'ethcfg.cfg'.

Parameter	Description
Command initiator	Application
Command number	0001h
Fragmented	No
Extended Header data	-
Message data	Contains IP address, Subnet mask, and Gateway address.
Response message	The response indicates if the command was accepted. 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	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 mailbox command is used for reading the current network settings. This command cannot be sent during module initialization.

Parameter	Description
Command initiator	Application
Command number	0002h
Fragmented	No
Extended Header data	-
Message data	-
Response message	The response indicate if the command was accepted. The response data shows the used Ethernet 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.

Parameter	Description
Command initiator	Application
Command number	0010h
Fragmented	No
Extended Header data	-
Message data	-
Response message	MAC Address of the module, 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	-	Fault information	
		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. Valid timeout values range from 1 to 255 seconds. This command can only be sent during module initialization. The default value is 75 seconds.

Parameter	Description
Command initiator	Application
Command number	0003h
Fragmented	No
Extended Header data	-
Message data	Contains the timeout value
Response message	The response indicates if the command was accepted. 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	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	

Set Host and Domain (SET_HOST_DOMAIN)

Description

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

Note: This command can only be sent during module initialisation.

Parameter	Description
Command initiator	Application
Command number	0032h
Fragmented	No
Extended Header data	-
Message data	Hostname and Domain name, NULL terminated
Response message	The response indicates if the command was accepted. 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	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 + NULL	Host name + NULL	
	Domain name + NULL	Domain name + NULL	

Get Host and Domain (GET_HOST_DOMAIN)

Description

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

Parameter	Description
Command initiator	Application
Command number	0034h
Fragmented	No
Extended Header data	-
Message data	-
Response message	The response indicates if the command was accepted. The response contains the host name and domain name as NULL terminated strings.

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 + NULL	**Response data**
		Domain name + NULL	

Disable / Enable ARP Config (ARP_CFG_CONTROL)

Description

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

Parameter	Description
Command initiator	Application
Command number	0014h
Fragmented	No
Extended Header data	-
Message data	0: Disable ARP, 1: Enable ARP
Response message	The response indicates if the command was accepted. 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	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 the HICP functionality.

HICP is an acronym for 'HMS IP Configuration Protocol', and will be used by a future Windows-based application that will be able to detect HMS modules on the network and configure their IP settings. Since the protocol is based on broadcast messages, it will be possible to detect and configure modules that are outside of the host's subnet. Please note that the required software is not yet available, and that this feature should be disabled if this functionality is not desired.

Parameter	Description
Command initiator	Application
Command number	0013h
Fragmented	No
Extended Header data	-
Message data	0: Disable HICP, 1: Enable HICP
Response message	The response indicates if the command was accepted. 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	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 the DHCP support. If this command is sent to the module it will not be capable of receiving its configuration from a DHCP or BOOTP server. This command overrides the DHCP setting in the configuration file 'ethcfg.cfg'. Also, the DHCP configuration checkbox on the default configuration web page will be excluded and the module will not report its DHCP status via HICP.

Note: This command can only be sent during module initialisation.

Parameter	Description
Command initiator	Application
Command number	0015h
Fragmented	No
Extended Header data	-
Message data	-
Response message	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	

Server Settings

Commands in this category:

Mailbox Commands	Description	Page
Set DNS Server Settings (SET_DNS_SERVERS)	Specifies the addresses of the DNS servers	12-13
Get DNS Server Settings (GET_DNS_SERVERS)	Returns the configured DNS server settings	12-14
Set SMTP Server (SET_SMTP_SERVER)	Configures the SMTP server IP address	12-15
Set SMTP Server by Name (SET_SMPT_SERVER_NAME)	Configures the SMTP server address	12-16
Get SMTP Server (GET_SMTP_SERVER)	Returns the currently used SMTP server IP address	12-17
Disable Web Server (DISABLE_WEB_SERVER)	This command disables the onboard web server	12-18
Enable Web Server (ENABLE_WEB_SERVER)	This command enabless the onboard wen server	12-19
Disable FTP Server (DISABLE_FTP_SERVER)	This command disables the onboard FTP server	12-20
Disable Telnet Server (DISABLE_TELNET_SERVER)	This command disables the onboard telnet server	12-21

Set DNS Server Settings (SET_DNS_SERVERS)

Description

This command is used to specify the address of the primary and secondary DNS servers. (If only a primare DNS is required, set the secondary DNS address to 0.0.0.0.)

Note: This command can only be sent during module initialisation.

Parameter	Description
Command initiator	Application
Command number	0031h
Fragmented	No
Extended Header data	-
Message data	Primary and secondary DNS server address
Response message	The response indicates if the command was accepted. 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	0031h	0031h	*SET_DNS_SERVERS*
Data size	0008h	0008h	*8 bytes of data (4 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	primary DNS address (hi)	primary DNS address (hi)	
Message dataword 2	primary DNS address (lo)	primary DNS address (lo)	
Message dataword 3	secondary DNS address(hi)	secondary DNS address (hi)	
Message dataword 4	secondary DNS address (lo)	secondary DNS address (lo)	

Get DNS Server Settings (GET_DNS_SERVERS)

Description

This function returns the configured DNS server settings. If a DNS server is not configured, it's IP address will be 0.

Parameter	Description
Command initiator	Application
Command number	0033h
Fragmented	No
Extended Header data	-
Message data	-
Response message	The response indicates if the command was accepted. The response data contains the addresses of the primary and secondary DNS servers.

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	*8 bytes of data (4 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	
		primary DNS address (hi)	**Response dataword 1**
		primary DNS address (lo)	**Response dataword 2**
		secondary DNS address (hi)	**Response dataword 3**
		secondary DNS address (lo)	**Response dataword 4**

Set SMTP Server (SET_SMTP_SERVER)

Description

This mailbox command configures the SMTP server IP address.

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

Note 2: This command can only be sent during module initialisation.

Parameter	Description
Command initiator	Application
Command number	000Eh
Fragmented	No
Extended Header data	-
Message data	Contains SMTP server's IP address.
Response message	The response indicate if the command was accepted. The response 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	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 mailbox command configures the SMTP server address by name. It is also accepts an IP address in ascii form.

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

Note 2: This command can only be sent during module initialisation.

Parameter	Description
Command initiator	Application
Command number	0016h
Fragmented	No
Extended Header data	-
Message data	String containing the SMTP server address.
Response message	The response indicate if the command was accepted. 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	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 name, NULL terminated	SMTP server name, NULL terminated	

Get SMTP Server (GET_SMTP_SERVER)

Description

This function returns the currently used SMTP server IP address.

Parameter	Description
Command initiator	Application
Command number	000Fh
Fragmented	No
Extended Header data	-
Message data	-
Response message	Contains SMTP server's 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.

Parameter	Description
Command initiator	Application
Command number	0004h
Fragmented	No
Extended Header data	-
Message data	-
Response message	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	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.

Parameter	Description
Command initiator	Application
Command number	0005h
Fragmented	No
Extended Header data	-
Message data	-
Response message	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	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 onboard FTP server.

Parameter	Description
Command initiator	Application
Command number	0006h
Fragmented	No
Extended Header data	-
Message data	-
Response message	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	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.

Parameter	Description
Command initiator	Application
Command number	0008h
Fragmented	No
Extended Header data	-
Message data	-
Response message	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	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	-	-	

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	12-23
Disable Modbus/TCP Server (DISABLE_MB_TCP)	Disables the Modbus/TCP server	12-24
Modbus ON/OFF Line Configuration (MB_ON_OFF_LINE_CONFIG)	Configures the Modbus ON/OFF line setting	12-25
Write Exception Status (WR_EXP_STAT)	Defines the response to the Modbus command 'Read Exception Status'	12-26

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 module.

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 can only be sent during module initialization.

Parameter	Description
Command initiator	Application
Command number	0020h
Fragmented	No
Extended Header data	-
Message data	Contains the timeout value for the Modbus/TCP connections.
Response message	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	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 can only be sent during module initialization.

