



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China





User Manual Anybus[®] Communicator[™] for PROFINET

Doc. Id. HMSI-27-309
Rev. 3.11

Important User Information

This document contains a general introduction as well as a description of the technical features provided by the Anybus Communicator, including the PC-based configuration software.

The reader of this document is expected to be familiar with PLC and software design, as well as communication systems in general. The reader is also expected to be familiar with the Microsoft® Windows® operating system.

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. Microsoft® and Windows® are registered trademarks of Microsoft, Inc. 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 Communicator PROFINET User Manual
Copyright© HMS Industrial Networks AB
Doc: HMSI-27-309

Table of Contents

Preface	About This Document	
	Related Documents	8
	Document History	8
	Conventions & Terminology	9
	<i>Glossary</i>	9
	Support.....	9
Chapter 1	About the Anybus Communicator for PROFINET	
	General.....	10
	External View.....	11
	Status LEDs	12
	Hardware Installation.....	13
	Software Installation	14
	<i>Anybus Configuration Manager</i>	14
Chapter 2	Basic Operation	
	General.....	15
	Data Exchange Model	16
	<i>Memory Map</i>	16
	<i>Data Exchange Example</i>	17
	Subnetwork Protocol.....	18
	<i>Protocol Modes</i>	18
	<i>Protocol Building Blocks</i>	18
	<i>Master Mode</i>	19
	<i>Generic Data Mode</i>	20
	<i>DF1 Master Mode</i>	20
	PROFINET IO	21
	<i>General</i>	21
	<i>I/O Configuration</i>	21
	<i>GSDML File</i>	21
	<i>Data Representation (IO Data & Record Data)</i>	22
	Modbus-TCP (Read-Only).....	23
	<i>General</i>	23
	<i>Data Representation (Modbus-TCP Register Map)</i>	23
	<i>Supported Exception codes</i>	23
Chapter 3	File System	
	General.....	24
	File System Overview	25
	System Files	25

Chapter 4	Basic Network Configuration	
	General Information.....	26
	Ethernet Configuration File (“ethcfg.cfg”).....	27
	<i>General</i>	27
	PROFINET Settings	28
	IP Access Control	28
	Anybus IPconfig (HICP)	29
Chapter 5	FTP Server	
	General.....	30
	FTP Connection Example (Windows Explorer).....	31
Chapter 6	Web Server	
	General.....	32
	Authorization	33
	Content Types.....	34
Chapter 7	Server Side Include (SSI)	
	General.....	35
	Functions	36
	Changing SSI output.....	45
	<i>SSI Output String File</i>	45
	<i>Temporary SSI Output Change</i>	46
Chapter 8	E-mail Client	
	General.....	47
	E-mail Definitions.....	48
Chapter 9	Navigating ACM	
	Main Window.....	49
	<i>Drop-down Menus</i>	50
	<i>Toolbar Icons</i>	53
Chapter 10	Basic Settings	
	Fieldbus Settings.....	54
	Communicator Parameters	55
	Sub-network Parameters	56

Chapter 11	Nodes	
	General.....	57
	Adding & Managing Nodes.....	57
	Node Parameters.....	57
	<i>Master Mode and Generic Data Mode</i>	57
Chapter 12	Transactions	
	General.....	58
	Adding & Managing Transactions.....	59
	Transaction Parameters (Master Mode).....	60
	<i>Parameters (Query & Broadcast)</i>	60
	<i>Parameters (Response)</i>	61
	Transaction Parameters (Generic Data Mode).....	62
	<i>Produce Transactions</i>	62
	<i>Consume Transactions</i>	63
	Transaction Editor.....	64
Chapter 13	Frame Objects	
	General.....	65
	Adding and Editing Frame Objects.....	65
	Constant Objects (Byte, Word, Dword).....	66
	Limit Objects (Byte, Word, Dword).....	67
	Data Object.....	68
	Variable Data Object.....	68
	Checksum Object.....	70
Chapter 14	Commands	
	General.....	71
	Adding & Managing Commands.....	71
	<i>Drop-down Menu</i>	72
	<i>Toolbar Icons</i>	72
	The Command Editor.....	73
	<i>General</i>	73
	<i>Basic Navigation</i>	73
	<i>Drop-down Menu</i>	74
	<i>Editing a Command</i>	74
	<i>Example: Specifying a Modbus-RTU Command in Master Mode</i>	75

