# Fieldbus Appendix

# Anybus-S EtherNet/IP

## Modbus/TCP, EtherNet/IP & IT Functionality

**Doc.Id. SCM-1200-075**
**Rev.1.63**

Connecting Devices™

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

HMS Industrial Networks
Mailing address: Box 4126, 300 04 Halmstad, Sweden
Visiting address: Stationsgatan 37, Halmstad, Sweden

E-mail: info@hms-networks.com
Web: www.anybus.com

# Table of Contents

## Preface About This Document

How To Use This Document .... P-1

Important User Information .... P-1

Related Documents .... P-2

Document History .... P-2

Conventions & Terminology .... P-3

Sales and Support .... P-4

## Chapter 1 About the Anybus-S EtherNet/IP

Features .... 1-1

Overview .... 1-2
- *Status Indicators* .... 1-2

## Chapter 2 Basic Operation

EtherNet/IP .... 2-1
- *General* .... 2-1
- *Data Exchange* .... 2-1
- *Implementation Notes* .... 2-1

Modbus/TCP .... 2-2
- *General* .... 2-2
- *Addressing Modes* .... 2-2
- *Supported Modbus/TCP Commands* .... 2-3
- *Supported Exception Codes* .... 2-3
- *Anybus Addressing Mode (Default)* .... 2-4
- *Modbus Addressing Mode* .... 2-5

Filesystem .... 2-7
- *General* .... 2-7
- *Filesystem Overview* .... 2-8
- *System Files* .... 2-8

## Chapter 3 Basic Network Configuration

TCP/IP Settings .... 3-1

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

IP Access Control .... 3-3

On/Off Line Configuration .... 3-4

Anybus IPconfig (HICP) .... 3-5

ARP Gleaning .... 3-5

## Chapter 4 FTP Server

General .... 4-1

## Chapter 5 Telnet Server

General ........ 5-1
General Commands ........ 5-2
Diagnostic Commands ........ 5-3
Filesystem Operations ........ 5-3

## Chapter 6 Web Server

General ........ 6-1
Authorization ........ 6-2
Content Types ........ 6-3

## Chapter 7 SMTP Client

General ........ 7-1
Email Definitions ........ 7-2

## Chapter 8 Server Side Include (SSI)

Functions ........ 8-2
Changing SSI output ........ 8-10
*SSI Output String File* ........ 8-10
*Temporary SSI Output change* ........ 8-11

## Chapter 9 CIP Object Implementation

General ........ 9-1
Identity Object, Class 01h ........ 9-2
*General Information* ........ 9-2
*Class Attributes* ........ 9-2
*Instance Attributes* ........ 9-2
*Device Status* ........ 9-3
*Reset Service* ........ 9-3
Message Router, Class 02h ........ 9-4
*General Information* ........ 9-4
*Class Attributes* ........ 9-4
*Instance Attributes* ........ 9-4
Assembly Object, Class 04h ........ 9-5
*General Information* ........ 9-5
*Class Attributes* ........ 9-5
*Instance C6h Attributes (Heartbeat, Input-Only)* ........ 9-5
*Instance C7h Attributes (Heartbeat, Listen-Only)* ........ 9-5
*Instance C5h Attributes (Configuration Data)* ........ 9-5
*Instance 64h... 69h Attributes (Producing Instances)* ........ 9-6
*Instance 96h... 9Bh Attributes (Consuming Instance)* ........ 9-6
Connection Manager, Class 06h ........ 9-7
*General Information* ........ 9-7
*Instance Descriptions* ........ 9-7
*Class 1 Connection Details* ........ 9-7
*Class 3 Connection Details* ........ 9-8

Diagnostic Object, Class AAh ........ 9-9
*General Information* ........ 9-9
*Class Attributes* ........ 9-9
*Instance Attributes, Instance 01h* ........ 9-9

I/O Data Input Mapping Object, Class A0h ........ 9-10
*General Information* ........ 9-10
*Class Attributes* ........ 9-10
*Instance Attributes, Instance 01h* ........ 9-10

I/O Data Output Mapping Object, Class A1h ........ 9-11
*General Information* ........ 9-11
*Class Attributes* ........ 9-11
*Instance Attributes, Instance 01h* ........ 9-11

Parameter Data Input Mapping Object, Class B0h ........ 9-12
*General Information* ........ 9-12
*Class Attributes* ........ 9-12
*Instance Attributes, Instance 01h* ........ 9-12

Parameter Data Output Mapping Object, Class B1h ........ 9-13
*General Information* ........ 9-13
*Class Attributes* ........ 9-13
*Instance Attributes, Instance 01h* ........ 9-13

Port Object, Class F4h ........ 9-14
*General Information* ........ 9-14
*Class Attributes* ........ 9-14
*Instance Attributes, Instance 02h* ........ 9-14

TCP/IP Interface Object, Class F5h ........ 9-15
*General Information* ........ 9-15
*Class Attributes* ........ 9-15
*Instance Attributes* ........ 9-16

Ethernet Link Object, Class F6h ........ 9-18
*General Information* ........ 9-18
*Class Attributes* ........ 9-18
*Instance Attributes* ........ 9-18

**Chapter 10 Fieldbus Specific Mailbox Commands**

Fault Information ........ 10-1

General Configuration Commands ........ 10-3
*Set Ethernet Configuration (SET_ETN_CONFIG)* ........ 10-4
*Read Ethernet Configuration (READ_ETN_CONFIG)* ........ 10-5
*Get MAC Address (GET_MAC_ADDR)* ........ 10-6
*Connection Timeout (CONNECT_TIMEOUT)* ........ 10-7
*Set Host and Domain (SET_HOST_DOMAIN)* ........ 10-8
*Get Host and Domain (GET_HOST_DOMAIN)* ........ 10-9
*Disable / Enable ARP Config (ARP_CFG_CONTROL)* ........ 10-10
*Disable / Enable HICP (HICP_CFG_CONTROL)* ........ 10-11
*Disable DHCP (DISABLE_DHCP)* ........ 10-12
*Set DNS Server Settings (SET_DNS_SERVERS)* ........ 10-13
*Get DNS Server Settings (GET_DNS_SERVERS)* ........ 10-14
*Set SMTP Server (SET_SMTP_SERVER)* ........ 10-15
*Set SMTP Server by Name (SET_SMTP_SERVER_NAME)* ........ 10-16
*Get SMTP Server (GET_SMTP_SERVER)* ........ 10-17
*Disable Web Server (DISABLE_WEB_SERVER)* ........ 10-18
*Enable Web Server (ENABLE_WEB_SERVER)* ........ 10-19
*Disable FTP server (DISABLE_FTP_SERVER)* ........ 10-20
*Disable Telnet Server (DISABLE_TELNET_SERVER)* ........ 10-21
*Global Admin Mode (GLOBAL_ADMIN_MODE)* ........ 10-22
*Disable Virtual File System (DISABLE_VFS)* ........ 10-23
*Set Serial Number (SET_SERIAL_NUMBER)* ........ 10-24
*Set HTTP Server Name (SET_HTTP_SERVER)* ........ 10-25
*Set SMTP Port Number (SET_SMTP_PORT)* ........ 10-26

Mailbox File System Interface ........ 10-27
*Open File (FILE_OPEN)* ........ 10-28
*Close File (FILE_CLOSE)* ........ 10-29
*Read File (FILE_READ)* ........ 10-30
*Write File (FILE_WRITE)* ........ 10-31
*Delete File (FILE_DELETE)* ........ 10-32
*Move File (FILE_MOVE)* ........ 10-32
*Rename File (FILE_RENAME)* ........ 10-33
*Copy File (FILE_COPY)* ........ 10-35
*Create Directory (DIR_CREATE)* ........ 10-35
*Delete Directory (DIR_DELETE)* ........ 10-36
*Open Directory (DIR_OPEN)* ........ 10-37
*Read Directory (DIR_READ)* ........ 10-38
*Close Directory (DIR_CLOSE)* ........ 10-40
*Create RAM disc (CREATE_RAM_DISC)* ........ 10-40
*Format file system (FORMAT_FS)* ........ 10-41
*File system Checksum (CRC_FS)* ........ 10-43

Modbus/TCP Specific Commands ........ 10-44
*Modbus Connection Timeout (MB_CONN_TIMEOUT)* ........ 10-44
*Disable Modbus/TCP server (DISABLE_MB_TCP)* ........ 10-45
*Modbus ON/OFF Line Configuration (MB_ON_OFF_LINE_CONFIG)* ........ 10-47
*Write Exception Status (WR_EXP_STAT)* ........ 10-48
*Set Modbus Coil Area Size (MB_SET_COIL_AREA_SIZE)* ........ 10-48

EtherNet/IP Specific Commands........10-50
*Set Product Info All (PRODUCT_INFO_ALL)*........10-52
*Set Product Info (SET_PRODUCT_INFO)*........10-52
*Set Product Code (SET_PRODUCT_CODE)*........10-53
*Parameter Data Input Mapping (PARAMETER_INPUT_MAP)*........10-54
*Parameter Data Output Area Mapping (PARAMETER_OUTPUT_MAP)*........10-56
*I/O Data Input Area Mapping (IO_INPUT_MAP)*........10-58
*I/O Data Output Area Mapping (IO_OUTPUT_MAP)*........10-60
*Send UCMM (SEND_UCMM)*........10-62
*UCMM Request (UCMM_REQUEST)*........10-64
*Enable Large UCMM Request (ENABLE_LARGE_UCMM_REQUEST)*........10-65
*Large UCMM Request (LARGE_UCMM_REQUEST)*........10-66
*Example 1*........10-67
*Example 2*........10-71
*Register Class (REGISTER_CLASS)*........10-71
*Deregister Class (DEREGISTER_CLASS)*........10-72
*Enable Routing (ENABLE_ROUTING)*........10-74
*Register Port (REGISTER_PORT)*........10-74
*Route Unconnected Send (ROUTE_REQUEST)*........10-77
*Enable Reset Notification (ENABLE_ID_RESET_NOTIFY)*........10-78
*Reset Notification (ID_RESET_NOTIFY)*........10-80
*Get Reset Parameter (GET_ID_RESET_PARAM)*........10-81
*Copy I/O Status (COPY_IO_STATUS)*........10-82
*Disable EtherNet/IP (DISABLE_ETHERNET_IP)*........10-82
*Change Ethernet Port (CHANGE_ETHERNET_PORT)*........10-83
*Reset on IP Change (RST_ON_IP_CHANGE)*........10-84
*Enable Configuration Assembly (ENABLE_CONFIG_SY)*........10-84
*Set Configuration Data (SET_CONFIG_DATA)*........10-85
*Enable Exact IO Match (ENABLE_IO_MATCH)*........10-87
*TCP/IP Object Read-only (TCP_IP_RO)*........10-87
*Get Configuration Data (GET_CONFIG_DATA)*........10-89
*Set Assembly Instances (SET_SY_INSTANCES)*........10-90

Mailbox Socket Interface........10-91
*Socket Non-Blocking (SOCKET_NB)*........10-92
*Socket Blocking (SOCKET_B)*........10-93
*Listen (LISTEN)*........10-93
*Accept (ACCEPT)*........10-95
*Connect (CONNECT)*........10-96
*Send (SEND)*........10-97
*Receive (RECV)*........10-98
*Send To (SEND_TO)*........10-99
*Receive From (RECV_FROM)*........10-100
*Close (CLOSE)*........10-101
*Send Fragment (SEND_FRAG)*........10-102
*Receive Fragment (RECV_FRAG)*........10-104
*Send Fragment To (SEND_FRAG_TO)*........10-105
*Receive Fragment From (RECV_FRAG_FROM)*........10-107
*Get Socket Option (GET_SOCKET_OPTION)*........10-109
*Set Socket Option (SET_SOCKET_OPTION)*........10-110
*Socket Options*........10-111

Other Commands ........ 10-114

*Alter LNK and ACT LED Functionality (ALT_LNK_ACT_LEDS)* ........ 10-115

*Alter Module Status LED Functionality (ALT_MS_LED_FUNCTION)* ........ 10-116

*Alter Network Status LED Functionality (ALT_NS_LED_FUNCTION)* ........ 10-117

*Get DIP Switch (GET_DIP_SWITCH)* ........ 10-118

*DNS Request (DNS_REQUEST)* ........ 10-119

*Send Email (SEND_EMAIL)* ........ 10-120

*Request SSI Data (REQUEST_SSI_DATA)* ........ 10-122

*Write SSI Data (WRITE_SSI_DATA)* ........ 10-123

*Write Output Area (WRITE_OUTPUT_AREA)* ........ 10-124

**Chapter 11 Fieldbus Specific Area**

Memory Map ........ 11-1

Socket Status Structure ........ 11-2

Network Status ........ 11-3

EtherNet/IP Status ........ 11-3

**Appendix A Miscellaneous**

Control Register Area ........ A-1

Firmware Upgrade ........ A-2

Formatting the File System ........ A-2

IP-related implementation details ........ A-2

**Appendix B Technical Specification**

Electrical Specification ........ B-1

*Protective Earth (PE) Requirements* ........ B-1

*Isolation* ........ B-1

*Power Supply* ........ B-1

Environmental Specification ........ B-2

EMC (CE) Pre-compliance ........ B-2

**Appendix C Connectors**

Application Connector ........ C-1

Ethernet ........ C-1

**Appendix D Mechanical Specification**

Measurements, Connectors & LEDs ........ D-1

**Appendix E Copyright Notices**

# About This Document

## How To Use This Document

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

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

## Important User Information

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

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

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

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

**Warning:** This is a class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

**ESD Note:** This product contains ESD (Electrostatic Discharge) sensitive parts that may be damaged if ESD control procedures are not followed. Static control precautions are required when handling the product. Failure to observe this may cause damage to the product.

# Related Documents

| Document name | Author |
|---|---|
| Open Modbus/TCP Specification, Release 1.0 | Schneider Automation |
| RFC 821 | Network Working Group |
| RFC 1918 | Network Working Group |
| ENIP Specifications | ODVA |
| Anybus-S Parallel Design Guide | HMS |
| Anybus-S Ethernet Development Specification v1.16 | HMS |

# Document History

## Summary of Recent Changes (1.62... 1.63)

| Change | Page(s) |
|---|---|
| Corrected port type in Port object (F4h) | 9-14 |
| Changed note on creation of ethcfg.cfg, updated description of ethcfg.cfg | 3-1, 3-2 |
| Added instance attributes to TCP/IP object (F5h) | 9-16 |
| Added mailbox SET_SMTP_PORT | 10-26 |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

## Revision List

| Revision | Date | Author(s) | Chapter(s) | Description |
|---|---|---|---|---|
| <1.50 | - | - | - | (See previous versions) |
| 1.50 | 2006-06-22 | PeP | All | Major update/rewrite |
| 1.51 | 2006-10-19 | PeP | 3, B, C | Misc. minor corrections & updates |
| 1.52 | 2007-01-19 | ToT | 2, 3, 4, 5, 6, 11 | Added connection limits for the onboard TCP servers, additions and corrections to the fieldbus-specific area |
| 1.60 | 2008-11-07 | HeS | All | Major update |
| 1.61 | 2009-05-26 | KeL | 2, 3, 9, 10 | Misc. minor corrections and updates |
| 1.62 | 2010-12-06 | KeL | 2, 9, P, B, 5, 10, E | Misc. updates |
| 1.63 | 2012-03-09 | KeL | 3, 9, 10 | Misc. corrections and updates |

# Conventions & Terminology

The following conventions are used throughout this document:

- Numbered lists provide sequential steps
- Bulleted lists provide information, not procedural steps
- Mailbox commands that must to be sent "during module initialization" must be sent between the "START_INIT" and "END_INIT" commands.
- The term 'module' is used when referring to the ABS-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
- The term 'byte' always refers to a string of 8 bits

