Fieldbus Appendix

Anybus-S PROFINET IRT

Doc. Id. HMSI-168-52
Rev. 1.33

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

Important User Information

This document is intended to provide a good understanding of the functionality offered by Anybus-S PROFINET IRT. The document only describes the features that are specific to the Anybus-S PROFINET IRT. For general information regarding the Anybus-S, consult the Anybus-S/M Parallel Design Guide.

Please consult the general Anybus-S Parallel Design Guide for further information about the Anybus-S platform.

The reader of this document is expected to be familiar with high level software design, and communication systems in general. The use of advanced PROFINET IRT-specific functionality may require in-depth knowledge in PROFINET IRT networking internals and/or information from the official PROFINET IRT specifications. In such cases, the people responsible for the implementation of this product should either obtain the PROFINET IRT specification to gain sufficient knowledge or limit their implementation in such a way that this is not necessary.

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.

Liability

Every care has been taken in the preparation of this manual. Please inform HMS Industrial Networks AB of any inaccuracies or omissions. 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 applications meet all performance and safety requirements including any applicable laws, regulations, codes, and standards.

HMS Industrial Networks AB will under no circumstances assume liability or responsibility for any problems that may arise as a result from the use of undocumented features, timing, or functional side effects found outside the documented scope of this product. The effects caused by any direct or indirect use of such aspects of the product are undefined, and may include e.g. compatibility issues and stability issues.

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 Industrial Networks AB cannot assume responsibility for actual use based on these examples and illustrations.

Intellectual Property Rights

HMS Industrial Networks AB has intellectual property rights relating to technology embodied in the product described in this document. These intellectual property rights may include patents and pending patent applications in the US and other countries.

Trademark Acknowledgements

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

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.

Anybus-S PROFINET IRT User Manual
Rev 1.33

Sept 2012 Doc Id HMSI-168-52

Table of Contents

Preface **About This Document**

Related Documents 9
Document History 9
Conventions & Terminology 10
Sales and Support 11

Chapter 1 **About the Anybus-S PROFINET IRT**

Features 12
Conformance Notes 13
GSD File 13
Overview 13

Chapter 2 **Basic Operation**

General Information 15
Software Requirements 15
Device Identity 15
Initialization 16
Modes of Operation 16
Initialization Sequence, Generic Anybus Mode 17
Initialization Sequence, Advanced Mode 18
Data Exchange (PROFINET) 19
I/O Data 19
Parameter Data (Record Data) 19
Example, Generic Anybus Mode 20
Example, Advanced Mode 21
Fast Start Up 22
Modbus-TCP Server (Read-only) 23
General Information 23
Supported Function Codes 23
Supported Exception codes 23
Register Map 23
Filesystem 24
General Information 24
Filesystem Overview 25
System Files 25

Chapter 3 PROFINET Implementation

General Information 26
Performance Characteristics 26
Device Model 27
Diagnostics & Alarms (Advanced Mode Only) 28
Initial Parameters (Advanced Mode Only) 29
Configuration Data 29
Identification & Maintenance (I&M) 30
Electronic Data Sheet (GSD-file) 31
General Information 31
HMS Standard GSD-file 31
How to Associate a Bitmap to a Device Access Point 32
PROFIenergy Profile 33
PROFIenergy States 33
Implementation 34

Chapter 4 Network Configuration

TCP/IP Settings 35
General 35
DCP (Discovery and Basic Configuration) 36
LLDP (Link Layer Discovery Protocol) 36
SNMP (Simple Network Management Protocol) 36
HICP (Anybus IPconfig) 37
IP Access Control 37
PROFINET Settings 38

Chapter 5 FTP Server

General Information 39
Security Levels 39
User Accounts 39
FTP Connection Example (Windows Explorer) 40

Chapter 6 Web Server

General 41
Authorization 42
Content Types 43

Chapter 7 SMTP Client

General 44
Email Definitions 45

Chapter 8 Server Side Include (SSI)

Functions....47

Changing SSI output....56
SSI Output String File....56
Temporary SSI Output change....57

Chapter 9 Fieldbus Specific Mailbox Commands

Fault Information....58

Extended Fault Information....60

General Configuration Commands....61
Set Ethernet Configuration (SET_ETN_CONFIG)....62
Read Ethernet Configuration (READ_ETN_CONFIG)....63
Get MAC Address (GET_MAC_ADDR)....64
Set MAC Address (SET_MAC_ADDR)....66
Set Host and Domain (SET_HOST_DOMAIN)....67
Get Host and Domain (GET_HOST_DOMAIN)....68
Disable / Enable HICP (HICP_CFG_CONTROL)....69
Set DNS Server Settings (SET_DNS_SERVERS)....70
Get DNS Server Settings (GET_DNS_SERVERS)....71
Set SMTP Server (SET_SMTP_SERVER)....72
Set SMTP Server by Name (SET_SMTP_SERVER_NAME)....73
Get SMTP Server (GET_SMTP_SERVER)....74
Disable Web Server (DISABLE_WEB_SERVER)....75
Enable Web Server (ENABLE_WEB_SERVER)....76
Disable FTP server (DISABLE_FTP_SERVER)....77
Global Admin Mode (GLOBAL_ADMIN_MODE)....78
Disable Virtual File System (DISABLE_VFS)....79

Mailbox File System Interface....80
Open File (FILE_OPEN)....81
Close File (FILE_CLOSE)....82
Read File (FILE_READ)....83
Write File (FILE_WRITE)....84
Delete File (FILE_DELETE)....85
Move File (FILE_MOVE)....86
Rename File (FILE_RENAME)....87
Copy File (FILE_COPY)....88
Create Directory (DIR_CREATE)....89
Delete Directory (DIR_DELETE)....90
Open Directory (DIR_OPEN)....91
Read Directory (DIR_READ)....92
Close Directory (DIR_CLOSE)....94
Create RAM disc (CREATE_RAM_DISC)....95
Format File System (FORMAT_FS)....96
File System Checksum (CRC_FS)....97

PROFINET Specific Commands 98
Advanced Mode (PNIO_ADV_MODE) 99
Set Stop Mode Action (PNIO_SET_STOP_MODE_ACTION) 101
Set Device Identity (PNIO_SET_DEVICE_IDENTITY) 102
Set Station Name (PNIO_SET_STATION_NAME) 103
Set Station Type (PNIO_SET_STATION_TYPE) 104
Plug Module (PNIO_PLUG_MODULE) 105
Plug Submodule (PNIO_PLUG_SUBMODULE) 107
Pull Module (PNIO_PULL_MODULE) 109
Pull Submodule (PNIO_PULL_SUBMODULE) 110
End of Parameterization (PNIO_IND_END_OF_PRM) 111
Application State Ready (PNIO_APPL_STATE_READY) 112
Record Data Read (PNIO_IND_RECORD_DATA_READ) 113
Record Data Write (PNIO_IND_RECORD_DATA_WRITE) 115
Add Channel Diagnostic Alarm (PNIO_ADD_CHANNEL_DIAG_ALARM) 117
Remove Diagnostic Alarm (PNIO_REMOVE_DIAG_ALARM) 120
Add Generic Diagnostic Alarm (ADD_GENERIC_DIAG_ALARM) 121
Add Process Alarm (PNIO_ADD_PROCESS_ALARM) 123
Abort AR (PNIO_AR_ABORT) 125
Set port MAC address (PNIO_SET_PORT_MAC_ADDR) 126
Set System Description (PNIO_SET_SYSTEM_DESCRIPTION) 127
Set Interface Description (PNIO_SET_INTERFACE_DESCRIPTION) 128
Set System Description MIB-II (PNIO_SET_SYSTEM_DESCRIPTION_MIB2) 129
Set System Contact MIB-II (PNIO_SET_SYSTEM_CONTACT_MIB2) 130
Set System Location MIB-II (PNIO_SET_SYSTEM_LOCATION_MIB2) 131
Get IM0 Information (PNIO_GET_IM0_INFO) 132
Set IM0 Information (PNIO_SET_IM0_INFO) 134
Alarm Acknowledge Received (PNIO_IND_ALARM_ACK_RECEIVED) 136
Configuration Mismatch (PNIO_IND_CFG_MISMATCH) 137
Check AR Indication (PNIO_IND_CHECK_AR) 139
AR Info Indication (PNIO_IND_AR_INFO) 141
Abort AR Indication (PNIO_IND_AR_ABORT) 143
AR Offline Indication (PNIO_IND_AR_OFFLINE) 145

PROFIenergy Specific Commands 146
Enable PROFIenergy (PNIO_ENABLE_PROFIENERGY) 147
Get PROFIenergy mode (PNIO_IND_PE_GET_ENERGY_MODE) 149
Start PROFIenergy Pause (PNIO_IND_PE_START_PAUSE) 151
End PROFIenergy Pause (PNIO_IND_PE_END_PAUSE) 152
Get PROFIenergy status (PNIO_IND_PE_GET_STATUS) 153

Modbus-TCP-related Commands 154
Modbus Connection Timeout (MB_CONN_TIMEOUT) 155
Disable Modbus-TCP server (DISABLE_MB_TCP) 156

Mailbox Socket Interface 157
Socket Non-Blocking (SOCKET_NB) 158
Socket Blocking (SOCKET_B) 159
Listen (LISTEN) 160
Accept (ACCEPT) 161
Connect (CONNECT) 162
Send (SEND) 164
Receive (RECV) 165
Send To (SEND_TO) 166
Receive From (RECV_FROM) 167
Close (CLOSE) 168
Send Fragment (SEND_FRAG) 169
Receive Fragment (RECV_FRAG) 170
Send Fragment To (SEND_FRAG_TO) 172
Receive Fragment From (RECV_FRAG_FROM) 173
Get Socket Option (GET_SOCKET_OPTION) 175
Set Socket Option (SET_SOCKET_OPTION) 176
Socket Options 177

Other Commands 179
DNS Request (DNS_REQUEST) 180
Send Email (SEND_EMAIL) 181
Request SSI Data (REQUEST_SSI_DATA) 183
Write SSI Data (WRITE_SSI_DATA) 184

Chapter 10 Fieldbus Specific Area

Memory Map 185

Socket Status Structure 186

Network Status 187

IO Controller Status (PROFINET) 187

PROFINET Layer Status 187

PROFINET Layer Fault Code 187

Appendix A Miscellaneous

Control Register Area 188

Firmware Upgrade 188

Formatting the File System 188

Appendix B Technical Specification

Electrical Specification 189
Protective Earth (PE) Requirements 189
Isolation 189
Power Supply 189

Environmental Specification 190

Regulatory Compliance 190

Appendix C Connectors

Application Connector 191
Ethernet Connectors 191

Appendix D Mechanical Specification

Measurements, Connectors & LEDs 192

Appendix E Conformance Test Guide

General 193
Reidentifying Your Product 194
Factory Default Reset - RDR Support Recommended 195
IP Address 195
Station Name 195
Certification in Generic Anybus Mode 195
Certification in Advanced Mode 196

Appendix F Copyright Notices

P. About This Document

For more information, documentation etc., please visit the HMS website, 'www.anybus.com'.

P.1 Related Documents

Document	Author
Open Modbus/TCP Specification	Schneider Automation
PROFINET Technology and Application	PI
GSDML Specification for PROFINET IO	
PROFINET IO specification	
PROFIBUS Guideline, Identification & Maintenance Functions	
SIMATIC NET PROFINET IO Softwarebeschreibung	ComDec
Structure of the Diagnostic Data Records (publication: A5E00337523-01)	SIEMENS
RFC 821	Network Working Group
RFC 1918	

P.2 Document History

Summary of Recent Changes (1.32 ... 1.33)

Change	Page(s)
Added LED functionality (Solid red LED 2, solid red LED 3)	14
Added fault codes to fault information	58
Some mailboxes no longer store information in the file pniocfg.cfg. Information about that has been removed	102, 104, 127, 128, 129 , 130, 131

Revision List

Revision	Date	Author(s)	Chapter(s)	Description
1.00	2007-10-29	PeP	All	First revision
1.10	2008-10-13	HeS	-	Major update
-	-	-	-	-
1.13	2008-11-05	HeS	4, 9	Minor updates
1.14	2009-07-20	KeL	1, 2, 3, 9, 10, B	Minor updates
1.15	2009-10-02	KeL	9	Minor updates
1.16	2010-04-09	KeL	2, 9, A	Minor updates
1.17	2010-04-26	KeL	4, 9	Added new mailboxes and minor corrections
1.20	2011-03-11	KeL	9, E	Added new mailboxes, added appendix and minor corrections
1.21	2011-04-18	KeL	9, E	Minor corrections
1.30	2011-06-13	KeL	B	New template, minor correction
1.31	2012-04-10	KaD	P, 2, 3, 9, 10, E	Minor additions and updates
1.32	2012-05-29	KaD	3, 9	Minor additions and corrections
1.33	2012-09-03	KaD	1, 9	Minor additions and corrections

P.3 Conventions & Terminology

The following conventions are used throughout this manual:

- Numbered lists provide sequential steps
- Bulleted lists provide information, not procedural steps
- Mailbox commands that "may only be issued during initialization" must be issued between the "START_INIT" and "END_INIT" commands.
- 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

The following terms are used throughout this document:

Term	Meaning
Anybus	Anybus-S PROFINET IO IRT module
Application	Hardware which is connected to the Anybus Application Connector.
Generic Mode	Basic modes of operation which determines how certain aspects of the communication are handled by the Anybus module
Advanced Mode	
AR	Application Relationship
Byte	A byte akways consists of 8 bits
DAP	Device Access Point
DCP	Discovery and basic Configuration Protocol. Used for IP configuration over PROFINET.
DHCP	De-facto standard for dynamic IP address management. This feature is not supported in the Anybus-S PROFINET IRT.
GSDML	XML-based descriptive language for GSD-files
Initial Record Data	Record Data write-requests destined for a sub-module. Comparable to Profibus-DP User Parameter Data.
IOCS	IO Consumer Status
IOPS	IO Provider Status
IO Controller	Controlling device, which acts as a client for several IO devices. Usually a PLC. Comparable to a Profibus-DP Class 1 master.
IO Device	Field device assigned to an IO Controller. Comparable to a Profibus DPV1 slave.
IO Supervisor	Programming device with commissioning and diagnostic functions. Comparable to a Profibus-DP Class 2 master.
IRT	Isochronous RealTime
Module	Hardware or logical component of a network device.
Submodule	Hardware or logical component of a module
PNIO	Short for PROFINET IO
PROFINET IO	PROFINET IO is a communication concept for the implementation of modular, decentralized applications. Comparable to Profibus-DP, where I/O data of field devices are cyclically transmitted to the process image of a PLC. The real time capabilities of PROFINET IO are further divided into RT and IRT (see below).
PROFINET IO RT	PROFINET IO with Real Time capabilities. Optimized real time communication channel for time critical I/O data and Alarms. Implemented in software.
PROFINET IRT	PROFINET IO with Isochronous Real Time capabilities. Necessary for motion control application which require an update rate of 1ms, or less, with no jitter. Implemented in hardware.
PROFINET CBA	PROFINET Component Based Automation. Comparable to Profibus FMS.
Record Data	Comparable to Profibus DPV1 acyclic Read/Write
A 'set' bit	The bit value is 1 (one)
A 'cleared' bit	The bit value is 0 (zero)

P.4 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

1. About the Anybus-S PROFINET IRT

The Anybus-S PROFINET IRT communication module provides instant integration to any Ethernet based LAN via SMTP, FTP, HTTP as well as PROFINET 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 e-mail messages. SSI (Server Side Include) technology enables web pages and e-mail 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 PROFINET IRT is compatible with any product that supports the Anybus-S application interface with only little or no software adjustments. Where applicable, the Anybus-S PROFINET IRT is designed to be compatible with the Anybus-S PROFINET IO.

1.1 Features

General

- Shielded (FTP) and unshielded (UTP) cables
- Built-in 2-port Ethernet switch
- Flexible file system providing both volatile and nonvolatile storage areas
- Security framework
- PROFIenergy support
- Integrated FTP server provides easy file management using standard FTP clients.
- Server Side Include (SSI) capability
- Web server
- E-mail client (Messages can be triggered by data events or directly by the application)
- Supports the Link Layer Discovery Protocol (LLDP)
- Supports the Simple Network Management Protocol (SNMP)

Note: The application cannot synchronize with the IRT bus cycle.

Industrial Protocols

- PROFINET IO
  - Isochronous Real Time operation
  - Up to 220 bytes cyclic I/O in each direction
  - Up to 17 slots, up to 4 subslots per slot (1 by default)
- Modbus-TCP server
  - read-only

1.2 Conformance Notes

When using the default settings of all parameters, the module is precertified for network compliance. However, any parameter changes which require deviations from the standard GSD file supplied by HMS will require recertification. For further information, please contact HMS.

See also...

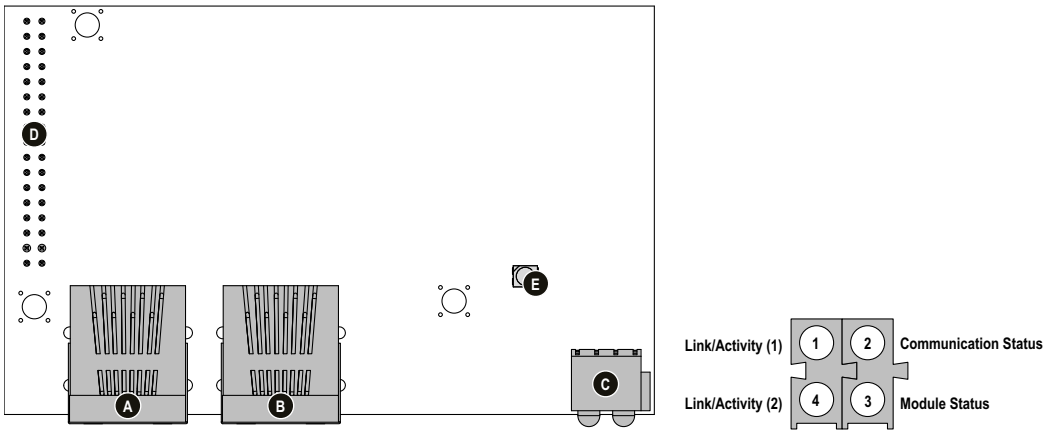
- "Conformance Test Guide" on page 193
- "Regulatory Compliance" on page 190

1.3 GSD File

On PROFINET, the characteristics of a device is stored in an XML data file. This file, commonly referred to as the GSD file, is used by PROFINET configuration tools when setting up the network.

HMS provides a generic GSD file, which corresponds to the default settings in the Anybus module. However, due to the flexible nature of the Anybus-S concept, it is possible to alter the behavior of the product in ways which invalidates this file. In such case, a custom file GSD file must be created.

1.4 Overview



#	Description	Comments
A	Ethernet Port 1	For more information, see Appendix "Connectors" on page 191
B	Ethernet Port 2	
C	Status Indicators	(below)
D	Application Connector	Consult the general Anybus-S Parallel Design Guide for further information.
E	Anybus Watchdog	

Status Indicators

#	Indication	State	Description
1	Link/Activity (1)	Green	Link established on Ethernet port 1
		Green, flashing	Receiving/Transmitting data on Ethernet port 1
		Off	Link not established on Ethernet port 1
2	Communication Status	Green	Online, Run - Connection with IO Controller established - IO Controller is in RUN state
		Green, 1 flash	Online, STOP - Connection with IO Controller established - IO Controller in STOP state
		Red	Almost finished installing firmware or repairing errors. Will be ready for use within 5 seconds.
		Off	Offline - No connection with IO Controller
3	Module Status	Green	Initialized, no error
		Green, 1 flash	Diagnostic data available
		Green, 2 flashes	Blink (used by engineering tools to identify the device)
		Red	Indicates any of the following: - Formatting filesystem - Installing or erasing firmware - Repairing errors
		Red, 1 flash	Configuration Error - Too many modules/submodules - I/O sizes derived from IO Controller configuration is too large - Configuration mismatch (no module, wrong module)
		Red, 3 flashes	No Station Name or no IP address assigned
		Red, 4 flashes	Internal error
		Off	No power or not initialized
4	Link/Activity (2)	Green	Link established on Ethernet port 2
		Green, flashing	Receiving/Transmitting data on Ethernet port 2
		Off	Link not established on Ethernet port 2

2. Basic Operation

2.1 General Information

2.1.1 Software Requirements

Generally, no additional network support code needs to be written in order to support the Anybus-S PROFINET IRT. However, due to the nature of the PROFINET networking system, certain restrictions must be taken into account:

- The total I/O size must exceed zero.
- The total I/O size for each direction cannot exceed 220 bytes (generic Anybus mode) or 254 bytes (advanced mode).
- Certain (optional) advanced functionality requires the use of spontaneous mailbox messaging.
- The use of advanced PROFINET-specific functionality may require in-depth knowledge in PROFINET networking internals and/or information from the official PROFINET specifications. In such cases, the people responsible for the implementation of this product should either obtain the PROFINET specification to gain sufficient knowledge or limit their implementation is such a way that this is not necessary.

For further information about the Anybus-S software interface, consult the general Anybus-S Parallel Design Guide.

2.1.2 Device Identity

By default, the module identifies itself on the network as a generic Anybus implementation as follows:

Station Name[ab]	‘ ’ (not assigned)
Station Type[b]	‘ABS-PIR’
Vendor ID[b]	010Ch (HMS Industrial Networks)
Device ID[b]	0006h (Anybus-S PROFINET IRT)

a. Can be specified via PROFINET by means of the Discovery and Basic Configuration protocol (DCP).
b. Can be specified by the application through the mailbox interface. Note that doing so invalidates the standard GSD file supplied by HMS and thus requires recertification of the end product.

See also...

- “Conformance Notes” on page 13
- “HMS Standard GSD-file” on page 31
- “Set Device Identity (PNIO_SET_DEVICE_IDENTITY)” on page 102
- “Set Station Name (PNIO_SET_STATION_NAME)” on page 103
- “Set Station Type (PNIO_SET_STATION_TYPE)” on page 104

2.2 Initialization

2.2.1 Modes of Operation

The Anybus offers two modes of operation which mainly affects the handling of the PROFINET protocol.