Chapter 15	DF1 Protocol Mode	
	General.....	76
	Communicator Parameters	77
	Sub-network Parameters	78
	Node Parameters	79
	Services.....	79
	<i>Available Services</i>	80
	Integrity Check	81
	Read Diagnostics	81
	Read Data	82
	Write Data	82
Chapter 16	Sub-network Monitor	
	General.....	83
	Operation.....	83
Chapter 17	Node Monitor	
	General.....	84
	Navigating the Node Monitor.....	85
	<i>Drop-down Menu</i>	86
	<i>Toolbar Icons</i>	87
Chapter 18	Data Logger	
	General.....	88
	Operation.....	88
	Configuration	89
Chapter 19	Configuration Wizards	
	General.....	90
	Selecting a Wizard Profile	90
	Wizard - Modbus RTU Master	91
Chapter 20	Control and Status Registers	
	General.....	92
	<i>Handshaking Procedure</i>	92
	<i>Data Consistency</i>	93
	Status Register Contents (Gateway to Control System).....	94
	<i>General Information</i>	94
	<i>Status Codes in Master Mode and DF1 Master Mode</i>	94
	<i>Status Code in Generic Data Mode</i>	95
	Control Register Contents (Control System to Gateway).....	96
	<i>General Information</i>	96
	<i>Control Codes in Master Mode and DF1 Master Mode</i>	96
	<i>Control Codes in Generic Data Mode</i>	96

Chapter 21	Advanced Fieldbus Configuration	
	General.....	97
	Mailbox Editor.....	97
Appendix A	Connector Pin Assignments	
	PROFINET Connector (Ethernet).....	98
	Power Connector	98
	PC Connector	99
	Subnetwork Interface	100
	<i>General Information</i>	100
	<i>Bias Resistors (RS485 Only)</i>	100
	<i>Termination (RS485 & RS422 Only)</i>	100
	<i>Connector Pinout (DB9F)</i>	100
	<i>Typical Connection (RS485)</i>	101
	<i>Typical Connection (RS422 & 4-Wire RS485)</i>	101
	<i>Typical Connection (RS232)</i>	101
Appendix B	Technical Specification	
	Mechanical Properties.....	102
	Electrical Characteristics	102
	Environmental Characteristics	102
	Regulatory Compliance	103
Appendix C	Troubleshooting	
Appendix D	ASCII Table	
Appendix E	Copyright Notices	

P. About This Document

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

P.1 Related Documents

Document name	Author
Anybus Communicator - PRT Installation Sheet	HMS
DF1 Protocol and Command Set - Reference Manual, 1770-6.5.16, October 1996	Allen-Bradley

P.2 Document History

Summary of Recent Changes (3.01... 3.10)

Change	Page(s)
Screenshots and descriptions of ABC Tool updated for Anybus Configuration Manager	Multiple
Changed "ABC" to "Communicator RS232/422/485"	Multiple
Amended description of "Update time" parameter	31, 32
Added description for Consume/Response to "Object Delimiter" parameter	39
Changed "Maximum Data Length" limit	39
Removed obsolete "Start Bits" parameter	48
Removed obsolete "ABCC ExtLink Wizard" entry	64
Replaced "Sales and Support" info with link to website	8
Added parameters to checksum object description	40
Added info about Modbus registers	23
Minor text edits, typo corrections	Multiple

Summary of Recent Changes (3.10... 3.11)