# Sales and Support

| **Sales** | | **Support** | |
|---|---|---|---|
| **HMS Sweden (Head Office)** | | | |
| E-mail: | sales@hms-networks.com | E-mail: | support@hms-networks.com |
| Phone: | +46 (0) 35 - 17 29 56 | Phone: | +46 (0) 35 - 17 29 20 |
| Fax: | +46 (0) 35 - 17 29 09 | Fax: | +46 (0) 35 - 17 29 09 |
| Online: | www.anybus.com | Online: | www.anybus.com |
| **HMS North America** | | | |
| E-mail: | us-sales@hms-networks.com | E-mail: | us-support@hms-networks.com |
| Phone: | +1-312 - 829 - 0601 | Phone: | +1-312-829-0601 |
| Toll Free: | +1-888-8-Anybus | Toll Free: | +1-888-8-Anybus |
| Fax: | +1-312-629-2869 | Fax: | +1-312-629-2869 |
| Online: | www.anybus.com | Online: | www.anybus.com |
| **HMS Germany** | | | |
| E-mail: | ge-sales@hms-networks.com | E-mail: | ge-support@hms-networks.com |
| Phone: | +49 (0) 721-989777-000 | Phone: | +49 (0) 721-989777-000 |
| Fax: | +49 (0) 721-989777-010 | Fax: | +49 (0) 721-989777-010 |
| Online: | www.anybus.de | Online: | www.anybus.de |
| **HMS Japan** | | | |
| E-mail: | jp-sales@hms-networks.com | E-mail: | jp-support@hms-networks.com |
| Phone: | +81 (0) 45-478-5340 | Phone: | +81 (0) 45-478-5340 |
| Fax: | +81 (0) 45-476-0315 | Fax: | +81 (0) 45-476-0315 |
| Online: | www.anybus.jp | Online: | www.anybus.jp |
| **HMS China** | | | |
| E-mail: | cn-sales@hms-networks.com | E-mail: | cn-support@hms-networks.com |
| Phone: | +86 (0) 10-8532-3183 | Phone: | +86 (0) 10-8532-3023 |
| Fax: | +86 (0) 10-8532-3209 | Fax: | +86 (0) 10-8532-3209 |
| Online: | www.anybus.cn | Online: | www.anybus.cn |
| **HMS Italy** | | | |
| E-mail: | it-sales@hms-networks.com | E-mail: | it-support@hms-networks.com |
| Phone: | +39 039 59662 27 | Phone: | +39 039 59662 27 |
| Fax: | +39 039 59662 31 | Fax: | +39 039 59662 31 |
| Online: | www.anybus.it | Online: | www.anybus.it |
| **HMS France** | | | |
| E-mail: | fr-sales@hms-networks.com | E-mail: | fr-support@hms-networks.com |
| Phone: | +33 (0) 3 68 368 034 | Phone: | +33 (0) 3 68 368 033 |
| Fax: | +33 (0) 3 68 368 031 | Fax: | +33 (0) 3 68 368 031 |
| Online: | www.anybus.fr | Online: | www.anybus.fr |
| **HMS UK & Eire** | | | |
| E-mail: | uk-sales@hms-networks.com | E-mail: | support@hms-networks.com |
| Phone: | +44 (0) 1926 405599 | Phone: | +46 (0) 35 - 17 29 20 |
| Fax: | +44 (0) 1926 405522 | Fax: | +46 (0) 35 - 17 29 09 |
| Online: | www.anybus.co.uk | Online: | www.anybus.com |
| **HMS Denmark** | | | |
| E-mail: | dk-sales@hms-networks.com | E-mail: | support@hms-networks.com |
| Phone: | +45 (0) 35 38 29 00 | Phone: | +46 (0) 35 - 17 29 20 |
| Fax: | +46 (0) 35 17 29 09 | Fax: | +46 (0) 35 - 17 29 09 |
| Online: | www.anybus.com | Online: | www.anybus.com |
| **HMS India** | | | |
| E-mail: | in-sales@hms-networks.com | E-mail: | in-support@hms-networks.com |
| Phone: | +91 (0) 20 40111201 | Phone: | +91 (0) 20 40111201 |
| Fax: | +91 (0) 20 40111105 | Fax: | +91 (0) 20 40111105 |
| Online: | www.anybus.com | Online: | www.anybus.com |

# About the Anybus-S EtherNet/IP

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

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

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

## Features

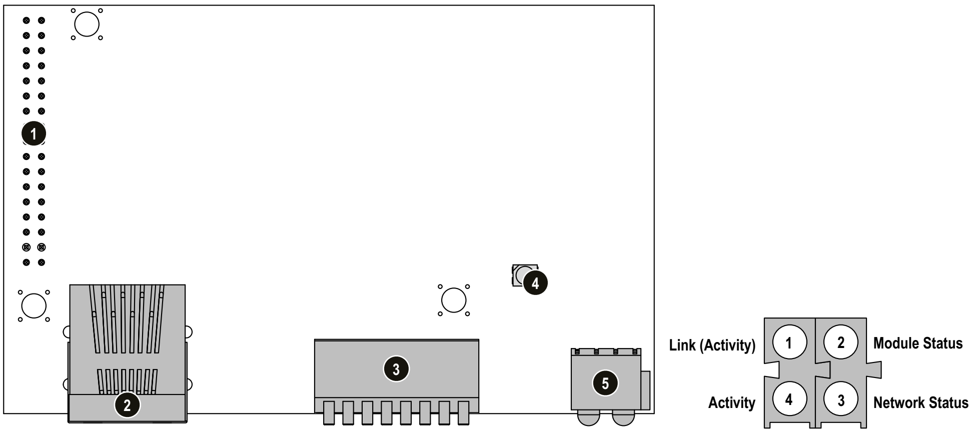
### General

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

### Industrial Protocols

- EtherNet/IP
  - Adapter class, Message server, Message client
- Modbus/TCP server
  - Conforms to Modbus/TCP v1.0

# Overview



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

## Status Indicators

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.

### Link (Activity)

The behaviour of this led can be altered, see 10-115 “Alter LNK and ACT LED Functionality (ALT_LNK_ACT_LEDS)”.

- **Led configuration 1 (Default)**

| State | Description |
|---|---|
| Off | Link not sensed |
| Green | Link sensed |

- **Led configuration 2**

| State | Description |
|---|---|
| Off | Link not sensed |
| Green | Link sensed |
| Green, flashing | Activity; receiving/transmitting ethernet packets |

### 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 10-116 "Alter Module Status LED Functionality (ALT_MS_LED_FUNCTION)".

- **Led configuration 1**

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

- **Led configuration 2**

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

- **Led configuration 3 (Default)**

| State | Description |
|---|---|
| Off | No power |
| Green | Controlled by a Scanner in Run state |
| Green, flashing | Not configured, or Scanner in Idle state |
| Red, flashing | A minor recoverable fault has been detected |
| Red | A major unrecoverable fault has been detected |
| Alternating Green/Red | Self-test in progress |

- **Led configuration 4**

  This configuration deactivates this led.

## 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 10-117 “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 | Description |
|---|---|
| Off | No power or no IP address |
| Green | On-line, one or more connections established (CIP Class 1 or 3) |
| Green, flashing | On-line, no connections established |
| Red | Duplicate IP address, fatal error |
| Red, flashing | One or more connections timed out (CIP Class 1 or 3) |
| Alternating Green/Red | Self test in progress |

## Activity

The behaviour of this led can be altered, see 10-115 “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.

# Basic Operation

## EtherNet/IP

### General

EtherNet/IP is based on the Common Industrial protocol (CIP) which is also the application layer for DeviceNet and ControlNet. The module acts as an adapter class product on the EtherNet/IP network.

The Input- and Output Data is accessed using I/O connections or Explicit messages towards the Assembly Object and the Parameter Input/Output Mapping Objects.

The following port numbers are used for EtherNet/IP communication:

| | |
|---|---|
| • Port 2222 | (I/O Data) |
| • Port 44818 | (Encapsulated CIP messages) |

See also...

- 9-1 “CIP Object Implementation”

### Data Exchange

I/O Data is exchanged on change of value, and can be accessed using I/O connections towards the Assembly Object. Parameter Data can be accessed acyclically via the Parameter Input- and Output Mapping Objects.

See also...

- 9-5 “Assembly Object, Class 04h”
- 9-12 “Parameter Data Input Mapping Object, Class B0h”
- 9-13 “Parameter Data Output Mapping Object, Class B1h”

### Implementation Notes

Rockwell Automation PLCs uses the first four bytes consumed by a device 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 behavior can be changed using the COPY_IO_STATUS mailbox command (see 10-82 “Copy I/O Status (COPY_IO_STATUS)”.

When initiating the module, its output area has to be extended by four bytes to allow for this status information, the 32-bit Run/Idle header. The application drive also needs to compensate for this, to start copying the output data at the correct location.

The run/idle bit is bit 0 in this 32-bit status information. As CIP is a little-endian system the run/idle bit will be located in bit 0 of the first byte in the output area.

# Modbus/TCP

## General

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

All Modbus/TCP messages are exchanged through TCP port no. 502 and the Modbus TCP server can handle a maximum of 8 simultaneous connections. For detailed information regarding the Modbus/TCP protocol, consult the Open Modbus Specification, Release 1.0.

## Addressing Modes

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

- **Anybus Addressing Mode (Default)**

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

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

  See also...

  - 2-4 “Anybus Addressing Mode (Default)”

- **Modbus Addressing Mode**

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

  See also...

  - 2-5 “Modbus Addressing Mode”

See also...

- 10-48 “Set Modbus Coil Area Size (MB_SET_COIL_AREA_SIZE)”

## Supported Modbus/TCP Commands

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

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

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

## Supported Exception Codes

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

See also...

- 10-48 "Write Exception Status (WR_EXP_STAT)"

# Anybus Addressing Mode (Default)

## Supported Function Codes

The following function codes can be used in this mode:

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

## Coil & Register Map

The data is mapped to Coils & Registers as follows:

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

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

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

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

## Modbus Addressing Mode

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

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

### Supported Function Codes

The following function codes can be used in this mode:

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

See also...

- 10-48 "Set Modbus Coil Area Size (MB_SET_COIL_AREA_SIZE)"

## Input Data

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

| Register # | Input Data Bytes | Comments |
|---|---|---|
| 1 | 000h + Coil Size In | Each register corresponds to two bytes of Input Data.<br>Coil Size In is a parameter of the mailbox "MB_SET_COIL_AREA_SIZE" which is used to create bit accessible data areas, see 10-48. |
| 2 | 002h + Coil Size In | |
| 3 | 004h + Coil Size In | |
| 5 | ... | |
| 1024 | 7FEh + Coil Size In | |

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

## Output Data

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

| Register # | Output Data Bytes | Comments |
|---|---|---|
| 1 | 000h + Coil Size Out | Each register corresponds to two bytes of Output Data.<br>Coil Size Out is a parameter of the mailbox "MB_SET_COIL_AREA_SIZE" which is used to create bit accessible data areas, see 10-48. |
| 2 | 002h + Coil Size Out | |
| 3 | 004h + Coil Size Out | |
| 5 | ... | |
| 1024 | 7FEh + Coil Size Out | |

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

# Filesystem

## General

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

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

### Storage Areas

The filesystem consists of the different storage areas:

- **Non-volatile area (approx. 1,4 Mb)**
  This section is intended for static files such as web files, configuration files etc.

- **Volatile area (approx. 1 Mb)**
  This area is intended for temporary storage; data placed here will be lost in case of power loss or reset. Note that this area is not available by default, and must be mounted by the application during initialisation (see 10-40 “Create RAM disc (CREATE_RAM_DISC)”)

### Conventions

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

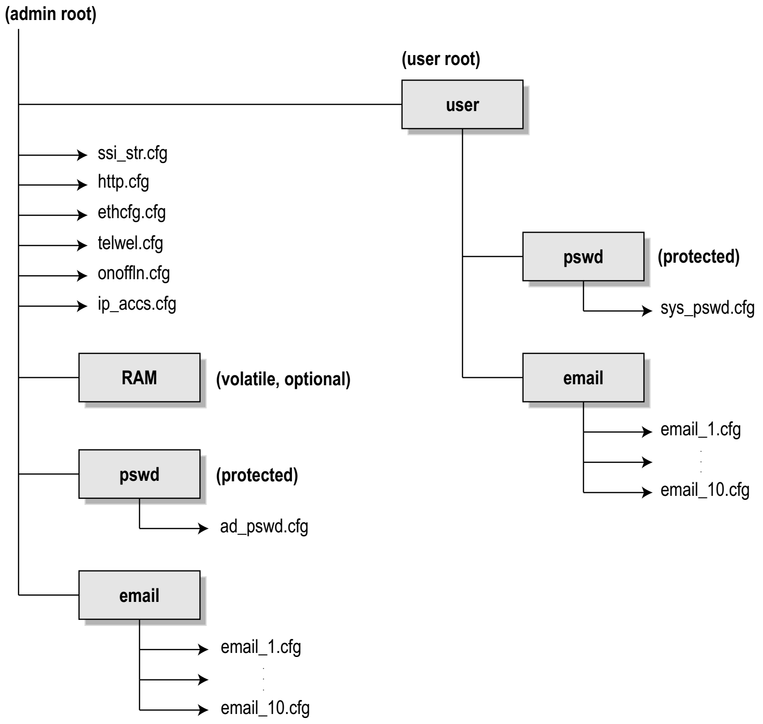
### Important Note:

The non-volatile storage is located in FLASH memory. Each FLASH segment can only be erased approximately 100 000 times due to the nature of this type of memory.

The following operations will erase one or more FLASH segments:

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

## Filesystem Overview



## System Files

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

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

*Example:*

```
[Key1]
value of key1

[Key2]
value of key2
```

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

The contents of the above files can be redirected:

*Example:*

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

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

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

# Basic Network Configuration

## TCP/IP Settings

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

- **Settings specified via Mailbox Interface**

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

  See also...

  - 10-4 “Set Ethernet Configuration (SET_ETN_CONFIG)”

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

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

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

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

  See also...

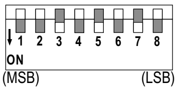
  - 3-2 “Ethernet Configuration File (‘ethcfg.cfg’)”

- **Settings specified by Configuration Switches**

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

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

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



### EtherNet/IP

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

See also...

- 9-1 “CIP Object Implementation”

### DHCP/BootP

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

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

# Ethernet Configuration File ('ethcfg.cfg')

## General

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

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

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

[a] When used in the configuration file, a parameter has to be enclosed by brackets.

[b] Read-only parameter. Used e.g. to detect why a restart was necessary.

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

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

See also...

- 3-1 “TCP/IP Settings”
- 4-1 “FTP Server”

# IP Access Control

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

***File Format:***

| | |
|---|---|
| `[Web]`<br>`xxx.xxx.xxx.xxx` | Nodes listed here may access the web server |
| `[FTP]`<br>`xxx.xxx.xxx.xxx` | Nodes listed here may access the FTP server |
| `[Modbus/TCP]`<br>`xxx.xxx.xxx.xxx` | Nodes listed here may access the module via Modbus/TCP |
| `[EtherNet/IP]`<br>`xxx.xxx.xxx.xxx` | Nodes listed here may access the module via EtherNet/IP |
| `[Telnet]`<br>`xxx.xxx.xxx.xxx` | Nodes listed here may access the module via Telnet |
| `[All]`<br>`xxx.xxx.xxx.xxx` | Fallback setting, used by the module when one or several of the keys above are omitted. If this value is not specified, the module will accept all connections regardless of the other |