- **Generic Anybus Mode**

  This is the simplest mode since it requires the least interaction from the application side.
  - Anybus handles plugging of modules/submodules automatically based on the sizes specified in ANYBUS_INIT. The algorithm inserts IRT-modules into subslot #1 by default. These modules may be replaced later on with the ones (RT- or IRT modules) defined in the IO Controller configuration. Unused slots are plugged with 'Empty slot' modules.
  - The actual configuration is adopted from the IO Controller
  - Record data is mapped as parameter data in the input/output data areas
  - Initial record data is *not* supported
  - Diagnostics and Alarm functionality is *not* supported
  - Up to 17 slots
  - Up to 4 subslots per slot
  - Max. module size = 128 bytes
  - Supported by the generic GSD file (see "HMS Standard GSD-file" on page 31)

- **Advanced Mode (Expert Users Only)**

  This mode offers tighter integration with the application, but is a bit more complex compared to generic Anybus mode.
  - Application handles plugging of modules/submodules.
  - The application can optionally retrieve information about the actual IO configuration from the IO controller
  - Record data is either mapped as parameter data in the input/output data areas, or routed to the application via the mailbox interface.
  - Alarm and Diagnostic functionality is handled by the application via the mailbox interface
  - Up to 17 slots
  - Up to 4 subslots per slot
  - Max. module size = 254 bytes
  - An application-specific GSD file must be created

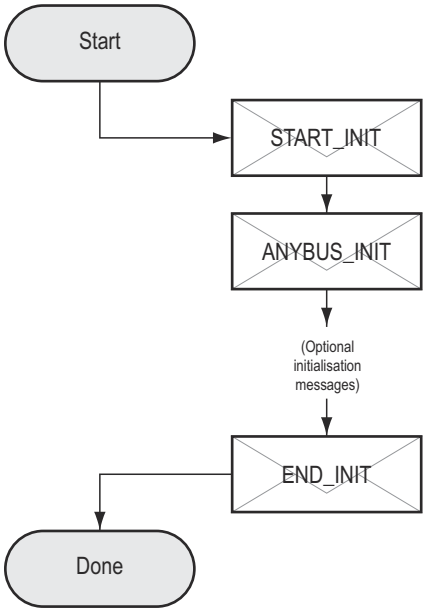
See also...

- "Initialization Sequence, Generic Anybus Mode" on page 17
- "Initialization Sequence, Advanced Mode" on page 18
- "Advanced Mode (PNIO_ADV_MODE)" on page 99 (enables Advanced Mode)
- "HMS Standard GSD-file" on page 31

2.2.2 Initialization Sequence, Generic Anybus Mode

To operate in this mode, the Anybus must be initialized with the following mailbox sequence:

- **START_INIT**
  (Initiates the initialization process)
- **ANYBUS_INIT**
  (Specifies the total I/O sizes)
- **Other initialization messages**
  (Optional)
- **END_INIT**
  (Finalizes the initialization process)



The Anybus will adopt the configuration issued by the IO Controller via the Connect service.

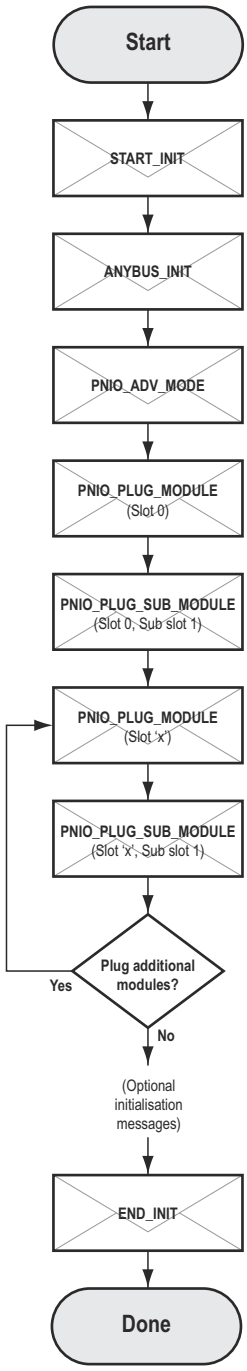
Note that the total size of the configuration must be less or equal to the corresponding I/O lengths specified in the ANYBUS_INIT mailbox message.

See also...

- "Modes of Operation" on page 16
- "Initialization Sequence, Advanced Mode" on page 18

2.2.3 Initialization Sequence, Advanced Mode

To operate in this mode, the Anybus must be initialized with the following mailbox sequence:



- **START_INIT**
  (Initiates the initialization process)
- **ANYBUS_INIT**
  (Specifies the total I/O sizes)
- **PNIO_ADV_MODE**
  (Instructs the Anybus to operate in advanced mode, and specifies advanced operational parameters)
- **PNIO_PLUG_MODULE**[1]
  (Slot #0, mandatory (DAP))
- **PNIO_PLUG_SUB_MODULE**[1]
  (Slot #0, Subslot #1, mandatory (DAP))
  ...
  (Plug additional modules/submodules)
  ....
- **Other initialization messages**
  (Optional)
- **END_INIT**
  (Finalizes the initialization process)

See also...

- "Modes of Operation" on page 16
- "Initialization Sequence, Generic Anybus Mode" on page 17

1. A module must be plugged in slot #0 before additional modules can be plugged. Slot #0 is the device access point (DAP) and acts as a proxy for the IO device.

2.3 Data Exchange (PROFINET)

2.3.1 I/O Data

IO data is exchanged via the I/O data portions of the Anybus input/output areas.

See also...

- "Example, Generic Anybus Mode" on page 20
- "Example, Advanced Mode" on page 21

2.3.2 Parameter Data (Record Data)

This is handled slightly differently depending on how the Anybus module has been initialized:

- **Generic Anybus Mode (or Advanced Mode with 'Record Data Request' bit cleared)**

Index Range	Mapped to...	Comments
1000h... 1XXXh	Parameter Input Area	Each index corresponds to a single byte in the Anybus module. The number of indexes used for data exchange depends on the data sizes specified in ANYBUS_INIT.
2000h... 2XXXh	Parameter Output Area	
6000h... 7FFFh	-	(reserved)
8000h... FFFFh	Diagnostics, I&M etc.	See... - "Diagnostics & Alarms (Advanced Mode Only)" on page 28 - "Identification & Maintenance (I&M)" on page 30

- **Advanced Mode ('Record Data Request' bit set)**

  The Anybus will act based on the settings specified in PNIO_ADV_MODE, see below.

Index Range	Mapped to...	Comments
0000h... 7fffh	Application	Routed to the application through the mailbox interface.
8000h... FFFFh	Diagnostics etc.	See... - "Diagnostics & Alarms (Advanced Mode Only)" on page 28 - "Identification & Maintenance (I&M)" on page 30

See also...

- "Modes of Operation" on page 16
- "Diagnostics & Alarms (Advanced Mode Only)" on page 28
- "Identification & Maintenance (I&M)" on page 30
- "Advanced Mode (PNIO_ADV_MODE)" on page 99
- "Record Data Read (PNIO_IND_RECORD_DATA_READ)" on page 113
- "Record Data Write (PNIO_IND_RECORD_DATA_WRITE)" on page 115

2.3.3 Example, Generic Anybus Mode

The modules in the configuration specified by the IO Controller will be mapped to the input/output data areas in the order of their slot number. Record data request will be mapped to the parameter data areas.

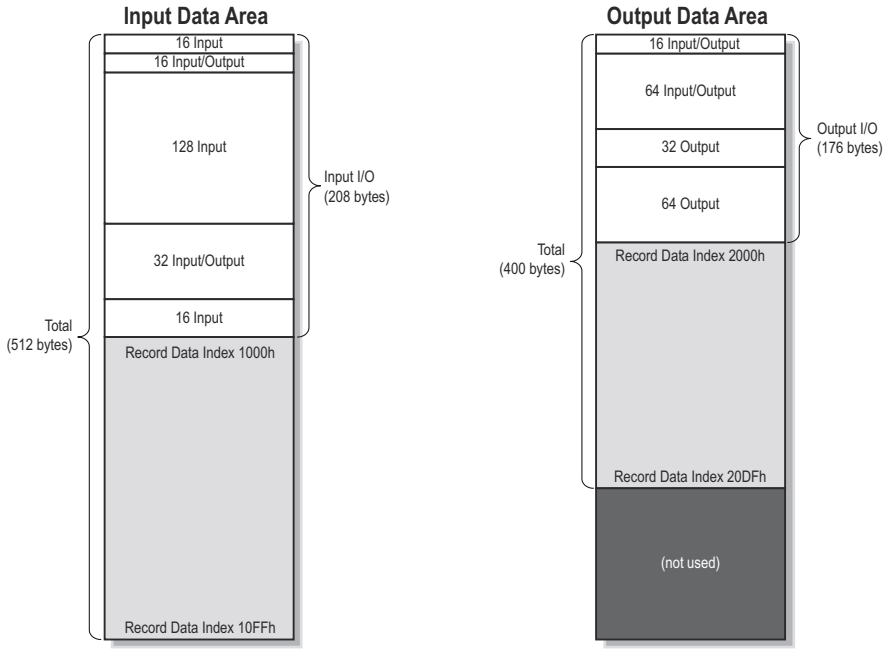
Settings in ANYBUS_INIT:

Input I/O Length	= 208 bytes
Total Input Length	= 512 bytes
Output I/O Length	= 176 bytes
Total Output Length	= 400 bytes

The following modules are specified in the IO Controller:

Slot	Module Size (bytes)	Direction	Notes
0	0	-	(Device Access Point, DAP)
1	16	Input	-
2	16	Input/Output	-
3	128	Input	-
4	32	Input/Output	-
5	16	Input	-
6	32	Output	-
7	64	Output	-

Resulting memory layout:



See also...

- "Modes of Operation" on page 16
- "Initialization Sequence, Generic Anybus Mode" on page 17

2.3.4 Example, Advanced Mode

The modules/submodules are specified by the application during initialization, and are mapped to the input/output Data Areas in the order of their slot number. In this example, the record data requests will be routed to the application via the mailbox interface.

Settings in ANYBUS_INIT:

Input I/O Length	= 192 bytes
Total Input Length	= 192 bytes
Output I/O Length	= 128 bytes
Total Input Length	= 128 bytes

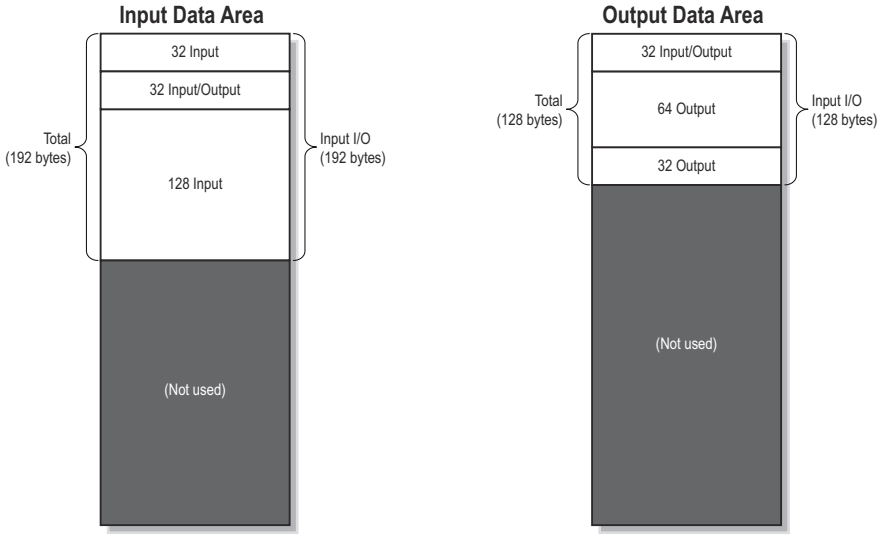
Settings in PNIO_ADV_MODE:

Max. No. Of Slots	= 10
Cfg Mismatch	= (not relevant in this example)
Record Data Request	= 1
End of Prm	= (not relevant in this example)
Alarm Acknowledge	= (not relevant in this example)
AR Information	= (not relevant in this example)
I&M Request	= (not relevant in this example)

The following modules are plugged by the application during initialization:

Slot	Module Size (bytes)	Direction	Notes
0	0	-	(Device Access Point, DAP)
1	32	Input	-
2	32	Input/Output	-
3	128	Input	-
4	64	Output	-
5	32	Output	-

Resulting memory layout:



See also...

- "Modes of Operation" on page 16
- "Initialization Sequence, Advanced Mode" on page 18

2.3.5 Fast Start Up

The Fast Start Up (FSU) function enables PROFINET IRT devices, connected to the network, to power up quickly. This is useful in for example robot applications, where rapid retooling is necessary. This function has to be activated when configuring the Anybus-S-PROFINET IRT module.

In the GSD file a few keywords for this functionality are used. The FSU time is defined as the number of milliseconds from hardware reset (or power on) until establishment of PROFINET IRT Communication. If the FSU time is measured to be larger than approximately 1500 ms it is recommended that this functionality is disabled.

The following keywords are used for this functionality (listed for the Device Access Point(s)):

- **PowerOnToCommReady**

  FSU time, in milliseconds (ms). Default value is 0 ms.
- **DCP_HelloSupported**

  Keyword stating whether or not the device will transfer "Hello" messages at power on. Default value: true.

To disable FSU, set the keywords to the following values:

- **PowerOnToCommReady**

  Remove this keyword from the GSD file.
- **DCP_HelloSupported**
- Value: false.

2.4 Modbus-TCP Server (Read-only)

2.4.1 General Information

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 read-only 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 received/transmitted on TCP port no. 502. For detailed information regarding the Modbus-TCP protocol, consult the Open Modbus Specification.

The Modbus-TCP server can be disabled, see "Disable Modbus-TCP server (DISABLE_MB_TCP)" on page 156.

2.4.2 Supported Function Codes

The following function codes are implemented:

Code	Function	Associated with Area
04h	Read Input Registers	Input Data Area
03h	Read Multiple Registers	Output Data Area

2.4.3 Supported Exception codes

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

2.4.4 Register Map

Register Type	Modbus Register	Mapped To...	
		Area	Offset
Input Registers (3xxxx)	0000h	Input Data Area	000h...001h
	0001h		002h...003h
	0002h		004h...005h
	0003h		006h...007h
	...		...
	03FFh		7FEh...7FFh
Output Registers (4xxxx)	0000h	Output Data Area	000h...001h
	0001h		002h...003h
	0002h		004h...005h
	0003h		006h...007h
	...		...
	03FFh		7FEh...7FFh

2.5 Filesystem

2.5.1 General Information

The Anybus 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, and by the application via the mailbox interface.

Storage Areas

The filesystem consists of the different storage areas:

- **Nonvolatile area (approx. 4Mb)**
  This section is intended for static files such as web files, configuration files etc.
- **Volatile area (approx. 1Mb)**
  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 initialization (see "Create RAM disc (CREATE_RAM_DISC)" on page 95)

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

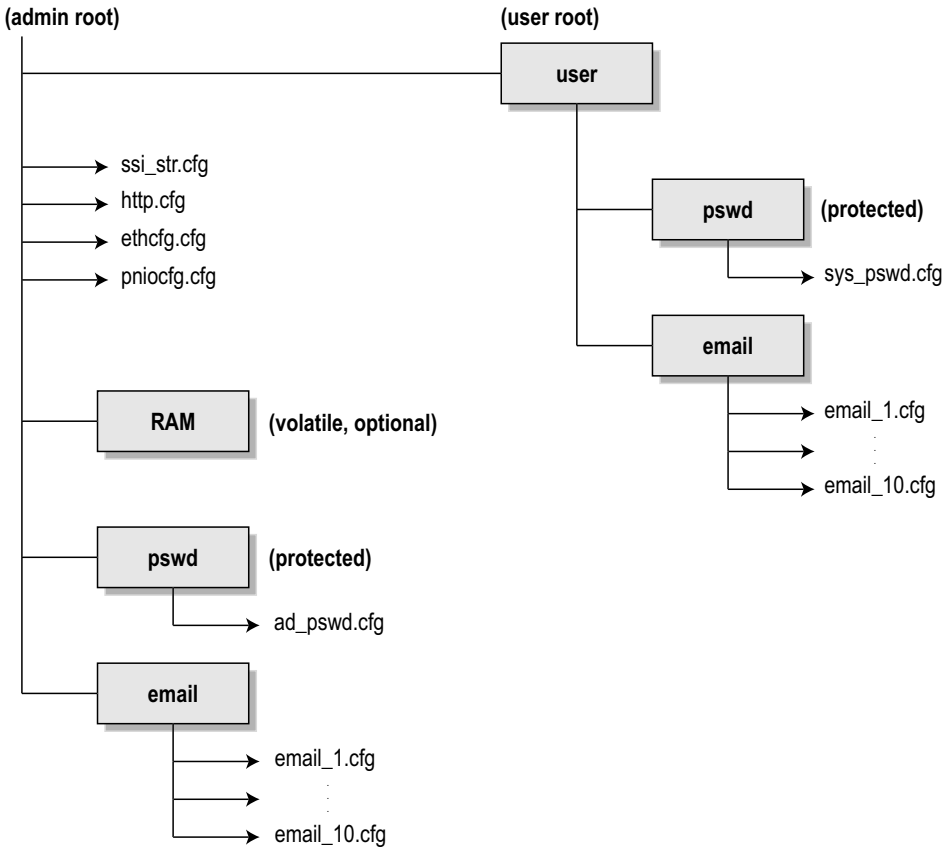
Important Notes

The nonvolatile storage is located in FLASH memory. Each FLASH segment can only be erased approximately 1000000 times due to the nature of this type of memory.

The following operations will erase one or more FLASH segments:

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

2.5.2 Filesystem Overview



2.5.3 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 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 "Server Side Include (SSI)" on page 46).

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

3. PROFINET Implementation

3.1 General Information

The Anybus module currently complies to conformance class C, and supports the following classes for RT communication:

- RT C1 (unsynchronized)
- RT C2 (unsynchronized/synchronized)
- RT C3 (synchronized)

The user determines which communication class that shall be enabled during network configuration.

HMS standard GSD-file contains two DAPs (Device Access Points):

- **DAP v1 RT**

  Complies with version 1.0 of the GSDML specification and may be used by IO Controllers that do not support the interface- and port concept specified in version 2.2 of the GSDML specification. Invoking this DAP enables RT C1 (unsynchronized RT communication).

- **DAP v2 RT**

  Complies with version 2.2 of the GSDML specification and supports the interface- and port concept. Invoking this DAP enables RT C2 (unsynchronized RT communication), RT C2 (synchronized communication) also known as IRT high flexibility "IRT flex", and RT C3 (synchronized communication) known as IRT high performance "IRT top".

Note 1: The communication class cannot be specified by the application. It is specified by the I/O Controller alone during network start-up.

Note 2: When using RT C2 (synchronized communication) or RT C3 make sure to use ethernet switches approved for this. If using RT C1 or RT C2 (unsynchronized communication) standard ethernet switches can be used.

See also...

- "HMS Standard GSD-file" on page 31

3.2 Performance Characteristics

The Anybus module supports:

- 100 Mbps, full duplex (fixed)
- Isochronous cycle times in the range 1 to 4 ms in increments of 250 us

3.3 Device Model

- **Application Process Instances (API)**
  The Anybus module implements the mandatory API 0 (zero).
- **Slots & Subslots**
  The Anybus module supports Up to 17 slots, with 4 subslots per slot.
- **IO Data**
  The Anybus module supports up to 220 bytes of IO Data in each direction.

3.4 Diagnostics & Alarms (Advanced Mode Only)

In the Anybus implementation, diagnostic entries are always created in conjunction with Diagnostic Alarms. The following alarm types are supported:

- **Diagnostic Alarm**

  There are two types of Diagnostic Alarms; Channel- and Generic Diagnostic Alarms. Generally, it's recommended to use Channel Diagnostic Alarms, since the semantics are fully defined by the PROFINET specification. Alternatively, vendor-specific diagnostics can be accomplished by means of Generic Diagnostic Alarms.

  Each time an event is reported as a Diagnostic Alarm, a corresponding diagnostic entry is created and stored by the Anybus module. The IO Controller/Supervisor may read the diagnostic entry using Record Data requests as follows:

Index	Contents
800Ah... 800Bh	Channel diagnostic data for a specific Submodule
800Ch	Generic diagnostic data for a specific Submodule
C00Ah... C00Bh	Channel diagnostic data for a specific Module
C00Ch	Generic diagnostic data for a specific Module
E00Ah... E00Bh	Channel diagnostic data for an AR
E00Ch	Generic diagnostic data for an API
F00Ah... F00Bh	Channel diagnostic data for an API
F00Ch	Generic diagnostic data for an API

  When an event has been resolved, an additional Diagnostic alarm shall be issued to inform the IO Controller that the event has been handled. As a result, the corresponding diagnostic entry is removed. Note that diagnostic data only can be obtained from configured modules/submodules.

  For more information about how to interpret the data, consult the PROFINET specification.

- **Process Alarm**

  A Process Alarm signals the occurrence of an event related to the process, e.g. over temperature, short circuit etc.

- **Pull/Plug Alarm**

  Issued each time the application pulls/plugs modules and submodules during runtime.

See also...

- "Advanced Mode (PNIO_ADV_MODE)" on page 99
- "Add Channel Diagnostic Alarm (PNIO_ADD_CHANNEL_DIAG_ALARM)" on page 117
- "Add Generic Diagnostic Alarm (ADD_GENERIC_DIAG_ALARM)" on page 121
- "Add Process Alarm (PNIO_ADD_PROCESS_ALARM)" on page 123
- "Alarm Acknowledge Received (PNIO_IND_ALARM_ACK_RECEIVED)" on page 136[1]
- "Remove Diagnostic Alarm (PNIO_REMOVE_DIAG_ALARM)" on page 120
- "Plug Module (PNIO_PLUG_MODULE)" on page 105
- "Plug Submodule (PNIO_PLUG_SUBMODULE)" on page 107
- "Pull Module (PNIO_PULL_MODULE)" on page 109
- "Pull Submodule (PNIO_PULL_SUBMODULE)" on page 110

1. If enabled, the Anybus module will issue this mailbox message each time the IO Controller has processed an alarm. See also "Advanced Mode (PNIO_ADV_MODE)" on page 99.

3.5 Initial Parameters (Advanced Mode Only)

During network startup, the module may optionally be loaded with initial parameters which are defined in the GSD-file. This is carried out using Record Data requests, which means that the data will either be written to the Parameter Data areas or sent to the application depending on how the module has been initialised.