Parameter	Description
Command initiator	Application
Command number	0021h
Fragmented	No
Extended Header data	-
Message data	-
Response message	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	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.

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

Parameter	Description
Command initiator	Application
Command number	0023h
Fragmented	No
Extended Header data	Time out value. A value of 1 equals 100ms. A value of 0 configures the ON/OFF line functionality to be triggered by the Link Status.
Message data	List of triggering modbus commands. Data size = 0 = trigger for all commands. The trigger list can be a maximum of 11 bytes long.
Response message	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	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	

- **Trigger Commands**

  Trigger Command = Modbus function code to trigger from

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 module.

Parameter	Description
Command initiator	Application
Command number	0022h
Fragmented	No
Extended Header data	-
Message data	Exception status byte
Response message	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	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 byte	Exception status byte	

EtherNet/IP Specific Commands

EtherNet/IP share the same object model as ControlNet and DeviceNet. Because of this it would have been preferable to use the same mailbox commands as in the AnyBus modules for these fieldbus systems.

However, compared to the AnyBus-S DeviceNet and ControlNet, the mailbox command numbers for this module has change slightly in order to keep the module backwards compatible with the old ABS-ETN-1.

In an effort to keep the mailbox commands as similar as possible to the ones in the DeviceNet and ControlNet modules, all mailbox commands that are applicable have been adopted. The only change that has been introduced, is that bit 7 in the command number has been set to one.

Commands in this category:

Mailbox Command	Description	Page
Set product info all (PRODUCT_INFO_ALL)	Changes all default EtherNet/IP specific product info into customer specific.	12-28
Set product info (SET_PRODUCT_INFO)	Sets the EtherNet/IP specific product info	12-29
Set product code (SET_PRODUCT_CODE)	Changes the default product code into a customer specific	12-30
Parameter data input mapping (PARAMETER_INPUT_MAP)	Maps parameter input data into the parameter input object	12-31
Parameter data output mapping (PARAMETER_OUTPUT_MAP)	Maps parameter output data into the parameter input object	12-33
I/O data input area mapping (I/O_INPUT_MAP)	Maps I/O input data in the DPRAM into the assembly object	12-35
I/O data output area mapping (I/O_OUTPUT_MAP)	Maps I/O output data in the DPRAM into the assembly object	12-37
Send UCMM (SEND_UCMM)	Sends an unconnected message to another EthernNet/IP module	12-39
UCMM request (UCMM_REQUEST)	Generated by the module when an UCMM request is addressed to a user-registered object	12-41
Register class (REGISTER_CLASS)	Registers an EtherNet/IP object in the message router object	12-42
Deregister class (DEREGISTER_CLASS)	Deregisters a class that is already registered within the message router	12-43
Enable routing (ENABLE_ROUTING)	Enables the routing and bridging functionality	12-44
Register port (REGISTER_PORT)	Registers a port in the port object	12-45
Route Unconnected Send (ROUTE_REQUEST)	Generated by the module when it receives an unconnected send message.	12-48
Enable Reset Notification (ENABLE_ID_RESET_NOTIFY)	Enables Identity Object reset notification	12-50
Reset Notification (ID_RESET_NOTIFY)	Reset notification	12-51
Get Reset Parameter (GET_ID_RESET_PARAM)	Get reset parameter	12-52
Copy I/O Status (COPY_IO_STATUS)	Copy I/O Status bytes	12-53
Disable EtherNet/IP (DISABLE_ETHERNET_IP)	Disables the EtherNet/IP protocol support	12-54

Set Product Info All (PRODUCT_INFO_ALL)

Description

This command is used to change Vendor ID, Device Type, Product Code, Major Revision, Minor Revision, and Product Name in the Identity object to customize the developed product. Note that the EDS-file must be modified accordingly.

Contact HMS for more information about obtaining a Product Code.

Note: This command can only be sent during module initialisation.

Parameter	Description
Command initiator	Application
Command number	0089h
Fragmented	No
Extended Header data	-
Message data	Vendor ID, Device type, Product code, Major revision, Minor revision, Name length, Name (ascii values)
Response message	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	0089h	0089h	*PRODUCT_INFO_ALL*
Data size	(size)	(size)	*Message data 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 byte 1	Vendor ID (high byte)	Vendor ID (high byte)	*Vendor ID*
Message data byte 2	Vendor ID (low byte)	Vendor ID (low byte)	
Message data byte 3	Device type (high byte)	Device type (high byte)	*Device Type*
Message data byte 4	Device type (low byte)	Device type (low byte)	
Message data byte 5	Product code (high byte)	Product code (high byte)	*Product Code*
Message data byte 6	Product code (low byte)	Product code (low byte)	
Message data byte 8	Major revision	Major revision	*Major revision*
Message data byte 9	Minor revision	Minor revision	*Minor revision*
Message data byte 10	Product name length	Product name length	*Length of prod. name*
Message data byte 11	Product name 1st character	Product name 1st character	*Product name 1st character*
...	...	...	*...*
Message data byte n	Product name last character	Product name last character	*Last character*

Set Product Info (SET_PRODUCT_INFO)

Description

This mailbox command is used to customize the Vendor ID, Product Code and Product Name in the Identity object. The EDS file needs to be adjusted accordingly.

Contact HMS for more information about obtaining a Product Code.

Note: This command can only be sent during module initialisation.

Parameter	Description
Command initiator	Application
Command number	0082h
Fragmented	No
Extended Header data	-
Message data	Vendor ID, Product code, Name length, Name
Response message	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	0082h	0082h	*SET_PRODUCT_INFO*
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	Vendor ID (high byte)	Vendor ID (high byte)	*Vendor ID*
Message data byte 2	Vendor ID (low byte)	Vendor ID (low byte)	"
Message data byte 3	Product code (high byte)	Product code (high byte)	*Product code*
Message data byte 4	Product code (low byte)	Product code (low byte)	"
Message data byte 5	Product name length	Product name length	*Product name, max 32 char.*
Message data byte 6	Product name 1st character	Product name 1st character	"
Message data byte 7	Product name 2st character	Product name 2st character	"
...	...	...	"
Message data byte n	Product name last character	Product name last character	"

Set Product Code (SET_PRODUCT_CODE)

Description

This mailbox command is used to customize the Product Code of the module. This enables a configuration tool to identify the product as a special implementation instead of a general AnyBus-S module. The EDS file needs to be adjusted accordingly.

Contact HMS for more information about obtaining a Product Code.

Note: This command can only be sent during module initialisation.

Parameter	Description
Command initiator	Application
Command number	0083h
Fragmented	No
Extended Header data	-
Message data	Unique number provided by HMS
Response message	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	0083h	0083h	*SET_PRODUCT_CODE*
Data size	0002h	0002h	*2 bytes of data (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	Product Code	Product Code	

Parameter Data Input Mapping (PARAMETER_INPUT_MAP)

Description

This mailbox command is used to set up blocks of data in the input data area to be used with the Class B0h "Parameter data input mapping object" (See 11-8 "Parameter Data Input Mapping Object, Class B0h"). This way, a 'Get_Atrribute_Single' command from the fieldbus can return a specified block of data from the input parameter area. It is possible to map up to 50 attributes.

Attributes are mapped beginning with attribute 1, followed by attributes 2 through 50. Offset in the input parameter area and number of bytes to map is specified for each attribute.

If zero length is specified, the attribute will not be mapped. This way, it is for example possible to map only attributes 1 and 10 by specifying zero length for attributes 2 through 9. It is only necessary to include information in the telegram up to the last used attributes; the remaining attributes will not be mapped.

If any offset or length is invalid, the length and offset will be set to zero in the response and the attribute will not be mapped.

To access the parameters from EtherNet/IP use Class attribute B0h, Instance attribute 01h and Attribute ID 01h through 32h. (See 11-8 "Parameter Data Input Mapping Object, Class B0h")

Note: This command can only be sent after the ANYBUS_INIT command, during module initialisation.