Revision	Change	Page(s)
3.11	Added compliance info	79

Revision List

Revision	Date	Author	Chapter	Description
1.00	2005-06-07	PeP	All	1st release
2.00	2006-03-27	PeP	All	2nd major release
2.01	2006-12-22	PeP	-	Misc. minor corrections & updates
2.02	2008-10-14	HeS	-	Misc. minor corrections & updates
2.03	2009-04-24	KeL	All	Misc. minor corrections and updates
3.00	2011-01-19	KaD	All	Misc. minor corrections, new template and DF1 functionality
3.01	2011-09-30	KaD	All	Misc corrections and updates, new Anybus Configuration Manager name
3.10	February 2015	ThN	All	Misc. corrections and updates, new Doc ID
3.11	March 2015	ThN	B	Added compliance info

P.3 Conventions & Terminology

The following conventions are used throughout this document:

- Numbered lists provide sequential steps
- Bulleted lists provide information, not procedural steps
- The term “user” refers to the person or persons responsible for installing the Anybus Communicator in a network.
- The term “ABC” refers to the Anybus Communicator.
- Hexadecimal values are written in the format 0xNNNN, where NNNN is the hexadecimal value.
- Decimal values are represented as NNNN where NNNN is the decimal value
- As in all communication systems, the terms “input” and “output” can be ambiguous, because their meaning depend on which end of the link is being referenced. The convention in this document is that “input” and “output” are always being referenced to the master/scanner end of the link.

P.3.1 Glossary

Term	Meaning
ABC	Anybus [®] Communicator™
PRT	PROFINET-IO
ACM	Anybus Configuration Manager
Broadcaster	A protocol specific node in the subnetwork scan- that hold transactions destined to all nodes.
Command	A protocol specific transaction.
Configuration	List of configured nodes with transactions on the subnetwork.
Fieldbus	The network to which the communicator is connected.
Fieldbus Control System	Fieldbus master
Frame	Higher level series of bytes forming a complete telegram on the subnetwork
Monitor	A tool for debugging the ABC and the network connections.
Node	A device in the configuration that defines the communication with a node on the subnetwork
Scan list	List of configured slaves with transactions on the subnetwork.
Subnetwork	The network that logically is located on a subsidiary level with respect to the fieldbus and to which the ABC acts as a gateway.
Transaction	A generic building block that is used in the subnetwork scan-list and defines the data that is sent and received on the subnetwork.
IO Controller	PROFINET device which acts as a client for several IO devices, usually a PLC. (Comparable to a PROFIBUS-DP class 1 master).
IO Supervisor	PROFINET programming device with commissioning and diagnostic functions (Comparable to a PROFIBUS-DP class 2 master).
IO Device	Field device assigned to an IO Controller. In this case the ABC.
Module (or I/O module)	Hardware or logical component of a PROFINET network device.
Higher Level Network	In this case, PROFINET
Network	
Fieldbus	