When all initial parameters have been loaded, the IO Controller signals this to the Anybus module. At this stage, the Anybus module will act on the settings specified in PNIO_ADV_MODE as follows:

- 'End of Prm'-bit = 1

  The Anybus module issues the spontaneous mailbox message PNIO_IND_END_OF_PRM.

  The application can then examine the initial parameters and decide whether or not to enter data exchange by providing an appropriate mailbox response.
- 'End of Prm'-bit = 0

  The Anybus module automatically enters data exchange.

See also...

- "Modes of Operation" on page 16
- "Advanced Mode (PNIO_ADV_MODE)" on page 99
- "End of Parameterization (PNIO_IND_END_OF_PRM)" on page 111
- "Application State Ready (PNIO_APPL_STATE_READY)" on page 112
- "Record Data Read (PNIO_IND_RECORD_DATA_READ)" on page 113
- "Record Data Write (PNIO_IND_RECORD_DATA_WRITE)" on page 115

3.6 Configuration Data

During network startup, the IO Controller sends its expected configuration to the Anybus module for validation. The Anybus module handles this slightly differently depending on how it has been initialized:

- **Generic Anybus Mode**

  The Anybus module will adopt to the expected configuration sent by the IO Controller.

- **Advanced Mode**

  In the event of a mismatch, the Anybus will act based on the settings in the PNIO_ADV_MODE, see below.

  - 'Cfg Mismatch'-bit = 1

    In the event of a mismatch, the Anybus issues PNIO_IND_CFG_MISMATCH. The application can then decide whether to reject the configuration or to adopt the expected configuration sent by the IO Controller.

  - 'Cfg Mismatch'-bit = 0

    In the event of a mismatch, the Anybus will reject the IO Controller configuration.

See also...

- "Modes of Operation" on page 16
- "Advanced Mode (PNIO_ADV_MODE)" on page 99
- "Configuration Mismatch (PNIO_IND_CFG_MISMATCH)" on page 137

3.7 Identification & Maintenance (I&M)

Identification & Maintenance (from now on referred to as I&M) provides a standardized way of gathering information about an IO device.

The I&M data is accessed using Record Data requests as follows:

Index	Contents	Comments
AFF0h	IM0	read-only
AFF1h	IM1	read/write
AFF2h	IM2	read/write
AFF3h	IM3	read/write
AFF4h	IM4	read/write

These requests are handled differently depending on how the Anybus module has been initialized:

- **Generic Mode (or Advanced Mode with 'I&O Request'-bit = 0)**

  By default, the Anybus module identifies itself as a generic HMS product. This can be customized by implementing PNIO_SET_IM0_INFO, see "Set IM0 Information (PNIO_SET_IM0_INFO)" on page 134.

  Note however that all slots/sub-slots will return the same information.

- **Advanced Mode ('I&M Request'-bit = 1)**

  I&M-related Record Data requests are routed to the application, allowing each slot and subslot to carry it's own I&M information.

See also...

- "Modes of Operation" on page 16
- "Data Exchange (PROFINET)" on page 19
- "Advanced Mode (PNIO_ADV_MODE)" on page 99
- "Record Data Read (PNIO_IND_RECORD_DATA_READ)" on page 113
- "Record Data Write (PNIO_IND_RECORD_DATA_WRITE)" on page 115
- "Set IM0 Information (PNIO_SET_IM0_INFO)" on page 134

3.8 Electronic Data Sheet (GSD-file)

3.8.1 General Information

On PROFINET, the characteristics of a device is stored in an XML-format data file with the suffix GSD. This file is used by the PROFINET configuration tool when setting up the network.

HMS provides a generic GSD-file, which corresponds to the default settings in the module (i.e. when operating in Generic Anybus Mode with the default identity settings).

See also...

- "Conformance Notes" on page 13
- "Modes of Operation" on page 16

3.8.2 HMS Standard GSD-file

The standard GSD-file has the following properties:

- Complies to v2.2 of the GSDML specification for PROFINET IO
- Features two Device Access Points (DAP):
  - 'DAP v1', used for RT C1
  - 'DAP v2', used for RT C2 and RT C3
- Available 'DAP v1' modules and their block size: 1, 2, 4, 8, 16, 32, 64, 128 bytes.
- Available 'DAP v2' modules and their block size: 1, 2, 4, 8, 16, 32, 64, 128 bytes.
- Modules have no assigned 'Initial Parameters'
  - See "Initial Parameters (Advanced Mode Only)" on page 29
- Modules are consistent over the entire block size
- All modules are available as input, output and bidirectional (input/output)
- An 'Empty slot'-module can be used to achieve a perfect configuration match in Generic Anybus Mode
- Each module is associated with exactly one submodule. Multiple submodules per module requires a custom GSD-file

---

IMPORTANT: *Due to the flexible nature of the Anybus-S concept, it is possible to alter the behavior of the product in a way that invalidates the standard GSD-file. In such case, it is necessary to recertify the end product with a custom GSD-file.*

For further information, contact HMS.

3.8.3 How to Associate a Bitmap to a Device Access Point

It is possible to associate a bitmap to a Device Access Point, using the GSD file.

For the Device Access Point, the following information needs to be added (add it right before the "</DeviceAccessPointItem>"):

```
<Graphics>
        <GraphicItemRef Type="DeviceSymbol" GraphicItemTarget="X"/>
</Graphics>
```

In addition to this, a list of graphics needs to be created. This list can be added directly after, for example, the "</DeviceAccessPointList>", or "</ValueList>" keywords. Please note that the "X" above and below shall be replaced with the proper value (if only one bitmap is used, replace X with 1).

```
<GraphicsList>
        <GraphicItem ID="X" GraphicFile="GSDML-VVVV-DDDD-N…N"/>
</GraphicsList>
```

The format of the name of the bitmap shall be as specified above, where VVVV corresponds to the Vendor ID (for example, "010C"), DDDD corresponds to the Device ID (for example, "0009") and "N…N" is a vendor specific extension (for example, "ABSPRTPIC1").

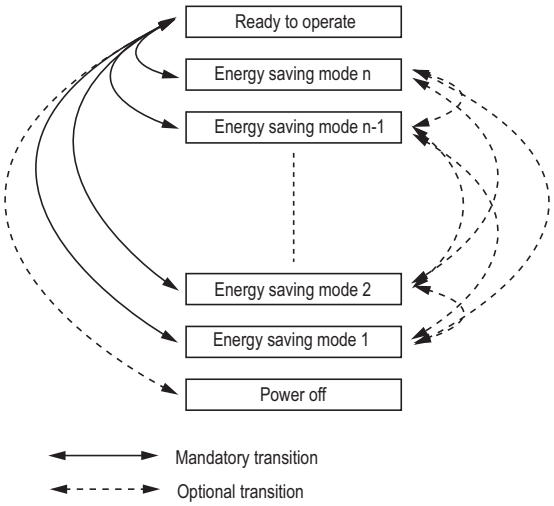
3.9 PROFIenergy Profile

The Anybus-S PROFINET IO module supports the PROFIenergy profile, according to the PROFIenergy Technical Specification, rev. 1.0. This profile makes it possible for a user to temporarily put a device in energy saving mode, e.g. during a lunch break or during weekends. The amount of power used by machines, when they are not in active use is thus reduced. Each device can be set individually to the energy saving mode that is the most optimal depending on the length of the production stop. Operators in factories, for example, can easily set all devices at the same time in the, for each device, optimal energy saving mode.

3.9.1 PROFIenergy States

Using PROFIenergy, it is possible to set the application to different energy modes from the network side. The network will specify the duration of the production stop, and the devices will then themselves determine the most efficient energy mode. The number of modes available is device specific and determined by the device.

The energy saving scheme defines several different energy levels (modes). It is possible to trigger transitions between these from the network side. The transitions from the "Ready to operate" mode to all available energy saving modes, and vice versa, are mandatory. All other transitions are optional. The transition from "Ready to operate" mode to "Power off" mode is not mandatory, as repowering the device may mean a hands-on restart. The host is allowed to change the energy saving mode without interacting with the Anybus-S module.



The number of available energy modes are set when enabling the PROFIenergy functionality by using the mailbox "Enable PROFIenergy (PNIO_ENABLE _PROFINENERGY)".

Energy mode 0 corresponds to the "Power off" mode (the device saves as much energy as possible). Energy mode 255 corresponds to the "Ready to operate" mode (the device is fully operational and saves no energy at all). For each increment of the energy mode the amount of energy that is consumed is increased, ie. the lower the energy mode is the more energy is saved.

In most cases, not the entire range of modes will be used (0 - 255). Only two modes are mandatory to implement: energy mode 0 ("Power off") and energy mode 255 ("Ready to operate").

3.9.2 Implementation

The PROFIenergy profile is implemented in the Anybus-S PROFINET IO module according to the state machine described in the PROFIenergy Technical Specification (available from PROFIBUS International). PROFIenergy commands arriving from the network will be translated into the Anybus-S implementation as follows:

PROFIenergy command	Subcommand	Anybus-S Implementation
Start_Pause	-	Translated into mailbox “Start PROFIenergy Pause (PNIO_IND_PE_START_PAUSE)”
End_Pause	-	Translated into mailbox “End PROFIenergy Pause (PNIO_IND_PE_END_PAUSE)”
Query_Modes	List_Energy_Saving_Modes	Depends on the amount of mailboxes stated to be supported by mailbox “Enable PROFIenergy (PNIO_ENABLE_PROFIENERGY)”
	Get_Mode	Translated into mailbox “Get PROFIenergy mode (PNIO_IND_PE_GET_ENERGY_MODE)”
PEM_Status	-	Translated into mailbox “Get PROFIenergy status (PNIO_IND_PE_GET_STATUS)”. Information about the current energy state can be read using mailbox “Get PROFIenergy mode (PNIO_IND_PE_GET_ENERGY_MODE)”
PE_Identity	-	Returns information set with mailbox “Enable PROFIenergy (PNIO_ENABLE_PROFIENERGY)” (Parameter PEFunctionality).
Query_Measurement	Get_Measurement_List	Not supported
	Get_Measurement_Values	

For more information about the PROFIenergy mailboxes, see “PROFIenergy Specific Commands” on page 137.

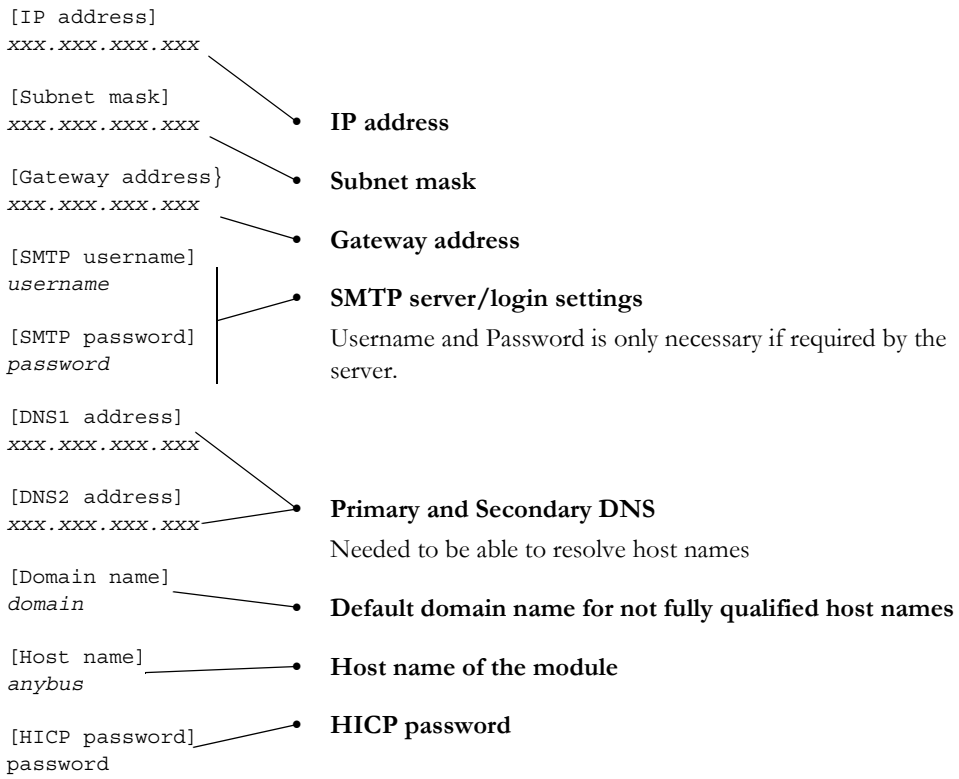
4. Network Configuration

4.1 TCP/IP Settings

4.1.1 General

To be able to participate on the network, the module needs a valid TCP/IP configuration. These settings are stored in the system file '\ethcfg.cfg'.

File Format:



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

- Mailbox Commands (See "General Configuration Commands" on page 61)
- Using DCP (See "DCP (Discovery and Basic Configuration)" on page 36).
- HICP (See "HICP (Anybus IPconfig)" on page 37)
- SSI (See "Server Side Include (SSI)" on page 46)

4.1.2 DCP (Discovery and Basic Configuration)

The Anybus module fully supports the DCP protocol, which allows an IO Controller/Supervisor to change the IP settings of the module during runtime.

4.1.3 LLDP (Link Layer Discovery Protocol)

The Link Layer Discover Protocol (LLDP) provides information about which "partner" that is connected to which ethernet port. This information is stored in the LLDP-MIB that can be read using SNMP (below).

4.1.4 SNMP (Simple Network Management Protocol)

The module supports the Simple Network Management Protocol (SNMP), which enables a Network Management Station to remotely configure, monitor and manage the Anybus module. The protocol uses a message-based communication scheme which is used to retrieve data from the Management Information Base (MIB) in the module.

Setting SNMP parameters

There are a couple of SNMP parameters that can be customized by the application through different mailboxes:

PNIO MIB:

- "System name" - Default setting: <none>. Override this parameter by sending mailbox PNIO_SET_STATION_NAME[1].
- "System Description" - Default setting: "HMS Industrial Networks Anybus-S". Override this parameter by sending mailbox PNIO_SET_SYSTEM_DESCRIPTION[1].
- "Interface Description" - Default setting: "PROFINET IO interface". Override this parameter by sending mailbox PNIO_SET_INTERFACE_DESCRIPTION[1].

MIB2:

- "System Description" - Default setting: "HMS Industrial Networks Anybus-S". Override this parameter by sending mailbox PNIO_SET_SYSTEM_DESCRIPTION_MIB2[1].
- "System Contact" - Default setting: "www.anybus.com". Override this parameter by sending mailbox PNIO_SET_SYSTEM_CONTACT_MIB2[1].
- "System Location" - Default setting: "Stationsgatan 37, 302 45 Halmstad, Sweden". Override this parameter by sending mailbox PNIO_SET_SYSTEM_LOCATION_MIB2[1].

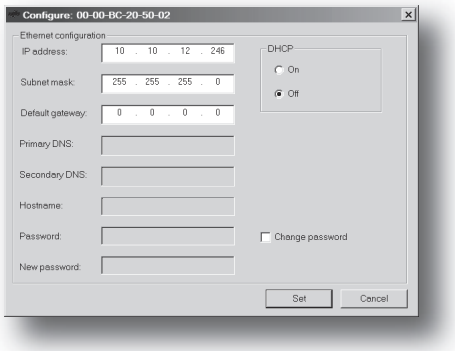
Note: These mailboxes can only be sent before "END_INIT".

1. See "PROFINET Specific Commands" on page 98.

4.1.5 HICP (Anybus IPconfig)

The module supports the HICP protocol used by the Anybus IPconfig utility, which can be downloaded free of charge from the HMS website. This utility may be used to access the network settings of any Anybus product connected to the network via UDP port 3250.

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-PIR'.



To alter the network settings of the module, double-click on its entry in the list. A window will appear, containing the settings for the module.

Validate the new settings by clicking 'Set'. The new IP configuration will be stored in '\ethcfg.cfg'.

Optionally, the configuration can 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'. The password is stored in the system file '\ethcfg.cfg'.

Note 1: This feature cannot be used if SET_ETN_CONFIG has been issued (see "Set Ethernet Configuration (SET_ETN_CONFIG)" on page 62).

Note 2: Unlike most other Anybus product, Anybus IPconfig cannot detect the Anybus-S PROFINET IRT if it's located on a different subnet (i.e. if the Anybus module has no IP address (i.e. 0.0.0.0), it will not be visible in Anybus IPconfig).

4.1.6 IP Access Control

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

File Format:

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

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

```
[Modbus/TCP]
xxx.xxx.xxx.xxx
```
Nodes listed here may connect to the module via Modbus/TCP

```
[All]
xxx.xxx.xxx.xxx
```
Fallback setting, used by the module when one or several of the keys above are omitted

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

4.2 PROFINET Settings

The file '\pniocfg.cfg' holds various PROFINET-related settings. The file is read once during startup, i.e. the Anybus module must be restarted in order for any changes to have effect (Unless it's contents has been changed by an IO Controller/Supervisor via the DCP protocol. In such case, the settings will have effect immediately).

Example:

```
[Station Name]
Nice Device

[Station Type]
ABS-PIR

[Vendor ID]
0x010C

[Device ID]
0x0006

[System Description]
"HMS Industrial Networks Anybus-S"

[System Interface]
"PROFINET IO interface"

[System Contact]
"www.anybus.com"

[System Location]
"Stationsgatan 37,
302 45 Halmstad,
Sweden"
```

Station Name

Station name as ASCII string, maximum 64 characters.

Station Type

Station type as ASCII string, maximum 64 characters.

Vendor ID

16 bit hexadecimal value, with the prefix 0x. Assigned by the PNO.

Device ID

16 bit hexadecimal value, with the prefix 0x. Assigned by vendor.

System Description

SNMP parameter

System Interface

SNMP parameter.

System Contact

SNMP parameter

System Location

SNMP parameter.

5. FTP Server

5.1 General Information

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)

5.2 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.

  If the mailbox command GLOBAL_ADMIN_MODE (see "Global Admin Mode (GLOBAL_ADMIN_MODE)" on page 78) is issued during startup, the Anybus will grant Admin-level access to all users. In such case, the FTP server accepts any username/password combination, and the root directory will be '\'.

5.3 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:

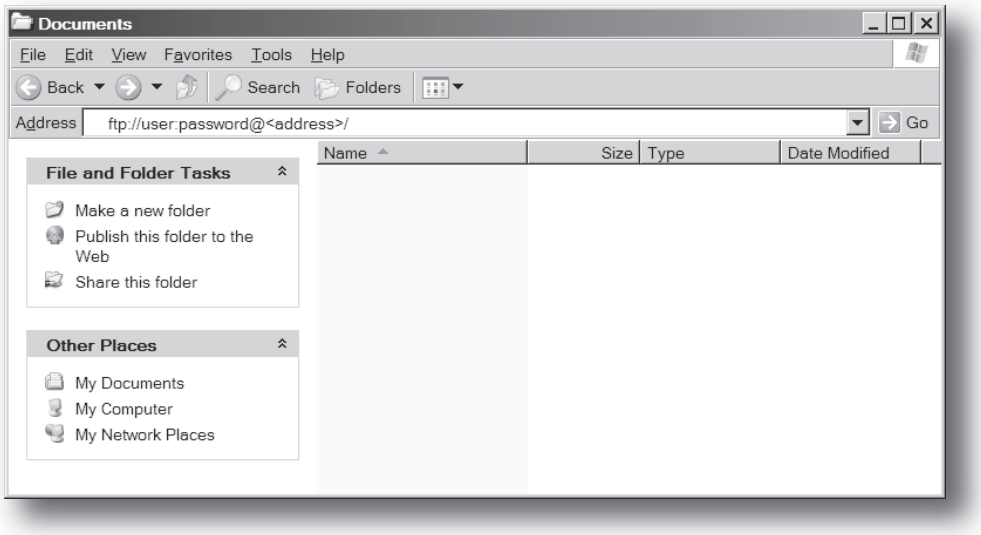
```
Username1:Password1
Username2:Password2
Username3:Password3
```

Note: If no valid user accounts have been defined, the Anybus will grant Admin-level access to all users. In such case, the FTP server accepts any username/password combination, and the root directory will be '\'.

5.4 FTP Connection Example (Windows Explorer)

The built in FTP client in Windows Explorer can easily be used to access the filesystem 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 Anybus module using the specified settings. If successful, the filesystem of the module is displayed in the Explorer window.



6. Web Server

6.1 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...

- "Disable Web Server (DISABLE_WEB_SERVER)" on page 75
- "Enable Web Server (ENABLE_WEB_SERVER)" on page 76

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'.

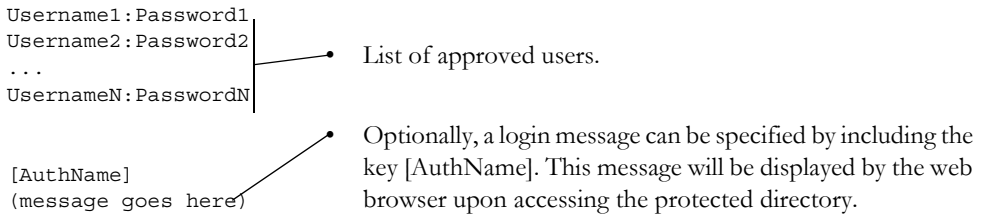
These virtual files are:

```
\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
```

6.2 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:



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 "Protected Files" on page 41.

6.3 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.

7. SMTP Client

7.1 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...

- "Server Side Include (SSI)" on page 46
- "Send Email (SEND_EMAIL)" on page 181

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...

- "Set SMTP Server (SET_SMTP_SERVER)" on page 72
- "Send Email (SEND_EMAIL)" on page 181

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 "Email Definitions" on page 45.

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.

7.2 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.

8. 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.