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

# On/Off Line Configuration

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

***File Format:***

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

[Timeout]
10

[Commands]
3, 16, 23

[ON-line method]
1
```

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

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

- **Timeout Value**

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

  A value of 10 equals 1000ms.

- **Commands (Optional)**

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

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

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

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

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

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

File example:

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

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

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

**Note 2:** The settings in this file will be ignored if the application has issued MB_ON_OFF_LINE_CONFIG.

See also...

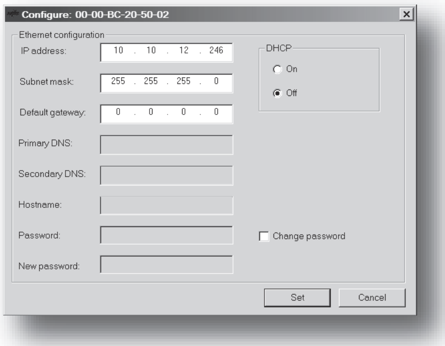
- 10-47 "Modbus ON/OFF Line Configuration (MB_ON_OFF_LINE_CONFIG)"
- A-1 "Event Notification Cause/Source Registers"

# Anybus IPconfig (HICP)

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

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

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



Configure: 00-00-BC-20-50-02

Ethernet configuration

IP address: 10 . 10 . 12 . 246

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 0 . 0 . 0 . 0

Primary DNS:

Secondary DNS:

Hostname:

Password:

New password:

DHCP: ○ On ● Off

□ Change password

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

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

**Note:** The HICP protocol communicates over UDP port 3250.

# ARP Gleaning

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

***Syntax:***

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

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

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

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

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

See also...

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

# FTP Server

## General

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

The following port numbers are used for FTP communication:

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

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

See also...

- 10-20 “Disable FTP server (DISABLE_FTP_SERVER)”

### Security Levels

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

- **Normal-level users**
  The root directory will be ‘\user’.
- **Admin-level users**
  The root directory will be ‘\’, i.e. the user has unrestricted access to the file system.

### User Accounts

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

- **‘\user\pswd\sys_pswd.cfg’**
  This file holds the user accounts for normal-level users.
- **‘\pswd\ad_pswd.cfg’**
  This file holds the user accounts for admin-level users.

***File Format:***

The format of these files are as follows:

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

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

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

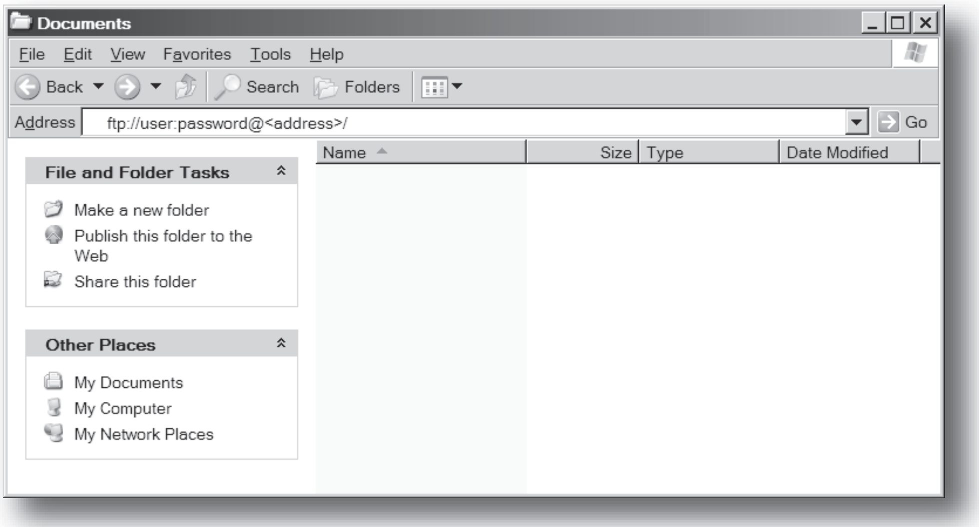
# FTP Connection Example (Windows Explorer)

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

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

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

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



Address ftp://user:password@<address>/

# Telnet Server

## General

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

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

See also...

- 10-21 “Disable Telnet Server (DISABLE_TELNET_SERVER)”
- 2-8 “Filesystem Overview”

### Security Levels

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

- **Normal-level users**

  The root directory will be ‘\user’.
- **Admin-level users**

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

### User Accounts

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

For more information, see 4-1 “User Accounts”

# General Commands

### admin

- **Syntax**

  `admin`

- **Description**

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

### exit

- **Syntax**

  `exit`

- **Description**

  This command closes the Telnet session.

### help

- **Syntax**

  `help [general|diagnostic|filesystem]`

- **Description**

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

```
General commands:

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

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

### version

- **Syntax**

  `version`

- **Description**

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

# Diagnostic Commands

### arps

- **Syntax**

  `arps`
- **Description**

  Display ARP stats and table

### iface

- **Syntax**

  `iface`
- **Description**

  Display net interface stats

### routes

- **Syntax**

  `routes`
- **Description**

  Display IP route table

### sockets

- **Syntax**

  `sockets`
- **Description**

  Display socket list

# Filesystem Operations

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

### append

- **Syntax**

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

  Appends a line to a file.

### cd

- **Syntax**

  `cd [path]`

- **Description**

  Changes current directory.

### copy

- **Syntax**

  `copy [source] [destination]`
- **Description**

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

### del

- **Syntax**

  `del [file]`
- **Description**

  Deletes a file.

### dir

- **Syntax**

  `dir [path]`
- **Description**

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

### df

- **Syntax**

  `df`
- **Description**

  Displays filesystem info.

### format

- **Syntax**

  `format`
- **Description**

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

## md

- **Syntax**

```
md [directory]
```

- **Description**

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

## mkfile

- **Syntax**

```
mkfile [filename]
```

- **Description**

  Creates an empty file.

## move

- **Syntax**

```
move [source] [destination]
```

- **Description**

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

## rd

- **Syntax**

```
rd [directory]
```

- **Description**

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

## ren

- **Syntax**

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

- **Description**

  Renames a file or directory.

## type

- **Syntax**

```
type [filename]
```

- **Description**

  Types the contents of a file.

# Web Server

## General

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

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

See also...

- 10-18 "Disable Web Server (DISABLE_WEB_SERVER)"

### Protected Files

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

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

### Default Web Pages

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

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

These virtual files are:

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

# Authorization

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

***File Format:***

```
Username1:Password1
Username2:Password2
...
UsernameN:PasswordN
```

- List of approved users.

```
[AuthName]
(message goes here)
```

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

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

***Example:***

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

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

[AuthName]
Yeah. Whatsda passwoid?
```

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

# Content Types

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

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

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

*File Format:*

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

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

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

# SMTP Client

## General

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

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

See also...

- 8-1 “Server Side Include (SSI)”
- 10-120 “Send Email (SEND_EMAIL)”

### Server Settings

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

- 10-15 “Set SMTP Server (SET_SMTP_SERVER)”
- 10-120 “Send Email (SEND_EMAIL)”

### Event-Triggered Messages

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

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

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

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

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

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

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

# Email Definitions

The email definitions are stored in the following two directories:

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

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

***File Format:***

```
[Register]
Area, Offset, Type

[Register Match]
Value, Mask, Operand

[To]
recipient

[From]
sender

[Subject]
subject line

[Headers]
Optional extra headers

[Message]
message body
```

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

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

# Server Side Include (SSI)

## General

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

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

### *Syntax*

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

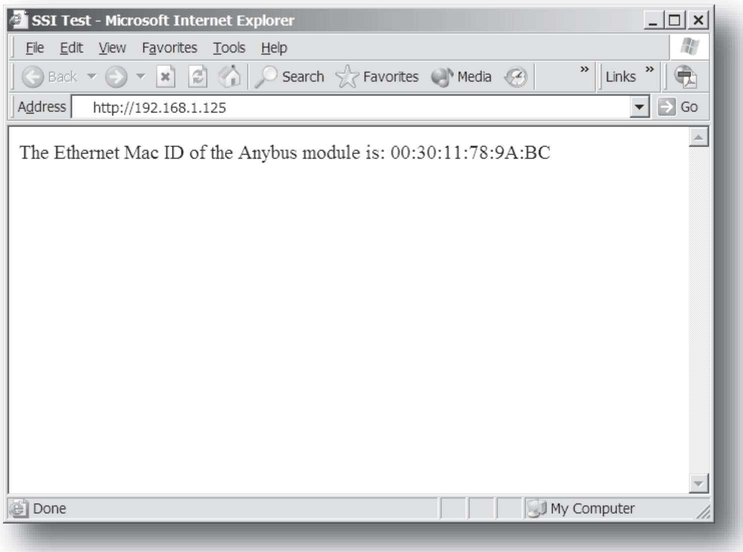
```
<?--#exec cmd_argument='XXXXXXXXXXXXXXXXXXXXXX'-->
```

### *Example*

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

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

Resulting webpage:



The Ethernet Mac ID of the Anybus module is: 00:30:11:78:9A:BC

SSI Test - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Links
Address
http://192.168.1.125
Go
Done
My Computer

# Functions

### DisplayMacID

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

***Syntax:***

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

### DisplaySerial

This function returns the serial number of the Anybus module.

***Syntax:***

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

### DisplayFWVersion

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

***Syntax:***

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

### DisplayBLVersion

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

***Syntax:***

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

### DisplayIP

This function returns the currently used IP address.

***Syntax:***

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

### DisplaySubnet

This function returns the currently used Subnet mask.

***Syntax:***

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

### DisplayGateway

This function returns the currently used Gateway address.

***Syntax:***

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

### DisplayDNS1

This function returns the address of the primary DNS server.

***Syntax:***

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

### DisplayDNS2

This function returns the address of the secondary DNS server.

***Syntax:***

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

### DisplayHostName

This function returns the hostname.

***Syntax:***

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

### DisplayDomainName

This function returns the default domain name.

***Syntax:***

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

### DisplayDhcpState

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

***Syntax:***

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

### DisplayDhcpSupport

DHCP support can be disabled using the mailbox command DISABLE_DHCP, see 10-12 “Disable DHCP (DISABLE_DHCP)”. This function returns ‘Arg1’ if it’s enabled and ‘Arg2’ if it’s disabled.

***Syntax:***

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

### DisplayEmailServer

This function returns the currently used SMTP server address.

***Syntax:***

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

### DisplaySMTPUser

This function returns the username used for SMTP authentication.

***Syntax:***

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

### DisplaySMTPPswd

This function returns the password used for SMTP authentication.

***Syntax:***

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

### StoreEtnConfig

**Note:** This function cannot be used in email messages.

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

***Syntax:***

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

Include this line in a HTML page and pass a form with new IP settings to it.

***Accepted fields in form:***

```
SetIp
SetSubnet
SetGateway
SetEmailServer
SetDhcpState - value "on" or "off"
SetDNS1
SetDNS2
SetHostName
SetDomainName
SetSMTPUser
SetSMTPPswd
```

***Default output:***

```
Invalid IP address!
Invalid Subnet mask!
Invalid Gateway address!
Invalid IP address or Subnet mask!
Invalid Email Server IP address!
Invalid DHCP state!
Invalid DNS1!
Invalid DNS2!
Configuration stored correctly.
Failed to store configuration.
```

### GetText

**Note:** This function cannot be used in email messages.

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

***Syntax:***

```
<?--#exec cmd_argument='GetText( "ObjName", OutWriteString ( offset ), n )'-->
```

| | |
|---|---|
| ObjName | - Name of object. |
| offset | - Specifies the offset from the beginning of the OUT area. |
| n | - Specifies maximum number of characters to read (Optional) |

***Default output:***

```
Success        - Write succeeded
Failure        - Write failed
```

## printf

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

***Syntax:***

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

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

- Flags (in any order), which modify the specification:

  - which specifies left adjustment of the converted argument in its field.

  + which specifies that the number will always be printed with a sign

  (space) if the first character is not a sign, a space will be prefixed.

  0 for numeric conversions, specifies padding to the field with leading zeroes.

  # which specifies an alternate output form. For o, the first digit will be zero. For x or X, 0x or 0X will be prefixed to a non-zero result. For e, E,f, g and G, the output will always have a decimal point; for g and G, trailing zeros will not be removed.

- A number specifying a minimum field width. The converted argument will be printed in a field at least this wide, and wider if necessary. If the converted argument has fewer characters than the field width it will be padded on the left (or right, if left adjustment has been requested) to make up the field width. The padding character is normally space, but can be 0 if the zero padding flag is present.
- A period, which separates the field width from the precision.
- A number, the precision, that specifies the maximum number of characters to be printed from a string, or the number of digits to be printed after the decimal point for e, E, or F conversions, or the number of significant digits for g or G conversion, or the minimum number of digits to be printed for an integer (leading 0s will be added to make up the necessary width)
- A length modifier h, l (letter ell), or L. "h" Indicates that the corresponding argument is to be printed as a short or unsigned short; "l" indicates that the argument is along or unsigned long.

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

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

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

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

## scanf

**Note:** This function cannot be used in email messages.

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

***Syntax:***

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

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

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

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

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

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

***Default output:***

```
Write succeeded
Write failed
```

## IncludeFile

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

***Syntax:***

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

***Default output:***

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

## SaveToFile

**Note:** This function cannot be used in email messages.

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

***Syntax:***

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

***Default output:***

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

## SaveDataToFile

**Note:** This function cannot be used in email messages.

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

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

### *Syntax:*