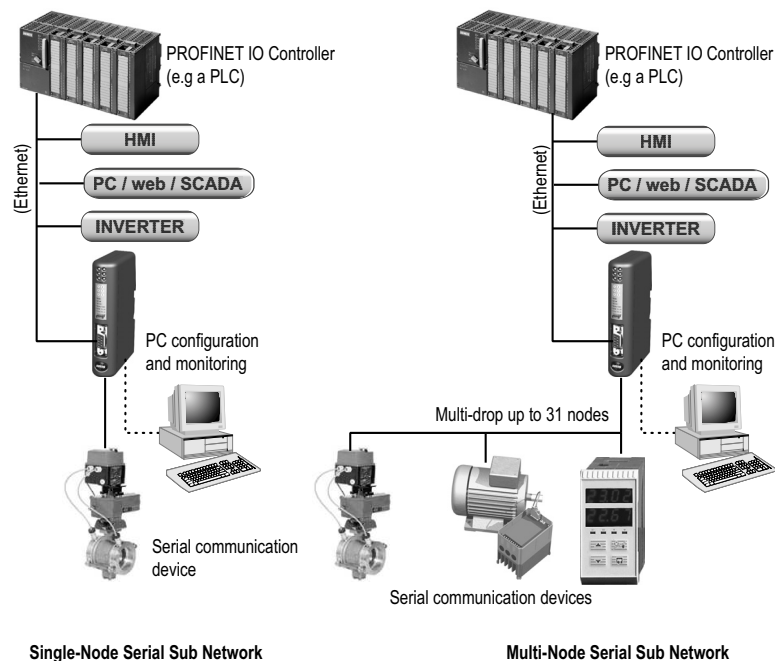
P.4 Support

For general contact information and support, please refer to the contact and support pages at www.anybus.com

1. About the Anybus Communicator for PROFINET

1.1 General

The Anybus Communicator module for PROFINET acts as a gateway between virtually any serial application protocol and a PROFINET IO-based network. Integration of industrial devices is enabled without loss of functionality, control and reliability, both when retro-fitting to existing equipment as well as when setting up new installations.



Subnetwork

The Anybus Communicator can address up to 31 nodes, and supports the following physical standards:

- RS-232
- RS-422
- RS-485

Ethernet Interface

Ethernet connectivity is provided through the patented Anybus technology; a proven industrial communication solution used all over the world by leading manufacturers of industrial automation products.

- PROFINET IO
- Modbus-TCP server (read only)
- Security framework with per user access rights and IP access control
- Server Side Include (SSI) functionality
- Web server and Email client capability
- Easy file management via FTP
- 10/100 Mbit/s, twisted pair

1.2 External View

For wiring and pin assignments, see “Connector Pin Assignments” on page 98.

A: PROFINET Connector (Ethernet)

This connector is used to connect the module to the network.

See also...

- “PROFINET Connector (Ethernet)” on page 98

B: Status LEDs

See also...

- “Status LEDs” on page 12

C: PC-connector

This connector is used to connect the module to a PC for configuration and monitoring purposes.

See also...

- “PC Connector” on page 99

D: Subnetwork Connector

This connector is used to connect the module to the serial subnetwork.

See also...

- “Subnetwork Interface” on page 100

E: Power Connector

This connector is used to apply power to the module.

See also...

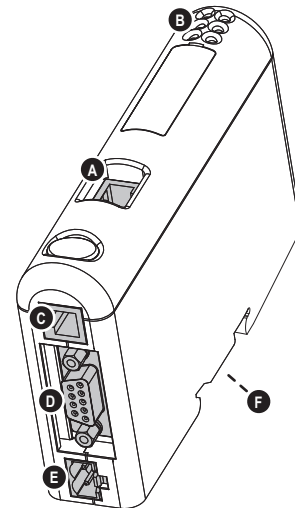
- “Power Connector” on page 98
- “Technical Specification” on page 128

F: DIN-rail Connector

The DIN-rail mechanism connects the module to PE (Protective Earth).

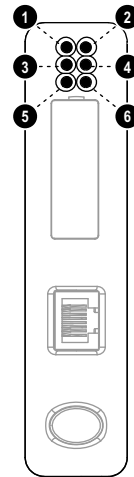
See also...

- “Hardware Installation” on page 13



1.3 Status LEDs

#	State	Status
1 - Comm. Status	Off	Off line - No connection with IO Controller
	Green	On line, Run - Connection with IO Controller established - IO Controller is in RUN state
	Green, flashing	On line, STOP - Connection with IO Controller established - IO Controller in STOP state
2 - Module Status	Off	No power or not initialized
	Green	Initialized, no error
	Green, 1 flash	Diagnostic data available
	Green, 2 flashes	Blink. Used by engineering tools for identification.
	Red, 1 flash	Configuration Error - Too many modules/submodules - I/O size or Configuration mismatch
	Red, 3 flashes	No Station Name or no IP address assigned
	Red, 4 flashes	Internal error
3 - Link/Activity	Off	No link or power off
	Green	Link established
	Green, flashing	Receiving/transmitting data
4 - (not used)	-	-
5 - Subnet Status ^a	Off	Power off
	Green, flashing	Running correctly, but one or more transaction error(s) have occurred
	Green	Running
	Red	Transaction error/timeout or subnet stopped
6 - Device Status	Off	Power off
	Alternating Red/Green	Invalid or missing configuration
	Green	Initializing
	Green, flashing	Running
	Red	Bootloader mode ^b
	Red, flashing	If the Device Status LED is flashing in a sequence starting with one or more red flashes, please note the sequence pattern and contact the HMS support department