Parameter	Description
Command initiator	Application
Command number	0084h
Fragmented	No
Extended Header data	-
Message data	Offset and length of the attributes to map
Response message	The response indicates if the command was accepted.

Command and response layout (example when only setting attribute 1-5)

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0084h	0084h	*PARAMETER_INPUT_MAP*
Data size	0014h	0014h	*20 bytes of data (10 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 data word 1	Offset	Offset	*Offset*
Message data word 2	Length	Length	*Number of bytes to map*
Message data word 3	Offset	Offset	*Attribute 2*
Message data word 4	Length	Length	
Message data word 5	Offset	Offset	*Attribute 3*
Message data word 6	Length	Length	
Message data word 7	Offset	Offset	*Attribute 4*
Message data word 8	Length	Length	
Message data word 9	Offset	Offset	*Attribute 5*
Message data word 10	Length	Length	

Parameter Data Out Area Mapping (PARAMETER_OUTPUT_MAP)

Description

This mailbox command is used to set up blocks of data in the output data area to be used with the Class B1h "Parameter data output mapping object" (See 11-9 "Parameter Data Output Mapping Object, Class B1h"). This way, a Get_Attribute_Single or Set_Attribute_Single command from the fieldbus can return/write a specified block of data from/to the output parameter area. It is possible to map up to 50 attributes.

Attributes are mapped beginning with attribute 1, followed by attributes 2 through 50. Offset in the input parameter area and number of bytes to map is specified for each attribute.

If zero length is specified, the attribute will not be mapped. This way, it is for example possible to map only attributes 1 and 10 by specifying zero length for attributes 2 through 9. It is only necessary to include information in the telegram up to the last used attributes; the remaining attributes will not be mapped.

If any offset or length is invalid, the length and offset will be set to zero in the response and the attribute will not be mapped.

To access the parameters from EtherNet/IP use Class attribute B1h, Instance attribute 01h and Attribute ID no. (See 11-9 "Parameter Data Output Mapping Object, Class B1h")

Note: This command can only be sent after the ANYBUS_INIT command, during module initialisation.

Parameter	Description
Command initiator	Application
Command number	0085h
Fragmented	No
Extended Header data	-
Message data	Offset and length of the attributes to map
Response message	The response indicates if the command was accepted.

Command and response layout (example when only setting attribute 1-5)

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0085h	0085h	*PARAMETER_OUTPUT_MAP*
Data size	0014h	0014h	*20 bytes of data (10 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 data word 1	Offset	Offset	*Offset*
Message data word 2	Length	Length	*Number of bytes to map*
Message data word 3	Offset	Offset	*Attribute 2*
Message data word 4	Length	Length	
Message data word 5	Offset	Offset	*Attribute 3*
Message data word 6	Length	Length	
Message data word 7	Offset	Offset	*Attribute 4*
Message data word 8	Length	Length	
Message data word 9	Offset	Offset	*Attribute 5*
Message data word 10	Length	Length	

I/O Data Input Area Mapping (I/O_INPUT_MAP)

Description

This mailbox command is used to set up blocks of I/O data in the output area using the Assembly Object (See 11-4 "Assembly Object, Class 04h"). This makes it is possible to select which area in the input area to use when writing I/O data from the application. Attributes are mapped with start at instance 64h. Offset is set from the start of the input data area, and length specifies the number of bytes to map.

If zero length is specified, the instance will not be mapped. This way, it is for example possible to map only instances 64h and 68h by specifying zero length for instances 65h through 67h. It is only necessary to include information in the telegram up to the last used instance number; the remaining instances will not be mapped.

If any offset or length is invalid, the length and offset will be set to zero in the response and the attribute will not be mapped.

To access the parameters from EtherNet/IP use Get_Atribute_Single and Class attribute 04h, Instance attributes 64h through 69h, and Attribute ID 03h or use Class attribute A0h, Instance attribute 04h and Attribute ID 01h through 06h. (See 11-4 "Assembly Object, Class 04h")

Note: This command can only be sent during module initialisation., after ANYBUS_INIT.

Parameter	Description
Command initiator	Application
Command number	0086h
Fragmented	No
Extended Header data	-
Message data	Offset and length of the instances to map
Response message	The response indicates if the command was accepted.

Command and response layout (example when setting input instances 64h-68h)

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0086h	0086h	*I/O_INPUT_MAP*
Data size	0014h	0014h	*20 bytes of data (10 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 data word 1	Offset	Offset	*Offset*
Message data word 2	Length	Length	*Number of bytes to map*
Message data word 3	Offset	Offset	*Instance 65h*
Message data word 4	Length	Length	
Message data word 5	Offset	Offset	*Instance 66h*
Message data word 6	Length	Length	
Message data word 7	Offset	Offset	*Instance 67h*
Message data word 8	Length	Length	
Message data word 9	Offset	Offset	*Instance 68h*
Message data word 10	Length	Length	

I/O Data Output Area Mapping (I/O_OUTPUT_MAP)

Description

This mailbox command is used to set up blocks of I/O data in the output area using the Assembly Object (See 11-4 "Assembly Object, Class 04h"). This makes it is possible to select which area in the output area to read data from EtherNet/IP. Attributes are mapped with start at instance 96h. Offset is set from the start of the output data area, and length specifies the number of bytes to map.

If zero length is specified, the instance will not be mapped. This way, it is for example possible to map only instances 96h and 9Bh by specifying zero length for instances 97h through 9Ah. It is only necessary to include information in the telegram up to the last used instance number; the remaining instances will not be mapped.

If any offset or length is invalid, the length and offset will be set to zero in the response and the attribute will not be mapped.

To access the parameters from EtherNet/IP, use Get_Atribute_Single and Class attribute 04h, Instance attributes 96h through 9Bh, and Attribute ID 03h or use Class attribute A1h, Instance attribute 01h and Attribute ID 01h through 06h. (See 11-4 "Assembly Object, Class 04h")

Note: This command can only be sent during module initialisation., after ANYBUS_INIT.

Parameter	Description
Command initiator	Application
Command number	0087h
Fragmented	No
Extended Header data	-
Message data	Offset and length of the instances to map
Response message	The response indicates if the command was accepted.

Command and response layout (example when setting input instances 96h-9Ah)

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0087h	0087h	*I/O_OUTPUT_MAP*
Data size	0014h	0014h	*20 bytes of data (10 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 data word 1	Offset	Offset	*Offset*
Message data word 2	Length	Length	*Number of bytes to map*
Message data word 3	Offset	Offset	*Instance 97h*
Message data word 4	Length	Length	
Message data word 5	Offset	Offset	*Instance 98h*
Message data word 6	Length	Length	
Message data word 7	Offset	Offset	*Instance 99h*
Message data word 8	Length	Length	
Message data word 9	Offset	Offset	*Instance 9Ah*
Message data word 10	Length	Length	

Send UCMM (SEND_UCMM)

Description

This mailbox command is used to send an explicit unconnected message from the application directly to a node in the network. The format of the message is the message router / request format. (For more information, consult the EtherNet/IP Specification volume 1, section 2-4). This command will not respond until the response from the remote host is received or a timeout has occurred.

Parameter	Description
Command initiator	Application
Command number	008Ah
Fragmented	No
Extended Header data	-
Message data	Destination host IP address and unconnected message.
Response message	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	008Ah	008Ah	*SEND_UCMM*
Data size	(data size)	(data size)	*Size of data*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	Destination IP high word	Destination IP high word	*Destination IP high word*
Extended word 2	Destination IP low word	Destination IP low word	*Destination IP low word*
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault Information	
Message data byte 1	Service Request	(Service Dependant)	
Message data byte 2	Request Path Size (in words)		
Message data byte 3	Padded EPATH[a]		
...			
...	(optional service data)		
Message data byte n			

a. See EtherNet/IP Specification, Appendix C - 'Data Management'

The following example (see next page) uses the SEND_UCMM command to retrieve data from a node on the network. The remote node is an Allen Bradley ControlLogix5000 with a 1756-ENBT/A Ethernet/IP module. It has VendorID 0001h, Product Type 000C, Product Code 003Ah, Version 1.33 and Serial Number 00121E63h. The request that is sent to the remote node is 'Get_Attribute_All' (0x01) to Class 0x01 and Instance 0x01. This is the identity object, see 5-2.2 in the EtherNet/IP specification for more information about the response.