```
<?--#exec cmd_argument='SaveDataToFile( "File name", "Object name",[Ap-
pend|Overwrite] )'-->
```

### *Default output:*

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

## DisplayRemoteUser

**Note:** This function cannot be used in email messages.

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

### *Syntax:*

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

# Changing SSI output

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

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

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

## SSI Output String File

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

The files shall have the following format:

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

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

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

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

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

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

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

***Example:***

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

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

[1] ‘%s’ includes the filename in the string

## Temporary SSI Output change

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

***Syntax:***

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

***Example:***

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

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

# CIP Object Implementation

## General

The following CIP-objects are implemented in this product:

### Mandatory Objects

| Object | Page |
|---|---|
| Identity Object, Class 01h | 9-2 |
| Message Router, Class 02h | 9-4 |
| Assembly Object, Class 04h | 9-5 |
| Connection Manager, Class 06h | 9-7 |
| Port Object, Class F4h | 9-14 |
| TCP/IP Interface Object, Class F5h | 9-15 |
| Ethernet Link Object, Class F6h | 9-18 |

### Vendor Specific Objects

| Object | Page |
|---|---|
| Diagnostic Object, Class AAh | 9-9 |
| I/O Data Input Mapping Object, Class A0h | 9-10 |
| I/O Data Output Mapping Object, Class A1h | 9-11 |
| Parameter Data Input Mapping Object, Class B0h | 9-12 |
| Parameter Data Output Mapping Object, Class B1h | 9-13 |

# Identity Object, Class 01h

## General Information

### Object Description

-

### Supported Services

| | |
|---|---|
| Class services: | Get Attribute All<br>Get Attribute Single |
| Instance services: | Get Attribute All<br>Get Attribute Single<br>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 |
| 2 | Get | Device Type | UINT | Default: 000Ch | Communication Adapter |
| 3 | Get | Product Code | UINT | 000Eh | Single port |
| 4 | Get | Revision | Struct of: | | - |
| | | | USINT | | Major fieldbus version |
| | | | USINT | | Minor fieldbus version |
| 5 | Get | Status | WORD | - | Device status, see table below |
| 6 | Get | Serial Number | UDINT | Serial number | (set at production) |
| 7 | Get | Product Name | SHORT_STRING | ‘Anybus-S EtherNet/IP’ | Name of product |

## Device Status

| bit(s) | Name |
|---|---|
| 0 | Module Owned |
| 1 | (reserved, ignore) |
| 2 | Configured |
| 3 | (reserved, ignore) |
| 4... 7 | Extended Device Status:<br>Value: Meaning:<br>0000b Unknown<br>0010b Faulted I/O Connection<br>0011b No I/O connection established<br>0100b Non-volatile configuration bad<br>0110b Connection in Run mode<br>0111b Connection in Idle mode<br>(other) (reserved) |
| 8 | Set for minor recoverable faults |
| 9 | Set for minor unrecoverable faults |
| 10 | Set for major recoverable faults |
| 11 | Set for major unrecoverable faults |
| 12... 15 | (reserved, ignore) |

## 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.

Upon receiving a reset request, the default behaviour for the module is to reset itself. If the request is of type 1 ('Out of box reset'), the configuration file 'ethcfg.cfg' is erased.

If the application shall be notified about the reset (i.e. for resetting itself), there are two options for this:

1. Issue ENABLE_ID_RESET_NOTIFY to the module during initialisation.

   The reset service will then cause the module to issue ID_RESET_NOTIFY, and if necessary delete the configuration file 'ethcfg.cfg'. ID_RESET_NOTIFY specifies the type of reset request that was issued.

   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 (ANYBUS_INIT.)

   The reset service will then generate an Event Notification. The type of reset request that was issued can be retrieved using GET_ID_RESET_PARAM.

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

See also...

- 10-80 "Reset Notification (ID_RESET_NOTIFY)"
- 10-81 "Get Reset Parameter (GET_ID_RESET_PARAM)"
- 10-84 "Reset on IP Change (RST_ON_IP_CHANGE)"
- A-1 "Event Notification Cause/Source Registers"

# Message Router, Class 02h

## General Information

### Object Description

-

### Supported Services

| | |
|---|---|
| Class services: | - |
| Instance services: | - |

## Class Attributes

-

## Instance Attributes

-

# Assembly Object, Class 04h

## General Information

### Object Description

This object provides access to I/O Data.

See also...

- 2-1 “EtherNet/IP”

### Supported Services

Class services: Get Attribute Single

Instance services: Get Attribute Single
Set Attribute Single

## 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 C6h Attributes (Heartbeat, Input-Only)

This instance is used as heartbeat for Input-Only connections, and does not carry any data.

## Instance C7h Attributes (Heartbeat, Listen-Only)

This instance is used as heartbeat for listen-only connections, and does not carry any data.

## Instance C5h Attributes (Configuration Data)

No attributes are implemented for this instance.

See also...

- 10-84 “Enable Configuration Assembly (ENABLE_CONFIG_SY)”
- 10-85 “Set Configuration Data (SET_CONFIG_DATA)”

## Instance 64h... 69h Attributes (Producing Instances)

| # | Name | Access | Type | Value |
|---|---|---|---|---|
| 3 | Produced Data | Get | Array of BYTE | This data corresponds to the Input I/O Data |

**Note:** By default, only instance 64h is enabled. The remainder can be created by the application using the 'I/O Data Input Area Mapping'-command.

See also...

- 2-1 "Data Exchange"
- 9-10 "I/O Data Input Mapping Object, Class A0h"
- 10-58 "I/O Data Input Area Mapping (IO_INPUT_MAP)"

## Instance 96h... 9Bh Attributes (Consuming Instance)

| # | Name | Access | Type | Value |
|---|---|---|---|---|
| 3 | Consumed Data | Set | Array of BYTE | This data corresponds to the Output I/O Data. |

**Note:** By default, only instance 96h is enabled. The remainder can be created by the application using the 'I/O Data Output Area Mapping'-command.

See also...

- 2-1 "Data Exchange"
- 2-1 "Implementation Notes"
- 9-11 "I/O Data Output Mapping Object, Class A1h"
- 10-60 "I/O Data Output Area Mapping (IO_OUTPUT_MAP)"
- 10-82 "Copy I/O Status (COPY_IO_STATUS)"

# Connection Manager, Class 06h

## General Information

### Object Description

-

### Supported Services

| | |
|---|---|
| Class services: | - |
| Instance services: | Forward_Open<br>Forward_Close<br>Unconnected_Send[1] |

## Instance Descriptions

(No supported instance attributes)

## Class 1 Connection Details

### General

Class 1 connections are used to transfer I/O data, and can be established to instances in the Assembly Object. Each Class 1 connection will establish two data transports; one consuming and one producing. The heartbeat instances can be used for connections that shall only access inputs. Class 1 connections use UDP transport.

- Total number of supported class 1 connections: 31
- Max input connection size: 509 bytes
- Max output connection size:505 bytes
- Supported API: 2... 3200ms
- T->O Connection type: Point-to-point, Multicast
- O->T Connection type: Point-to-point
- Supported trigger type: Cyclic, Change of state

[1] Only if routing has been enabled. See 10-74 “Enable Routing (ENABLE_ROUTING)”.

### Connection Types

- **Exclusive-Owner connection**

  This type of connection controls the outputs of the Anybus module and does not depend on other connections.

| | |
|---|---|
| - Max. no. of Exclusive-Owner connections: | 1 per connection point |
| - Connection point O ⇒ T: | Assembly Object, instance 96h (Default) |
| - Connection point T ⇒ O: | Assembly Object, instance 64h (Default) |

- **Input-Only connection**

  This type of connection is used to read data from the Anybus module without controlling the outputs. It does not depend on other connections.

| | |
|---|---|
| - Max. no. of Input-Only connections: | Up to 31[1] |
| - Connection point O ⇒ T: | Assembly Object, instance 03h (Default) |
| - Connection point T ⇒ O: | Assembly Object, instance 64h (Default) |

**Note:** If an Exclusive-Owner connection has been opened towards the module and times out, the Input-Only connection times out as well. If the Exclusive-Owner connection is properly closed, the Input-Only connection remains unaffected.

- **Listen-Only connection**

  This type of connection requires another connection in order to exist. If that connection (Exclusive-Owner or Input-Only) is closed, the Listen-Only connection will be closed as well.

| | |
|---|---|
| - Max. no. of Input-Only connections: | Up to 31[2] |
| - Connection point O ⇒ T: | Assembly Object, instance 04h (Default) |
| - Connection point T ⇒ O: | Assembly Object, instance 64h (Default) |

- **Redundant-Owner connection**

  This connection type is not supported by the module.

## Class 3 Connection Details

- **Explicit message connection**

  Class 3 connections are used to establish connections towards the message router. Thereafter, the connection is used for explicit messaging. Class 3 connections use TCP transport.

| | |
|---|---|
| - No. of simultaneous Class 3 connections: | 16 |
| - Supported API: | ≥2ms |
| - T->O Connection type: | Point-to-point |
| - O->T Connection type: | Point-to-point |
| - Supported trigger type: | Application |

[1] Shared with Exclusive-Owner and Listen-Only connections

[2] Shared with Exclusive-Owner and Input-Only connections

# Diagnostic Object, Class AAh

## General Information

### Object Description

This object groups diagnostic information.

### Supported Services

Class services: Get Attribute All

Instance services: Get Attribute Single

## Class Attributes

| # | Access | Name | Type | Value |
|---|---|---|---|---|
| 1 | Get | Revision | UINT | 0001h |

## 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 |
| 0Ah | Get | Module Type | UINT | Module Type |
| 0Fh | Get | IN cyclic I/O length | UINT | Size of I/O Input data (in bytes) |
| 11h | Get | IN total length | UINT | Total input data size in bytes (I/O + Parameter Data) |
| 12h | Get | OUT cyclic I/O length | UINT | Size of I/O Output data (in bytes) |
| 14h | Get | OUT total length | UINT | Total Output data size in bytes (I/O + Parameter Data) |

# I/O Data Input Mapping Object, Class A0h

## General Information

### Object Description

This object can be used to access I/O Data, and is set up dynamically by the application. The mapped attribute id:s are also available as vendor specific instance attributes (64h to 69h) in the Assembly Object.

See also...

- 9-5 “Assembly Object, Class 04h”
- 9-11 “I/O Data Output Mapping Object, Class A1h”
- 10-58 “I/O Data Input Area Mapping (IO_INPUT_MAP)”

### Supported Services

| | |
|---|---|
| Class services: | Get Attribute All |
| Instance services: | Get Attribute Single |

## Class Attributes

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

## Instance Attributes, Instance 01h

| # | Access | Name | Type | Description |
|---|---|---|---|---|
| 1 | Get | Data | Array of USINT | Mapped I/O data (also available in Assembly Instance 64h) |
| 2 | Get | Data | Array of USINT | Mapped I/O data (also available in Assembly Instance 65h) |
| 3 | Get | Data | Array of USINT | Mapped I/O data (also available in Assembly Instance 66h) |
| 4 | Get | Data | Array of USINT | Mapped I/O data (also available in Assembly Instance 67h) |
| 5 | Get | Data | Array of USINT | Mapped I/O data (also available in Assembly Instance 68h) |
| 6 | Get | Data | Array of USINT | Mapped I/O data (also available in Assembly Instance 69h) |

# I/O Data Output Mapping Object, Class A1h

## General Information

### Object Description

This object can be used to access I/O Data, and is set up dynamically by the application. The mapped attribute id:s are also available as vendor specific instance attributes (96h to 9Bh) in the Assembly Object.

See also...

- 9-5 “Assembly Object, Class 04h”
- 9-10 “I/O Data Input Mapping Object, Class A0h”
- 10-60 “I/O Data Output Area Mapping (IO_OUTPUT_MAP)”

### Supported Services

| | |
|---|---|
| Class services: | Get Attribute All |
| Instance services: | Get Attribute Single<br>Set Attribute Single |

## Class Attributes

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

## Instance Attributes, Instance 01h

| # | Access | Name | Type | Description |
|---|---|---|---|---|
| 1 | Get/Set | Data | Array of USINT | Mapped I/O data (also available in Assembly Instance 96h) |
| 2 | Get/Set | Data | Array of USINT | Mapped I/O data (also available in Assembly Instance 97h) |
| 3 | Get/Set | Data | Array of USINT | Mapped I/O data (also available in Assembly Instance 98h) |
| 4 | Get/Set | Data | Array of USINT | Mapped I/O data (also available in Assembly Instance 99h) |
| 5 | Get/Set | Data | Array of USINT | Mapped I/O data (also available in Assembly Instance 9Ah) |
| 6 | Get/Set | Data | Array of USINT | Mapped I/O data (also available in Assembly Instance 9Bh) |

# Parameter Data Input Mapping Object, Class B0h

## General Information

### Object Description

This object can be used to access Parameter Data acyclically, and is set up dynamically by the application.

See also...

- 9-13 “Parameter Data Output Mapping Object, Class B1h”
- 10-54 “Parameter Data Input Mapping (PARAMETER_INPUT_MAP)”

### Supported Services

Class services: Get Attribute All

Instance services: Get Attribute Single

## Class Attributes

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

## Instance Attributes, Instance 01h

Each attribute corresponds to a block of Input Parameter Data. Note that the size and location of each block must be specified using by the application.

See also...

- 10-54 “Parameter Data Input Mapping (PARAMETER_INPUT_MAP)”

| # | Access | Name | Type | Description |
|---|---|---|---|---|
| 01h | Get | Data | Array of USINT | Mapped block if Input Parameter Data |
| 02h | Get | Data | Array of USINT | Mapped block if Input Parameter Data |
| 02h | Get | Data | Array of USINT | Mapped block if Input Parameter Data |
| 02h | Get | Data | Array of USINT | Mapped block if Input Parameter Data |
| 02h | Get | Data | Array of USINT | Mapped block if Input Parameter Data |
| 02h | Get | Data | Array of USINT | Mapped block if Input Parameter Data |
| ... | ... | ... | ... | ... |
| 32h | Get | Data | Array of USINT | Mapped block if Input Parameter Data |

# Parameter Data Output Mapping Object, Class B1h

## General Information

### Object Description

This object can be used to access Output Data acyclically, and is set up dynamically by the application.

See also...

- 9-12 “Parameter Data Input Mapping Object, Class B0h”
- 10-56 “Parameter Data Output Area Mapping (PARAMETER_OUTPUT_MAP)”

### Supported Services

Class services: Get Attribute All

Instance services: Get Attribute Single
Set Attribute Single

## Class Attributes

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

## Instance Attributes, Instance 01h

Each attribute corresponds to a block of Output Parameter Data. Note that the size and location of each block must be specified using by the application.

See also...

- 10-56 “Parameter Data Output Area Mapping (PARAMETER_OUTPUT_MAP)”

| # | Access | Name | Type | Description |
|---|---|---|---|---|
| 01h | Get/Set | Data | Array of USINT | Mapped block of Output Parameter Data |
| 02h | Get/Set | Data | Array of USINT | Mapped block of Output Parameter Data |
| 01h | Get/Set | Data | Array of USINT | Mapped block of Output Parameter Data |
| 02h | Get/Set | Data | Array of USINT | Mapped block of Output Parameter Data |
| 01h | Get/Set | Data | Array of USINT | Mapped block of Output Parameter Data |
| 02h | Get/Set | Data | Array of USINT | Mapped block of Output Parameter Data |
| ... | ... | ... | ... | ... |
| 32h | Get/Set | Data | Array of USINT | Mapped block of Output Parameter Data |

# Port Object, Class F4h

## General Information

### Object Description

-

### Supported Services

| | |
|---|---|
| Class services: | Get Attribute All<br>Get Attribute Single |
| Instance services: | Get Attribute All<br>Get Attribute Single |

## 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 |
| 3 | Get | No. of instances | UINT | 0001h | 1 instance is implemented |
| 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<br>0000h 0000h<br>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 | Comments |
|---|---|---|---|---|---|
| 1 | Get | Port Type | UINT | 0000h | TCP/IP |
| 2 | Get | Port Number | UINT | 0002h | Port 2 |
| 3 | Get | Port Object | Struct of: | | |
| | | Path Size | UINT | 0002h | - |
| | | Path | Padded EPATH | 20 F5 24 01h | TCP class, Instance 1 |
| 4 | Get | Port Name | SHORT_STIRNG | ‘TCP/IP’ | Name of port |
| 7 | Get | Node Address | Padded EPATH | - | - |

# TCP/IP Interface Object, Class F5h

## General Information

### Object Description

This object groups TCP/IP-related settings.

See also...

- 3-1 "Basic Network Configuration"

### Supported Services

| | |
|---|---|
| Class services: | Get Attribute All<br>Get Attribute Single |
| Instance services: | Get Attribute All<br>Get Attribute Single<br>Set Attribute Single |

## Class Attributes

| # | Access | Name | Type | Value | Description |
|---|---|---|---|---|---|
| 1 | Get | Revision | UINT | 0002h | Revision 2 |

## Instance Attributes

| # | Access | Name | Type | Value | Comments |
|---|---|---|---|---|---|
| 1 | Get | Status | DWORD | - | - |
| 2 | Get | Configuration Capability | DWORD | Bit: Value: | |
| | | | | 0 0 | BootP client |
| | | | | 1 1 | DNS client |
| | | | | 2 - | DHCP Client<br>Value: Meaning:<br>0 DISABLE_DHCP mailbox received<br>1 Other |
| | | | | 3 0 | DHCP-DNS Update |
| | | | | 4 - | Configuration settable<br>Value: Meaning:<br>0 TCP_IP_RO mailbox received<br>1 Other |
| | | | | 5 - | Hardware Configurable<br>Value: Meaning:<br>0 SET_ETN_CONFIG mailbox<br>1 Other |
| | | | | 6 0 | Interface config requires reset |
| | | | | 7 1 | ACD capable |
| | | | | 8 - 31 | (not used, set to 0) |
| 3 | Get/Set[a] | Configuration Control | DWORD | - | Value: Meaning:<br>0 Configuration from non-volatile memory (i.e. ‘ethcfg.cfg’)<br>2 Configuration from DHCP |
| 4 | Get | Port Object | Struct of: | | Path to Ethernet Class, Instance 1 |
| | | Path Size | UINT | 0002h | |
| | | Path | Padded EPATH | 20 F6 24 01h | |
| 5 | Get/Set[a] | Interface Configuration | Struct of: | | |
| | | IP Address | UDINT | - | IP address |
| | | Subnet Mask | UDINT | - | Subnet mask |
| | | Gateway Address | UDINT | - | Gateway Address |
| | | Name Server 1 | UDINT | - | Primary DNS |
| | | Name Server 2 | UDINT | - | Secondary DNS |
| | | Domain Name | STRING | - | Default domain name |
| 6 | Get/Set[a] | Host Name | STRING | - | Host name |
| 8 | Get/Set | TTL Value | USINT | - | Time-To-Live value for IP multicast packets. Default value is 1. Minimum is 1. Maximum 255. |
| 9 | Get/Set | Mcast Config | Struct of: | | Physical link -> Ethernet object |
| | | Alloc Control | USINT | - | Multicast address allocation control word. Determines how addresses are allocated |
| | | Reserved | USINT | 0 | Shall be 0 |
| | | Num Mcast | UINT | - | Number of IP multicast addresses to allocate for EtherNet/IP (1 - 32) |
| | | Mcast Start Addr | UDINT | - | Starting multicast address from which to begin allocation. |
| 10 | Get/Set | SelectACD | BOOL | - | Enable ACD |

| # | Access | Name | Type | Value | Comments |
|---|---|---|---|---|---|
| 11 | Get/Set | LastConflictDetected | Struct of: | | Last detected address conflict. |
| | | AcdActivity | USINT | - | State of ACD activity when last conflict detected. |
| | | RemoteMAC | Array of 6 USINT | - | MAC address of remote node from the ARP PDU in which a conflict was detected. |
| | | ArpPdu | Array of 28 USINT | - | Copy of the raw ARP PDU in which a conflict was detected. |

# Ethernet Link Object, Class F6h

## General Information

### Object Description

This object groups diagnostic information for the Ethernet interface.

See also...

- 3-1 “Basic Network Configuration”

### Supported Services

| | |
|---|---|
| Class services: | Get Attribute All<br>Get Attribute Single |
| Instance services: | Get Attribute All<br>Get Attribute Single<br>Set Attribute Single |

## Class Attributes

| # | Access | Name | Type | Value | Description |
|---|---|---|---|---|---|
| 1 | Get | Revision | UINT | 0003h | Revision 3 |

## Instance Attributes

| # | Access | Name | Type | Value | Comments |
|---|---|---|---|---|---|
| 1 | Get | Interface Speed | UDINT | 10 or 100 | Actual ethernet interface speed |
| 2 | Get | Interface Flags | DWORD | - | - |
| 3 | Get | Physical Address | Array of 6 USINTS | (MAC ID) | Physical network address |
| 6 | Get/Set | Interface Control | Struct of: | - | Auto-negotiation, duplex, and speed. |
| | | | WORD | | |
| | | | UINT | | |

# Fieldbus Specific Mailbox Commands

## Fault Information

When a mailbox command cannot be processed, the Message Information register in the header of the response will indicate that an error occurred (Consult the Anybus-S Parallel Design Guide for more 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 unsupported |
| 001Dh | Too many open files |
| 001Eh | Failed to create directory |
| 001Fh | Failed to delete directory |
| 0020h | Failed to rename file |
| 0021h | Failed to move file |
| 0022h | Failed to copy file |
| 0023h | Too many open directories |
| 0024h | Failed to open directory or directory not found |
| 0025h | Failed to resolve hostname with DNS |
| 0026h | Timed out resolving hostname with DNS |

# General Configuration Commands

Commands in this category:

| Mailbox Commands | Description | Page |
|---|---|---|
| Set Ethernet Configuration (SET_ETN_CONFIG) | Set network settings | 10-4 |
| Read Ethernet Configuration (READ_ETN_CONFIG) | Retrieve the currently used network settings | 10-5 |
| Get MAC Address (GET_MAC_ADDR) | Retrieve the ethernet MAC address from the module | 10-6 |
| Connection Timeout (CONNECT_TIMEOUT) | Set the connection timeout value | 10-7 |
| Set Host and Domain (SET_HOST_DOMAIN) | Set host and default domain names | 10-8 |
| Get Host and Domain (GET_HOST_DOMAIN) | Returns the currently used host and domain | 10-9 |
| Disable / Enable ARP Config (ARP_CFG_CONTROL) | Disable/enable ARP support | 10-10 |
| Disable / Enable HICP (HICP_CFG_CONTROL) | Disable/enable HICP support | 10-11 |
| Disable DHCP (DISABLE_DHCP) | Disable DHCP support | 10-12 |
| Set DNS Server Settings (SET_DNS_SERVERS) | Configures the DNS server settings | 10-13 |
| Get DNS Server Settings (GET_DNS_SERVERS) | Returns the currently used DNS server settings | 10-14 |
| Set SMTP Server (SET_SMTP_SERVER) | Configures the SMTP server IP address | 10-15 |
| Set SMTP Server by Name (SET_SMTP_SERVER_NAME) | Configures the SMTP server address | 10-16 |