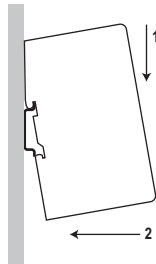
- a. This LED turns green when all transactions have been active at least once. This includes any transactions using “change of state” or “change of state on trigger”. If a timeout occurs on a transaction, this LED will turn red.
- b. The gateway is in bootloader mode, and firmware must be restored in order for it to work properly. Start up the Anybus Configuration Manager and connect to the Anybus Communicator. Choose Tools/Options/Module. Click “Factory Restore” to restore firmware. See “Tools” on page 61.

1.4 Hardware Installation

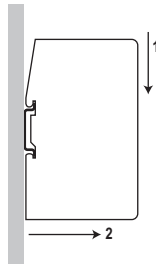
Perform the following steps when physically installing the Anybus Communicator module:

1. Snap the module on to the DIN-rail.

The DIN-rail mechanism works as follows:



To snap the module *on*, first press it downwards (1) to compress the spring in the DIN-rail mechanism, then push it against the DIN-rail as to make it snap on (2)



To snap the module *off*, push it downwards (1) and pull it out from the DIN-rail (2), as to make it snap off from the DIN-rail

2. Connect the module to the PROFINET (Ethernet) network
3. Connect the module to the serial subnetwork
4. Connect the module to the PC via the configuration cable
5. Connect the power cable and apply power
6. Start the Anybus Configuration Manager program on the PC
(The Anybus Configuration Manager software attempts to detect the serial port automatically. If not successful, select the correct port manually in the "Port"-menu)
7. Configure the Anybus Communicator using the Anybus Configuration Manager and download the configuration
8. Set up the PROFINET communication in accordance with the module configuration

1.5 Software Installation

1.5.1 Anybus Configuration Manager

System requirements

- Pentium 133 MHz or higher
- 650 MB of free space on the hard drive
- 32 MB RAM
- Screen resolution 800 x 600 (16 bit color) or higher
- Microsoft Windows® 2000 / XP / Vista / 7 (32- or 64-bit)
- Internet Explorer 4.01 SP1 or newer (or any equivalent browser)

Installation

- **Anybus Communicator resource CD**
 - Insert the CD and follow the on-screen instructions.
 - If the installation does not start automatically: right-click on the CD drive icon and select “Explore” to show the contents of the CD. Locate the installation executable and double-click on it to start the installation, then follow the on-screen instructions.
- **From HMS website**
 - Download the latest version of Anybus Configuration Manager from www.anybus.com.
 - Unzip the archive on your computer and double-click on the installation executable.