Send UCMM Example

	Command	Expected response	
Message ID	ID	ID	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	008Ah	008Ah	*SEND_UCMM*
Data size	0006h	001Eh	*Size of data*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	0A0Ah	0A0Ah	*Destination IP (high word)*
Extended word 2	0E50h	0E50h	*Destination IP (low word)*
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault Information	
Data byte 1 (Service Request)	01h (Get_Attribute_All)	81h	*Get_Attiribute_All reply*
Data byte 2 (Request Path Length)	02h (2 words)	00h	*(reserved)*
Data byte 3 (Segment Type)	20h[a]	00h	*General Status*
Data byte 4 (Segment Data)	01h (Class #1)	00h	*Additional Status*
Data byte 5 (Segment Type)	24h[b]	01h	*Vendor ID (LSB)*
Data byte 6 (Segment Data)	01h (Instance #1)	00h	*Vendor ID (MSB)*
		0Ch	*Product Type (LSB)*
		00h	*Product Type (MSB)*
		3Ah	*Product Code (LSB)*
		00h	*Product Code (MSB)*
		01h	*Version (Major)*
		21h	*Version (Minor)*
		30h	*Status (LSB)*
		00h	*Status (MSB)*
		63h	*Serial no. (LSB)*
		1Eh	*Serial no.*
		12h	*Serial no.*
		00h	*Serial no. (MSB)*
		0Bh (11)	*Product Name Length*
		31h (‘1’)	*Product Name Char #1*
		37h (‘7’)	*Product Name Char #2*
		35h (‘5’)	*Product Name Char #3*
		36h (‘6’)	*Product Name Char #4*
		2Dh (‘-’)	*Product Name Char #5*
		45h (‘E’)	*Product Name Char #6*
		4Eh (‘N’)	*Product Name Char #7*
		42h (‘B’)	*Product Name Char #8*
		54h (T1’)	*Product Name Char #9*
		2Fh (‘/’)	*Product Name Char #10*
		41h (‘A’)	*Product Name Char #11*

a. Segment Type = Logical Segment
Logical Type = Class ID
Logical Format = 8bit Logical Address

b. Segment Type = Logical Segment
Logical Type = Instance ID
Logical Format = 8bit Logical Address.

UCMM Request (UCMM_REQUEST)

Description

This message is used when the application has registered an EtherNet/IP class (See 12-42 "Register Class (REGISTER_CLASS)") in the module, and an explicit message request has been generated to this class from a node in the EtherNet/IP network.

The format of the message is the message router / request format. (See EtherNet/IP Specification volume 1 section 2-4). The application will have to process the message, and respond to the module with the data necessary to generate a response on the explicit message request for the object.

Parameter	Description
Command initiator	AnyBus
Command number	008Dh
Fragmented	No
Extended Header data	-
Message data	Explicit message data
Response message	The response is the data that the explicit request message asks for (or error code). The response 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	008Dh	008Dh	*UCMM_REQUEST*
Data size	(size)	(size)	*Data 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	Service Request	Reply Service	
Message data byte 2	Request Path Size (in words)	Reserved (00h)	
Message data byte 3	Padded EPATH[a]	General Status	
...		Size of Additional Status	
...	(optional service data)	Additional Status	
Message data byte n		Response_data	

a. Request path according to the EtherNet/IP Specification, Appendix C - 'Data Management'

Register Class (REGISTER_CLASS)

Description

This mailbox command makes it possible for the application to register objects inside the message router object. If there is a node on the network that sends an explicit message request to the module, address to the registered class, the explicit message will generate an explicit message request telegram, which will be sent from the module to the application. See 12-41 "UCMM Request (UCMM_REQUEST)".

Parameter	Description
Command initiator	Application
Command number	008Bh
Fragmented	No
Extended Header data	-
Message data	Contains the class ID of the class to be registered in the module.
Response message	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	008Bh	008Bh	*REGISTER_CLASS*
Data size	0002h	0002h	*2 bytes of data (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	-	Fault information	
Message data byte 1	Class ID (high byte)	Class ID (high byte)	
Message data byte 2	Class ID (low byte)	Class ID (low byte)	

Deregister Class (DEREGISTER_CLASS)

Description

This mailbox command makes it possible for the application to deregister objects inside the message router object.

The following classes cannot be deregistered with this command;

- Class 02h - Message Router
- Class 04h - Assembly Object

Parameter	Description
Command initiator	Application
Command number	008Eh
Fragmented	No
Extended Header data	-
Message data	Contains the class ID of the class to be deregistered in the module.
Response message	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	008Eh	008Eh	*DEREGISTER_CLASS*
Data size	0002h	0002h	*2 bytes of data (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	-	Fault information	
Message data byte 1	Class ID (high byte)	Class ID (high byte)	
Message data byte 2	Class ID (low byte)	Class ID (low byte)	

Enable Routing (ENABLE_ROUTING)

Description

This mailbox command enables the routing functionality of the module.

Note: This command can only be sent during module initialization.

Parameter	Description
Command initiator	Application
Command number	0091h
Fragmented	No
Extended Header data	-
Message data	-
Response message	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	0091h	0091h	*ENABLE_ROUTING*
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	-	-	

Register Port (REGISTER_PORT)

Description

This command is used to register a port in the Port Object (See EtherNet/IP specification vol. 1 chapters 3 - 7). This must be done for each port in the application if routing is enabled. (See 12-44 "Enable Routing (ENABLE_ROUTING)"). The message data shall contain the instance attributes 1, 2, 3, 4 and 7, in that order. The class attributes will be updated automatically after each received mailbox command. Port 2 is reserved for the AnyBus-S module, and it is not possible to register the same port twice.

Note: This command can only be sent during module initialization.

Parameter	Description
Command initiator	Application
Command number	0090h
Fragmented	No
Extended Header data	-
Message data	Instance attributes 1,2,3,4 and 7.
Response message	The response indicates if the command was accepted. 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	0090h	0090h	*REGISTER_PORT*
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	-	Fault Information	
Data byte 1	Port Type (high)	Port Type (high)	
Data byte 2	Port Type (low)	Port Type (low)	
Data byte 3	Port Number (high)	Port Number (high)	
Data byte 4	Port Number (low)	Port Number (low)	
Data byte 5	Port Object Size (high)	Port Object Size (high)	
Data byte 6	Port Object Size (low)	Port Object Size (low)	
Data byte 7 **...** **Data byte n**	Port Object EPATH (Padded)[a]	Port Object EPATH (Padded)[a]	
Data byte n+1	Port Name Length	Port Name Length	
Data byte n+2	Port Name Char #1	Port Name Char #1	
Data byte n+3	Port Name Char #2	Port Name Char #2	
...	...	...	*(Continued on next page)*

Data byte z	Port Name Char #n	Port Name Char #n
Data byte z+1	Node address	Node address
Data byte z+2	Node address	Node address

a. See EtherNet/IP Specification, Appendix C - 'Data Management'

Register Port Example

The following example registers a ControlNet redundant port (3) with port number 3. The port object points to class F0h (ControlNet object) instance 01h. The name of the port is "ControlNet", and the node address is 8 on port 3.