| Get SMTP Server (GET_SMTP_SERVER) | Returns the currently used SMTP server IP address | 10-17 |
| Disable Web Server (DISABLE_WEB_SERVER) | This command disables the built in web server | 10-18 |
| Enable Web Server (ENABLE_WEB_SERVER) | This command enables the built in web server | 10-19 |
| Disable FTP server (DISABLE_FTP_SERVER) | This command disables the built in FTP server | 10-20 |
| Disable Telnet Server (DISABLE_TELNET_SERVER) | This command disables the built in telnet server | 10-21 |
| Global Admin Mode (GLOBAL_ADMIN_MODE) | This command instruct the module to run in global admin mode | 10-22 |
| Disable Virtual File System (DISABLE_VFS) | Disable the virtual file system | 10-23 |
| Set Serial Number (SET_SERIAL_NUMBER) | Set the serial number of the module | 10-24 |
| Set HTTP Server Name (SET_HTTP_SERVER) | Set the web server name reported in the http header | 10-25 |
| Set SMTP Port Number (SET_SMTP_PORT) | Change the port number used for SMTP | 10-26 |

# Set Ethernet Configuration (SET_ETN_CONFIG)

## Description

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

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

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

## Command and response layout

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

# Read Ethernet Configuration (READ_ETN_CONFIG)

## Description

This command returns the currently used network settings.

**Note:** This command may only be issued during runtime.

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

## Command and response layout

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

## Get MAC Address (GET_MAC_ADDR)

### Description

This command returns the MAC address of the module.

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

### Command and response layout

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

# Connection Timeout (CONNECT_TIMEOUT)

## Description

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

**Note:** This command may only be issued during initialization.

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

## Command and response layout

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

- **Timeout Value**

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

# Set Host and Domain (SET_HOST_DOMAIN)

## Description

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

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

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

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

## Command and response layout

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

# Get Host and Domain (GET_HOST_DOMAIN)

## Description

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

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

## Command and response layout

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

# Disable / Enable ARP Config (ARP_CFG_CONTROL)

## Description

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

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

## Command and response layout

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

- **Config Value**

| | |
|---|---|
| 00h: | Disable |
| 01h: | Enable |

# Disable / Enable HICP (HICP_CFG_CONTROL)

## Description

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

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

## Command and response layout

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

- **Config Value**

| | |
|---|---|
| 00h: | Disable |
| 01h: | Enable |

# Disable DHCP (DISABLE_DHCP)

## Description

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

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

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

## Command and response layout

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

# Set DNS Server Settings (SET_DNS_SERVERS)

## Description

This command configures the DNS server settings.

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

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

## Command and response layout

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

- **Primary DNS**
  IP address to primary DNS server.
- **Secondary DNS**
  IP address to secondary DNS server, or 0.0.0.0 when using primary DNS only.

## Get DNS Server Settings (GET_DNS_SERVERS)

### Description

This command returns the currently used DNS server settings.

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

### Command and response layout

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

- **Primary DNS**
  IP address to primary DNS server, or 0.0.0.0 if unused.
- **Secondary DNS**
  IP address to secondary DNS server, or 0.0.0.0 if unused.

# Set SMTP Server (SET_SMTP_SERVER)

## Description

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

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

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

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

## Command and response layout

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

## Set SMTP Server by Name (SET_SMTP_SERVER_NAME)

### Description

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

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

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

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

### Command and response layout

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

- **SMTP Server**

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

# Get SMTP Server (GET_SMTP_SERVER)

## Description

This function returns the currently used SMTP server IP address.

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

## Command and response layout

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

# Disable Web Server (DISABLE_WEB_SERVER)

## Description

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

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

## Command and response layout

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

# Enable Web Server (ENABLE_WEB_SERVER)

## Description

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

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

## Command and response layout

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

# Disable FTP server (DISABLE_FTP_SERVER)

## Description

This command disables the FTP server.

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

## Command and response layout

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

# Disable Telnet Server (DISABLE_TELNET_SERVER)

## Description

This command disables the onboard Telnet server.

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

## Command and response layout

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

# Global Admin Mode (GLOBAL_ADMIN_MODE)

## Description

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

**Note:** This command may only be issued during initialization.

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

## Command and response layout

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

# Disable Virtual File System (DISABLE_VFS)

## Description

This command disables the virtual files in the file system.

**Note:** This command may only be issued during initialization.

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

## Command and response layout

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

# Set Serial Number (SET_SERIAL_NUMBER)

## Description

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

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

## Command and response layout

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

# Set HTTP Server Name (SET_HTTP_SERVER)

## Description

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

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

## Command and response layout

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

# Set SMTP Port Number (SET_SMTP_PORT)

## Description

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

**Note:** This command may only be issued during initialization.

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

## Command and response layout

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

# Mailbox File System Interface

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

Commands in this category:

| Mailbox Command | Description | Page |
|---|---|---|
| Open File (FILE_OPEN) | Open a file for reading, writing, or appending. | 10-28 |
| Close File (FILE_CLOSE) | Close a file previously opened using FILE_OPEN | 10-29 |
| Read File (FILE_READ) | Read data from a file | 10-30 |
| Write File (FILE_WRITE) | Write data to a file. | 10-31 |
| Delete File (FILE_DELETE) | Delete a file | 10-32 |
| Move File (FILE_MOVE) | Moves a file | 10-32 |
| Rename File (FILE_RENAME) | Rename a file | 10-33 |
| Copy File (FILE_COPY) | Copy a file | 10-35 |
| Create Directory (DIR_CREATE) | Create a new directory | 10-35 |
| Delete Directory (DIR_DELETE) | Delete an empty directory | 10-36 |
| Open Directory (DIR_OPEN) | Open a directory | 10-37 |
| Read Directory (DIR_READ) | Read contents of a directory previously opened using DIR_OPEN | 10-38 |
| Close Directory (DIR_CLOSE) | Close a directory previously opened using DIR_OPEN | 10-40 |
| Create RAM disc (CREATE_RAM_DISC) | Mounts the RAM disc into a specified directory in the file system. | 10-40 |
| Format file system (FORMAT_FS) | Formats the filesystem. | 10-41 |
| File system CRC (CRC_FS) | Calculate and return the CRC for the Used Sector Field and the File System | 10-43 |

# Open File (FILE_OPEN)

## Description

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

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

## Command and response layout

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

- **Mode**

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

- **Filesize**

  Current filesize (if applicable).

- **File Handle**

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

# Close File (FILE_CLOSE)

## Description

This command closes a file previously opened using FILE_OPEN.

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

## Command and response layout

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

- **File Handle**

  Handle of the file to close. See also 10-28 “File Handle”.

- **Filesize**

  Size of the file.

# Read File (FILE_READ)

## Description

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

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

## Command and response layout

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

- **File Handle**

  File handle of the file to read data from. See also 10-28 “File Handle”.

- **No. of bytes**

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

- **Data**

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

# Write File (FILE_WRITE)

## Description

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

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

## Command and response layout

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

- **File Handle**

  File handle of the file to write data to. See also 10-28 "File Handle".

- **No. of bytes**

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

- **Data**

  The actual data that shall be written.

# Delete File (FILE_DELETE)

## Description

This mailbox command deletes a file from the file system.

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

## Command and response layout

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

# Move File (FILE_MOVE)

## Description

This command renames a file in the filesystem.

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

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

### Command and response layout

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

## Rename File (FILE_RENAME)

### Description

This command renames a file in the filesystem.

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

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

### Command and response layout

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

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

# Copy File (FILE_COPY)

## Description

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

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

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

## Command and response layout

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

# Create Directory (DIR_CREATE)

## Description

This command creates a directory in the file system.

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

### Command and response layout

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

## Delete Directory (DIR_DELETE)

### Description

This command deletes an empty directory from the file system.

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

### Command and response layout

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

| | | |
|---|---|---|
| **Extended word 7** | - | - |
| **Extended word 8** | - | Fault information |
| **Message data** | Path + name<br>(String, null-terminated) | Path + name<br>(String, null-terminated) |

## Open Directory (DIR_OPEN)

### Description

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

See also 10-39 “Reading the Contents of a Directory”.

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

### Command and response layout

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

- **Directory Handle**
  Unique identifier which must be used on all further operations associated with the directory.

# Read Directory (DIR_READ)

## Description

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

See also 10-39 “Reading the Contents of a Directory”.

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

## Command and response layout

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

- **Directory Handle**

  Unique identifier which must be used on all further operations associated with the directory.
- **Object Size**

  Size of object (i.e. filesize).
- **Object Flags**

  Various flags specifying the nature of the object:

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

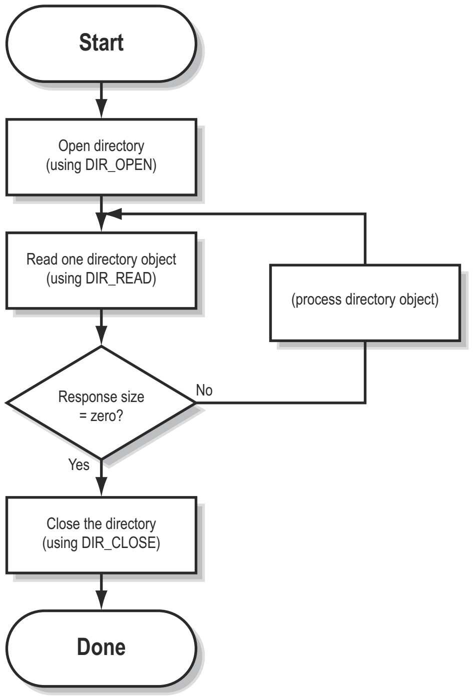
| Bit | Description | |
|---|---|---|
| DIR | Directory flag | 0: Object is a file<br>1: Object is a directory |
| RO | Read only | 0: Object can be read or written<br>1: Object is read-only |
| H | Hidden | 0: Object is visible<br>1: Object is hidden |
| SYS | System | 0: User object<br>1: System object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

- **Object Name**

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

## Reading the Contents of a Directory

The following flowchart illustrates the process of reading the contents of a directory:



# Close Directory (DIR_CLOSE)

## Description

This command closes a directory previously opened using DIR_OPEN.

See also 10-39 “Reading the Contents of a Directory”.

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

## Command and response layout

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

# Create RAM disc (CREATE_RAM_DISC)

## Description

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

- The directory must be empty

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

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

### Command and response layout

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

## Format file system (FORMAT_FS)

### Description

This command formats the file system.

- The directory must be empty

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

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

### Command and response layout

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

# File system Checksum (CRC_FS)

## Description

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

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

## Command and response layout

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

# Modbus/TCP Specific Commands

Commands in this category:

| Mailbox Commands | Description | Page |
|---|---|---|
| Modbus Connection Timeout (MB_CONN_TIMEOUT) | Set the timeout value for Modbus/TCP connections | 10-44 |
| Disable Modbus/TCP server (DISABLE_MB_TCP) | Disables the Modbus/TCP server | 10-45 |
| Modbus ON/OFF Line Configuration (MB_ON_OFF_LINE_CONFIG) | Configures the Modbus ON/OFF line setting | 10-47 |
| Write Exception Status (WR_EXP_STAT) | Defines the response to the Modbus command 'Read Exception Status' | 10-48 |
| Set Modbus Coil Area Size (MB_SET_COIL_AREA_SIZE) | This command affects the Modbus addressing scheme and defines part of the I/O data as coil-based | 10-48 |

## Modbus Connection Timeout (MB_CONN_TIMEOUT)

### Description

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

Valid timeout values are:

- 0 - No timeout is used for Modbus connections

- 10 - 65535 - Timeout value in seconds.

The default value is 60 seconds.

**Note:** This command may only be issued during initialization.

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

### Command and response layout

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

## Disable Modbus/TCP server (DISABLE_MB_TCP)

### Description

This mailbox command disables the Modbus/TCP server.

**Note:** This may only be issued during initialization.

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

### Command and response layout

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

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

# Modbus ON/OFF Line Configuration (MB_ON_OFF_LINE_CONFIG)

## Description

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

**Note:** This command overrides the configuration stored in the configuration file ‘onoffln.cfg’.

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

## Command and response layout

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

- **Timeout Value**

  A value of 1 equals 100ms.

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

- **Trigger Commands**

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

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

## Write Exception Status (WR_EXP_STAT)

### Description

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

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

### Command and response layout

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

## Set Modbus Coil Area Size (MB_SET_COIL_AREA_SIZE)

### Description

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

See also...

- 2-2 “Modbus/TCP”

**Note:** This command may only be issued during initializtion.

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

### Command and response layout

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

- **Coil Size In**

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

- **Coil Size Out**

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

# EtherNet/IP Specific Commands

Commands in this category:

| Command | Description | Page |
|---|---|---|
| Set Product Info All (PRODUCT_INFO_ALL) | Customizes EtherNet/IP product info | 10-52 |
| Set Product Info (SET_PRODUCT_INFO) | | 10-52 |
| Set Product Code (SET_PRODUCT_CODE) | | 10-53 |
| Parameter Data Input Mapping (PARAMETER_INPUT_MAP) | Maps Parameter Data to the Parameter Input/Output Objects | 10-54 |
| Parameter Data Output Area Mapping (PARAMETER_OUTPUT_MAP) | | 10-56 |
| I/O Data Input Area Mapping (IO_INPUT_MAP) | Maps I/O data to the Assembly Object | 10-58 |
| I/O Data Output Area Mapping (IO_OUTPUT_MAP) | | 10-60 |
| Send UCMM (SEND_UCMM) | Sends an unconnected message to another node | 10-62 |
| UCMM Request (UCMM_REQUEST) | Issued by the module when a UCMM request is addressed to a user-registered object | 10-64 |
| Enable Large UCMM Request (ENABLE_LARGE_UCMM_REQUEST) | Enables the possibility to send UCMM requests larger than 256 | 10-65 |
| Large UCMM Request (LARGE_UCMM_REQUEST) | Issued by the module when a large UCMM request is addressed to a user-registered object | 10-66 |
| Register Class (REGISTER_CLASS) | Registers an EtherNet/IP object in the message router object | 10-71 |
| Deregister Class (DEREGISTER_CLASS) | De-registers a previously registered class from the message router | 10-72 |
| Enable Routing (ENABLE_ROUTING) | Enables the routing and bridging functionality | 10-74 |
| Register Port (REGISTER_PORT) | Registers a port in the port object | 10-74 |
| Route Unconnected Send (ROUTE_REQUEST) | Issued by the module upon receiving an unconnected send message | 10-77 |
| Enable Reset Notification (ENABLE_ID_RESET_NOTIFY) | Enables Identity Object reset notification | 10-78 |
| Reset Notification (ID_RESET_NOTIFY) | Reset notification | 10-80 |
| Get Reset Parameter (GET_ID_RESET_PARAM) | Get reset parameter | 10-81 |
| Copy I/O Status (COPY_IO_STATUS) | Changes the default behaviour regarding the I/O Status bytes | 10-82 |
| Disable EtherNet/IP (DISABLE_ETHERNET_IP) | Disables the EtherNet/IP protocol support | 10-82 |
| Change Ethernet Port (CHANGE_ETHERNET_PORT) | Changes the port number reported in the Port Object | 10-83 |
| Reset on IP Change (RST_ON_IP_CHANGE) | Causes the module to issue a reset request to the identity object when the IP configuration has been changed over EtherNet/IP | 10-84 |
| Enable Configuration Assembly (ENABLE_CONFIG_SY) | Enables the application to take advantage of the configuration data in the Forward_Open-request. | 10-84 |
| Set Configuration Data (SET_CONFIG_DATA) | Issued by the module upon receiving the Forward_Open-request | 10-85 |
| Enable Exact IO Match (ENABLE_IO_MATCH) | Forces the module to only accept IO connection requests which exactly matches the sizes specified in Anybus_INIT | 10-87 |

| Command | Description | Page |
|---|---|---|
| TCP/IP Object Read-only (TCP_IP_RO) | Disable possibility to set IP address etc in TCP/IP object | 10-87 |
| Get Configuration Data (GET_CONFIG_DATA) | Sent upon receiving a Get_Single_Attribute service command to the configuration data instance. | 10-89 |
| Set assembly instances SET_SY_INSTANCES | Set custom instance numbers for the Assembly object. | 10-90 |

## 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 issued during module initialisation.

| | |
|---|---|
| **Initiated by** | Application |
| **Command no.** | 0089h |
| **Extended Header** | - |
| **Message data** | Vendor ID, Device Type, Product Code, Major/Minor Revision, Name length, Name (ascii) |
| **Response data** | (the response holds a copy of the command data) |

### Command and response layout

| | Command | Expected response | |
|---|---|---|---|
| **Message ID** | (ID) | (ID) | |
| **Message information** | 4002h | 0002h | *Fieldbus Specific Message* |
| **Command** | 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 issued during module initialisation.

| | |
|---|---|
| **Initiated by** | Application |
| **Command no.** | 0082h |
| **Extended Header** | - |
| **Message data** | Vendor ID, Product code, Name length, Name |
| **Response data** | (the response holds a copy of the command data) |

### Command and response layout

| | Command | Expected response | |
|---|---|---|---|
| **Message ID** | (ID) | (ID) | |
| **Message information** | 4002h | 0002h | *Fieldbus Specific Message* |