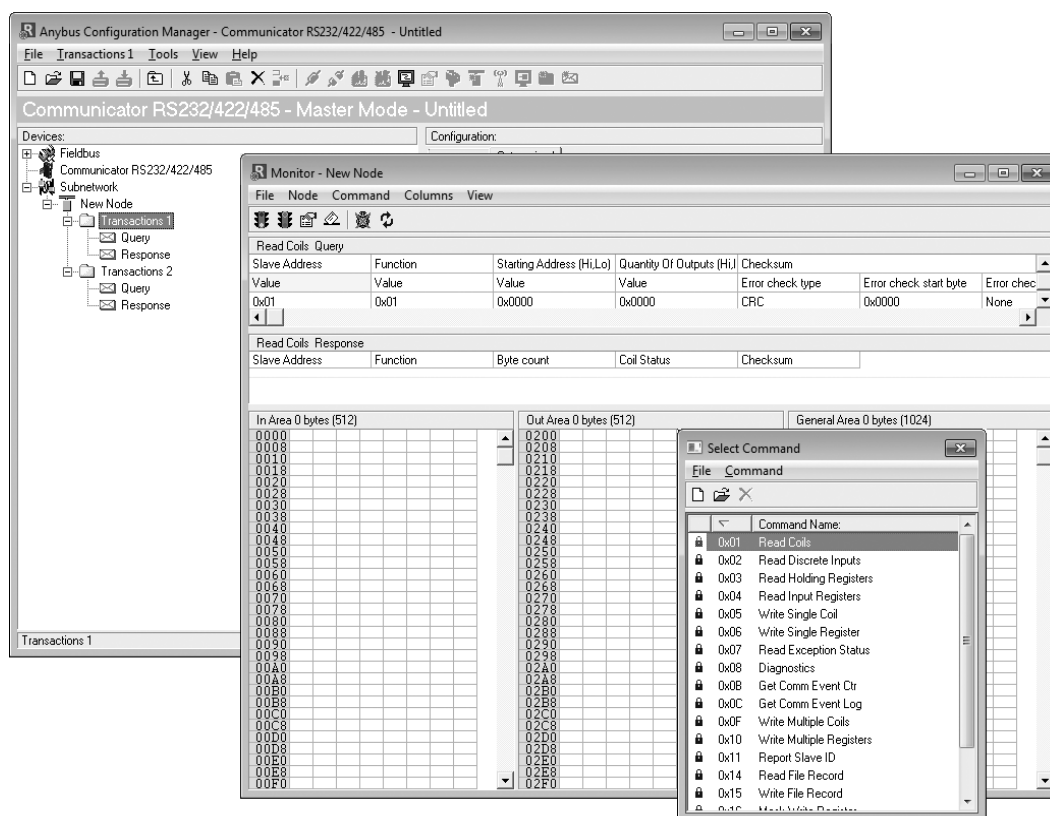
2. Basic Operation

2.1 General

The Anybus Communicator gateway is designed to exchange data between a serial sub-network and a higher level network. Unlike most other gateway devices of similar kind, it does not have a fixed protocol for the sub-network, and can be configured to handle almost any form of serial communication.

The gateway can issue serial telegrams cyclically, on change of state, or based on trigger events issued by the control system of the higher level network (i.e. the fieldbus master or PLC). It can also monitor certain aspects of the sub-network communication and notify the higher level network when data has changed.

An essential part of the Anybus Communicator package is Anybus Configuration Manager (ACM), a Windows®-based application used to supply the gateway with a description of the sub-network protocol. No programming skills are required; instead, a visual protocol description-system is used to specify the different parts of the serial communication.



2.2 Data Exchange Model

Internally, the data exchanged on the subnetwork, and the data exchanged on the higher level network, reside in the same memory.

This means that in order to exchange data with the subnetwork, the higher level network simply reads and writes data to memory locations specified using the Anybus Configuration Manager. The very same memory locations can then be exchanged on the subnetwork.

The internal memory buffer is divided into three areas based on their function:

- **Input Data (512 bytes)**

This area can be read by the higher level network, the web server and the e-mail client.

(How this data is represented on the higher level network will be described later in this chapter).

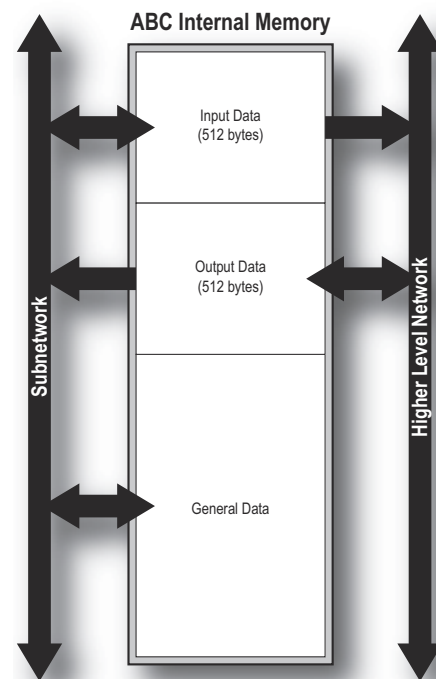
- **Output Data (512 bytes)**

This area can be read from/written to by the higher level network, the web server and the e-mail client.