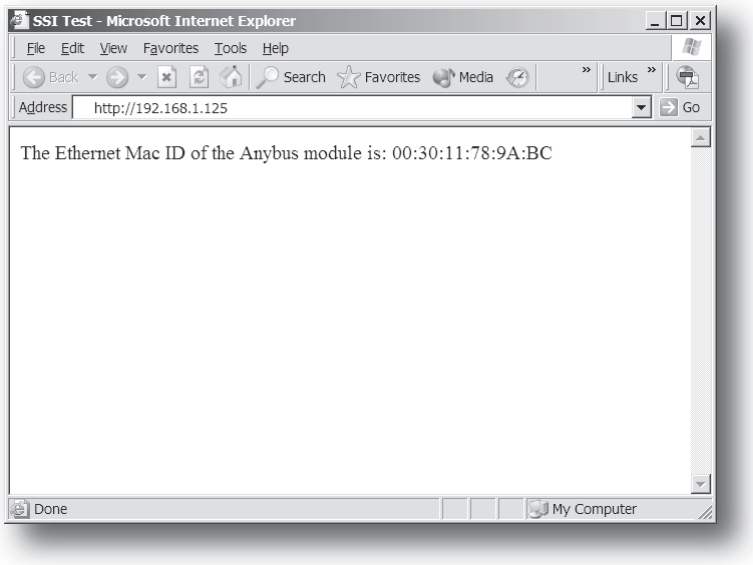
```
<?--#exec cmd_argument='XXXXXXXXXXXXXXXXXXXXX'-->
```

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:



8.1 Functions

DisplayMacID

This function returns the MAC ID of the Anybus module in format xx:xx:xx:xx:xx:xx.

Syntax:

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

DisplayMacIDPort1

This function returns the MAC ID of ethernet port 1 in format xx:xx:xx:xx:xx:xx.

Syntax:

```
<?--#exec cmd_argument='DisplayMacIdPort1'-->
```

DisplayMacIDPort2

This function returns the MAC ID of ethernet port 2 in format xx:xx:xx:xx:xx:xx.

Syntax:

```
<?--#exec cmd_argument='DisplayMacIdPort2'-->
```

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 host name.

Syntax:

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

DisplayDomainName

This function returns the default domain name.

Syntax:

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

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'-->
```

DisplayStationName

This function returns the PROFINET Station Name.

Syntax:

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

DisplayStationType

This function returns the PROFINET Station Type.

Syntax:

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

DisplayVendorID

This function returns the PROFINET Vendor ID.

Syntax:

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

DisplayDeviceID

This function returns the PROFINET DeviceID.

Syntax:

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

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 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'-->
```

8.2 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()".

8.2.1 SSI Output String File

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

The files shall have the following format:

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

[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

8.2.2 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) )'-->
```

9. Fieldbus Specific Mailbox Commands

9.1 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
0027h	Mailbox command not allowed in this state
0028h	Reserved value not set to zero

Register Value	Description
0029h	Value out-of-range
002Ah	Slot number out-of-range
002Bh	Subslot number out-of-range
002Ch	Channel number out-of-range
002Eh	Channel error type out-of-range
002Fh	Channel type out-of-range
0030h	Channel spec out-of-range
0031h	Channel dir out-of-range
0032h	User structure identifier out-of-range
0033h	Slot occupied
0034h	Subslot occupied
0035h	No module
0036h	No submodule
0037h	Memory overlap
0038h	Memory out-of-range
0039h	Extended fault information available (See “Extended Fault Information” on page 60)
003Ah	Slot 0 is not plugged
003Bh	Invalid application handle
003Ch	Module/submodule cannot be pulled/plugged
003Dh	Failed to set operation state
003Eh	Failed to send alarm
003Fh	Failed to add/remove diagnostic entry
0040h	I&M X not supported
0041h	AR handle not valid
0042h	Failed to abort the AR
0043h	EnergyModes out-of-range
0044h	PEFunctionality out-of-range
0045h	EnergyModes not supported
0046h	No suitable energy mode available

9.2 Extended Fault Information

When 'fault information' indicates 'Extended Fault Information available', additional information from the PROFINET stack is available in Extended Word 7.

The fault codes are:

Register Value	Description
1000h	No resources available
1002h	System adaptation error
1003h	Sequence error
1004h	Parameter error
1005h	Incorrect opcode
1006h	Memory allocation error
1007h	Error in lower layer of the PNIO stack
1008h	Buffer cannot be locked
1009h	Operation aborted
100Ah	Request belongs to an old session
100Bh	RPC timeout error
100Ch	RPC communication error
100Dh	RPC peer signalled busy (try again later)
100Eh	Function call-up not possible at the moment
1100h	Attempted pull from an empty slot
1101h	Attempted pull from an empty subslot
1102h	Attempted plug into an occupied slot
1103h	Attempted plug into an occupied subslot
1104h	Invalid parameter
1105h	Function cannot be called since no valid Application Relation (AR) is available

9.1 General Configuration Commands

Commands in this category:

Mailbox Commands	Description	Page
Set Ethernet Configuration (SET_ETN_CONFIG)	Set network settings	62
Read Ethernet Configuration (READ_ETN_CONFIG)	Retrieve the currently used network settings	63
Get MAC Address (GET_MAC_ADDR)	Retrieve the ethernet MAC address from the module	64
Set MAC Address (SET_MAC_ADDR)	Set the ethernet MAC address of the module	66
Set Host and Domain (SET_HOST_DOMAIN)	Set host and default domain names	67
Get Host and Domain (GET_HOST_DOMAIN)	Returns the currently used host and domain	68
Disable / Enable HICP (HICP_CFG_CONTROL)	Disable/enable HICP support	69
Set DNS Server Settings (SET_DNS_SERVERS)	Configures the DNS server settings	70
Get DNS Server Settings (GET_DNS_SERVERS)	Returns the currently used DNS server settings	71
Set SMTP Server (SET_SMTP_SERVER)	Configures the SMTP server IP address	72
Set SMTP Server by Name (SET_SMTP_SERVER_NAME)	Configures the SMTP server address	73
Get SMTP Server (GET_SMTP_SERVER)	Returns the currently used SMTP server IP address	74
Disable Web Server (DISABLE_WEB_SERVER)	This command disables the built in web server	75
Enable Web Server (ENABLE_WEB_SERVER)	This command enables the built in web server	76
Disable FTP server (DISABLE_FTP_SERVER)	This command disables the built in FTP server	77
Global Admin Mode (GLOBAL_ADMIN_MODE)	This command instruct the module to run in global admin mode	78
Disable Virtual File System (DISABLE_VFS)	Disable the virtual file system	79

9.1.1 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)	

9.1.2 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**

9.1.3 Get MAC Address (GET_MAC_ADDR)

Description

This command returns the MAC addresses of the module and it's ethernet ports.

Initiated by	Application
Command no.	0010h
Extended Header	MAC Selection
Message data	-
Response data	MAC Data

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	(size)	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	MAC Selection	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	-	
		MAC Data	**Response data word 1** **...** **Response data word N**

- **MAC Selection & MAC Data**

  MAC Data holds 6... 18 bytes of data depending on the value of MAC Selection.

MAC Selection	MAC Data[0...5]	MAC Data[6...11]	MAC Data[12...17]
0	Module MAC Address	-	-
1	Port 1 MAC Address	-	-
2	Port 2 MAC Address	-	-
FFFFh	Module MAC Address	Port 1 MAC Address	Port 2 MAC Address

Note: MAC Selection 2 (Port 2 MAC Address) is only available on two-port versions of the product. Also note that for one-port versions, MAC Selection FFFFh will only return 12 bytes (Module MAC Address + Port 1 MAC Address)

9.1.4 Set MAC Address (SET_MAC_ADDR)

Description

This command sets the MAC address of the module.

Note: This command may only be issued during initialisation.

Initiated by	Application
Command no.	0019h
Extended Header	-
Message data	The new MAC 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	0019h	0019h	*SET_MAC_ADDR*
Data size	0006h	0006h	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	MAC Selection	-	
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	MAC address byte 1, 2	MAC address byte 1, 2	**Response data word 1**
Message data word 2	MAC address byte 3, 4	MAC address byte 3, 4	**Response data word 2**
Message data word 3	MAC address byte 5, 6	MAC address byte 5, 6	**Response data word 3**

9.1.5 Set Host and Domain (SET_HOST_DOMAIN)

Description

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

Note: This command may only be issued during initialisation.

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

Command and response layout

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

9.1.6 Get Host and Domain (GET_HOST_DOMAIN)

Description

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

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

Command and response layout

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

9.1.7 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

9.1.8 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.

9.1.9 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.

9.1.10 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)	

9.1.11 Set SMTP Server by Name (SET_SMTP_SERVER_NAME)

Description

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

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

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

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

Command and response layout

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

- **SMTP Server**

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

9.1.12 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**

9.1.13 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	-	-	

9.1.14 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	-	-	

9.1.15 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	-	-	

9.1.16 Global Admin Mode (GLOBAL_ADMIN_MODE)

Description

This command instructs the module to run in Global Admin Mode. For more information, see "User Accounts" on page 39.

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	-	-	

9.1.17 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	-	-	

9.1 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.	81
Close File (FILE_CLOSE)	Close a file previously opened using FILE_OPEN	82
Read File (FILE_READ)	Read data from a file	83
Write File (FILE_WRITE)	Write data to a file.	84
Delete File (FILE_DELETE)	Delete a file	85
Move File (FILE_MOVE)	Moves a file	86
Rename File (FILE_RENAME)	Rename a file	87
Copy File (FILE_COPY)	Copy a file	88
Create Directory (DIR_CREATE)	Create a new directory	89
Delete Directory (DIR_DELETE)	Delete an empty directory	90
Open Directory (DIR_OPEN)	Open a directory	91
Read Directory (DIR_READ)	Read contents of a directory previously opened using DIR_OPEN	92
Close Directory (DIR_CLOSE)	Close a directory previously opened using DIR_OPEN	94
Create RAM disc (CREATE_RAM_DISC)	Mounts the RAM disc into a specified directory in the file system.	95
Format File System (FORMAT_FS)	Formats the file system	96
File System Checksum (CRC_FS)	Calculates checksums related to the file system	97

9.1.1 Open File (FILE_OPEN)

Description

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

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

Command and response layout

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

- **Mode**

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

- **Filesize**

  Current filesize (if applicable).

- **File Handle**

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

9.1.2 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 "File Handle" on page 81.
- **Filesize**
  Size of the file.

9.1.3 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 220 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 "File Handle" on page 81.
- **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).

9.1.4 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. 220 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 "File Handle" on page 81.

- **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.

9.1.5 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 220 bytes*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message data	Path + filename (String, null-terminated)	Path + filename (String, null-terminated)	

9.1.6 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)	

9.1.7 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)	

9.1.8 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)	

9.1.9 Create Directory (DIR_CREATE)

Description

This command creates a directory in the file system.

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

Command and response layout

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

9.1.10 Delete Directory (DIR_DELETE)

Description

This command deletes an empty directory from the file system.

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

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0069h	0069h	*DIR_DELETE*
Data size	(size)	(size)	*Size of path string*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault information	
Message data	Path + name (String, null-terminated)	Path + name (String, null-terminated)	

9.1.11 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 "Reading the Contents of a Directory" on page 93.

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

Command and response layout

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

- **Directory Handle**

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

9.1.12 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 "Reading the Contents of a Directory" on page 93.

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

Command and response layout

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

- **Directory Handle**

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

- **Object Size**

  Size of object (i.e. filesize).

- **Object Flags**

  Various flags specifying the nature of the object:

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

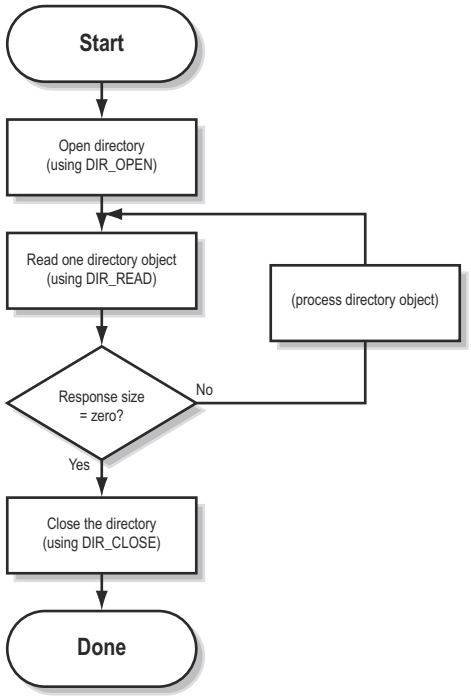
Bit	Description	
DIR	Directory flag	0: Object is a file 1: Object is a directory
RO	Read only	0: Object can be read or written 1: Object is read-only
H	Hidden	0: Object is visible 1: Object is hidden
SYS	System	0: User object 1: System object

- **Object Name**

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

Reading the Contents of a Directory

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



9.1.13 Close Directory (DIR_CLOSE)

Description

This command closes a directory previously opened using DIR_OPEN.

See also "Reading the Contents of a Directory" on page 93.

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	

9.1.14 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	*CREATE_RAM_DISC*
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)	

- **Path**

  RAM-disc location. Must be empty, or the command will fail.

9.1.15 Format File System (FORMAT_FS)

Description

This command formats the file system.

Initiated by	Application
Command no.	006Dh
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	006Dh	006Dh	*FORMAT_FS*
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	-	-	

9.1.16 File System Checksum (CRC_FS)

Description

This command calculates checksums related to the file system.

Initiated by	Application
Command no.	006Eh
Extended Header	-
Message data	-
Response data	Used Sector CRC, File System CRC

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	0000h	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	-	-	
		Used Sector CRC	**Response data word 1**
		File System CRC	**Response data word 2**

9.1 PROFINET Specific Commands

General

Commands in this category:

Mailbox Command	Page
Advanced Mode (PNIO_ADV_MODE)	99
Set Stop Mode Action (PNIO_SET_STOP_MODE_ACTION)	101
Set Device Identity (PNIO_SET_DEVICE_IDENTITY)	102
Set Station Name (PNIO_SET_STATION_NAME)	103
Set Station Type (PNIO_SET_STATION_TYPE)	104
Plug Module (PNIO_PLUG_MODULE)	105
Plug Submodule (PNIO_PLUG_SUBMODULE)	107
Pull Module (PNIO_PULL_MODULE)	109
Pull Submodule (PNIO_PULL_SUBMODULE)	110
Application State Ready (PNIO_APPL_STATE_READY)	112
Add Channel Diagnostic Alarm (PNIO_ADD_CHANNEL_DIAG_ALARM)	117
Remove Diagnostic Alarm (PNIO_REMOVE_DIAG_ALARM)	120
Add Generic Diagnostic Alarm (ADD_GENERIC_DIAG_ALARM)	121
Add Process Alarm (PNIO_ADD_PROCESS_ALARM)	123
Abort AR (PNIO_AR_ABORT)	125
Set port MAC address (PNIO_SET_PORT_MAC_ADDR)	126
Set System Description (PNIO_SET_SYSTEM_DESCRIPTION)	127
Set Interface Description (PNIO_SET_INTERFACE_DESCRIPTION)	128
Set System Description MIB-II (PNIO_SET_SYSTEM_DESCRIPTION_MIB2)	129
Set System Contact MIB-II (PNIO_SET_SYSTEM_CONTACT_MIB2)	130
Set System Location MIB-II (PNIO_SET_SYSTEM_LOCATION_MIB2)	131
Get IM0 Information (PNIO_GET_IM0_INFO)	132
Set IM0 Information (PNIO_SET_IM0_INFO)	134

Spontaneous Messages

The following messages may be issued by the Anybus when operating in Advanced Mode.

Mailbox Command	Page
End of Parameterization (PNIO_IND_END_OF_PRM)	111
Record Data Read (PNIO_IND_RECORD_DATA_READ)	113
Record Data Write (PNIO_IND_RECORD_DATA_WRITE)	115
Alarm Acknowledge Received (PNIO_IND_ALARM_ACK_RECEIVED)	136
Configuration Mismatch (PNIO_IND_CFG_MISMATCH)	137
Check AR Indication (PNIO_IND_CHECK_AR)	139
AR Info Indication (PNIO_IND_AR_INFO)	141
Abort AR Indication (PNIO_IND_AR_ABORT)	143
AR Offline Indication (PNIO_IND_AR_OFFLINE)	145

9.1.1 Advanced Mode (PNIO_ADV_MODE)

Description

This command instructs the Anybus module to operate in Advanced Mode, and extends the PROFINET specific functionality.

Note: This command may only be issued during initialisation, preceded by Anybus Init.

Command initiator	Application
Command number	0100h
Extended Header data	Spontaneous Mailbox settings, Max.no. of slots, Max.no. of subslots
Message data	-
Response message	-

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0100h	0100h	*PNIO_ADV_MODE*
Data size	0000h	0000h	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	Spontaneous Mailbox	Spontaneous Mailbox	
Extended word 2	Max.no. of slots	Max.no. of slots	
Extended word 3	Max.no. of subslots	Max.no. of subslots	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault Information	

- **Spontaneous Mailbox**

  This bitfield enables/disables certain PROFINET-related mailbox communication.

Bit(s)	Name	Meaning	Related Mailbox Message(s)
0	Cfg Mismatch	1: Notify the application in the event of a configuration mismatch 0: (do not notify the application)	- Configuration Mismatch (PNIO_IND_CFG_MISMATCH)
1	Record Data Request	1: Forward acyclic data through mailbox interface 0: Use the Parameter Data areas for acyclic data exchange	- Record Data Read (PNIO_IND_RECORD_DATA_READ) - Record Data Write (PNIO_IND_RECORD_DATA_WRITE)
2	End of Prm	1: Notify the application when the IO Controller has finished parameterization 0: (do not notify the application)	- End of Parameterization (PNIO_IND_END_OF_PRM)
3	Alarm Ack	1: Notify the application when the IO controller has acknowledged an alarm 0: (do not notify the application)	- Alarm Acknowledge Received (PNIO_IND_ALARM_ACK_RECEIVED)
4	AR Indication	1: Issue mailbox messages when an AR is established or disconnected 0: (do not issue these messages)	- Check AR Indication (PNIO_IND_CHECK_AR) - AR Info Indication (PNIO_IND_AR_INFO) - Abort AR Indication (PNIO_IND_AR_ABORT) - AR Offline Indication (PNIO_IND_AR_OFFLINE)
5	I&M Requests	1: Forward I&M-related requests to the application 0: Handle I&M-related requests internally	- Set IM0 Information (PNIO_SET_IM0_INFO) - Record Data Read (PNIO_IND_RECORD_DATA_READ) - Record Data Write (PNIO_IND_RECORD_DATA_WRITE)
6... 15	(reserved, set to zero)	(reserved, set to zero)	-

See also...

- "Modes of Operation" on page 16
- "Parameter Data (Record Data)" on page 19
- "Diagnostics & Alarms (Advanced Mode Only)" on page 28
- "Initial Parameters (Advanced Mode Only)" on page 29
- "Configuration Data" on page 29
- "Identification & Maintenance (I&M)" on page 30

- **Max.no. of slots**

  Maximum number of slots that can be occupied. Range: 1...17.
  (e.g. a value of 17 means that modules can be plugged into slots 0...16).

- **Max.no. of subslots**

  Maximum number of subslots that can be occupied per slot. Range: 1...4.
  (e.g. a value of 3 means that submodules can be plugged into subslots 1...3).

- **Fault Information**

  (see "Fault Information" on page 58)

9.1.2 Set Stop Mode Action (PNIO_SET_STOP_MODE_ACTION)

Description

This command defines what action to be taken with the Out I/O data when the IO Controller shift operation mode from RUN to STOP.

Command initiator	Application
Command number	0101h
Extended Header data	Output Action when IO Controller is in STOP state.
Message data	-
Response message	-

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0101h	0101h	
Data size	0000h	0000h	*Set Stop Mode Action*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	Output Action	Output Action	*(see below)*
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault Information	

- **Output Action**

  This word defines what to do when the IO Controller shifts operation state.

Value	Action
0000h	Out area cleared (0). Default setting.
0001h	Out area freeze
0002h	Out area set (1)

- **Fault Information**

  (see "Fault Information" on page 58)

9.1.3 Set Device Identity (PNIO_SET_DEVICE_IDENTITY)

Description

This command alters the Vendor- and Device ID.

Command initiator	Application
Command number	0102h
Extended Header data	Desired Vendor ID and Device ID
Message data	-
Response message	The response indicates if the command was accepted.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0102h	0102h	*Set I/O Device Identity*
Data size	0000h	0000h	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	Vendor ID	Vendor ID	
Extended word 2	Device ID	Device ID	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault Information	

- **Vendor ID**
  Identifies the manufacturer of the device. Assigned by PNO.
- **Device ID**
  Identifies the device. Assigned by the manufacturer.
- **Fault Information**
  (see "Fault Information" on page 58)

9.1.4 Set Station Name (PNIO_SET_STATION_NAME)

Description

This command changes the station name stored in the system file 'pniocfg.cfg' (See "PROFINET Settings" on page 38).

Note: This command should only be used when an end user wants to change the station name, using for example a keypad. Normally any changes to the name are done via the network.

Command initiator	Application
Command number	0103h
Extended Header data	-
Message data	Station Name, null terminated.
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	0103h	0103h	*Set Station Name*
Data size	(data size)	(data size)	*Length 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	Station Name	Station Name	

- **Station Name**

  Station name as ASCII-string (NULL terminated).

- **Fault Information**

  (see "Fault Information" on page 58)

9.1.5 Set Station Type (PNIO_SET_STATION_TYPE)

Description

This command changes the station type.

Command initiator	Application
Command number	0104h
Extended Header data	-
Message data	Station Type, null terminated.
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	0104h	0104h	*Set Station Type*
Data size	(data size)	(data size)	*Length 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	Station Type	Station Type	

- **Station Type**

  Station type as ASCII-string (NULL terminated).

- **Fault Information**

  (see “Fault Information” on page 58)

9.1.6 Plug Module (PNIO_PLUG_MODULE)

Description

This command adds modules to the configuration. At least one module must be plugged during startup. Additional modules may be plugged as desired either at startup or during runtime.

If this command is issued during runtime, the Anybus will send a 'Plug Alarm' to the IO Controller. This will in turn cause the IO controller to reload 'Initial parameters' for the module and signal 'End of parameterization'.

Command initiator	Application
Command number	0107h
Extended Header data	-
Message data	Information specifying where to plug the module
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	0107h	0107h	*PNIO_PLUG_MODULE*
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	-	Fault reference	
Extended word 7	-	Extended Fault Information	
Extended word 8	-	Fault Information	
Message data word 1	(reserved, set to zero)	(reserved, ignore)	
Message data word 2	SlotNo	SlotNo	
Message data word 3	ModIdent (msb)	ModIdent (msb)	
Message data word 4	ModIdent (lsb)	ModIdent (lsb)	
Message data word 5	(reserved, set to zero)	(reserved, ignore)	
Message data word 6	(reserved, set to zero)	(reserved, ignore)	

- **SlotNo**
  Number of the slot where to plug the module.
  Range 0...(N-1) (N = max number of slots allowed, specified in PNIO_ADV_MODE).
- **ModIdent**
  Module identifier as stated in GSD file.
- **Fault Information**
  (see “Fault Information” on page 58)
- **Extended Fault Information**
  (see “Extended Fault Information” on page 60)
- **Fault reference**
  Indicates which Message Data word that caused the error (when applicable).

9.1.7 Plug Submodule (PNIO_PLUG_SUBMODULE)

Description

This command adds submodules to the configuration, and can be issued either at startup or during runtime. The application decides where in the parameter data area to locate Record Data requests destined to the submodule itself.

If this command is issued during runtime, the Anybus will automatically send a 'Plug Alarm' and a 'Return of SubModule Alarm' to the IO Controller. This will in turn cause the IO Controller to reload 'Initial parameters' for the submodule and signal 'End of parameterization'.

Note 1: Sub-modules plugged into slot #0 cannot have I/O data assigned to them.

Note 2: The Anybus module will not respond to this command until the IO Controller has acknowledged the 'Plug Alarm'. When using the CPU317-2 PN/DP, this takes approximately 8 bus cycles, which for a 512 ms bus cycle time equals 4 seconds.

Command initiator	Application
Command number	0108h
Extended Header data	-
Message data	Information specifying where to plug the submodule and how to map its record data to the parameter input/output areas
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	0108h	0108h	*PNIO_PLUG_SUBMODULE*
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	-	Fault reference	
Extended word 7	-	Extended Fault Information	
Extended word 8	-	Fault Information	
Message data word 1	(reserved, set to zero)	(reserved, ignore)	
Message data word 2	SlotNo	SlotNo	
Message data word 3	(reserved, set to zero)	(reserved, ignore)	
Message data word 4	SubSlotNo	SubSlotNo	
Message data word 5	SubModIdent (msb)	SubModIdent (msb)	
Message data word 6	SubModIdent (lsb)	SubModIdent (lsb)	
Message data word 7	SubMod Record In offset	SubMod Record In offset	
Message data word 8	SubMod Record In length	SubMod Record In length	
Message data word 9	SubMod Record Out offset	SubMod Record Out offset	
Message data word 10	SubMod Record Out length	SubMod Record Out length	

- **SlotNo**

  Number of the slot where to plug the sub-module.

  Range 0...(N-1) (N = max number of slots allowed, specified in PNIO_ADV_MODE)

- **SubSlotNo**

  Number of the Sub-slot where to plug the sub-module.

  Range 1... (M-1) (M =max number of subslots allowed, specified in PNIO_ADV_MODE)

- **SubModIdent**

  Identifier for sub-module as stated in GSD-file.

- **SubMod Record In offset**

  Offset from start of Parameter Input Area where Record data associated with the sub-module is located.

  Note: This parameter is not relevant when Record Data Request has been enabled in PNIO_ADV_MODE, or if the command is issued during runtime.

- **SubMod Record In length**

  Max Record data length (in bytes) related to sub-module.

  Note: This parameter is not relevant when Record Data Request has been enabled in PNIO_ADV_MODE, or if the command is issued during runtime.

- **SubMod Record Out offset**

  Offset from start of Parameter Output Area where Record data associated with the sub-module is located.

  Note: This parameter is not relevant when Record Data Request has been enabled in PNIO_ADV_MODE, or if the command is issued during runtime.

- **SubMod Record Out length**

  Max Record data length (in bytes) related to sub-module.

  Note: This parameter is not relevant when Record Data Request has been enabled in PNIO_ADV_MODE, or if the command is issued during runtime.

- **Fault Information**

  (see “Fault Information” on page 58)

- **Extended Fault Information**

  (see “Extended Fault Information” on page 60)

- **Fault reference**

  Indicates which Message Data word that caused the error (when applicable).

9.1.8 Pull Module (PNIO_PULL_MODULE)

Description