| **Command** | 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 issued during module initialisation.

| | |
|---|---|
| **Initiated by** | Application |
| **Command no.** | 0083h |
| **Extended Header** | - |
| **Message data** | Product code |
| **Response data** | (the response holds a copy of the command data) |

### Command and response layout

| | Command | Expected response | |
|---|---|---|---|
| **Message ID** | (ID) | (ID) | |
| **Message information** | 4002h | 0002h | *Fieldbus Specific Message* |
| **Command** | 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 command maps blocks of Input Parameter Data to the Parameter Data Input Mapping Object, allowing a ‘Get_Attribute_Single’-request to return a specified block of data. Up to 50 blocks can be mapped this way.

Attributes are mapped in sequence, i.e. the first mapping specified corresponds to attribute #1, the second to attribute #2 etc. The size and location of each block is specified separately for each attribute.

If zero length is specified for an attribute, that attribute will not be mapped. This way, it’s possible to for example map only attributes 1 and 10 by specifying zero length for attributes 2 through 9. It is only necessary to specify mapping information up to the last used attribute; the remainder will not be mapped.

If the mapping information for an attribute is invalid, the length and offset for that attribute will be set to zero in the response, and the attribute will not be mapped.

See also...

- 9-12 “Parameter Data Input Mapping Object, Class B0h”
- 10-56 “Parameter Data Output Area Mapping (PARAMETER_OUTPUT_MAP)”

**Note:** This command may only be issued during module initialisation, after Anybus_INIT.

| | |
|---|---|
| **Initiated by** | Application |

| | |
|---|---|
| **Command no.** | 0084h |
| **Extended Header** | Fault information |
| **Message data** | Mapping information |
| **Response data** | (the response holds a copy of the command data) |

### 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 Output Area Mapping (PARAMETER_OUTPUT_MAP)

### Description

This command maps blocks of Output Parameter Data to the Parameter Data Output Mapping Object, allowing a ‘Set_Attribute_Single’- or ‘Get_Attribute_Single’-request to set/return a specified block of data. Up to 50 blocks can be mapped this way.

Attributes are mapped in sequence, i.e. the first mapping specified corresponds to attribute #1, the second to attribute #2 etc. The size and location of each block is specified separately for each attribute.

If zero length is specified for an attribute, that attribute will not be mapped. This way, it’s possible to for example map only attributes 1 and 10 by specifying zero length for attributes 2 through 9. It is only necessary to specify mapping information up to the last used attribute; the remainder will not be mapped.

If the mapping information for an attribute is invalid, the length and offset for that attribute will be set to zero in the response, and the attribute will not be mapped.

See also...

- 9-13 “Parameter Data Output Mapping Object, Class B1h”

- 10-54 “Parameter Data Input Mapping (PARAMETER_INPUT_MAP)”

**Note:** This command may only be issued during module init.

| | |
|---|---|
| **Initiated by** | Application |
| **Command no.** | 0085h |
| **Extended Header** | Fault information |
| **Message data** | Offset and length of the instances to map |
| **Response data** | (the response holds a copy of the command data) |

### 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 (IO_INPUT_MAP)

### Description

This command maps blocks of Input I/O Data to the I/O Data Input Mapping Object. Attributes are mapped in sequence, i.e. the first mapping specified corresponds to attribute #1, the second to attribute #2 etc. The size and location of each block is specified separately for each attribute. Up to 6 blocks can be mapped this way.

The mapped attribute will be mirrored to the Assembly Object, instances 64h... 69h. If no mapping information is specified (i.e. if this command is never issued), all Input I/O data will be grouped in Assembly Object instance 64h.

If zero length is specified for an attribute, that attribute will not be mapped. This way, it's possible to for example map only attributes 1 and 6 by specifying zero length for attributes 2 through 5. It is only necessary to specify mapping information up to the last used attribute; the remainder will not be mapped.

If the mapping information for an attribute is invalid, the length and offset for that attribute will be set to zero in the response, and the attribute will not be mapped.

See also...

- 9-5 "Assembly Object, Class 04h"

- 9-10 “I/O Data Input Mapping Object, Class A0h”
- 10-60 “I/O Data Output Area Mapping (IO_OUTPUT_MAP)”

**Note:** This command may only be issued during module initialisation., after Anybus_INIT.

| | |
|---|---|
| **Initiated by** | Application |
| **Command no.** | 0086h |
| **Extended Header** | Fault information |
| **Message data** | Offset and length of the instances to map |
| **Response data** | (the response holds a copy of the command data) |

### Command and response layout

| | Command | Expected response | |
|---|---|---|---|
| **Message ID** | (ID) | (ID) | |
| **Message information** | 4002h | 0002h | *Fieldbus Specific Message* |
| **Command** | 0086h | 0086h | *IO_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 | |
| **Message data word 11** | Offset | Offset | *Instance 69h* |
| **Message data word 12** | Length | Length | |

## I/O Data Output Area Mapping (IO_OUTPUT_MAP)

### Description

This command maps blocks of Output I/O Data to the I/O Data Output Mapping Object. Attributes are mapped in sequence, i.e. the first mapping specified corresponds to attribute #1, the second to attribute #2 etc. The size and location of each block is specified separately for each attribute. Up to 6 blocks can be mapped this way.

The mapped attribute will be mirrored to the Assembly Object, instances 96h... 6Bh. If no mapping information is specified (i.e. if this command is never issued), all Output I/O data will be grouped in Assembly Object instance 96h.

If zero length is specified for an attribute, that attribute will not be mapped. This way, it's possible to for example map only attributes 1 and 6 by specifying zero length for attributes 2 through 5. It is only necessary to specify mapping information up to the last used attribute; the remainder will not be mapped.

If the mapping information for an attribute is invalid, the length and offset for that attribute will be set to zero in the response, and the attribute will not be mapped.

See also...

- 9-5 "Assembly Object, Class 04h"
- 9-11 "I/O Data Output Mapping Object, Class A1h"

- 10-58 “I/O Data Input Area Mapping (IO_INPUT_MAP)”

**Note:** This command may only be issued during module initialisation., after Anybus_INIT.

| | |
|---|---|
| **Initiated by** | Application |
| **Command no.** | 0087h |
| **Extended Header** | Fault information |
| **Message data** | Offset and length of the instances to map |
| **Response data** | (the response holds a copy of the command data) |

### Command and response layout

| | Command | Expected response | |
|---|---|---|---|
| **Message ID** | (ID) | (ID) | |
| **Message information** | 4002h | 0002h | *Fieldbus Specific Message* |
| **Command** | 0087h | 0087h | *IO_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 | |
| **Message data word 11** | Offset | Offset | *Instance 9Bh* |
| **Message data word 12** | 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.

| | |
|---|---|
| **Initiated by** | Application |
| **Command no.** | 008Ah |
| **Extended Header** | Destination IP address |
| **Message data** | Unconnected message request |
| **Response data** | Unconnected message response |

### 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**<br>**...** | Padded EPATH[a] | | |
| **...**<br>**Message data byte n** | (optional service data) | | |

[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 10-71 “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.

| | |
|---|---|
| **Initiated by** | Anybus |
| **Command no.** | 008Dh |
| **Extended Header** | - |
| **Message data** | Explicit message request |
| **Response data** | Requested data (or an error code) |

### 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’

## Enable Large UCMM Request (ENABLE_LARGE_UCMM_REQUEST)

### Description

This mailbox command enables the possibility for the module to send UCMM requests larger than 256 bytes to application registered CIP classes (See 10-66 “Large UCMM Request (LARGE_UCMM_REQUEST)”).

When this command has been sent to the module, all UCMM_REQUEST mailbox commands will be replaced by LARGE_UCMM_REQUEST mailbox commands.

This command must be sent to the module during initalization.

| | |
|---|---|
| **Initiated by** | Anybus |
| **Command no.** | 00D0h |
| **Extended Header** | - |
| **Message data** | No message data |
| **Response message** | (the response holds a copy of the command data) |

### Command and response layout

| | Command | Expected response | |
|---|---|---|---|
| **Message ID** | (ID) | (ID) | |
| **Message information** | 4002h | 0002h | *Fieldbus Specific Message* |
| **Command** | 00D0h | 00D0h | *ENABLE_LARGE_UCMM_ REQUEST* |
| **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** | - | - | |

## Large UCMM Request (LARGE_UCMM_REQUEST)

### Description

This message is used when the application has registered a CIP class (See 10-71 “Register Class (REGISTER_CLASS)” in the Anybus module, and an explicit message request has been sent to this class from a EtherNet/IP client.

The application will have to process the message, and respond to the Anybus module with the data necessary to generate a response to the explicit message request for the class.

### Command and response layout

| | Command | Expected response | |
|---|---|---|---|
| **Message ID** | (ID) | (ID) | |
| **Message information** | 4002h | 0002h | *Fieldbus Specific Message* |
| **Command** | 00D1h | 00D1h | *LARGE_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** | Fragment type | Fragment type | *Fragmentation information* |
| **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 | Service Reply | |
| 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 | | Optional data | |

[a] Request path according to the EtherNet/IP Specification, Appendix C - ‘Data Management’

### Fragment Type

| Fragment Type | Description |
|---|---|
| 0000h | First fragment of new message |
| 0001h | Subsequent fragment |
| 0002h | Last fragment. Signals the end of the fragmented data. |

## Example 1

This example mailbox is from a CIP UCMM request to the application registered class CCh, instance 1, attribute 2. The object specific service code 54h is used. 6 bytes of data are sent (11h, 22h, 33h, 44h, 55h and 66h) in 3 mailbox fragments. The application returns 6 bytes of payload data (12h, 34h, 56h, 78h, 9Ah and BCh) in 3 mailbox fragments.

Please note that this is an example. The actual fragmentation from the module will be in chunks of 256 bytes.

| | Command | Expected response | |
|---|---|---|---|
| **Message ID** | 0001h | 0001h | |
| **Message information** | 4002h | 0002h | |
| **Command** | 00D1h | 00D1h | |
| **Data size** | 000Ah | 0000h | *Number of bytes* |
| **Frame count** | 0001h | 0001h | |
| **Frame number** | 0001h | 0001h | |
| **Offset high** | 0000h | 0000h | |
| **Offset low** | 0000h | 0000h | |
| **Extended word 1** | 0000h | 0000h | *Fragmentation information* |
| **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** | 54h | | |
| **Message data byte 2** | 03h | | |
| **Message data byte 3** | 20h | | |
| **Message data byte 4** | CCh | | |
| **Message data byte 5** | 24h | | |
| **Message data byte 6** | 01h | | |
| **Message data byte 7** | 30h | | |
| **Message data byte 8** | 02h | | |
| **Message data byte 9** | 11h | | |
| **Message data byte 10** | 22h | | |

| | Command | Expected response | |
|---|---|---|---|
| **Message ID** | 0001h | 0001h | |
| **Message information** | 4002h | 0002h | |
| **Command** | 00D1h | 00D1h | |
| **Data size** | 0002h | 0000h | *Number of bytes* |
| **Frame count** | 0001h | 0001h | |
| **Frame number** | 0001h | 0001h | |
| **Offset high** | 0000h | 0000h | |
| **Offset low** | 0000h | 0000h | |
| **Extended word 1** | 00001h | 0000h | *Fragmentation information* |
| **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** | 33h | | |
| **Message data byte 2** | 44h | | |

| | Command | Expected response | |
|---|---|---|---|
| **Message ID** | 0001h | 0001h | |
| **Message information** | 4002h | 0002h | |
| **Command** | 00D1h | 00D1h | |
| **Data size** | 0002h | 0006h | *Number of bytes* |
| **Frame count** | 0001h | 0001h | |
| **Frame number** | 0001h | 0001h | |
| **Offset high** | 0000h | 0000h | |
| **Offset low** | 0000h | 0000h | |
| **Extended word 1** | 0002h | 0000h | *Fragmentation information* |
| **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** | 55h | D4h | |
| **Message data byte 2** | 66h | 00h | |
| **Message data byte 3** | | 00h | |
| **Message data byte 4** | | 00h | |
| **Message data byte 5** | | 12h | |
| **Message data byte 6** | | 34h | |

|  | Command | Expected response |  |
|---|---|---|---|
| **Message ID** | 0001h | 0001h |  |
| **Message information** | 4002h | 0002h |  |
| **Command** | 00D1h | 00D1h |  |
| **Data size** | 0000h | 0002h | *Number of bytes* |
| **Frame count** | 0001h | 0001h |  |
| **Frame number** | 0001h | 0001h |  |
| **Offset high** | 0000h | 0000h |  |
| **Offset low** | 0000h | 0000h |  |
| **Extended word 1** | 0000h | 0001h | *Fragmentation information* |
| **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** |  | 56h |  |
| **Message data byte 2** |  | 78h |  |

|  | Command | Expected response |  |
|---|---|---|---|
| **Message ID** | 0001h | 0001h |  |
| **Message information** | 4002h | 0002h |  |
| **Command** | 00D1h | 00D1h |  |
| **Data size** | 000Ah | 0000h | *Number of bytes* |
| **Frame count** | 0001h | 0001h |  |
| **Frame number** | 0001h | 0001h |  |
| **Offset high** | 0000h | 0000h |  |
| **Offset low** | 0000h | 0000h |  |
| **Extended word 1** | 0000h | 0002h | *Fragmentation information* |
| **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** |  | 9Ah |  |
| **Message data byte 2** |  | BCh |  |

## Example 2

This example is from a CIP UCMM request to the application registered class CCh, instance 1, attribute 3. The service code 0Eh is used. The application returns 6 bytes of payload data (AAh, BBh, CCh, DDh, EEh and FFh).

| | Command | Expected response | |
|---|---|---|---|
| **Message ID** | 0001h | 0001h | |
| **Message information** | 4002h | 0002h | |
| **Command** | 00D1h | 00D1h | |
| **Data size** | 0008h | 000Ah | *Number of bytes* |
| **Frame count** | 0001h | 0001h | |
| **Frame number** | 0001h | 0001h | |
| **Offset high** | 0000h | 0000h | |
| **Offset low** | 0000h | 0000h | |
| **Extended word 1** | 0002h | 0002h | *Fragmentation information* |
| **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** | 0Eh | 8Eh | |
| **Message data byte 2** | 03h | 00h | |
| **Message data byte 3** | 20h | 00h | |
| **Message data byte 4** | CCh | 00h | |
| **Message data byte 5** | 24h | AAh | |
| **Message data byte 6** | 01h | BBh | |
| **Message data byte 7** | 30h | CCh | |
| **Message data byte 8** | 03h | DDh | |
| **Message data byte 9** | | EEh | |
| **Message data byte 10** | | FFh | |

## 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 10-64 “UCMM Request (UCMM_REQUEST)”.

| | |
|---|---|
| **Initiated by** | Application |
| **Command no.** | 008Bh |
| **Extended Header** | - |
| **Message data** | Class ID |
| **Response data** | (the response holds a copy of the command data) |

### Command and response layout

| | Command | Expected response | |
|---|---|---|---|
| **Message ID** | (ID) | (ID) | |
| **Message information** | 4002h | 0002h | *Fieldbus Specific Message* |
| **Command** | 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

| | |
|---|---|
| **Initiated by** | Application |
| **Command no.** | 008Eh |
| **Extended Header** | - |
| **Message data** | Class ID |
| **Response data** | (the response holds a copy of the command data) |

### Command and response layout

| | Command | Expected response | |
|---|---|---|---|
| **Message ID** | (ID) | (ID) | |
| **Message information** | 4002h | 0002h | *Fieldbus Specific Message* |
| **Command** | 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.

| | |
|---|---|
| **Initiated by** | Application |
| **Command no.** | 0091h |
| **Extended Header** | - |
| **Message data** | - |
| **Response data** | - |

### Command and response layout

| | **Command** | **Expected response** | |
|---|---|---|---|
| **Message ID** | (ID) | (ID) | |
| **Message information** | 4002h | 0002h | *Fieldbus Specific Message* |
| **Command** | 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 10-74 “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.

| | |
|---|---|
| **Initiated by** | Application |
| **Command no.** | 0090h |
| **Extended Header** | - |
| **Message data** | Instance attributes 1,2,3,4 and 7. |
| **Response data** | (the response holds a copy of the command data) |

### Command and response layout

| | Command | Expected response | |
|---|---|---|---|
| **Message ID** | (ID) | (ID) | |
| **Message information** | 4002h | 0002h | *Fieldbus Specific Message* |
| **Command** | 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**<br>**...**<br>**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 10-74 "Enable Routing (ENABLE_ROUTING)".

| | |
|---|---|
| **Initiated by** | Anybus |
| **Command no.** | 008Fh |
| **Extended Header** | - |
| **Message data** | The unconnected send message received by the module |
| **Response data** | 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 | |
| | Request Path Size | | **Response data byte n** |
| | Req. Path (Padded EPATH) | | |
| | Request Data | *Unsuccessful response:* | |
| | | Transaction ID (low) | **Response data byte 1** |
| | | Transaction ID (high) | **Response data byte 2** |
| | 00h (PAD)<br>(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 10-80 “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.

| | |
|---|---|
| **Initiated by** | Application |
| **Command no.** | 0092h |
| **Extended Header** | - |
| **Message data** | Notify type |

| | |
|---|---|
| **Response data** | (the response holds a copy of the command data) |

## Command and response layout

| | Command | Expected response | |
|---|---|---|---|
| **Message ID** | (ID) | (ID) | |
| **Message information** | 4002h | 0002h | *Fieldbus Specific Message* |
| **Command** | 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 10-78 “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.

| | |
|---|---|
| **Initiated by** | Anybus |
| **Command no.** | 0093h |
| **Extended Header** | - |
| **Message data** | Reset Type |
| **Response data** | (the response holds a copy of the command data) |

## Command and response layout

| | Command | Expected response | |
|---|---|---|---|
| **Message ID** | (ID) | (ID) | |
| **Message information** | 4002h | 0002h | *Fieldbus Specific Message* |
| **Command** | 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

| | |
|---|---|
| **Initiated by** | Application |
| **Command no.** | 0095h |
| **Extended Header** | - |
| **Message data** | - |
| **Response data** | Reset Type |

## 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.

| | |
|---|---|
| **Initiated by** | Application |
| **Command no.** | 0094h |
| **Extended Header** | - |
| **Message data** | - |
| **Response data** | - |

## Command and response layout

| | Command | Expected response | |
|---|---|---|---|
| **Message ID** | (ID) | (ID) | |
| **Message information** | 4002h | 0002h | *Fieldbus Specific Message* |
| **Command** | 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.

| | |
|---|---|
| **Initiated by** | Application |
| **Command no.** | 000Ah |
| **Extended Header** | - |
| **Message data** | - |
| **Response data** | - |

## Command and response layout

| | Command | Expected response | |
|---|---|---|---|
| **Message ID** | (ID) | (ID) | |
| **Message information** | 4002h | 0002h | *Fieldbus Specific Message* |

| | | | |
|---|---|---|---|
| **Command** | 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 | |

## Change Ethernet Port (CHANGE_ETHERNET_PORT)

### Description

This command changes the port number reported by the Port Object.

| | |
|---|---|
| **Initiated by** | Application |
| **Command no.** | 0096h |
| **Extended Header** | - |
| **Message data** | - |
| **Response data** | - |

### Command and response layout

| | **Command** | **Expected response** | |
|---|---|---|---|
| **Message ID** | (ID) | (ID) | |
| **Message information** | 4002h | 0002h | *Fieldbus Specific Message* |
| **Command** | 0096h | 0096h | *Change Ethernet Port* |
| **Data size** | 0001h | 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** | - | - | |
| **Message Databyte** | Port no. | Port no. | |

- **Port no.**
  Desired port number.

## Reset on IP Change (RST_ON_IP_CHANGE)

### Description

This command instructs the module to issue a reset request towards the Identity Object when attribute #5 in the TCP/IP Interface Object (TCP/IP Interface Object, Class F5h) has been altered. If the Identity Object is registered in the application, this causes a reset message to be sent to the application.

| | |
|---|---|
| **Initiated by** | Application |
| **Command no.** | 0097h |
| **Extended Header** | - |
| **Message data** | - |
| **Response data** | - |

### Command and response layout

| | Command | Expected response | |
|---|---|---|---|
| **Message ID** | (ID) | (ID) | |
| **Message information** | 4002h | 0002h | *Fieldbus Specific Message* |
| **Command** | 0097h | 0097h | *RST_ON_IP_CHANGE* |
| **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 Configuration Assembly (ENABLE_CONFIG_SY)

### Description

This command enables the module (application) to take advantage of the configuration data in a Forward_Open-request, and will cause the module to issue SET_CONFIG_DATA upon reception of a Forward_Open-request that contains a data segment.

**Note**: This command may only be issued during initialization.

| | |
|---|---|
| **Initiated by** | Application |
| **Command no.** | 0098h |
| **Extended Header** | - |
| **Message data** | - |
| **Response data** | The response indicates if the command was accepted. |

### Command and response layout

| | Command | Expected response | |
|---|---|---|---|
| **Message ID** | (ID) | (ID) | |
| **Message information** | 4002h | 0002h | *Fieldbus Specific Message* |
| **Command** | 0098h | 0098h | *ENABLE_CONFIG_SY* |
| **Data size** | 0000h | 0000h | |
| **Frame count** | 0001h | 0001h | |
| **Frame number** | 0001h | 0001h | |
| **Offset high** | 0000h | 0000h | |
| **Offset low** | 0000h | 0000h | |
| **Extended word 1** | - | - | |
| **Extended word 2** | - | - | |
| **Extended word 3** | - | - | |
| **Extended word 4** | - | - | |
| **Extended word 5** | - | - | |
| **Extended word 6** | - | - | |
| **Extended word 7** | - | - | |
| **Extended word 8** | - | - | |

## Set Configuration Data (SET_CONFIG_DATA)

### Description

If enabled, this command is issued spontaneously by the module upon reception of a Forward_Open-request that contains a data segment or a set single attribute service command to the configuration instance.