(How this data is represented on the higher level network will be described later in this chapter).

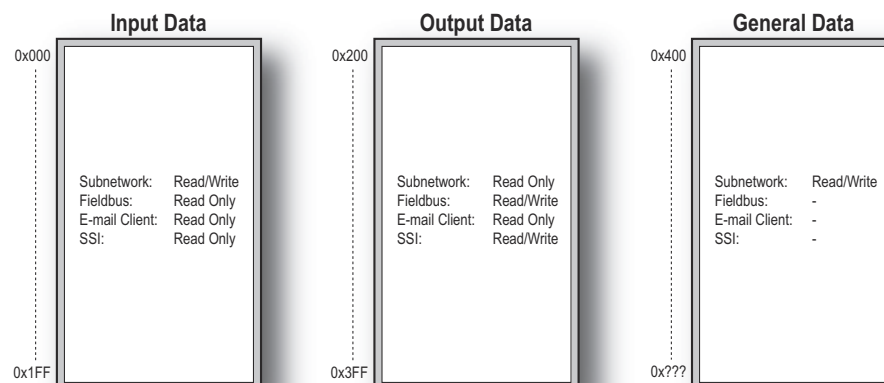
- **General Data (Up to 1024 bytes)**

This area cannot be accessed from the higher level network, but can be used for transfers between individual nodes on the subnetwork, or as a general “scratch pad” for data. The actual size of this area depends on the amount of data that is exchanged on the subnetwork. The Anybus Communicator can handle up to 1024 bytes of general data.