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0090h	0090h	*REGISTER_PORT*
Data size	0017h	0017h	*23 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	-	Fault Information	
Data byte 1	00h	00h	*Port Type (high)*
Data byte 2	03h	03h	*Port Type (low)*
Data byte 3	00h	00h	*Port Number (high)*
Data byte 4	03h	03h	*Port Number (low)*
Data byte 5	00h	00h	*Port Object Size (high)*
Data byte 6	02h	02h	*Port Object Size (low)*
Data byte 7	20h	20h	*Port Object EPATH*
Data byte 8	F0h	F0h	*Port Object EPATH*
Data byte 9	24h	24h	*Port Object EPATH*
Data byte 10	01h	01h	*Port Object EPATH*
Data byte 11	0Ah	0Ah	*Port Name Length*
Data byte 12	43h	43h	*Port name: "C"*
Data byte 13	6Fh	6Fh	*Port name: "o"*
Data byte 14	6Eh	6Eh	*Port name: "n"*
Data byte 15	74h	74h	*Port name: "t"*
Data byte 16	72h	72h	*Port name: "r"*
Data byte 17	6Fh	6Fh	*Port name: "o"*
Data byte 18	6Ch	6Ch	*Port name: "l"*
Data byte 19	4Eh	4Eh	*Port name: "N"*
Data byte 20	65h	65h	*Port name: "e"*
Data byte 21	74h	74h	*Port name: "t"*
Data byte 22	03h	03h	*Port to leave node = 3*
Data byte 23	08h	08h	*On ControlNet = 8*

Continued on next page...

The [Port] section in the .EDS file should look like this to fit the example above:

```
[Port]
   Port1 =
      TCP,                    $ Port type
      "TCP/IP",               $ Port name
      "20 F5 24 01",          $ Path to object supporting this port
      2;                      $ Port number

   Port2 =
      ControlNet,             $ Port type
      "ControlNet",           $ Port name
      "20 F0 24 01",          $ Path to object supporting this port
      3;                      $ Port number
```

Route Unconnected Send (ROUTE_REQUEST)

Description

This mailbox message is generated by the module when it receives a valid unconnected send message, i.e. a message addressed to a port registered by the application. The message data contains the whole unconnected send message (See EtherNet/IP spec. Vol. 1 3-5.5.4). The response message from the application shall contain either a successful or unsuccessful unconnected send response (See EtherNet/IP spec. Vol. 1 3-5.5.4).

If 16 or more message requests are waiting to be processed by the application, the module will answer with a "No resource" error code for all new requests until there are less than 16 unprocessed requests.

Note: This function requires routing to be enabled, see 12-44 "Enable Routing (ENABLE_ROUTING)".

Parameter	Description
Command initiator	AnyBus
Command number	008Fh
Fragmented	No
Extended Header data	-
Message data	The unconnected send message received by the module
Response message	The application shall respond with the data returned by the target device.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	008Fh	008Fh	*ROUTE_REQUEST*
Data size	(size)	(size)	*Datasize*
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	Transaction ID (low)		
Message data byte 2	Transaction ID (high)	*Successful response:*	
	Priority / Time tick	Transaction ID (low)	**Response data byte 1**
	Time-out ticks	Transaction ID (high)	**Response data byte 2**
	Msg. req. size (low)	General Status = 00h	
	Msg. req. size (high)	00h (Reserved)	
Message Request	Service Code	Service Response Data	**Response data byte n**
	Request Path Size		
	Req. Path (Padded EPATH)		
	Request Data	*Unsuccessful response:*	
		Transaction ID (low)	**Response data byte 1**
		Transaction ID (high)	**Response data byte 2**
	00h (PAD) (Only if Msg.req.size is odd)	General Status = 00h	
		Size of additional status	
	Route Path Size	Additional status	
	00h (reserved)		
Message databyte n	Route Path	Remaining path size	**Response data byte n**

Enable Reset Notification (ENABLE_ID_RESET_NOTIFY)

Description

This command enables reset notification. When reset notification is enabled and the identity object receives a valid reset request, the module will send an ID_RESET_NOTIFY to the application. (See 12-51 "Reset Notification (ID_RESET_NOTIFY)")

It is possible to change how the module should react when the identity object receives a valid reset request, by altering the Notify Type Value (See below)

Note: This command can only be sent during module initialisation.

Parameter	Description
Command initiator	Application
Command number	0092h
Fragmented	No
Extended Header data	-
Message data	Notification type
Response message	The response indicates if the data was accepted.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0092h	0092h	*ENABLE_ID_RESET_NOTIFY*
Data size	0001h	0001h	*1 byte 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 Databyte	Notify type	Notify type	*(See below)*

- **Notify Type Value**

  00h: When the identity object receives a valid reset request, the module sends an ID_RESET_NOTIFY to the application.

  01h: When the identity object receives a valid reset request, the configuration file is erased, and the module sends an ID_RESET_NOTIFY to the application.

Reset Notification (ID_RESET_NOTIFY)

Description

If reset notification is enabled (see 12-50 "Enable Reset Notification (ENABLE_ID_RESET_NOTIFY)"), this mailbox message is sent to the applications when the identity object receives a valid reset request.

The message data contains the type of reset received, see ENIP spec. 5-2.3.1 Reset Service.

Parameter	Description
Command initiator	AnyBus
Command number	0093h
Fragmented	No
Extended Header data	-
Message data	Type of reset
Response message	The response indicates if the data was accepted.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0093h	0093h	*ID_RESET_NOTIFY*
Data size	0001h	0001h	*1 byte 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 Databyte	Reset Type	Reset Type	*(See below)*

- **Reset Type Value**

  00h: Power on reset

  01h: Out of box configuration

Get Reset Parameter (GET_ID_RESET_PARAM)

Description

Using this command, it is possible to determine what type of reset that was received via EtherNet/IP

Parameter	Description
Command initiator	Application
Command number	0095h
Fragmented	No
Extended Header data	-
Message data	-
Response message	The response indicates if the data was accepted. The response contains the type of reset.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0095h	0095h	*GET_ID_RESET_PARAM*
Data size	0000h	0001h	
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	-	-	
		Reset Type	**Response Databyte**

- **Reset Type Value**

  00h: Power on reset

  01h: Out of box configuration

Copy I/O Status (COPY_IO_STATUS)

Description

When this message is sent to the module, the first four bytes in the connection a.k.a. Run/Idle header is not stripped of the data but is passed to the DPRAM memory in front of the actual I/O data.

Parameter	Description
Command initiator	Application
Command number	0094h
Fragmented	No
Extended Header data	-
Message data	-
Response message	The response indicates if the data was accepted.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0094h	0094h	*COPY_IO_STATUS*
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 EtherNet/IP (DISABLE_ETHERNET_IP)

Description

This mailbox command disables support for the EtherNet/IP protocol.

Note: This command can only be sent during module initialization.

Parameter	Description
Command initiator	Application
Command number	000Ah
Fragmented	No
Extended Header data	-
Message data	-
Response message	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	000Ah	000Ah	*DISABLE_ETHERNET_IP*
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	-	Fault information	

Mailbox Socket Interface

The 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.

  Status information for all non-blocking sockets are available in the fieldbus specific area, see 12-1 "Fieldbus Specific Area". The module can handle up to 16 non-blocking sockets simultaneously.

- **Blocking**

  Blocking sockets means that the module will not respond to a socket command until the command is completed. However, other commands can still be sent to the module.

  These sockets do not have any status information in the fieldbus specific area. The module can handle up to 32 blocking sockets simultaneously.

Commands in this category:

Mailbox Command	Description	Page
Socket non-blocking (SOCKET_NB)	Creates a socket in non-blocking mode.	12-56
Socket blocking (SOCKET_B)	Creates a socket in blocking mode.	12-57
Listen (LISTEN)	Starts listen on a socket for incoming connections.	12-58
Accept (ACCEPT)	Accepts connections for sockets in blocking mode.	12-59
Connect (CONNECT)	Tries to connect a socket to a client.	12-60
Send (SEND)	Sends a message to a connected socket.	12-62
Receive (RECEIVE)	Receives a message form a connected socket.	12-63
Send To (SEND_TO)	Sends a message to an unconnected UDP socket to a specified host.	12-64
Receive From (RECV_FROM)	Receives a message from an unconnected UDP socket.	12-65
Close (CLOSE)	Closes a socket (and connection).	12-66
Send Fragment (SEND_FRAG)	Sends a fragment of a message with a total maximum size of 4096 bytes.	12-67
Receive Fragment (RECV_FRAG)	Receives a fragment of a message with a total maximum size of 4096 bytes.	12-69
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.	12-71
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.	12-72

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 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.