| | |
|---|---|
| **Initiated by** | Anybus |
| **Command no.** | 0099h |
| **Extended Header** | - |
| **Message data** | - |
| **Response data** | - |

### Command and response layout

| | Command | Expected response | |
|---|---|---|---|
| **Message ID** | (ID) | (ID) | |
| **Message information** | 4002h | 0002h | *Fieldbus Specific Message* |
| **Command** | 0099h | 0099h | *SET_CONFIG_DATA* |
| **Data size** | 0000h | 0002h | |
| **Frame count** | 0001h | 0001h | |
| **Frame number** | 0001h | 0001h | |
| **Offset high** | 0000h | 0000h | |
| **Offset low** | 0000h | 0000h | |
| **Extended word 1** | Fragment type | Fragment type | *Fragmentation information* |
| **Extended word 2** | Producing Connection Point | Producing Connection Point | *If forwarded from forward open request* |
| **Extended word 3** | Consuming Connection Point | Consuming Connection Point | *If forwarded from forward open request* |
| **Extended word 4** | - | - | |
| **Extended word 5** | - | - | |
| **Extended word 6** | - | - | |

| | | |
|---|---|---|
| **Extended word 7** | - | - |
| **Extended word 8** | - | - |
| **Message Data Word 1** | Configuration Data | Error type |
| **Message Data Word 2** | - | Additional error code |

- **Fragment Type**

| Fragment Type | Description |
|---|---|
| 0000h | First fragment of new message |
| 0001h | Subsequent fragment |
| 0002h | Last fragment. Signals the end of the fragmented data. |

- **Fragmented Message Data**

  The data in this message is fragmented, which means that the message will be issued repeatedly until all data has been transferred. The first fragment contains the Producing/Consuming Connection Point, followed by a block of Configuration Data. The remaining fragments only contains Configuration Data.

  The application must monitor the Fragment Type-register. Upon reception of the last fragment, the application shall assemble and process the Configuration Data as a complete message.

- **Error Type**

| Error Type | Description | Additional error code |
|---|---|---|
| 0000h | The configuration data was accepted by the application | - |
| 0001h | The configuration data was invalid | Offset to which data byte that made the data invalid |
| 0002h | Ownership conflict | - |
| 0003h | Not enough configuration data | - |
| 0004h | Too much configuration data | - |

# Enable Exact IO Match (ENABLE_IO_MATCH)

## Description

This command forces the module to only accept IO connection requests which exactly matches the sizes specified in Anybus_INIT.

**Note**: This command may only be issued during module initialization.

| | |
|---|---|
| **Initiated by** | Application |
| **Command no.** | 009Ah |
| **Extended Header** | - |
| **Message data** | - |
| **Response data** | - |

## Command and response layout

| | Command | Expected response | |
|---|---|---|---|
| **Message ID** | (ID) | (ID) | |
| **Message information** | 4002h | 0002h | |
| **Command** | 009Ah | 009Ah | *ENABLE_IO_MATCH* |
| **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** | - | - | |

# TCP/IP Object Read-only (TCP_IP_RO)

## Description

This command disables the possibility to set IP address etc in the EtherNet/IP TCP/IP Object.

**Note**: This command may only be issued during module initialization.

| | |
|---|---|
| **Initiated by** | Application |
| **Command no.** | 009Bh |
| **Extended Header** | - |
| **Message data** | - |
| **Response data** | - |

## Command and response layout

| | Command | Expected response |
|---|---|---|
| **Message ID** | (ID) | (ID) |

| | | | |
|---|---|---|---|
| **Message information** | 4002h | 0002h | |
| **Command** | 009Bh | 009Bh | *TCP_IP_RO* |
| **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** | - | - | |

# Get Configuration Data (GET_CONFIG_DATA)

## Description

This command is sent to the application upon reception of a get_single_attribute service command to the configuration data instance.

**Note**: This command may only be issued during module initialization.

| | |
|---|---|
| **Initiated by** | Anybus |
| **Command no.** | 009Dh |
| **Extended Header** | - |
| **Message data** | - |
| **Response data** | - |

## Command and response layout

| | Command | Expected response | |
|---|---|---|---|
| **Message ID** | (ID) | (ID) | |
| **Message information** | 4002h | 0002h | *Fieldbus Specific Message* |
| **Command** | 009Dh | 009Dh | *GET_CONFIG_DATA* |
| **Data size** | nnnnh | nnnnh | *n byte of data* |
| **Frame count** | 0001h | 0001h | |
| **Frame number** | 0001h | 0001h | |
| **Offset high** | 0000h | 0000h | |
| **Offset low** | 0000h | 0000h | |
| **Extended word 1** | - | Fragment type | |
| **Extended word 2** | - | Total config data size | |
| **Extended word 3** | - | - | |
| **Extended word 4** | - | - | |
| **Extended word 5** | - | - | |
| **Extended word 6** | - | - | |
| **Extended word 7** | - | - | |
| **Extended word 8** | - | - | |
| **Message Data Word** | | Configuration data | |

| Fragment type | Description |
|---|---|
| 0x0000 | First fragment of a new message. |
| 0x0001 | Subsequent fragment of the message. |
| 0x0002 | Last fragment of the message. When this fragment is received the entire message will be sent to the socket. |

# Set Assembly Instances (SET_SY_INSTANCES)

## Description

This command can be used to change the instance numbers in the EtherNet/IP Assembly object.

**Note:** Only available in firmware version 2.xx and later.

**Note:** The  command must be sent to the module during initialization before the I/O_INPUT_MAP or I/O_OUTPUT_MAP mailboxes.

| | |
|---|---|
| **Initiated by** | Anybus |
| **Command no.** | 009Eh |
| **Extended Header** | - |
| **Message data** | Five UINT16, one for each assembly instance |
| **Response data** | - |

## Command and response layout

| | Command | Expected response | |
|---|---|---|---|
| **Message ID** | (ID) | (ID) | |
| **Message information** | 4002h | 0002h | |
| **Command** | 009Eh | 009Eh | *SET_SY_INSTANCES* |
| **Data size** | 000Ah | 000Ah | |
| **Frame count** | 0001h | 0001h | |
| **Frame number** | 0001h | 0001h | |
| **Offset high** | 0000h | 0000h | |
| **Offset low** | 0000h | 0000h | |
| **Extended word 1** | - | Fragment type | |
| **Extended word 2** | - | Total config data size | |
| **Extended word 3** | - | - | |
| **Extended word 4** | - | - | |
| **Extended word 5** | - | - | |
| **Extended word 6** | - | - | |
| **Extended word 7** | - | - | |
| **Extended word 8** | - | - | |
| **Message Data Word 1** | Consuming instance no. | Consuming instance no. | |
| **Message Data Word 2** | Producing instance no. | Producing instance no. | |
| **Message Data Word 3** | Listen Only instance no. | Listen Only instance no. | |
| **Message Data Word 4** | Input Only instance no. | Input Only instance no. | |
| **Message Data Word 5** | Configuration instance no. | Configuration instance no. | |

# Mailbox Socket Interface

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

- **Non-blocking**

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

  **Note:** Status information for all non-blocking sockets are available in the fieldbus specific area, see 11-1 “Fieldbus Specific Area”.

- **Blocking**

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

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

Commands in this category:

| Mailbox Command | Description | Page |
|---|---|---|
| Socket non-blocking<br>(SOCKET_NB) | Creates a socket in non-blocking mode. | 10-92 |
| Socket blocking<br>(SOCKET_B) | Creates a socket in blocking mode. | 10-93 |
| Listen<br>(LISTEN) | Starts listen on a socket for incoming connections. | 10-93 |
| Accept<br>(ACCEPT) | Accepts connections for sockets in blocking mode. | 10-95 |
| Connect<br>(CONNECT) | Tries to connect a socket to a client. | 10-96 |
| Send<br>(SEND) | Sends a message to a connected socket. | 10-97 |
| Receive<br>(RECEIVE) | Receives a message form a connected socket. | 10-98 |
| Send To<br>(SEND_TO) | Sends a message to an unconnected UDP socket to a specified host. | 10-99 |
| Receive From<br>(RECV_FROM) | Receives a message from an unconnected UDP socket. | 10-100 |
| Close<br>(CLOSE) | Closes a socket (and connection). | 10-101 |
| Send Fragment<br>(SEND_FRAG) | Sends a fragment of a message with a maximum total size of 4096 bytes. | 10-102 |
| Receive Fragment<br>(RECV_FRAG) | Receives a fragment of a message with a total maximum size of 4096 bytes. | 10-104 |
| Send Fragment To<br>(SEND_FRAG_TO) | Sends a fragment of a message with a total maximum size of 4096 bytes to an unconnected UDP socket. | 10-105 |
| Receive Fragment From<br>(RECV_FRAG_FROM) | Receives a fragment of a message with a total maximum size of 4096 bytes from an unconnected UDP socket. | 10-107 |
| Get Socket Option<br>(GET_SOCKET_OPTION) | Read options from a socket. | 10-109 |
| Set Socket Option<br>(SET_SOCKET_OPTION) | Sets options to a socket | 10-110 |

# Socket Non-Blocking (SOCKET_NB)

## Description

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

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

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

## Command and response layout

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

- **Socket Type**

| Value | Socket type |
|---|---|
| 0001h | TCP socket |
| 0002h | UDP socket |

# Socket Blocking (SOCKET_B)

## Description

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

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

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

## Command and response layout

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

- **Socket Type**

| Value | Socket type |
|---|---|
| 0001h | TCP socket |
| 0002h | UDP socket |

# Listen (LISTEN)

## Description

This mailbox command makes a socket listen for new connections. If the Anybus module detects a connection request on the specified socket, a new connected socket will be created, and the current socket will continue listening for new connections. This means that multiple hosts can connect to one listening socket simultaneously.

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

- **Non-blocking sockets**

  Information about active connections on this socket can be read in the fieldbus specific area, see 11-1 “Memory Map” and 11-2 “Socket Status Structure”.

- **Blocking sockets**

  Socket descriptors for new connections connected to this socket can be received by the mailbox command ACCEPT, see 10-95 “Accept (ACCEPT)”.

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

### Command and response layout

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

## Accept (ACCEPT)

### Description

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

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

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

### Command and response layout

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

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

## Connect (CONNECT)

### Description

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

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

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

- **Non-blocking sockets**
  If this command is correctly sent, it will be accepted regardless it's possible to establish a connection or not. The result of the operation is available in the fieldbus specific area, see 11-1 "Fieldbus Specific Area".

- **Blocking sockets**

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

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

### Command and response layout

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

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

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

## Send (SEND)

### Description

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

- **Non-blocking sockets**

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

- **Blocking sockets**

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

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

### Command and response layout

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

## Receive (RECV)

### Description

This mailbox command receives data from a connected socket.

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

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

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

- **Non-blocking sockets**

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

- **Blocking sockets**

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

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

### Command and response layout

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

## Send To (SEND_TO)

### Description

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

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

### Command and response layout

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

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

## Receive From (RECV_FROM)

### Description

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

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

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

- **Non-blocking sockets**

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

- **Blocking sockets**

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

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

### Command and response layout

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

## Close (CLOSE)

### Description

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

- **Blocking sockets**

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

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

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

### Command and response layout

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

## Send Fragment (SEND_FRAG)

### Description

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

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

- **Non-blocking sockets**

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

- **Blocking sockets**

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

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

## Command and response layout

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

- **Fragment Type Value**

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

## Receive Fragment (RECV_FRAG)

### Description

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

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

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

The maximum size of a fragmented message is 4096 bytes.

- **Non-blocking sockets**
  If no data is available on the socket the response will indicate that 0 bytes of data was received.

- **Blocking sockets**

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

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

### Command and response layout

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

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

- **Fragment Type Value**

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

## Send Fragment To (SEND_FRAG_TO)

### Description

This mailbox command sends a UDP datagram to a specified IP address and port number. This command is used when sending a fragment of a message larger than 256 byte. Internally the fragments are stored in a buffer until the last fragment is received. The message is then sent to the socket. The maximum size of a fragmented message is 4096 bytes.

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

### Command and response layout

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

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

- **Fragment Type Value**

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

## Receive Fragment From (RECV_FRAG_FROM)

### Description

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

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

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

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

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

- **Non-blocking sockets**
  If no data is available on the socket the response will indicate that 0 bytes of data was received.

- **Blocking sockets**

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

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

## Command and response layout

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

- **Fragment Type Value**

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

# Get Socket Option (GET_SOCKET_OPTION)

## Description

This command reads options from a socket.

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

## Command and response layout

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

## Socket Options

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

SO_LINGER<br>
SO_KEEPALIVE<br>
SO_REUSEADDR<br>
IP_MULTICAST_TTL<br>
IP_MULTICAST_LOOP

For more information see section Socket Options page10-111.

# Set Socket Option (SET_SOCKET_OPTION)

## Description

This command changes the settings for a specified socket.

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

## Command and response layout

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

## Socket Options

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

SO_LINGER
SO_KEEPALIVE
SO_REUSEADDR
IP_MULTICAST_TTL
IP_MULTICAST_LOOP
IP_ADD_MEMBERSHIP
IP_DROP_MEMBERSHIP
TCP_NODELAY

For more information see section Socket Options page10-111.

## Socket Options

| Name | Option Value | DataType | Description |
|---|---|---|---|
| SO_LINGER | 0x00000080 | Struct of:<br>UINT32 l_onoff<br>UINT32 l_linger | Controls the action taken when unsent data is queued on a socket that is being closed. This option is only valid for TCP sockets.<br>l_onoff 0:Linger OFF (default) Other:Linger ON<br>l_linger Normally defines the linger timeout. NOT SUPPORTED, ALWAYS SET TO 0.<br>If SO_LINGER is disabled, Socket Close returns immediately and the connection is gracefully closed in the background.<br>If SO_LINGER is enabled with a zero timeout, Socket Close returns immediately and the connection is reset. |
| SO_KEEPALIVE | 0x00000008 | UINT32 l_keepalive | Enables/disables keep alive probes on a socket. This option is only valid for TCP sockets.<br>l_keepalive 0:Keep alive OFF (default) Other:Keep alive ON<br>Keep alive can be used to detect if the host is still active, and if not close down the connection.<br>If keep alive is enabled a keep alive probe will be sent to the host after 2 hours with no data being sent or received on a connection. This packet is designed to provoke an ACK response from the host. If no ACK is received another 8 keep alive probes will be sent with 75 seconds interval, and if non of them is ACKed the connection will be reset. |
| SO_REUSEADDR | 0x00000004 | UINT32 l_reuseaddr | Enables/disables reuse address option on a socket. This option is only valid for TCP sockets.<br>l_reuseaddr 0:Reuse address OFF (default) Other:Reuse address ON<br>When reuse address option is enabled it is possible to reuse a TCP port even if the port is busy in TIME_WAIT state. If the port is busy in other states an error will still be generated. This can be useful for a server implementation that is shut down and directly restarted while sockets are still active on its port. |

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

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

# Other Commands

Commands in this category:

| Mailbox Command | Description | Page |
|---|---|---|
| Alter LNK and ACT Led Functionality (ALT_LNK_ACT_LEDS) | Alter the behaviour of the Link and Activity LEDs. | 10-115 |
| Alter Module Status LED Functionality (ALT_MS_LED_FUNCTION) | Alter the behaviour of the Module Status LED | 10-116 |
| Alter Network Status LED Functionality (ALT_NS_LED_FUNCTION) | Alter the behaviour of the Network Status LED | 10-117 |
| Get DIP Switch<br>GET_DIP_SWITCH | Returns the setting of the onboard DIP switch | 10-118 |
| DNS Request (DNS_REQUEST) | Asks the configured DNS server for the IP address of a specified host | 10-119 |
| Send Email (SEND_EMAIL) | Sends an email message to a specified recipient | 10-120 |
| Request SSI Data (REQUEST_SSI_DATA) | Requests SSI data from the application<br>(issued by the Anybus module) | 10-122 |
| Write SSI Data (WRITE_SSI_DATA) | Writes SSI data to the application<br>(issued by the Anybus module) | 10-123 |
| Write Output Area (WRITE_OUTPUT_AREA) | Writes data to the Output Area | 10-124 |

# Alter LNK and ACT LED Functionality (ALT_LNK_ACT_LEDS)

## Description

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

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

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

## Command and response layout

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

- **LED Configuration Value**

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

# Alter Module Status LED Functionality (ALT_MS_LED_FUNCTION)

## Description

This function alters the behaviour of the Module Status LED.

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

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

## Command and response layout

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

- **LED Configuration byte value**

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

# Alter Network Status LED Functionality (ALT_NS_LED_FUNCTION)

## Description

This function alters the behaviour of the Network Status LED.

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

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

## Command and response layout

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

- **LED Configuration byte value**

| **Value** | **Description** |
|---|---|
| 01h | Led configuration 1 |
| 02h | Led configuration 2 |
| 03h | Led configuration 3 |

# Get DIP Switch (GET_DIP_SWITCH)

## Description

This command returns the setting of the onboard switch.

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

## Command and response layout

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

- **Switch Value**

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

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

## DNS Request (DNS_REQUEST)

### Description

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

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

### Command and response layout

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

# Send Email (SEND_EMAIL)

## Description

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

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

## Command and response layout

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

- **Fragment Type**

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

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

- **SMTP Error**

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

- **Fragment Data**

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

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

# Request SSI Data (REQUEST_SSI_DATA)

## Description

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

***Example:***

The following SSI...