This command removes a previously plugged module. When this happens, the Anybus will automatically issue a 'Pull Alarm' to the IO Controller.

Command initiator	Application
Command number	0109h
Extended Header data	-
Message data	SlotNo of the module to unplug/pull.
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	0109h	0109h	*PNIO_PULL_MODULE*
Data size	0004h	0004h	*4 bytes (2 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 Fault Information	
Extended word 8	-	Fault Information	
Message data word 1	(reserved, set to zero)	(reserved, ignore)	
Message data word 2	SlotNo	SlotNo	

- **SlotNo**

  Slot where the module to pull is located.

  Range 1...(N-1) (N = max number of slots allowed, specified in PNIO_ADV_MODE)

  Note: A slot value of 0 (zero) is not allowed.

- **Fault Information**

  (see "Fault Information" on page 58)

- **Extended Fault Information**

  (see "Extended Fault Information" on page 60)

9.1.9 Pull Submodule (PNIO_PULL_SUBMODULE)

Description

This command removes a previously plugged submodule. When this happens, the Anybus will automatically issue a 'Pull Alarm' to the IO Controller.

Command initiator	Application
Command number	010Ah
Extended Header data	-
Message data	SlotNo and SubslotNo of the submodule to unplug/pull.
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	010Ah	010Ah	*PNIO_PULL_SUBMODULE*
Data size	0008h	0008h	*8 bytes of data (4 words)*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	Extended Fault Information	
Extended word 8	-	Fault Information	
Message data word 1	(reserved, set to zero)	(reserved, ignore)	
Message data word 2	SlotNo	SlotNo	
Message data word 3	(reserved, set to zero)	(reserved, ignore)	
Message data word 4	SubslotNo	SubslotNo	

- **SlotNo**

  Slot where the submodule to pull is located.

  Range 1...(N-1) (N = max number of slots allowed, specified in PNIO_ADV_MODE)

  Note: A slot value of 0 (zero) is not allowed.

- **SubslotNo**

  Subslot where the submodule to pull is located.

  Range 1... M (M =max number of subslots allowed, specified in PNIO_ADV_MODE)

- **Fault Information**

  (see "Fault Information" on page 58)

- **Extended Fault Information**

  (see "Extended Fault Information" on page 60)

9.1.10 End of Parameterization (PNIO_IND_END_OF_PRM)

Description

The Anybus issues this message to indicate to the application that the parameterization phase is completed. Upon receiving this message, the application must provide a response to indicate whether it is ready for data exchange or not.

Note 1: This message may be issued during run-time when modules/submodules are re-inserted.

Note 2: This functionality must be enabled in PNIO_ADV_MODE ('End of Prm'-bit).

Command initiator	Anybus
Command number	010Bh
Extended Header data	AR Handle, Slot, Subslot, Application State (indicates to the Anybus if the application is ready)
Message data	-
Response message	-

Command and response layout

	Message	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	010Bh	010Bh	*PNIO_IND_END_OF_PRM*
Data size	0000h	0000h	*(no message data)*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	AR Handle	-	
Extended word 2	Slot	-	
Extended word 3	Subslot	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Application state	

- **AR Handle**

  Handle for the Application Relationship.

- **Slot & Subslot**

  These words indicate which slots/subslots that are affected by the message:

Subslot	Slot	Meaning
0 (zero)	(don't care)	Message applies to all modules in the configuration
>0	>0	Message applies to the specified slot/subslot

- **Application state**

  0: Application is ready for data exchange.

  1: Application is not yet ready for data exchange. (PNIO_APPL_STATE_READY must be issued at a later stage to indicate when the application is ready.)

9.1.11 Application State Ready (PNIO_APPL_STATE_READY)

Description

This command indicates to the Anybus that the application is ready for data exchange, and must be issued if the application has previously responded "Not ready for data exchange yet" to the PNIO_IND_END_OF_PRM request.

Note: This functionality must be enabled in PNIO_ADV_MODE ('End of Prm'-bit).

Command initiator	Application
Command number	0106h
Extended Header data	AR Handle
Message data	-
Response message	-

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0106h	0106h	*Application State Ready*
Data size	0000h	0000h	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	AR Handle	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault Information	

- **AR Handle**
  Handle for the Application Relationship.
- **Fault Information**
  (see "Fault Information" on page 58)

9.1.12 Record Data Read (PNIO_IND_RECORD_DATA_READ)

Description

This message may be issued by the Anybus when a Record Data Read request has been received from the IO Controller. Upon receiving this message, the application must assemble the requested data and provide a response.

Note: This functionality must be enabled in PNIO_ADV_MODE ('Record Data Request'- and/or 'I&M Request'-bit).

Command initiator	Anybus
Command number	010Ch
Extended Header data	The message contains details about the issued record data request.
Message data	-
Response message	Read Data

Command and response layout

	Message	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	010C	010Ch	*Record Data Read*
Data size	0000h	(data size)	*Length of* Read Data
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	AR Handle	-	
Extended word 2	-	-	
Extended word 3	Sequence No.	-	
Extended word 4	Slot No..	-	
Extended word 5	Subslot No..	-	
Extended word 6	Index	Error code 1 | Error code 2	
Extended word 7	Length	Add data 1	
Extended word 8	-	Add data 2	
		Read Data	**Message data bytes 1...n**

- **AR Handle**

  Handle for the Application Relationship.

- **Sequence No.**

  The parameter Sequence number is used by the client (IO Controller- /supervisor) to identify the outstanding service.

- **Slot No.**

  The parameter Slot number is used in the application for addressing the desired Record Data object in a specific slot (typically a module).

  Range 0...(N-1) (N = max number of slots allowed, specified in PNIO_ADV_MODE)

- **Subslot No.**

  The parameter Subslot number is used in the application for addressing the desired Record Data object in a specific subslot (typically a submodule).

  Range 1... M (M =max number of subslots allowed, specified in PNIO_ADV_MODE)

- **Index**

  The parameter Index is used in the application for addressing the desired Record Data object.

  Range: 0000h-7FFFh (Acyclic IO), AFF0h-AFFF (I&M)

- **Length**

  The parameter Length indicates the number of bytes of a Record data object that is to be read. Maximum length that can be transferred is 256 bytes (i.e. max size of a mailbox).

- **Error code 1**

  (Consult the PROFINET specification).

- **Error code 2**

  (User specific, consult the PROFINET specification).

- **Add data 1**

  The parameter Add Data 1 is API specific (profile specific). Set to zero if no Additional data 1 is defined.

- **Add data 2**

  The parameter Additional Data 2 is user specific. Set to zero if no Additional data 2 is defined.

- **Read Data**

  The parameter Read data contains the value of the object which has been read and consists of the number of bytes indicated by “Data size” in the header of the response.

9.1.13 Record Data Write (PNIO_IND_RECORD_DATA_WRITE)

Description

This message may be issued by the Anybus when a Record Data Write request has been received from the IO Controller. Upon receiving this message, the application must interpret the request and provide a response.

Note: This functionality must be enabled in PNIO_ADV_MODE ('Record Data Request'- and/or 'I&M Request'-bit).

Command initiator	Anybus
Command number	010Dh
Extended Header data	The message contains details about the issued record data request.
Message data	Write Data
Response message	-

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	010Dh	010Dh	*Record Data Write*
Data size	(data size)	0000h	*Length of* Write Data
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	AR Handle	-	
Extended word 2	-	-	
Extended word 3	Sequence No.	-	
Extended word 4	Slot No..	-	
Extended word 5	Subslot No.	-	
Extended word 6	Index	Error code 1 / Error code 2	
Extended word 7	-	Add data 1	
Extended word 8	-	Add data 2	
Message data bytes 1...n	Write Data		

- **AR Handle**

  Handle for the Application Relationship.

- **Sequence No.**

  The parameter Sequence number is used by the client (IO Controller- /supervisor) to identify the outstanding service.

- **Slot No.**

  The parameter Slot number is used in the application for addressing the desired Record Data object in a specific slot (typically a module).

  Range 0...(N-1) (N = max number of slots allowed, specified in PNIO_ADV_MODE)

- **Subslot No.**

  The parameter Subslot number is used in the application for addressing the desired Record Data object in a specific subslot (typically a submodule).

  Range 1... (M-1) (M =max number of subslots allowed, specified in PNIO_ADV_MODE)

- **Index**

  The parameter Index is used in the application for addressing the desired Record Data object.

  Range: 0000h-7FFFh (Acyclic IO), AFF0h-AFFF (I&M)

- **Write Data**

  The parameter Write data contains the value of the object which shall be written and consists of the number of bytes indicated by "Data size" in the header of the command.

- **Error code 1**

  (Consult the PROFINET specification).

- **Error code 2**

  (User specific, consult the PROFINET specification).

- **Add data 1**

  The parameter Add Data 1 is API specific (profile specific).

  (Set to zero if no Add data 1 data 1 is defined).

- **Add data 2**

  The parameter Additional Data 2 is user specific.

  (Set to zero if no Add data 2 is defined).

9.1.14 Add Channel Diagnostic Alarm (PNIO_ADD_CHANNEL_DIAG_ALARM)

Description

This command reports a Channel Diagnostic Alarm to the IO Controller. Additionally, a diagnostic entry is created in the Anybus module so that the IO Controller/Supervisor can fetch the diagnostic data by accessing the corresponding Diagnostic data record.

Note: This functionality is only available in Advanced Mode (see "Advanced Mode (PNIO_ADV_MODE)" on page 99)

Command initiator	Application
Command number	010Eh
Extended Header data	Details regarding the event, it's origin, and a unique Alarm Handle to identify it later on
Message data	-
Response message	-

Command and response layout

	Command		Expected response	
Message ID	(ID)		(ID)	
Message information	4002h		0002h	*Fieldbus Specific Message*
Command	010Eh		010Eh	*Add Channel Diag. Alarm*
Data size	0000h		0000h	*(no message data)*
Frame count	0001h		0001h	
Frame number	0001h		0001h	
Offset high	0000h		0000h	
Offset low	0000h		0000h	
Extended word 1	Alarm Handle		-	
Extended word 2	Slot No.		-	
Extended word 3	Subslot No.		-	
Extended word 4	Channel No.		-	
Extended word 5	Channel error type		-	
Extended word 6	Channel-Prop.type (high byte)	Channel-Prop.dir (low byte)	-	
Extended word 7	-		Extended Fault Information	
Extended word 8	-		Fault Information	

- **Alarm Handle**

  Unique handle for the alarm, specified by the application. This handle is used on all further references to that particular alarm event.

  See also "Remove Diagnostic Alarm (PNIO_REMOVE_DIAG_ALARM)" on page 120.

- **Slot No.**

  This parameter is used in the application for addressing the desired diagnostic object in a specific slot (typically a module).

  Range 0...(N-1) (N = max number of slots allowed, specified in PNIO_ADV_MODE)

- **Subslot No.**

  This parameter is used in the application for addressing the desired diagnostic object in a specific subslot (typically a submodule).

  Range 1... M (M =max number of subslots allowed, specified in PNIO_ADV_MODE)

- **Channel No.**

  The channel number for which the diagnostic data applies.

  Range: 0-8000h (0-7FFFh Manufacturer specific, 8000h refers to the Submodule itself).

- **Channel error type**

  The type of error that has occurred.

Value	Description
0001h	Short circuit
0002h	Under voltage
0003h	Over voltage
0004h	Overload
0005h	Over temperature
0006h	Line break
0007h	Upper limit value exceeded
0008h	Lower limit value exceeded
0009h	Error
000A...FFFFh	(Consult the PROFINET IO specification)

- **ChannelProp.type (high byte)**

  Indicate channel size to which the channel diagnosis object is related.

Value	Description
00h	Shall be used if 'Channel No' equals 8000h (submodule), or if none of the types below are appropriate
01h	1 Bit.
02h	2 Bit.
03h	4 Bit.
04h	8 Bit.
05h	16 Bit.
06h	32 Bit.
07h	64 Bit.
08h-FFh	(reserved)

- **ChannelProp.dir (low byte)**

  Specifies the direction of the channel.

Value	Description
00h	Manufacturer specific
01h	Input
02h	Output
03h	Input/Output
Other	(reserved)

- **Fault Information**

  (see "Fault Information" on page 58)

- **Extended Fault Information**
  (see “Extended Fault Information” on page 60)

9.1.15 Remove Diagnostic Alarm (PNIO_REMOVE_DIAG_ALARM)

Description

When a diagnostic alarm event has been resolved, this command is used to remove its diagnostic entry and report this to the IO Controller.

Note: This functionality is only available in Advanced Mode (see "Advanced Mode (PNIO_ADV_MODE)" on page 99)

Command initiator	Application
Command number	010Fh
Extended Header data	Alarm Handle for the diagnostic alarm to remove
Message data	-
Response message	-

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	010Fh	010Fh	*Remove Diag. Alarm*
Data size	0000h	0000h	*(no message data)*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	Alarm Handle	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	Extended Fault Information	
Extended word 8	-	Fault Information	

- **Alarm Handle**

  Handle of the alarm event; must be set to match the handle specified when the alarm was issued.

  See also...

  - "Add Channel Diagnostic Alarm (PNIO_ADD_CHANNEL_DIAG_ALARM)" on page 117
  - "Add Generic Diagnostic Alarm (ADD_GENERIC_DIAG_ALARM)" on page 121

- **Fault Information**

  (see "Fault Information" on page 58)

- **Extended Fault Information**

  (see "Extended Fault Information" on page 60)

9.1.16 Add Generic Diagnostic Alarm (ADD_GENERIC_DIAG_ALARM)

Description

This command reports a vendor specific diagnostic alarm to the IO Controller. In addition, a diagnostic entry is created in the Anybus so that the IO Controller/Supervisor can fetch the diagnostic data by accessing the corresponding Diagnostic data record.

Note: This functionality is only available in Advanced Mode (see "Advanced Mode (PNIO_ADV_MODE)" on page 99)

Command initiator	Application
Command number	0110h
Extended Header data	Details regarding the event, it's origin, and a unique Alarm Handle to identify it later on
Message data	Diagnostic Data (Vendor specific)
Response message	-

Command and response layout

	Command		Expected response	
Message ID	(ID)		(ID)	
Message information	4002h		0002h	*Fieldbus Specific Message*
Command	0110h		0110h	*Add Generic Diagnostic Alarm*
Data size	(data size)		0000h	*(length of diagnostic data)*
Frame count	0001h		0001h	
Frame number	0001h		0001h	
Offset high	0000h		0000h	
Offset low	0000h		0000h	
Extended word 1	Alarm Handle		-	
Extended word 2	Slot No.		-	
Extended word 3	Subslot No.		-	
Extended word 4	Channel No.		-	
Extended word 5	User structure identifier		-	
Extended word 6	Channel-Prop.type (high byte)	Channel-Prop.dir (low byte)	-	
Extended word 7	-		Extended Fault Information	
Extended word 8	-		Fault Information	
Message Data Words 1...n	Diagnostic Data			

- **Alarm Handle**

  Unique handle for the alarm, specified by the applications. This handle is used on all further references to that particular alarm event. See also "Remove Diagnostic Alarm (PNIO_REMOVE_DIAG_ALARM)" on page 120.

- **Slot No.**

  This parameter is used in the application for addressing the desired diagnostic object in a specific slot (typically a module).

  Range 0... (N-1) (N = max number of slots allowed, specified in PNIO_ADV_MODE)

- **Subslot No.**

  This parameter is used in the application for addressing the desired diagnostic object in a specific subslot (typically a submodule).

  Range 1... M (M =max number of subslots allowed, specified in PNIO_ADV_MODE)

- **Channel No.**

  Channel number for which the diagnostic data applies.

  Range: 0-8000h (0-7FFFh Manufacturer specific, 8000h refers to the Submodule itself).

- **User structure identifier**

  Is used to identify the structure of the diagnostic data (Message data word 1-x).

Value	Description
0000h-7FFFh	Manufacturer specific diagnostics
8000h	Channel diagnostics
8001h	Manufacturer specific multiple
9000h-9FFFh	(reserved for profiles)
Other	(reserved)

- **ChannelProp.type (high byte)**

  Indicate channel size to which the channel diagnosis object is related.

Value	Description
00h	Shall be used if 'Channel No' equals 8000h (submodule), or if none of the types below are appropriate
01h	1 Bit.
02h	2 Bit.
03h	4 Bit.
04h	8 Bit.
05h	16 Bit.
06h	32 Bit.
07h	64 Bit.
08h-FFh	(reserved)

- **ChannelProp.dir (low byte)**

  Specifies the direction of the channel.

Value	Description
00h	Manufacturer specific
01h	Input
02h	Output
03h	Input/Output
Other	(reserved)

- **Fault Information**

  (see "Fault Information" on page 58)

- **Extended Fault Information**

  (see "Extended Fault Information" on page 60)

- **Diagnostic Data**

  Contains the generic diagnostic data that shall be sent. Range: 1 – 256 bytes.

9.1.17 Add Process Alarm (PNIO_ADD_PROCESS_ALARM)

Description

This command issues a Process Alarm to the IO Controller. Since no diagnostic entry is created for this type of alarm, no corresponding "removal" command is needed.

Note: This functionality is only available in Advanced Mode (see "Advanced Mode (PNIO_ADV_MODE)" on page 99)

Command initiator	Application
Command number	0112h
Extended Header data	Details regarding the alarm, it's origin, and a unique Alarm Handle to identify it later on
Message data	-
Response message	-

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0112h	0112h	*Add Process Alarm*
Data size	(data size)	0000h	*(Length of Alarm Data)*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	Alarm Handle	-	
Extended word 2	Slot No.	-	
Extended word 3	Subslot No.	-	
Extended word 4	User structure identifier	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	Extended Fault Information	
Extended word 8	-	Fault Information	
Message Data word 1...n	Alarm Data		

- **Alarm Handle**

  Unique handle for the alarm, specified by the application. This handle is used on all further references to that particular alarm event.

- **Slot No.**

  The parameter Slot number is used in the application for addressing the desired diagnostic object in a specific slot (typically a module).

  Range 0...(N-1) (N = max number of slots allowed, specified in PNIO_ADV_MODE)

- **Subslot No.**

  The parameter Subslot number is used in the application for addressing the desired diagnostic object in a specific subslot (typically a submodule).

  Range 1... M (M =max number of subslots allowed, specified in PNIO_ADV_MODE)

- **User structure identifier**

  Is used to identify the structure of the process alarm data (Message data word 1-x).

Value	Description
0000h-7FFFh	Manufacturer specific diagnostics
Other	Reserved

- **Alarm Data**

  Contains the process alarm data that shall be sent.

- **Fault Information**

  (see "Fault Information" on page 58)

- **Extended Fault Information**

  (see "Extended Fault Information" on page 60)

9.1.18 Abort AR (PNIO_AR_ABORT)

Description

This command aborts a previously established AR.

Command initiator	Application
Command number	011Ah
Extended Header data	AR Handle
Message data	-
Response message	-

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	011Ah	011Ah	*PNIO_AR_ABORT*
Data size	0000h	0000h	*(No message data)*
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	AR Handle	AR Handle	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault Information	

- **AR Handle**
  Handle for the Application Relationship.
- **Fault Information**
  (see "Fault Information" on page 58)

9.1.19 Set port MAC address (PNIO_SET_PORT_MAC_ADDR)

Description

This command aborts a previously established AR.

Command initiator	Application
Command number	0123h
Extended Header data	Port number
Message data	Port MAC address to be used
Response message	The response indicates if the command was accepted

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0123h	0123h	*PNIO_SET_PORT_MAC_ADDR*
Data size	0006h	0006h	
Frame count	0001h	0001h	*One frame*
Frame number	0001h	0001h	*This is frame 1*
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	PortNr	PortNr	
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	Port MAC address byte 1, 2	Port MAC address byte 1, 2	
Message data word 2	Port MAC address byte 3, 4	Port MAC address byte 3, 4	
Message data word 3	Port MAC address byte 5, 6	Port MAC address byte 5, 6	

- **PortNr**

  The parameter Port number addresses which port that should attain the MAC ID.

  Range: 1 or 2

- **Fault information**

  Invalid port number (see "Fault Information" on page 58)

9.1.20 Set System Description (PNIO_SET_SYSTEM_DESCRIPTION)

Description

This command changes the SNMP parameter "System Description". Sending a mailbox with only a NULL character erases the system description.

Note: This command may only be issued during initialization.

Command initiator	Application
Command number	0120h
Extended Header data	No extended header data is associated with this command
Message data	System description
Response message	The response indicates if the command was accepted

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0120h	0120h	*PNIO_SET_SYSTEM_DESCRIPTION*
Data size	(data size)	(data size)	*(Length of System Description, including NULL termination)*
Frame count	0001h	0001h	*One frame*
Frame number	0001h	0001h	*This is frame 1*
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-x	System Description	System Description	

- **System Description**

  The parameter System Description holds an ASCII string with up to 256 bytes including NULL termination.

- **Fault information** (see "Fault Information" on page 58)
  - Failed to store the parameter into the flash memory
  - Mailbox not allowed in current state

9.1.21 Set Interface Description (PNIO_SET_INTERFACE_DESCRIPTION)

Description

This command changes the SNMP parameter "Interface Description". Sending a mailbox with only a NULL character erases the interface description.

Note: This command may only be issued during initialization.

Command initiator	Application
Command number	0121h
Extended Header data	No extended header data is associated with this command
Message data	Interface description
Response message	The response indicates if the command was accepted

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0121h	0121h	*PNIO_SET_INTERFACE_DESCRIPTION*
Data size	(data size)	(data size)	*(Length of Interface Description, including NULL termination)*
Frame count	0001h	0001h	*One frame*
Frame number	0001h	0001h	*This is frame 1*
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-x	Interface Description	Interface Description	

- **Interface Description**

  The parameter Interface Description holds an ASCII string with up to 256 bytes including NULL termination.

- **Fault information** (see "Fault Information" on page 58)
  - Failed to store the parameter into the flash memory
  - Mailbox not allowed in current state

9.1.22 Set System Description MIB-II (PNIO_SET_SYSTEM_DESCRIPTION_MIB2)

Description

This command changes the MIB-II parameter "System Description". Sending a mailbox with only a NULL character erases the system description.

Note: This command may only be issued during initialization.

Command initiator	Application
Command number	0124h
Extended Header data	No extended header data is associated with this command
Message data	System description
Response message	The response indicates if the command was accepted

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0124h	0124h	*PNIO_SET_SYSTEM_DESCRIPTION _MIB2*
Data size	(data size)	(data size)	*(Length of System Description, including NULL termination)*
Frame count	0001h	0001h	*One frame*
Frame number	0001h	0001h	*This is frame 1*
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-x	System Description	System Description	

- **System Description**

  The parameter System Description holds an ASCII string with up to 256 bytes including NULL termination.

- **Fault information** (see "Fault Information" on page 58)
  - Failed to store the parameter into the flash memory
  - Mailbox not allowed in current state

9.1.23 Set System Contact MIB-II (PNIO_SET_SYSTEM_CONTACT_MIB2)

Description

This command changes the MIB-II parameter "System Contact". Sending a mailbox with only a NULL character erases the system contact.

Note: This command may only be issued during initialization.

Command initiator	Application
Command number	0125h
Extended Header data	No extended header data is associated with this command
Message data	System contact
Response message	The response indicates if the command was accepted

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0125h	0125h	*PNIO_SET_SYSTEM_CONTACT_MIB2*
Data size	(data size)	(data size)	*(Length of System Description, including NULL termination)*
Frame count	0001h	0001h	*One frame*
Frame number	0001h	0001h	*This is frame 1*
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-x	System Contact	System Contact	

- **System Contact**

  The parameter System Contact holds an ASCII string with up to 256 bytes including NULL termination.

- **Fault information** (see "Fault Information" on page 58)
  - Failed to store the parameter into the flash memory
  - Mailbox not allowed in current state

9.1.24 Set System Location MIB-II (PNIO_SET_SYSTEM_LOCATION_MIB2)

Description

This command changes the MIB-II parameter "System Location". Sending a mailbox with only a NULL character erases the system contact.

Note: This command may only be issued during initialization.

Command initiator	Application
Command number	0126h