Parameter	Description
Command initiator	Application
Command number	0040h
Fragmented	No
Extended Header data	-
Message data	The socket type (TCP or UDP) and the port number to bind the socket to.
Response message	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 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.

Parameter	Description
Command initiator	Application
Command number	003Fh
Fragmented	No
Extended Header data	-
Message data	The socket type (TCP or UDP) and the port number to bind the socket to.
Response message	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 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 12-1 "Memory Map" and 12-2 "Socket Status Structure".

- **Blocking sockets**

  Socket descriptors for new connections connected to this socket can be received by the mailbox command ACCEPT, see 12-59 "Accept (ACCEPT)".

Parameter	Description
Command initiator	Application
Command number	0041h
Fragmented	No
Extended Header data	Contains the descriptor of the socket this command is affective for.
Message data	-
Response message	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	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[a]	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	

a. Reserved - Must be set to 0000h

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.

Parameter	Description
Command initiator	Application
Command number	0050h
Fragmented	No
Extended Header data	Contains the descriptor of the socket this command is affective for.
Message data	-
Response message	The response indicates if the command was accepted. The newly received socket descriptor will be present in the response.

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[a]	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**

a. Reserved - must be set to 0000h

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 12-1 "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.

Parameter	Description
Command initiator	Application
Command number	0042h
Fragmented	No
Extended Header data	Contains the descriptor of the socket this command is affective for.
Message data	Contains IP-address and Port number to connect to.
Response message	The response indicates if the command was accepted. The newly received socket descriptor will be present in the response.

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	Socket Descriptor	
Extended word 2	Reserved[a]	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	

a. Reserved - must be set to 0000h

- **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.

Parameter	Description
Command initiator	Application
Command number	0043h
Fragmented	No
Extended Header data	Contains the docket descriptor of the socket this command is affective for.
Message data	Contains the data to send.
Response message	The response indicates if the command was accepted. The response data is the sent 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[a]	-	
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	

a. Reserved - must be set to 0000h

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.

Parameter	Description
Command initiator	Application
Command number	0044h
Fragmented	No
Extended Header data	Contains the descriptor of the socket this command is affective for and how many bytes to receive.
Message data	-
Response message	The response indicates if the command was accepted. The response data is the data that was read from the connection.

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	Data size (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)

Parameter	Description
Command initiator	Application
Command number	0045h
Fragmented	No
Extended Header data	Contains the socket descriptor this command is affective for and the destination IP-address and port number to send to.
Message data	Contains the data to send.
Response message	The response indicates if the command was accepted. The response data is the 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[a]	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message data	Data to send	Sent data	

a. Reserved - must be set to 0000h

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.

Parameter	Description
Command initiator	Application
Command number	0046h
Fragmented	No
Extended Header data	Contains the descriptor of the socket this command is affective for and how many bytes to receive.
Message data	-
Response message	Address information about the sender is returned in the extended header area of the response. The response data is the 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[a]	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**

a. Reserved - must be set to 0000h

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".

Parameter	Description
Command initiator	Application
Command number	0047h
Fragmented	No
Extended Header data	Contains the descriptor of the socket this command is affective for.
Message data	-
Response message	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	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.

Parameter	Description
Command initiator	Application
Command number	005Eh
Fragmented	Yes
Extended Header data	Contains the socket descriptor of the socket this command is affective for and the fragment type.
Message data	Contains the data to send.
Response message	The response data is the 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[a]	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	

a. Reserved - must be set to 0000h

- **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.

Parameter	Description
Command initiator	Application
Command number	005Fh
Fragmented	Yes
Extended Header data	Contains the descriptor of the socket this command is affective for, the data size to be receive from the network buffers in the module and the fragment type. The response contains the number of remaining bytes to read.
Message data	-
Response message	The response contains the 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[b]	-	
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
b. Reserved - must be set to 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.

Parameter	Description
Command initiator	Application
Command number	005Ch
Fragmented	Yes
Extended Header data	Contains the descriptor of the socket this command is affective for, the destination IP address, port number, and fragment type
Message data	Contains the data to send.
Response message	The response indicates if the command was accepted. The response data is the data that was sent.

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[b]	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.
b. Reserved - must be set to 0000h

- **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.

Parameter	Description
Command initiator	Application
Command number	005Dh
Fragmented	Yes
Extended Header data	Socket descriptor, how many bytes to receive from the network buffers in the module, and the fragment type.
Message data	-
Response message	The response indicates if the command was accepted. Address information about the sender is returned in the extended header of the response.

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[a]	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**

a. Reserved - must be set to 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.

Mailbox File System Interface

The filesystem is available to the application through the mailbox interface. 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.	12-75
Close File (FILE_CLOSE)	Close a file previously opened using FILE_OPEN	12-77
Read File (FILE_READ)	Read data from a file	12-78
Write File (FILE_WRITE)	Write data to a file. Depending on how the file was opened, data is either overwritten or added to the end of the file.	12-79
Delete File (FILE_DELETE)	Delete a file	12-80
Move File (FILE_MOVE)	Moves a file	12-81
Rename File (FILE_RENAME)	Rename a file	12-82
Copy File (FILE_COPY)	Copy a file	12-83
Create Directory (DIR_CREATE)	Create a new directory	12-84
Delete Directory (DIR_DELETE)	Delete an empty directory	12-85
Open Directory (DIR_OPEN)	Open a directory	12-86
Read Directory (DIR_READ)	Read contents of a directory previously opened using DIR_OPEN	12-87
Close Directory (DIR_OPEN)	Close a directory previously opened using DIR_OPEN	12-88
Disable Virtual File System (VFS_DISABLE)	Disable the virtual file system	12-89
Create RAM disc (CREATE_RAM_DISC)	Mounts the RAM disc into a specified directory in the file system.	12-90

Open File (FILE_OPEN)

Description

This mailbox command opens a file for reading, writing, or appending.

- **Read mode**

  Opens a file for read only access.

- **Write mode**

  Opens a file for write only access. If the specified file does not exist, it will be created. If the specified file already exists, it will be overwritten.

- **Append mode**

  Opens a file for writing at end-of-file. 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.

Note: The module can handle up to 10 open files simultaneously.

Parameter	Description
Command initiator	Application
Command number	0060h
Fragmented	No
Extended Header data	Mode (Read, Write or Append)
Message data	Name and path to the file to open (NULL terminated)
Response message	The response indicates if the command was accepted. The response contains a file descriptor that shall be used for any further operations on the file.

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, see below	File size (high)	
Extended word 2	-	File size (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 + NULL	File descriptor (high)	**Response data word 1**
		File descriptor (low)	**Response data word 2**

- **Mode Value**

Value	Mode
0000h	Open a file in read mode
0001h	Open a file in write mode
0002h	Open a file in append mode

Close File (FILE_CLOSE)

Description

This mailbox command closes a file previously opened with FILE_OPEN.

Parameter	Description
Command initiator	Application
Command number	0061h
Fragmented	No
Extended Header data	File descriptor of the file to close
Message data	-
Response message	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	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 descriptor (high)	File descriptor (high)	
Extended word 2	File descriptor (low)	File descriptor (low)	
Extended word 3	-	File size (high)	
Extended word 4	-	File size (low)	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	

Read File (FILE_READ)

Description

This mailbox command reads data from a file previously opened in read mode using FILE_OPEN. It is possible to read a maximum of 256 bytes at a time using this command.

Parameter	Description
Command initiator	Application
Command number	0062h
Fragmented	No
Extended Header data	File descriptor and number of bytes to read
Message data	-
Response message	The response indicates if the command was accepted. 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 descriptor (high)	File descriptor (high)	
Extended word 2	File descriptor (low)	File descriptor (low)	
Extended word 3	Number of bytes to read	Number of requested bytes	*Maximum 256 bytes.*
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
		File data	**Response data**

Write File (FILE_WRITE)

Description

This mailbox command writes data to a file previously opened in write or append mode with FILE_OPEN. It is possible to write a maximum of 256 bytes at a time using this command.