```
<?--#exec cmd_argument=’printf( “Data: %u”, MbReadWord( 42 ) )’-->
```

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

See also 8-5 “printf”.

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

## Command and response layout

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

- **SSI Identifier**

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

- **SSI Data**

  Data associated with the specified SSI Identifier.

# Write SSI Data (WRITE_SSI_DATA)

## Description

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

***Example:***

The following SSI...

```
<?--#exec cmd_argument=’scanf( “Input”, ”%i”, MbWriteWord( 24 ) )’-->
```

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

See also 8-6 “scanf”.

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

## Command and response layout

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

- **SSI Identifier**

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

- **SSI Data**

  Data associated with the specified SSI Identifier.

## Write Output Area (WRITE_OUTPUT_AREA)

### Description

This command writes data to the Output Area.

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

### Command and response layout

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

- **Offset**

  Destination in Output Area.
- **No. of Bytes**

  Number of bytes to write.
- **Data**

  Data that shall be written.

# Fieldbus Specific Area

## Memory Map

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

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

# Socket Status Structure

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

## Socket Type

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

## Socket Status

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

## Socket Information

<table>
<tr><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td colspan="15">(reserved)</td><td>DA</td></tr>
</table>

| Bit | Description |
|---|---|
| DA | 0: Data Not Available<br>1: Data Available |

## Local Port Number

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

## Host Port Number

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

## Host IP-address

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

# Network Status

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

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

# EtherNet/IP Status

| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Connection Status | | | | | | | | Run/Idle Status | | | | | | | |
| 7B0h (MSB) | | | | | | | | | | | | | | | 7B1h (LSB) |

## Connection Status

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

## Run/Idle Status

| Value | Description |
|---|---|
| 00h | Idle |
| 01h | Run |

### Num Modbus connections

Number of established ModbusTCP connections.

### Num active Class1 connections

Indicates number of error free Class1 connections.

### Num active Class3 connections

Indicates number of error free Class3 connections.

### Num timed out Class1 connections

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

### Num timed out Class3 connections

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

# Miscellaneous

## Control Register Area

### Fieldbus Type

The fieldbus type value for this product is 0083h.

### Module Type

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

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

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

### Event Notification Cause/Source Registers

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

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

  For other options, see...

  - 3-4 "On/Off Line Configuration"
  - 10-47 "Modbus ON/OFF Line Configuration (MB_ON_OFF_LINE_CONFIG)"

- **Network Reset Functionality (RST)**

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

  See also...

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

# Firmware Upgrade

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

1. As a precaution, make a backup copy of the filesystem contents before proceeding.
2. Upload the new firmware file(s) to the system root ("\"), or to the 'user\'-directory.
3. Reset the module and wait until the watchdog LED flashes 2Hz green (may take up to 1 minute).
4. Reset the module again. The new firmware is now operational.

# Formatting the File System

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

1. Short jumper named 'J2'
2. Apply power
3. Wait until the watchdog LED turns red
4. Disconnect power
5. Remove jumper
6. Apply power
7. Wait approx. 1 minute while the filesystem is being formatted.

# IP-related implementation details

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

# Technical Specification

## Electrical Specification

### Protective Earth (PE) Requirements

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

### Isolation

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

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

(Tests performed according to EN 60950-1)

### Power Supply

#### Supply Voltage

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

#### Power Consumption

The maximum power consumption is 450mA.

# Environmental Specification

### Temperature

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

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

### Humidity

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

Tests performed according to EN 60068.

# EMC (CE) Pre-compliance

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

# Connectors

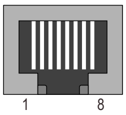
## Application Connector

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

## Ethernet

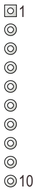
### RJ45 (Standard Connector)

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



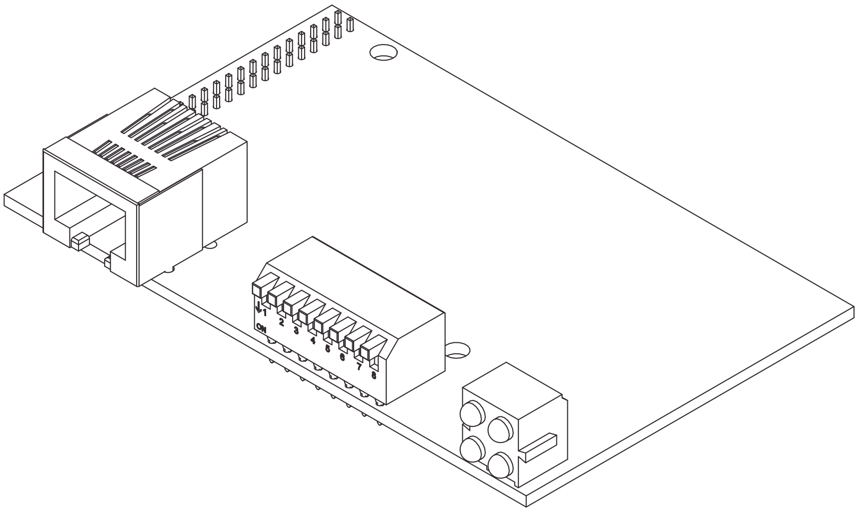
### Board to Board

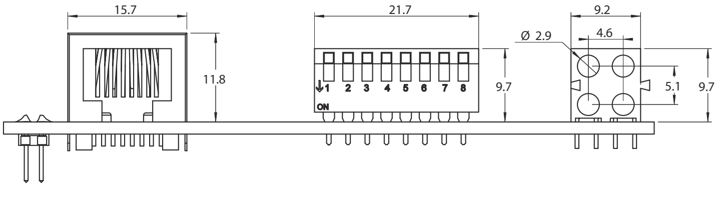
| Pin | Signal | Connect to RJ45 pin... | Notes |
|---|---|---|---|
| 1 | Shield | Housing | - |
| 2 | - | 4 | (See notes for pins 4 and 5 in RJ45 connector) |
| 3 | - | 5 | |
| 4 | - | - | (not used) |
| 5 | TD+ | 1 | - |
| 6 | TD- | 2 | - |
| 7 | RD+ | 3 | - |
| 8 | - | 7 | (See notes for pins 7 and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

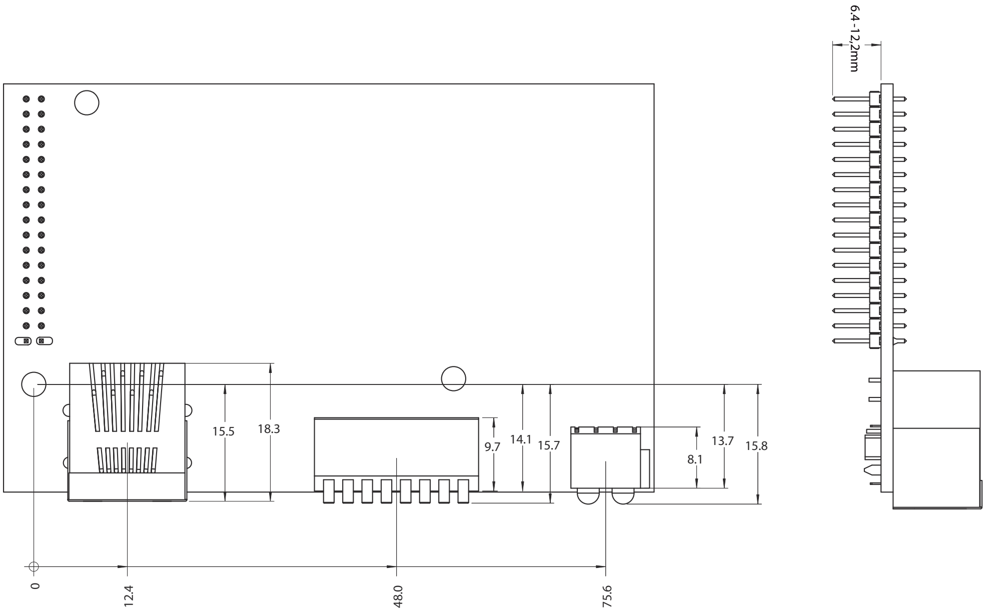


# Mechanical Specification

## Measurements, Connectors & LEDs







# Copyright Notices

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