Extended Header data	No extended header data is associated with this command
Message data	System location
Response message	The response indicates if the command was accepted

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0126h	0126h	*PNIO_SET_SYSTEM_LOCATION _MIB2*
Data size	(data size)	(data size)	*(Length of System Location, including NULL termination)*
Frame count	0001h	0001h	*One frame*
Frame number	0001h	0001h	*This is frame 1*
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-x	System Location	System Location	

- **System Location**

  The parameter System Location holds an ASCII string with up to 256 bytes including NULL termination.

- **Fault information** (see "Fault Information" on page 58)
  - Failed to store the parameter into the flash memory
  - Mailbox not allowed in current state

9.1.25 Get IM0 Information (PNIO_GET_IM0_INFO)

Description

This command reads the information in the I&M (IM0) structure.

Please note that this command can be issued at any time.

Command initiator	Application
Command number	0127h
Extended Header data	No extended header data is associated with this command
Message data	-
Response message	The response indicates if the command was accepted Manufacturer ID, Order ID, Serial Number, Hardware Revision, Software Revision, Revision Counter, Profile ID, Profile-Specific Type, IM Supported

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0127h	0127h	*PNIO_GET_IM0_INFO*
Data size	0000h	0034h	*52 bytes of data in response*
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 bytes 1... 2		Manufacturer ID	
Message Data bytes 3... 22		Order ID [1... 20]	
Message Data bytes 23... 38		Serial Number [1... 16]	
Message Data bytes 39... 40		Hardware Revision	
Message Data bytes 41... 44		Software Revision [1... 4]	
Message Data bytes 45... 46		Revision Counter	
Message Data bytes 47... 48		Profile ID	
Message Data bytes 49... 50		Profile-Specific Type	
Message Data bytes 51... 52		IM Supported	

- **Fault Information**

  (see "Fault Information" on page 58)

- **Manufacturer ID**

  The Manufacturer ID (a.k.a. the 'Vendor ID') is administered by the PNO.

- **Serial Number**

  Serial number.

- **Order ID**

  Order ID of the product. If the Order ID is shorter than 20 characters, the unused characters shall be filled with blanks (i.e. 20h).

- **Hardware Revision**

  Consult the I&M specification for details.

- **Software Revision**

  Consult the I&M specification for details.

- **Revision Counter**

  Consult the I&M specification for details.

- **Profile ID**

  Consult the I&M specification for details.

- **Profile-Specific Type**

  The Profile-Specific Type of the product.

  Consult the I&M specification.

- **IM Supported**

  Shows which I&M records are enabled/disabled.

  Consult the I&M specification for details.

9.1.26 Set IM0 Information (PNIO_SET_IM0_INFO)

Description

This command alters the information in the I&M (IM0) structure.

Please note that this command can be issued at any time to update the I&M information. To ensure that the I&M information is up to date when starting up the network communication, it must be issued once during initialization.

Command initiator	Application
Command number	0115h
Extended Header data	Fault Information
Message data	Manufacturer ID, Order ID, Serial Number, Hardware Revision, Software Revision, Revision Counter, Profile ID, Profile-Specific Type, IM Supported
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	0115h	0115h	*PNIO_SET_IM0_INFO*
Data size	0034h	0034h	*52 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	
Message Data bytes 1... 2	Manufacturer ID	Manufacturer ID	
Message Data bytes 3... 22	Order ID [1... 20]	Order ID [1... 20]	
Message Data bytes 23... 38	Serial Number [1... 16]	Serial Number [1... 16]	
Message Data bytes 39... 40	Hardware Revision	Hardware Revision	
Message Data bytes 41... 44	Software Revision [1... 4]	Software Revision [1... 4]	
Message Data bytes 45... 46	Revision Counter	Revision Counter	
Message Data bytes 47... 48	Profile ID	Profile ID	
Message Data bytes 49... 50	Profile-Specific Type	Profile-Specific Type	
Message Data bytes 51... 52	IM Supported	IM Supported	

- **Fault Information**

  (see "Fault Information" on page 58)

- **Manufacturer ID**

  The Manufacturer ID (a.k.a. the 'Vendor ID') is administered by the PNO.

  Default: 010Ch (HMS)

- **Serial Number**

  Serial number.

  Default: (set at production)

  If the Serial Number is set to all zeroes, the default Serial Number of the Anybus module will be used.

- **Order ID**

  Order ID of the product. If the Order ID is shorter than 20 characters, the unused characters shall be filled with blanks (i.e. 20h).

  Default: "ABS-PIR"

  If the Order ID is set to all zeroes, the default Order ID of the Anybus module will be used.

- **Hardware Revision**

  Consult the I&M specification for details.

  Default: (Anybus revision, assigned during manufacturing)

- **Software Revision**

  Consult the I&M specification for details.

  Default: "V", XXh, YYh, ZZh (revision of the Anybus module).

  If the software revision is set to all zeroes, the default software revision of the Anybus module will be used.

- **Revision Counter**

  Consult the I&M specification for details.

  Default: 0000h

- **Profile ID**

  Consult the I&M specification for details.

  Default: F600h (Generic Device)

- **Profile-Specific Type**

  The Profile-Specific Type must be set to a valid value. Consult the I&M specification for details or use default (0004h).

- **IM Supported**

  By setting/clearing bits in this word it's possible to enable/disable support for specific I&M records. Please note that only I&M # 1...4 can be enabled/disabled. I&M #0 is always enabled.

  Consult the I&M specification for details.

  Default: 001Eh (support I&M 0-4)

  Note: As of version 2.3 of the PROFINET specification, it is mandatory to support I&M 0-4. Therefore, it is not recommended to change the default value of 001Eh.

9.1.27 Alarm Acknowledge Received (PNIO_IND_ALARM_ACK_RECEIVED)

Description

This message is issued by the Anybus to inform the application that the IO Controller has acknowledged a previously reported alarm.

Note: This functionality must be enabled in PNIO_ADV_MODE ('Alarm Acknowledge'-bit)

Command initiator	Anybus
Command number	0113h
Extended Header data	Alarm Handle of the acknowledge event.
Message data	-
Response message	The application must not respond to this message.

Command and response layout

	Message	
Message ID	(ID)	
Message information	4002h	*Fieldbus Specific Message*
Command	0113h	*Alarm Acknowledge Received*
Data size	0000h	
Frame count	0001h	
Frame number	0001h	
Offset high	0000h	
Offset low	0000h	
Extended word 1	Alarm Handle	
Extended word 2	-	
Extended word 3	-	
Extended word 4	-	
Extended word 5	-	
Extended word 6	-	
Extended word 7	-	
Extended word 8	-	

- **Alarm Handle**

  Handle of the alarm event; specified by the application when the alarm was issued.

9.1.28 Configuration Mismatch (PNIO_IND_CFG_MISMATCH)

Description

This message is issued by the Anybus to inform the application that the configuration in the IO Controller does not match the application configuration (i.e. the configuration created during initialisation). The application must decide whether to adopt or reject the IO Controller configuration.

Please note that the Record Data Lengths specified when the sub-module was plugged is not updated if a module/submodule is substituted.

Note: This functionality must be enabled in PNIO_ADV_MODE ('Cfg Mismatch'-bit)

Command initiator	Anybus
Command number	0114h
Extended Header data	Slot no, Subslot no, Desired Mod. Identifier, Desired Sub. Identifier, Actual Module Identifier, Actual Sub. Identifier, Module State, Submodule State, Perfect Adaptation, AR Handle
Message data	-
Response message	-

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	*Fieldbus Specific Message*
Command	0114h	0114h	*Configuration Mismatch*
Data size	0000h	0000h	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	AR Handle	-	
Extended word 2	Slot No.	Module state	
Extended word 3	Subslot No.	Submodule state	
Extended word 4	Desired Mod. identifier (msb)	Actual Module identifier(msb)	
Extended word 5	Desired Mod. identifier (lsb)	Actual Module identifier(lsb)	
Extended word 6	Desired Sub. identifier (msb)	Actual Sub. identifier (msb)	
Extended word 7	Desired Sub. identifier (lsb)	Actual Sub. identifier (lsb)	
Extended word 8	-	Perfect Adaptation	

- **AR Handle**

  Handle for the Application Relationship.

- **Slot No.**

  The parameter Slot number is used in the application for addressing the desired diagnostic object in a specific slot (typically a module).

- **Subslot No.**

  The parameter Subslot number is used in the application for addressing the desired diagnostic object in a specific subslot (typically a submodule).

- **Desired Mod. identifier**

  Module identifier (derived from the IO Controller configuration) as stated in GSD file.

- **Desired Sub. identifier**

  Submodule identifier (derived from the IO Controller configuration) as stated in GSD file.

- **Actual Module identifier**

  Module identifier as stated in GSD file.

- **Actual Sub. identifier**

  Submodule identifier as stated in GSD file.

- **Module state**

  Here the application must decide how the mismatching slot shall be handled.

Value	State	Description
0	NO_MODULE	There is no module in the specified slot.
1	WRONG_MODULE	Wrong module in the specified slot. Specify correct module in 'Actual Module identifier' **Note:** This value should be avoided. Instead, set Module state to SUBSTITUTED_MODULE and set Submodule state for each submodule to WRONG_SUBMODULE.
2	PROPER_MODULE	Correct module specified in slot
3	SUBSTITUTED_MODULE	A replacement module is used in the slot. Specify module in 'Actual Module identifier'
Other	-	(reserved)

- **Submodule state**

  Here the application must decide how the mismatching subslot shall be handled.

Value	State	Description
0	NO_SUBMODULE	There is no module in the specified slot.
1	WRONG_SUBMODULE	Wrong module in the specified slot. Specify correct submodule in 'Actual Sub. identifier'
7	SUBSTITUTED_SUBMODULE	A replacement module is used in the slot. Specify submodule in 'Actual Sub. identifier'
FFFFh	PROPER_SUBMODULE	Correct submodule specified in slot
Other	-	(reserved)

- **Perfect Adaptation**

Value	Description
0	Use data specified in extended words 2...7 of the response
1	Make a perfect adaptation of the module; the module will automatically plug the current module and insert the module/submodule specified by the IO Controller. Extended words 2... 7 are not used.
other	undefined

9.1.29 Check AR Indication (PNIO_IND_CHECK_AR)

Description

This message is issued by the Anybus module each time a new AR has been established.

Note: This functionality must be enabled in PNIO_ADV_MODE ('AR Information'-bit)

Command initiator	Anybus
Command number	0116h
Extended Header data	AR Type, AR Properties, AR Handle
Message data	-
Response message	The application must not respond to this message.

Command and response layout

	Command	
Message ID	(ID)	
Message information	4002h	*Fieldbus Specific Message*
Command	0116h	*Check AR Indication*
Data size	0000h	
Frame count	0001h	
Frame number	0001h	
Offset high	0000h	
Offset low	0000h	
Extended word 1	AR Type	
Extended word 2	AR Properties (high)	
Extended word 3	AR Properties (low)	
Extended word 4	AR Handle	
Extended word 5	-	
Extended word 6	-	
Extended word 7	-	
Extended word 8	-	

- **AR Type**

  Value indicates the type of Application Relationship (AR) as follows:

Value	Type
0001h	CM_AR_TYPE_SINGLE
0003h	CM_AR_TYPE_CIR
0004h	CM_AR_TYPE_IOC_REDUNDANT
0005h	CM_AR_TYPE_IOD_REDUNDANT
0006h	CM_AR_TYPE_SUPERVISOR

- **AR Properties**

  Bitfield indicating the properties of the AR as follows:

Bits	Contents	Interpretation
0... 2	State	0: CM_AR_PROP_STATE_BACKUP 1: CM_AR_PROP_STATE_PRIMARY
3	Supervisor takeover allowed	0: CM_AR_SUPERVISOR_TAKEOVER_NOT_ALLOWED 1: CM_AR_SUPERVISOR_TAKEOVER_ALLOWED
4	Parameterization server	0: CM_AR_PROP_PRM_SERVER_EXTERNAL 1: CM_AR_PROP_PRM_SERVER_CMI
5... 6	Data rate	0: CM_AR_PROP_DATARATE_100MBPS_OR_MORE 1: CM_AR_PROP_DATARATE_100MBPS 2: CM_AR_PROP_DATARATE_1GBPS 3: CM_AR_DATARATE_10GBPS
7	-	-
8	Device access	0: CM_AR_PROP_DEVICE_ACCESS_NO 1: CM_AR_PROP_DEVICE_ACCESS_YES
9... 10	Companion AR	0: CM_AR_PROP_COMPANION_AR_SINGLE 1: CM_AR_PROP_COMPANION_AR_FIRST 2: CM_AR_PROP_COMPANION_AR_COMPANION
11... 31	-	-

- **AR Handle**

  Handle for the Application Relationship.

9.1.30 AR Info Indication (PNIO_IND_AR_INFO)

Description

This command reports the current IO configuration, allowing the application to see exactly how the data is mapped in the DPRAM. Note that this information may be fragmented, which means that this command may be issued multiple times by the module, each time containing different parts of the configuration.

Note: This functionality must be enabled in PNIO_ADV_MODE ('AR Information'-bit)

Command initiator	Anybus
Command number	0117h
Extended Header data	AR Handle, Current Fragment, Total Fragments
Message data	-
Response message	The application must not respond to this message.

Command and response layout

	Command	
Message ID	(ID)	
Message information	4002h	*Fieldbus Specific Message*
Command	0117h	*AR Info Indication*
Data size	(data size)	
Frame count	0001h	
Frame number	0001h	
Offset high	0000h	
Offset low	0000h	
Extended word 1	Current Fragment	
Extended word 2	Total Fragments	
Extended word 3	AR Handle	
Extended word 4	-	
Extended word 5	-	
Extended word 6	-	
Extended word 7	-	
Extended word 8	-	
	Data	
Message Data		

- **Current Fragment**

  Current fragment number; when this word equals the value of 'Total Fragments' (see below), all data has been received.

- **Total Fragments**

  The number of fragments needed for the data.

- **AR Handle**

  Handle for the Application Relationship.

- **Data**

  As mentioned previously, the data is fragmented. The Anybus will issue this mailbox message repeatedly, each time with a new block of data, until all configuration details has been transferred to the application. The application should wait until all data has been transferred before interpreting it.

  The data consists of 3 types of information blocks:

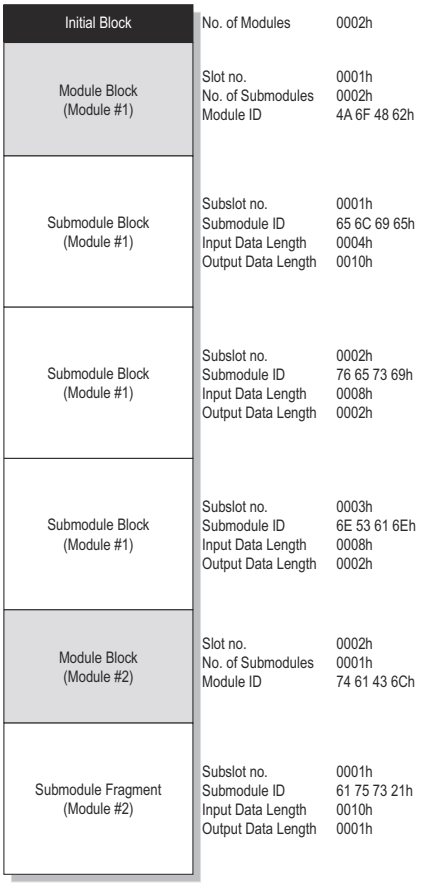
Block Type	Size	Contents		Comments
Initial Block	2 bytes	No. of Modules	(1 word)	Total number of modules
Module Block	8 bytes	Slot no.	(1 word)	Module slot number
		No. of Sub-modules	(1 word)	No. of sub-modules in the module
		Module ID	(2 words)	Module ID
Submodule Block	10 bytes	Subslot no.	(1 word)	Submodule subslot number
		Submodule ID	(2 words)	Submodule ID
		Input Data Length	(1 word)	I/O data sizes associated with the module in bytes
		Output Data Length	(1 word)	

  The first 2 bytes in the data indicates the number of modules in the configuration. Each module is then represented by a Module Block, followed by a number of Submodule Blocks (provided that the module in question contains submodules).

Example

In this example, the configuration contains two modules with the following properties:

- Module #1 contains three submodules
- Module #2 contains one submodule



9.1.31 Abort AR Indication (PNIO_IND_AR_ABORT)

Description

This message signals that an Application Relationship (AR) has been aborted.

Note: This functionality must be enabled in PNIO_ADV_MODE ('AR Information'-bit)

Command initiator	Anybus
Command number	0118h
Extended Header data	AR Handle, Reason Code
Message data	-
Response message	The application must not respond to this message.

Command and response layout

	Command	
Message ID	(ID)	
Message information	4002h	*Fieldbus Specific Message*
Command	0118h	*Abort AR Indication*
Data size	0000h	
Frame count	0001h	
Frame number	0001h	
Offset high	0000h	
Offset low	0000h	
Extended word 1	AR Handle	
Extended word 2	Reason Code	
Extended word 3	-	
Extended word 4	-	
Extended word 5	-	
Extended word 6	-	
Extended word 7	-	
Extended word 8	-	

- **AR Handle**

  Handle for the Application Relationship.

- **Reason Code**

  Reason for aborting the connection.

Value	Reason
3	Out of mem
4	Add provider or consumer failed
5	Miss (consumer)
6	Cmi timeout
7	Alarm-open failed
8	Alarm-send.cnf(-)
9	Alarm-ack-send.cnf(-)
10	Alarm-data too long
11	Alarm.ind(err)
12	Rpc-client call.cnf(-)
13	Ar-abort.req
14	Re-run aborts existing
15	Got release.ind
16	Device passivated
17	Device / AR removed
18	Protocol violation
19	NARE error
20	RPC-Bind error
21	RPC-Connect error
22	RPC-Read error
23	RPC-Write error
24	RPC-Control error
25	Forbidden pull or plug after check.rsp and before in-data.ind
26	AP removed
27	Link “down”
28	Could not register multicast-mac
29	Not synchronized (cannot start companion-ar)
30	Wrong topology (cannot start companion-ar)
31	Dcp, station-name changed
32	Dcp, reset to factory-settings

9.1.32 AR Offline Indication (PNIO_IND_AR_OFFLINE)

Description

This message signals that an Application Relationship (AR) has gone off-line.

Note: This functionality must be enabled in PNIO_ADV_MODE ('AR Information'-bit)

Command initiator	Anybus
Command number	0119h
Extended Header data	AR Handle, Reason Code
Message data	-
Response message	The application must not respond to this message.

Command and response layout

	Command	
Message ID	(ID)	
Message information	4002h	*Fieldbus Specific Message*
Command	0119h	*AR Offline Indication*
Data size	0000h	
Frame count	0001h	
Frame number	0001h	
Offset high	0000h	
Offset low	0000h	
Extended word 1	AR Handle	
Extended word 2	Reason Code	
Extended word 3	-	
Extended word 4	-	
Extended word 5	-	
Extended word 6	-	
Extended word 7	-	
Extended word 8	-	

- **AR Handle**
  Handle for the Application Relationship.
- **Reason Code**
  (see "Reason Code" on page 144)

9.2 PROFIenergy Specific Commands

Commands in this category:

Mailbox Command	Page
Enable PROFIenergy (PNIO_ENABLE_PROFIENERGY)	147
Get PROFIenergy mode (PNIO_IND_PE_GET_ENERGY_MODE)	149
Start PROFIenergy Pause (PNIO_IND_PE_START_PAUSE)	151
End PROFIenergy Pause (PNIO_IND_PE_END_PAUSE)	152
Get PROFIenergy status (PNIO_IND_PE_GET_STATUS)	153

9.2.1 Enable PROFIenergy (PNIO_ENABLE_PROFIENERGY)

Description

This command enables the PROFIenergy support.

Command initiator	Application
Command number	0130h
Extended Header data	See below
Message data	-
Response message	The response indicates if the command was successful

Command and response layout

	Command	Expected response	
Message ID	0001h	0001h	*Any integer value*
Message information	4002h	0002h	
Command	0130h	0130h	
Data size	0000h	0000h	
Frame count	0001h	0001h	*One frame*
Frame number	0001h	0001h	*This is frame one*
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	EnergyModes	EnergyModes	
Extended word 2	PEFunctionality	PEFunctionality	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault Information	

Command

- **Extended word1 (EnergyModes)**

  This parameter specifies the number of energy saving modes available. It always includes the "Ready to operate" and "Power off" modes. Thus the minimum value is 3, since at least one vendor specific energy mode should always be available.

- **Extended word2 (PEFunctionality)**

  This parameter specifies the PROFIenergy functionality enabled. Setting a bit to zero (0) will disable the functionality. Unused bits should be set to zero (0).

  Bit 0 = 1: PE function "Query_Modes" enabled.

  Bit 1 = 1: PE function "PEM_Status" enabled.

  Bit 2 = 1: PE function "PE_Identity" enabled.

Response

- **Extended word8 (Fault information)**
  - Current states does not allow mailbox. Not allowed after initialization.
  - EnergyModes out of range.
  - PEFunctionality out of range.

  See "Fault Information" on page 58.

9.2.2 Get PROFIenergy mode (PNIO_IND_PE_GET_ENERGY_MODE)

Description

This command tells the Anybus-S to query the host application for the PROFIenergy mode. Please note that the modes "Ready to operate" (ReqEnergyMode = 255) and "Power off" (ReqEnergyMode = 0) must be supported.

Command initiator	Anybus-S
Command number	0131h
Extended Header data	See below
Message data	The response mailbox contains the data for the requested energy mode
Response message	The response indicates if the command was accepted

Command and response layout

	Command	Expected response	
Message ID	0001h	0001h	*Any integer value*
Message information	4002h	0002h	
Command	0131h	0131h	
Data size	0000h	0022h	
Frame count	0001h	0001h	*One frame*
Frame number	0001h	0001h	*This is frame one*
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	EnergyMode	-	
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	-	Mode Attributes	
Message data word 2-3	-	TimeMinPause	
Message data word 4-5	-	TimeToPause	
Message data word 6-7	-	TimeToOperate	
Message data word 8-9	-	TimeMinLengthOfStay	
Message data word 10-11	-	TimeMaxLengthOfStay	
Message data word 12-13	-	ModePowerConsumption	
Message data word 14-15	-	EnergyConsumption ToPause	
Message data word 16-17	-	EnergyConsumption ToOperate	

Command

- **Extended word1 (EnergyMode)**

  The parameter specifies the number of the energy saving mode for which the Anybus-S would like to read data. It always includes the "Ready to operate" and "Power off" modes, and thus the minimum value is 2.

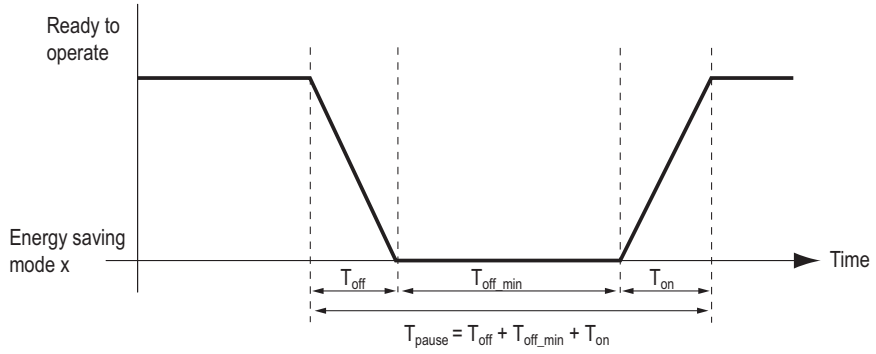
Response

- **Extended word8 (Fault information)**
  EnergyMode not supported. See "Fault Information" on page 58.

- **Message data**

Message Data Word	Name	Description	Comment
Word 1	Mode attributes	Bit field.	0
	Bit 0:	0 - Only static time and energy values are available.	
		1 - Dynamic time and energy values are available.	
	Bit 1 - 15:	Reserved.	
Word 2 - 3	TimeMinPause	Minimum pause in milliseconds (t_{pause}).	If the value is undefined, the maximum value FFFFFFFFh shall be used. If the value is zero (0), use the value 00000000h.
Word 4 - 5	TimeToPause	Expected time to go to this energy saving mode, in milliseconds (t_{off}).	
Word 6 - 7	TimeToOperate	Time needed to go to the "Ready to operate" mode, in milliseconds (t_{on}).	
Word 8 - 9	TimeMinLengthOfStay	The minimum time for which the device must stay in this mode, in milliseconds (t_{off_min}).	
Word 10 - 11	TimeMaxLengthOfStay	The maximum time allowed to stay in this mode, in milliseconds. If no maximum value is available, the value FFFFFFFFh shall be used.	
Word 12 - 13	ModePower Consumption	Amount of power consumed in this mode, in kWh. (**Note**: this parameter is interpreted as a four byte float value).	If the value is undefined, the value 0.0 shall be used.
Word 14 - 15	EnergyConsumption ToPause	Amount of energy to go to this mode, in kWh. (**Note**: this parameter is interpreted as a four byte float value).	
Word 16 - 17	EnergyConsumption ToOperate	Amount of energy required to go to this mode, in kWh. (**Note**: this parameter is interpreted as a four byte float value).	

The image below illustrates a timing diagram for when a device is switched from "Ready to operate" to an energy saving mode, and back again.



9.2.3 Start PROFIenergy Pause (PNIO_IND_PE_START_PAUSE)

Description

This command is used to make the Anybus-S tell the host application to switch to the most suitable energy saving mode, given the time of the standstill.

Command initiator	Anybus-S
Command number	0132h
Extended Header data	See below
Message data	No message data is associated with this mailbox
Response message	The response message contains the destination energy mode selected by the host given the time for the standstill (PauseTime)

Command and response layout

	Command	Expected response	
Message ID	0001h	0001h	*Any integer value*
Message information	4002h	0002h	
Command	0132h	0132h	
Data size	0000h	0000h	
Frame count	0001h	0001h	*One frame*
Frame number	0001h	0001h	*This is frame one*
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	PauseTime	DestinationEnergyMode	
Extended word 2		-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault Information	

Command

- **PauseTime**

  Pause time in milliseconds. Range: 00000000h - FFFFFFFFh.

Response

Extended Word	Name	Description	Comment
Word 1	DestinationEnergyMode	The energy mode which the application has selected.	Value range: 0 - 255
Word 8	Fault information	No suitable energy mode available. See “Fault Information” on page 58.	-

9.2.4 End PROFIenergy Pause (PNIO_IND_PE_END_PAUSE)

Description

This command is used to make the Anybus-S tell the host application to switch back to "Ready to operate" mode.

Command initiator	Anybus-S
Command number	0133h
Extended Header data	See below
Message data	There is no message data associated with this mailbox
Response message	The response message contains the time it will take until the host application has shifted to "Ready to operate" mode

Command and response layout

	Command	Expected response	
Message ID	0001h	0001h	*Any integer value*
Message information	4002h	0002h	
Command	0133h	0133h	
Data size	0000h	0022h	
Frame count	0001h	0001h	*Frame one*
Frame number	0001h	0001h	*This is frame one*
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	TimeToOperate	
Extended word 2	-		
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault Information	

Response

Extended Word	Name	Description	Comment
Word 1 - 2	TimeToOperate	Time needed to switch to "Ready to operate" mode, in milliseconds.	0 - FFFFFFFFh
Word 8	Fault information	No suitable energy mode available. See "Fault Information" on page 58.	-

If the host application is unable to end the pause, the error "No suitable energy mode available" shall be returned.

9.2.5 Get PROFIenergy status (PNIO_IND_PE_GET_STATUS)

Description

With this command, the Anybus-S requests the host application to report back about the PROFIenergy state switch progress.

Command initiator	Anybus-S
Command number	0134h
Extended Header data	See below
Message data	There is no message data associated with this mailbox
Response message	The response message contains the current energy saving state number and the time/energy remaining until a (possible) transition has been completed

Command and response layout

	Command	Expected response	
Message ID	0001h	0001h	*Any integer value*
Message information	4002h	0002h	