Parameter	Description
Command initiator	Application
Command number	0063h
Fragmented	No
Extended Header data	File descriptor of the destination file
Message data	Data to write
Response message	A 'Data size' value of 0 (zero) indicates that the command was unsuccessful, possibly due to a faulty descriptor, 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 descriptor (high)	File descriptor (high)	
Extended word 2	File descriptor (low)	File descriptor (low)	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message data	Data to write	Written data	

Delete File (FILE_DELETE)

Description

This mailbox command deletes a file in the file system.

Parameter	Description
Command initiator	Application
Command number	0064h
Fragmented	No
Extended Header data	-
Message data	Name and path to the file to delete (NULL terminated)
Response message	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	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 + NULL	Path + filename + NULL	

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.

Parameter	Description
Command initiator	Application
Command number	0065h
Fragmented	No
Extended Header data	-
Message data	Name + Path of source and destination, both NULL terminated
Response message	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	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	sourcepath + sourcefile + NULL	sourcepath + sourcefile + NULL	
	destpath + NULL	destpath + NULL	

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.

Parameter	Description
Command initiator	Application
Command number	0066h
Fragmented	No
Extended Header data	-
Message data	Name + Path of source and destination, both NULL terminated
Response message	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	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	path + old filename + NULL	path + old filename + NULL	
	path + new filename + NULL	path + new filename + NULL	

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.

Parameter	Description
Command initiator	Application
Command number	0067h
Fragmented	No
Extended Header data	-
Message data	Name + Path of source and destination, both NULL terminated
Response message	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	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	path + sourcefile + NULL	path + sourcefile + NULL	
	path + dest.file + NULL	path + dest.file + NULL	

Create Directory (DIR_CREATE)

Description

This command creates a directory in the file system.

Parameter	Description
Command initiator	Application
Command number	0068h
Fragmented	No
Extended Header data	-
Message data	Object name and path of the directory, NULL terminated
Response message	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	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 + object name + NULL	path + object name + NULL	

Delete Directory (DIR_DELETE)

Description

This command deletes an empty directory from the file system.

Parameter	Description
Command initiator	Application
Command number	0069h
Fragmented	No
Extended Header data	-
Message data	Object name and path of the directory, NULL terminated
Response message	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	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 + object name + NULL	path + object name + NULL	

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.

Note: The module can handle up to 10 open directories simultaneously.

Parameter	Description
Command initiator	Application
Command number	006Ah
Fragmented	No
Extended Header data	-
Message data	Object name and path of the directory, NULL terminated
Response message	The response indicates if the command was accepted. If the command succeeded, the response contains a 4 byte descriptor that should be used on all further operations on the directory.

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 + object name + NULL	descriptor (high)	**Response data word 1**
		descriptor (low)	**Response data word 2**

Read Directory (DIR_READ)

Description

This command reads a filesystem object from a directory previously opened using DIR_OPEN. This process must be repeated until the response is empty (i.e. Data size = 0). See Appendix A-2 "Reading a Directory")

Parameter	Description
Command initiator	Application
Command number	006Bh
Fragmented	No
Extended Header data	Directory descriptor
Message data	-
Response message	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	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	descriptor (high)	descriptor (high)	*(See DIR_OPEN)*
Extended word 2	descriptor (low)	descriptor (low)	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
		Size of objects (long)	**Response data byte 1** … **Response data byte 4**
		Object flags	**Response data byte 5**
		Object name + NULL	**Response data byte 6 ... n**

- **Object Flags**

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

Bit	Description
DIR	Object is a directory
RO	Object is read only

Bit	Description
H	Object is hidden
SYS	Object is a system object

Close Directory (DIR_CLOSE)

Description

This command closes a directory previously opened using DIR_OPEN

Parameter	Description
Command initiator	Application
Command number	006Ch
Fragmented	No
Extended Header data	Directory descriptor
Message data	-
Response message	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	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	descriptor (high)	descriptor (high)	*(See DIR_OPEN)*
Extended word 2	descriptor (low)	descriptor (low)	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	

Disable Virtual File System (DISABLE_VFS)

Description

This command disables the virtual file system.

Note: This command can only be sent during module initialization.

Parameter	Description
Command initiator	Application
Command number	0011h
Fragmented	No
Extended Header data	-
Message data	-
Response message	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	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	-	-	

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.

Parameter	Description
Command initiator	Application
Command number	0018h
Fragmented	No
Extended Header data	-
Message data	String containing RAM disc location, null terminated
Response message	The response indicates if the command was accepted. The command will fail if the passed directory is not empty.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0018h	0018h	*CREATE_RAM_DISC*
Data size	(size)	(size)	*Length of string 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 + NULL	Path + NULL	

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.	12-92
Alter Module Status LED Functionality (ALT_MS_LED_FUNCTION)	Alter the behaviour of the Module Status LED	12-93
Alter Network Status LED Functionality (ALT_NS_LED_FUNCTION)	Alter the behaviour of the Network Status LED	12-94
Get DIP Switch GET_DIP_SWITCH	Returns the setting of the onboard DIP switch	12-95
DNS Request (DNS_REQUEST)	Asks the configured DNS server for the IP address of a specified host	12-96
Global Administration Mode (GLOBAL_ADMIN_MODE)	Instruct the module to run in global administration mode	12-97
Send Email (SEND_EMAIL)	Sends an email message to a specified recipient	12-98

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 can only be sent during module initialisation.

Parameter	Description
Command initiator	Application
Command number	0017h
Fragmented	No
Extended Header data	-
Message data	LED function
Response message	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	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 change the behaviour of the Module Status LED according to 1-3 "Status Indicators".

Note: This command can only be sent during module initialisation.

Parameter	Description
Command initiator	Application
Command number	000Ch
Fragmented	No
Extended Header data	-
Message data	LED function
Response message	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	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 change the behaviour of the Network Status LED according to 1-3 "Status Indicators".

Note: This command can only be sent during module initialisation.

Parameter	Description
Command initiator	Application
Command number	000Dh
Fragmented	No
Extended Header data	-
Message data	LED function
Response message	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	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 DIP switch. This command enables alternative usage of the dipswitch.

Parameter	Description
Command initiator	Application
Command number	0012h
Fragmented	No
Extended Header data	-
Message data	-
Response message	The response indicates if the command was accepted. The response data contains the DIP switch setting.

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.

For more information about DNS, see 4-4 "DNS Support".

Parameter	Description
Command initiator	Application
Command number	0030h
Fragmented	No
Extended Header data	-
Message data	Hostname and Domain name, NULL terminated
Response message	The response indicates if the command was accepted. The response data contains the IP number if successful. If the host cannot be found, the returned IP address will be 0.0.0.0.

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 name + NULL	IP number of host	**Response databyte**

Global Administration Mode (GLOBAL_ADMIN_MODE)

Description

This command instructs the module to run in Global Admin Mode. For more information about normal/administration mode, see 3-2 "Security"

Note: This command can only be sent during module initialization.

Parameter	Description
Command initiator	Application
Command number	000Bh
Fragmented	No
Extended Header data	-
Message data	-
Response message	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	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	-	-	

Send Email (SEND_EMAIL)

Description

This command sends an email to a specified recipient. The module will not be able to complete this command without a configured SMTP server. 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.

- Fragment 1: Recipients email addresses, separated by semicolon. NULL terminated.
- Fragment 2: Senders email address. NULL terminated.
- Fragment 3: Subject line. NULL terminated.
- Fragment 4 - n: Message body.

Parameter	Description
Command initiator	Application
Command number	0070h
Fragmented	Yes
Extended Header data	-
Message data	The senders and recipients email addresses + subject and message body
Response message	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	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 Value**

Value	Description
0000h	First fragment. (Recipient)
0001h	Subsequent fragment. (Sender, Subject, Message body)
0002h	Last fragment. (Message body)

- **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.