Command	0134h	0134h	
Data size	0000h	0000h	
Frame count	0001h	0001h	*Frame one*
Frame number	0001h	0001h	*This is frame one*
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	CurrentEnergyMode	
Extended word 2	-	RemainingTime ToDestination	
Extended word 3	-		
Extended word 4	-	EnergyConsumption ToDestination	
Extended word 5	-		
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault Information	

Response

Extended Word	Name	Description	Comment
Word 1	CurrentEnergyMode	Number of the currently used energy saving mode.	0 - 255
Word 2 - 3	RemainingTime ToDestination	When changing between modes, this parameter will reflect the actual time, in milliseconds, remaining until the shift is complete. If a dynamic value cannot be generated, the static value for the transition from the source to the destination mode shall be used.	If the value is undefined, the maximum value FFFFFFFFh shall be used. If the value is zero (0), the value 00000000h shall be used.
	EnergyConsumption ToDestination	When changing between modes, this parameter will reflect the actual energy, in kWh, that will be consumed until the shift is complete. If a dynamic value cannot be generated, the static value for the transition from the source to the destination mode shall be used. (**Note**: this parameter is interpreted as a four byte float value).	If the value is undefined, the value 0.0 shall be used.
	Fault information	-	-

9.1 Modbus-TCP-related Commands

Commands in this category:

Mailbox Commands	Description	Page
Modbus Connection Timeout (MB_CONN_TIMEOUT)	Set the timeout value for Modbus-TCP connections	155
Disable Modbus-TCP server (DISABLE_MB_TCP)	Disables the Modbus-TCP server	156

9.1.1 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	

9.1.2 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	-	-	

9.1 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 "Fieldbus Specific Area" on page 185.

- **Blocking**

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

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

Commands in this category:

Mailbox Command	Description	Page
Socket non-blocking (SOCKET_NB)	Creates a socket in non-blocking mode.	158
Socket blocking (SOCKET_B)	Creates a socket in blocking mode.	159
Listen (LISTEN)	Starts listen on a socket for incoming connections.	160
Accept (ACCEPT)	Accepts connections for sockets in blocking mode.	161
Connect (CONNECT)	Tries to connect a socket to a client.	162
Send (SEND)	Sends a message to a connected socket.	164
Receive (RECEIVE)	Receives a message form a connected socket.	165
Send To (SEND_TO)	Sends a message to an unconnected UDP socket to a specified host.	166
Receive From (RECV_FROM)	Receives a message from an unconnected UDP socket.	167
Close (CLOSE)	Closes a socket (and connection).	168
Send Fragment (SEND_FRAG)	Sends a fragment of a message with a maximum total size of 4096 bytes.	169
Receive Fragment (RECV_FRAG)	Receives a fragment of a message with a total maximum size of 4096 bytes.	170
Send Fragment To (SEND_FRAG_TO)	Sends a fragment of a message with a total maximum size of 4096 bytes to an unconnected UDP socket.	172
Receive Fragment From (RECV_FRAG_FROM)	Receives a fragment of a message with a total maximum size of 4096 bytes from an unconnected UDP socket.	173
Get Socket Option (GET_SOCKET_OPTION)	Read options from a socket.	175
Set Socket Option (SET_SOCKET_OPTION)	Sets options to a socket	176

9.1.1 Socket Non-Blocking (SOCKET_NB)

Description

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

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

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

Command and response layout

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

- **Socket Type**

Value	Socket type
0001h	TCP socket
0002h	UDP socket

9.1.2 Socket Blocking (SOCKET_B)

Description

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

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

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

Command and response layout

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

- **Socket Type**

Value	Socket type
0001h	TCP socket
0002h	UDP socket

9.1.3 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 "Memory Map" on page 185 and "Socket Status Structure" on page 186.

- **Blocking sockets**

  Socket descriptors for new connections connected to this socket can be received by the mailbox command ACCEPT, see "Accept (ACCEPT)" on page 161.

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	

9.1.4 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**

9.1.5 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 "Fieldbus Specific Area" on page 185.

- **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

9.1.6 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	

9.1.7 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**

9.1.8 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	

9.1.9 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**

9.1.10 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	

9.1.11 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.

9.1.12 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.

9.1.13 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.

9.1.14 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.

9.1.15 Get Socket Option (GET_SOCKET_OPTION)

Description

This command reads options from a socket.

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

Command and response layout

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

Socket Options

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

SO_LINGER
SO_KEEPALIVE
SO_REUSEADDR
IP_MULTICAST_TTL
IP_MULTICAST_LOOP

For more information see section Socket Options page177.

9.1.16 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 page177.

9.1.17 Socket Options

Name	Option Value	DataType	Description
SO_LINGER	0x00000080	UINT32 UINT32	Controls the action taken when unsent data is queued on a socket that is being closed. This option is only valid for TCP sockets. l_onoff 0:Linger OFF (default) Other:Linger ON l_linger Normally defines the linger timeout. NOT SUPPORTED, ALWAYS SET TO 0. If SO_LINGER is disabled, Socket Close returns immediately and the connection is gracefully closed in the background. If SO_LINGER is enabled with a zero timeout, Socket Close returns immediately and the connection is reset.
SO_KEEPALIVE	0x00000008	UINT32	Enables/disables keep alive probes on a socket. This option is only valid for TCP sockets. l_keepalive 0:Keep alive OFF (default) Other:Keep alive ON Keep alive can be used to detect if the host is still active, and if not close down the connection. If keep alive is enabled a keep alive probe will be sent to the host after 2 hours with no data being sent or received on a connection. This packet is designed to provoke an ACK response from the host. If no ACK is received another 8 keep alive probes will be sent with 75 seconds interval, and if non of them is ACKed the connection will be reset.
SO_REUSEADDR	0x00000004	UINT32	Enables/disables reuse address option on a socket. This option is only valid for TCP sockets. l_reuseaddr 0:Reuse address OFF (default) Other:Reuse address ON When reuse address option is enabled it is possible to reuse a TCP port even if the port is busy in TIME_WAIT state. If the port is busy in other states an error will still be generated. This can be useful for a server implementation that is shut down and directly restarted while sockets are still active on its port.
IP_MULTICAST_TTL	0x0000000A	UINT8	Sets the TTL value for multicast packets. This option is only valid for UDP sockets. b_ttl 1-255 (Default 1) The TTL value is part of the IP packet header and specifies the number of routers a packet is allowed to pass before it shall be deleted. The default value of 1 prevents multicast packets from being forwarded beyond the local network.
IP_MULTICAST_LOOP	0x0000000B	UINT8	Enables/disables multicast packet loopback. This option is only valid for UDP sockets. l_reuseaddr 0:Multicast loopback OFF 1:Multicast loopback ON (default)

IP_ADD_MEMBERSHIP	0x0000000C	UINT32 UINT32	Adds membership to a multicast group. This option is only valid for UDP sockets. l_multiaddr IP address of multicast group to join. l_interface IP address of interface to join (own IP address) By joining a multicast group the local multicast router will be notified about the multicast membership (using IGMP) and the local interface network driver will enable reception of multicast datagrams destined for this multicast address.
IP_DROP_MEMBERSHIP	0x0000000D	UINT32 UINT32	Drops membership from a multicast group. This option is only valid for UDP sockets. l_multiaddr IP address of multicast group to leave. l_interface IP address of interface (own IP address) By leaving a multicast group the local multicast router will be notified and the local interface network driver will disable reception of multicast datagrams destined for this multicast address.
TCP_NODELAY	0x00002002	UINT32	Enables/disables the Nagle algorithm on a socket. This option is only valid on TCP sockets. l_nodelay 0:Nagle algorithm ON (default) l_nodelay Other:Nagle algorithm OFF For some applications, especially request/response applications, the performance over a TCP connection may be poor due to the interaction between the Nagle algorithm and the delayed acknowledgment functionality. Then the TCP_NODELAY option can be used to disable the Nagle algorithm to increase performance. For more information about Nagle algorithm see RFC 896.

9.1 Other Commands

Commands in this category:

Mailbox Command	Description	Page
DNS Request (DNS_REQUEST)	Asks the configured DNS server for the IP address of a specified host	180
Send Email (SEND_EMAIL)	Sends an email message to a specified recipient	181
Request SSI Data (REQUEST_SSI_DATA)	Requests SSI data from the application (issued by the Anybus module)	183
Write SSI Data (WRITE_SSI_DATA)	Writes SSI data to the application (issued by the Anybus module)	184

9.1.1 DNS Request (DNS_REQUEST)

Description

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

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

Command and response layout

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

9.1.2 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	

9.1.3 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 "printf" on page 51.

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.

9.1.4 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 "scanf" on page 53.

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.

10. Fieldbus Specific Area

10.1 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 - 687h	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
700h - 7A7h	Reserved	-
7B0h - 7B1h	IO Controller Status	RO
7B2h - 7B3h	PROFINET Layer Status	RO
7B4h - 7B5h	PROFINET Layer Fault Code	RO
7B6h - 7B9h	Reserved	-
7B6h - 7FFh	Reserved	-

10.2 Socket Status Structure

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

Socket Type

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

Socket Status

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

Socket Information

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

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

Local Port Number

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

Host Port Number

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

Host IP-address

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

10.3 Network Status

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

Bit	Description
LINK	0: Ethernet hardware link is not established 1: Ethernet hardware link is established
USE	0: No IP address configured - *or* - IP address not used by the module 1: IP address configured and used by the module (passed address collision detection)

10.4 IO Controller Status (PROFINET)

Value	Status
0000h	"No connection made"
0001h	STOP
0002h	RUN
0004h	STATION OK
0008h	STATION PROBLEM
0010h	PRIMARY
0020h	BACKUP

10.5 PROFINET Layer Status

This registers reflects the overall status of the PROFINET software layer in the Anybus module, and is primarily intended to be used as an aid during product development.

Value	Status
0000h	PROFINET layer not yet initialized
0001h	PROFINET layer successfully initialized
0002h	PROFINET layer failed to initialize (Contact HMS support)

10.6 PROFINET Layer Fault Code

In the event of a PROFINET layer initialization problem (see above), this register holds information that may be useful when contacting the HMS support department. Note that this register is primarily intended to be used as an aid during product development.

A. Miscellaneous

A.1 Control Register Area

Fieldbus Type

The fieldbus type value for this product is 0089h (PROFINET IRT).

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 Anybus module will go offline.

Event Notification Cause/Source Registers

- **ON/OFF Line Indication (FBON/FBOF)**
  This bit reflects the PROFINET online/offline status.
- **Network Reset Functionality (RST)**

A.2 Firmware Upgrade

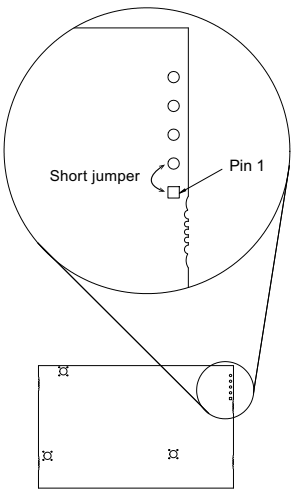
The Anybus module supports firmware updates via FTP. Follow the steps below:

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.

A.3 Formatting the File System

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

1. Attach a short jumper to the PCB as shown in figure. Apply power.
2. Wait until the watchdog LED turns red.
3. Disconnect power.
4. Remove jumper.
5. Apply power.
6. Wait approx. 1 minute while the filesystem is being formatted.



B. Technical Specification

B.1 Electrical Specification

B.1.1 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.

B.1.2 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)

B.1.3 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.

B.2 Environmental Specification

Temperature

Test performed according to IEC 60068-2-1 and IEC 60068-2-2.

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

Humidity

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

Test performed according to IEC 60068-2-30.

B.3 Regulatory Compliance

EMC (CE)

EMC compliance testing has been conducted according to the Electromagnetic Compatibility Directive 2004/108/EC. For more information please consult the EMC compliance document, see product/support pages for Anybus-S PROFINET IRT at www.anybus.com, "CE - Declaration of Pre-Conformity".

The tests have been performed to ensure that the final product, after new tests, may be fully compliant to the directive.

C. Connectors

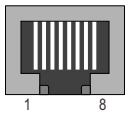
C.1 Application Connector

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

C.2 Ethernet Connectors

RJ45 (Standard Connector)

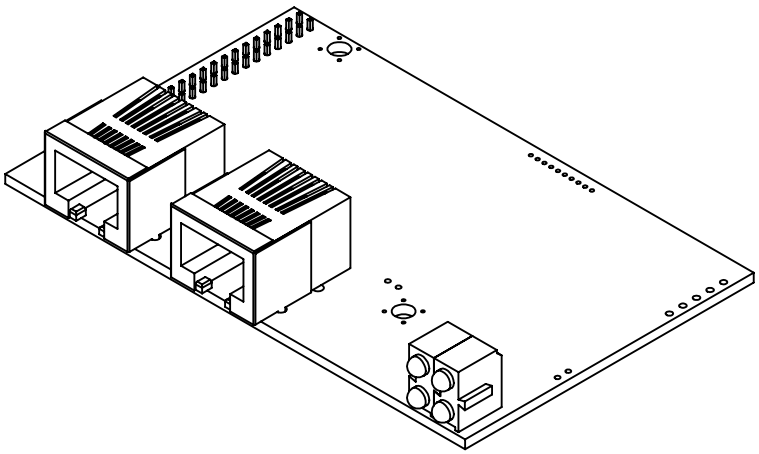
Pin	Signal	Notes
1	TD+	-
2	TD-	-
3	RD+	-
4	-	Normally left unused; to ensure signal integrity, these pins are tied together and terminated to PE via a filter circuit in the module.
5	-	
6	RD-	-
7	-	Normally left unused; to ensure signal integrity, these pins are tied together and terminated to PE via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

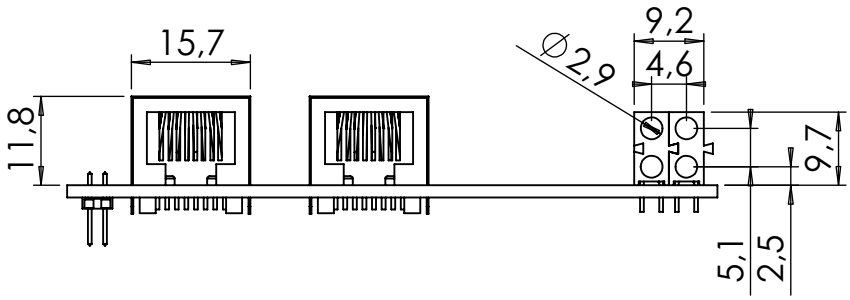


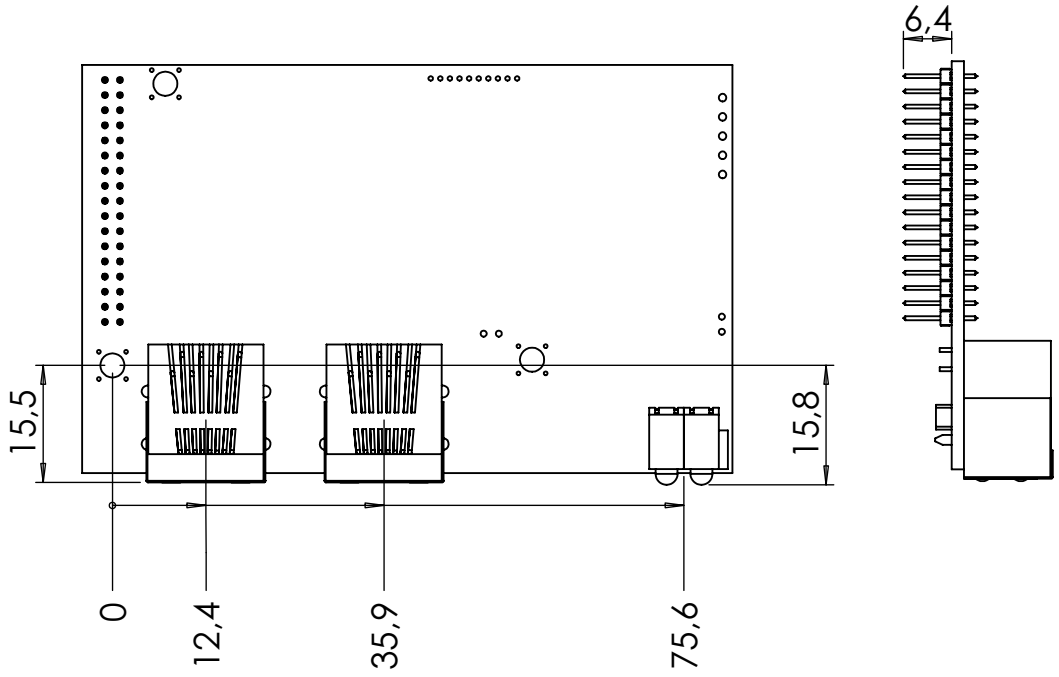
D. Mechanical Specification

D.1 Measurements, Connectors & LEDs

All measurements are in millimetres.







E. Conformance Test Guide

E.1 General

When using the default settings of all parameters, the Anybus-S PROFINET IO module is precertified for network compliance. This precertification is done to ensure that your product *can* be certified, but it does not mean that your product will not require certification.

Any change in the parameters in the GSD file, supplied by HMS, will require a certification. A vendor ID can be obtained from PNO and is compulsory for certification. This chapter provides a guide for successful conformance testing your product, containing the Anybus-S PROFINET IO module, to comply with the demands for network certification set by the PNO.

Independent of selected operation mode, the actions described in this appendix have to be accounted for in the certification process. The identity of the product needs to be changed to match your company and device.

IIMPORTANT: *This appendix provides guidelines and examples of what is needed for certification. Depending on the functionality of your application, there may be additional steps to take. Please contact HMS Industrial Networks at www.anybus.com for more information.*

E.2 Reidentifying Your Product

A number of mailbox messages have to be sent during initialization to change the HMS default values to values reidentifying the product. The messages have to be sent at each power on/reset before the mailbox END_INIT signals the completion of the initialization. These mailboxes are listed in the table below.

Mailbox (See also PROFINET Specific Commands on 98)	**Parameter set**	**Explanation**	**Default**	**Customer sample**	**Comment**
PNIO_SET_DEVICE_IDENTITY	Vendor ID Device ID	With this mailbox you set the Vendor ID and the Device ID of the product	Vendor ID: 010Ch (HMS Industrial Networks) Device ID: 0006h	Vendor ID: XXXXh Device ID: YYYYh	This information must match the keys of the "DeviceIdentity" of the GSD-file. Note that the GSD file keyword "VendorName" must correspond to the Vendor ID value.
PNIO_SET_STATION_TYPE	Station Type	With this mailbox you set the station type of the device	"ABS-PIR"	"Cust-PNIO-Dev"	This information matches, in the case of ABS-PIR, GSD keywords "DNS_CompatibleName" and "OrderNumber". The Station Type must be equal to the "DNS_CompatibleName", but it is allowed to have a completely different "OrderNumber", see also PNIO_SET_IM0_INFO below.
PNIO_SET_SYSTEM_DESCRIPTION PNIO_SET_SYSTEM_DESCRIPTION_MIB2	MIB and MIB-II information	With this mailbox you set the description of the system.	"HMS Industrial Networks Anybus-S"	"Customer HMI Interface Module"	This information kan be read by means of SNMP from the network side.
PNIO_SET_INTERFACE_DESCRIPTION	MIB-II information	With this mailbox you set the description of the interface	"PROFINET IO interface"	"PROFINET IO interface"	
PNIO_SET_SYSTEM_CONTACT_MIB2		With this mailbox you set the system contact information	"www.anybus.com"	"www.customer.com"	This information kan be read by means of SNMP from the network side, via the MIB-II.
PNIO_SET_SYSTEM_LOCATION_MIB2	MIB-II information	With this mailbox you set the system location information	"Stationsgatan 37, 30245 Halmstad, Sweden."	"Address, City, Country"	
PNIO_SET_IM0_INFO	I&M0 information, including the Manufacturer ID and Order ID. For complete list of attributes see "Set IM0 Information (PNIO_SET_IM0_INFO)" on page 134.	With this mailbox you set the information that shall be returned when an Identification and Maintenance request is received from the PROFINET network.	Manufacturer ID: 010Ch Order ID: "ABS-PIR"	Manufacturer ID: XXXXh Order ID: "Cust-PNIO-Dev"	The Manufacturer ID must match the Vendor ID set with PNIO_SET_DEVICE_IDENTITY and Order ID must be changed (original information can be fetched with PNIO_GET_IM0_INFO). The Order ID must match the GSD file keyword "OrderNumber". The hardware revision must match the GSD file keyword "HardwareRelease". The software revision must match the GSD file keyword "SoftwareRelease".

Additional GSD File Information

The GSD file keyword "ProductFamily" shall correspond to the vendor's name of the device.

The GSD file keyword "MainFamily" lists the kinds of devices for which the product shall be listed. As of GSD specification v2.25, the following "families" are available:

"General", "Drives", "Switching Devices", "I/O", "Valves", "Controllers", "HMI", "Encoders", "NC/RC", "Gateway", "PLCs", "Ident Systems", "PA Profiles", "Network Components", "Sensors".

E.3 Factory Default Reset - RDR Support Recommended

When PROFINET IO modules are delivered, they are required to be in their "Factory Default" state. For PROFINET devices this means that their Station Name is empty ("") and that the IP-suite is not assigned (IP 0.0.0.0). When a Factory Default Reset command is received from the network, the Anybus module will erase all IP and Station Name information and inform the host application that hardware or software reset of the Anybus module is required. This is done by using the RDR functionality (Field-bus Reset Device Request Notification). For information on how to enable this functionality please refer to the Anybus-S Parallel Design Guide.

When the RDR functionality is enabled, the Anybus module will set the RDR bit in the Module Status register (when a Factory default reset command has arrived) and wait for the host application to perform a hardware or software reset of the Anybus module (i.e., the Anybus module will not reset itself) and then reinitialize the module.

Note: It is strongly recommended to enable RDR support.

E.4 IP Address

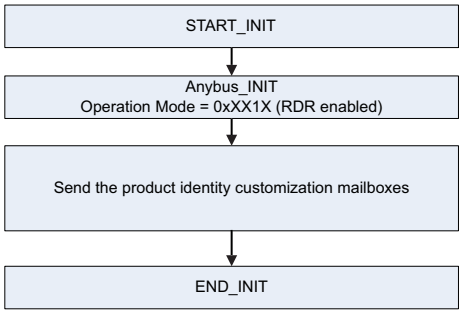
Normally the IP numbers of PROFINET IO devices are assigned via the PROFINET network via DCP (Discovery and Configuration Protocol). HMS recommends not using the mailbox SET_ETN_CONFIG during the initialization phase for PROFINET modules, unless the end user has requested the IP address to be set to a specific value (by for example using a keypad). The reason is that when a factory default reset command is received from the PROFINET network (via DCP) the node msut be available after a hardware or software reset with the default IP-address (0.0.0.0).

E.5 Station Name

Normally the Station Name of a PROFINET device is assigned by the end user via the PROFINET DCP protocol. HMS recommends not using the mailbox SET_STATION_NAME during the initiali-zation phase for PROFINET modules. If this mailbox is used, it is recommended that it is sent explicitly when the end user changes the Station Name with e.g. a keypad. The reason is that when a factory default reset command is received from the PROFINET network (via DCP), the node must available after a hardware or software reset with the default Station Name ("").

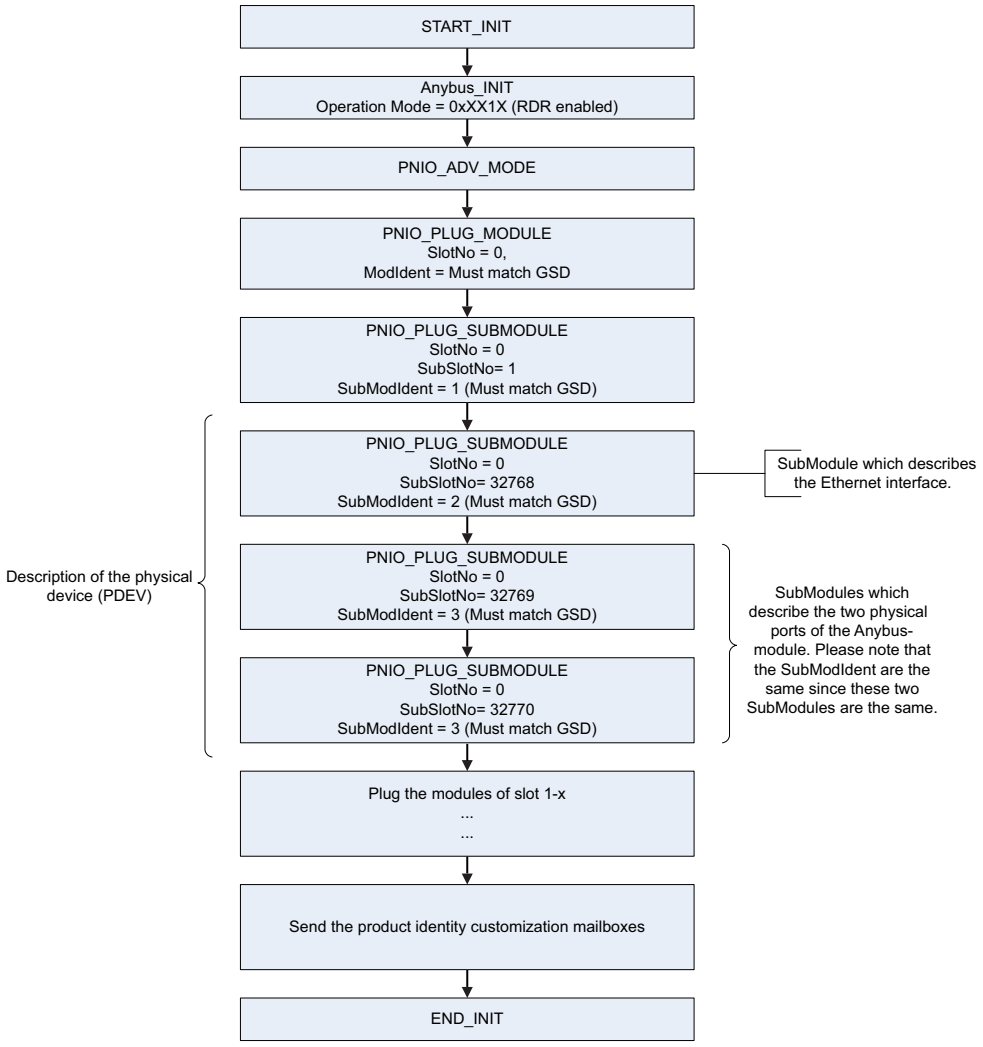
E.6 Certification in Generic Anybus Mode

In Generic Anybus Mode (when the PNIO_ADV_MODE is not used) there is normally nothing that needs to be considered apart from what is mentioned earlier in this appendix. The default HMS GSD file has to be modified with respect to the identity of the product and this requires a certification of the product.



E.7 Certification in Advanced Mode

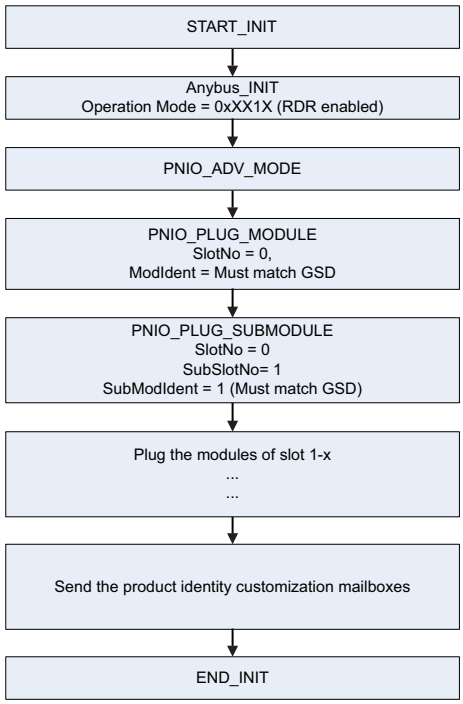
In advanced mode (mailbox PNIO_ADV_MODE is used), the most important thing is to use a Device Access Point (DAP) that conform to PROFINET IO Specification v2.0 or later (DAP2). From specification version 2.0 it is possible to describe the physical Ethernet interface and its ports (PDEV, or Physical Device) with a special mechanism. This is done with special submodules at slot 0 (the module at slot 0 is the access point for the device). HMS recommends following the flow below for setting up a DAP2.



The figure shows how to set up a PROFINET compatible DAP. Please note that for some mailboxes only the relevant parameters are shown.

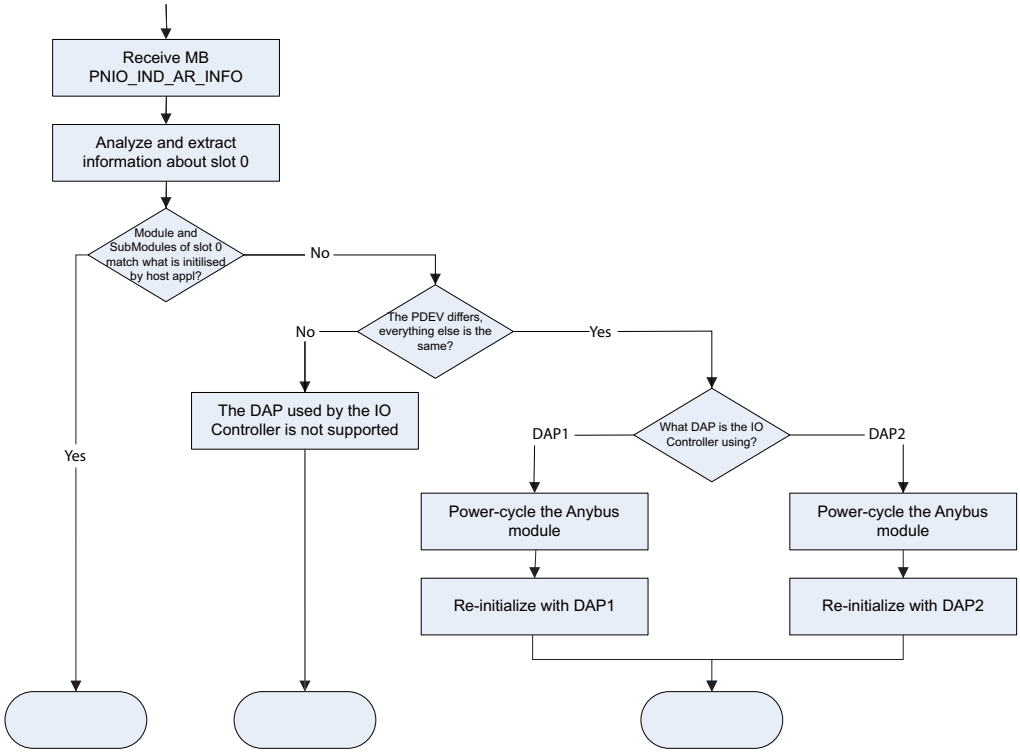
Please note that the values of "SubModIdent" in the above flowchart are the values of the default HMS GSD file. They can be changed if necessary, but there si no real need for it, the important thing is that it matches the GSD file. To be able to pass the PROFINET conformance test a "DAP2" is mandatory. On the market there still are some PROFINET IO controllers not supporting PROFINET IO specification v2.0 or later. These controlllers cannot use a DAP2. These controllers cannot use a DAP2. Therefore, it might be necessary to support also a DAP containing no PDEV (i.e. the three last PNIO_PLUG_SUBMODULE mailboxes are not issued). This is called a "migration" DAP. In the default GSD file there is such functionality. In the case of advanced mode this can be implemented in either of these two ways:

1. The end user decides that reverse compatibility is necessary and selects this for example with a parameter on a hand panel. The host application perfomrs a hardware or a softwarer reset of the Anybus module and skips the last three PNIO_PLUG_SUBMODULE mailboxes as shown in the figure on 196, resulting in the flow shown in the figure below:



The figure shows a DAP without a PDEV, for reverse compatibility only (please note that for some mailboxes only the relevant parameters are shown.)

2. The host application has enabled the PNIO_IND_AR_INFO mailbox and can thus analyze the connection which is being established by the IO Controller. If the IO Controller is trying to use the DAP which has not been plugged the host application can do a hardware or software reset of the Anybus module and re-initialize the Anybus module with the correct DAP (with or without PDEV), as described in the figure below:



Note: To enable the PNIO_IND_AR_INFO mailbox the PNIO_ADV_MODE mailbox shall be sent with the "Spontaneous Mailbox" field set to 0x0010 (if also other mailboxes shall be enabled please or the individual bits toghether)

The figure shows a flowchart of the functionality to swap DAPs depending on what the IO Controller is using.

Once the DAP has been plugged into slot 0, the other slots can be populated. Of some importance with these other modules, is that the Module Identification Number must uniquely define the kind of module (for example, a digital input module must not have the same module identification number as a digital ouput module). There is one exception to this rule for the DAP. It is allowed to have a DAP with or without a PDEV, but with the same module identification number.

HMS recommends that the host application, if possible, store, in nonvolatile memory, the DAP used last time and uses that DAP after power-cycle. The reason for doing so is to reduce time for connection establishment. If no DAP is stored DAP2 shall be used. If it is not possible for the host application to store the most recently used DAP, the host application should always plug DAP2 initially.

F. Copyright Notices

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