Fieldbus Specific Area

Memory Map

The module can handle 16 non-blocking sockets simultaneously. These can be accessed using the mail-box 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 - 7A7h	Reserved	-
7B0h - 7B1h	EtherNet/IP Connection Status	RO
7B1h - 7FFh	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.

EtherNet/IP Connection Status

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

7B0h (MSB) 7B1h (LSB)

Status

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

Control Area

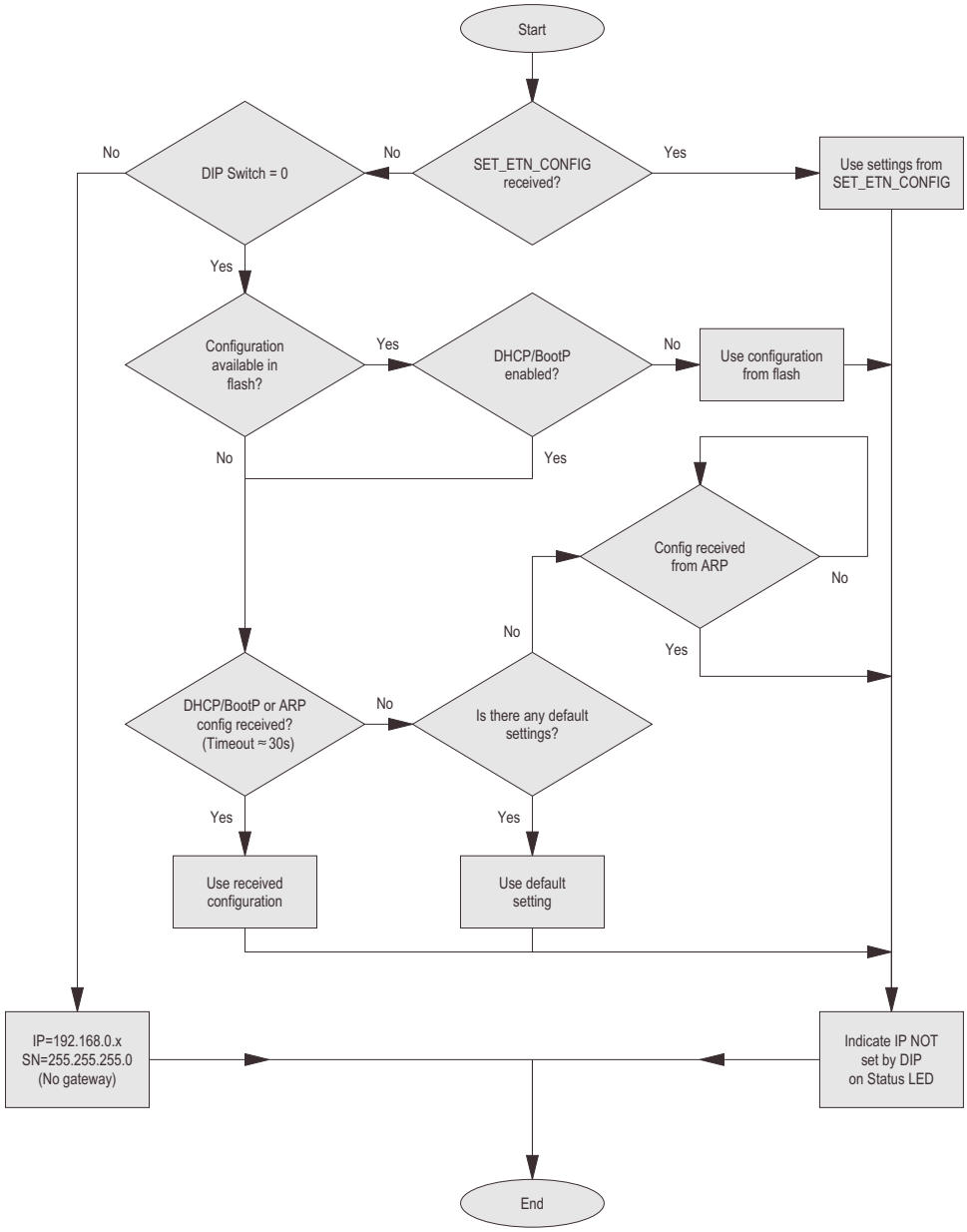
Fieldbus Type

The AnyBus-S Ethernet module is available in three different versions. Each version has it's own module type value depending on functionality.

Value	Module Type
0082h	ABS-EIT-2 (10/100Mbit, Modbus/TCP, IT Functionality)
0083h	ABS-EIP-2 (10/100Mbit, Modbus/TCP, EtherNet/IP, IT Functionality)

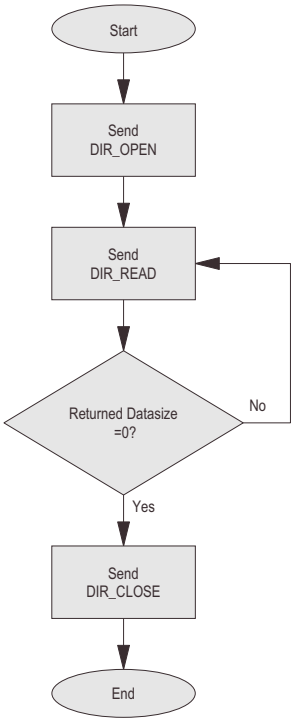
Flow Charts

IP Configuration Sequence



Reading a Directory

(See 12-74 “Mailbox File System Interface”)



Environmental Specification

Temperature

Operating

±0 to +70 degrees Celsius

Test performed according to IEC-68-2-1 and IEC 68-2-2.

Non Operating

-25 to +85 degrees Celsius

Test performed according to IEC-68-2-1 and IEC 68-2-2.

Relative Humidity

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

Test performed according to IEC 68-2-30.

EMC compliance

Emission

According to EN 50 081-2:1993

Tested per EN 55022 Class A: 1997

Immunity

According to EN 61000-6-2: 1999

Tested per EN 61000-4-2:1995

EN 61000-4-3:1996

EN 61000-4-4:1995

EN 61000-4-5:1995

EN 61000-4-6:1996

Miscellaneous

Firmware Upgrade

The 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 firmware file to the system root ("\"), or to the user root ("\user\") of the module.
3. Perform a module reset
   During startup, the module will check for a new firmware file. If a valid file is found, the module will reprogram the flash. The file will be deleted automatically after programming.
4. Perform a module reset
   The new firmware is now operational.

Formatting the File System

If the module refuses to start up, this may be due to an error in the file system. The steps below can be used to format and restore the filesystem to it's initial state.

1. Short jumper named "J2"
2. Apply power to the module
3. Wait for the watchdog led to turn red
4. Disconnect the power
5. Remove jumper
6. Again, apply power to the module
7. Wait approximately 1 minute as the module is formatting the file system

   The module should now be working properly, however all files and folders in the file system has been erased.

Connectors

Application Connector

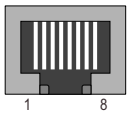
(Consult the generel AnyBus-S Parallel Design Guide for more information)

Ethernet

The module supports RJ45 and board to board connector types. The RJ45 is the standard connector.

RJ45 (Standard)

Pin	Signal
1	TD+
2	TD-
3	RD+
4	Termination
5	Termination
6	RD-
7	Termination
8	Termination



Board to Board

Pin	Signal
1	GND
2	Termination (Connected to RJ45 pin 4 and 5)
3	Termination (Connected to RJ45 pin 4 and 5)
4	NC
5	TD+
6	TD-
7	RD+
8	Termination (Connected to RJ45 pin 7 and 8)
9	RD-
10	Termination (Connected to RJ45 pin 7 and 8)

Electrical Characteristics

Supply Voltage

Both the module electronics and the fieldbus interface shall be supplied with regulated 5V DC. For more information regarding power requirements, consult the AnyBus-S Design Guide.

Power Consumption

The maximum current consumption is 450mA.

Mechanical Specification

Measurements, Connectors & Switches