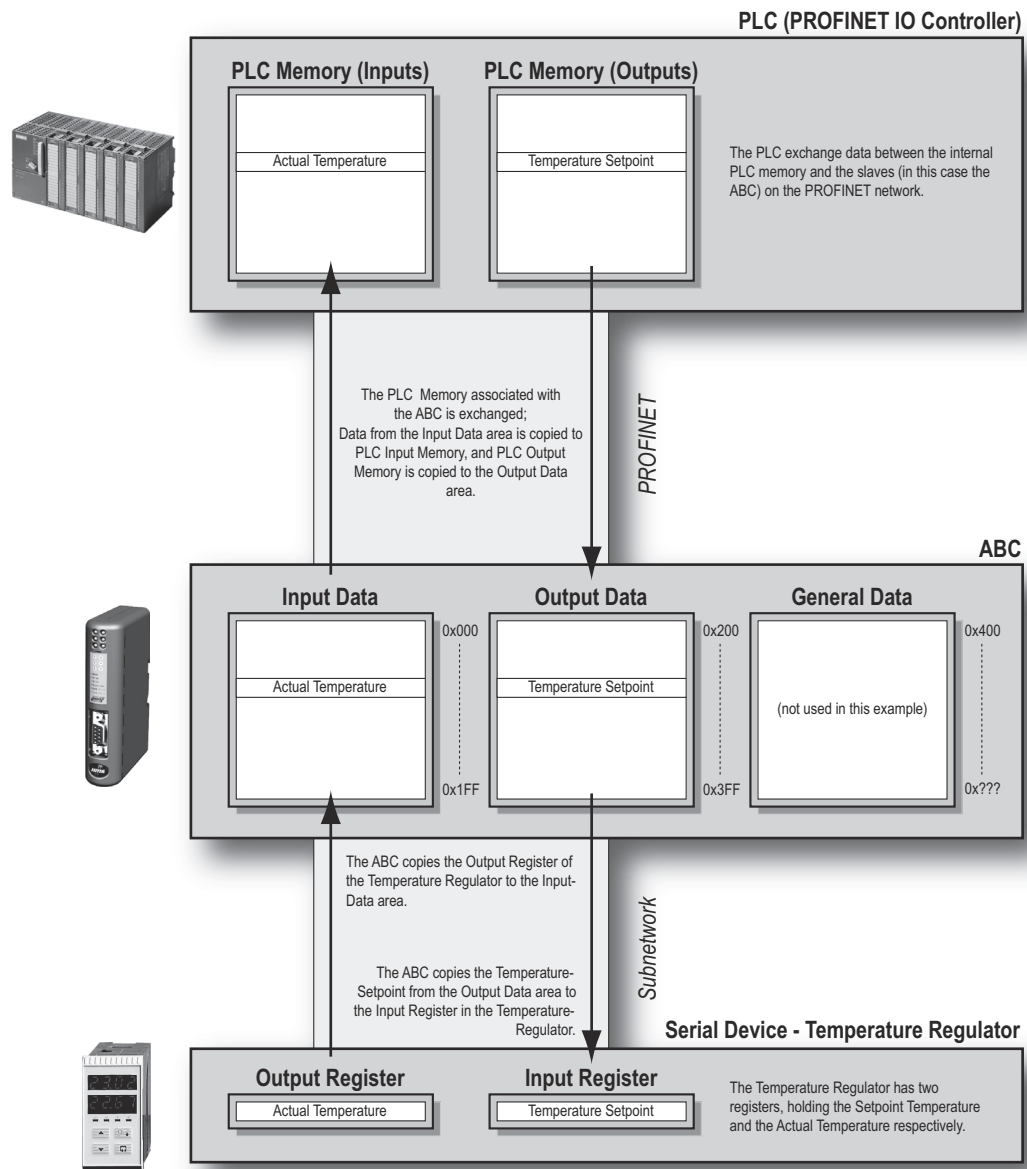
2.2.1 Memory Map

When building the subnetwork configuration using the Anybus Configuration Manager, the different areas described above are mapped to the memory locations (addresses) specified below.



2.2.2 Data Exchange Example

In the following example, a temperature regulator on the subnetwork exchanges information with a PLC on the higher level network, via the internal memory buffers in the module.



2.3 Subnetwork Protocol

2.3.1 Protocol Modes

The Anybus Communicator features three distinct modes of operation regarding the subnetwork communication, called “Master Mode”, “Generic Data Mode” and “DF1 Master Mode”. Note that the protocol mode only specifies the basic communication model, not the actual subnetwork protocol.

- **Master Mode**

In this mode, the module acts as a master on the subnetwork, and the serial communication takes place in a query-response fashion. The nodes on the network are not permitted to issue messages unless they have been addressed by the module first.

See also “Master Mode” on page 19.

- **Generic Data Mode**

In this mode, there is no master-slave relationship between the subnetwork nodes and the module; any node on the subnetwork, including the Anybus Communicator, may spontaneously produce or consume messages.

See also “Generic Data Mode” on page 20.

- **DF1 Master Mode**

In this mode, the module acts as a master on the subnetwork, using the DF1 protocol. The serial communication takes place in a query-response fashion.

See also “DF1 Protocol Mode” on page 76.

2.3.2 Protocol Building Blocks

The following building blocks are used in Anybus Configuration Manager to describe the subnetwork communication. How these blocks apply to the three protocol modes will be described later in this document.

- **Node**

A node represents a single device on the subnetwork. Each node can be associated with a number of transactions, see below.

- **Transaction**

A “transaction” represents a complete serial telegram, and consists of a number of frame objects (below). Each transaction is associated with a set of parameters controlling how and when to use it on the subnetwork.

- **Commands**

A “command” is simply a predefined transaction stored in a list in the Anybus Configuration Manager. This simplifies common operations by allowing transactions to be stored and reused.

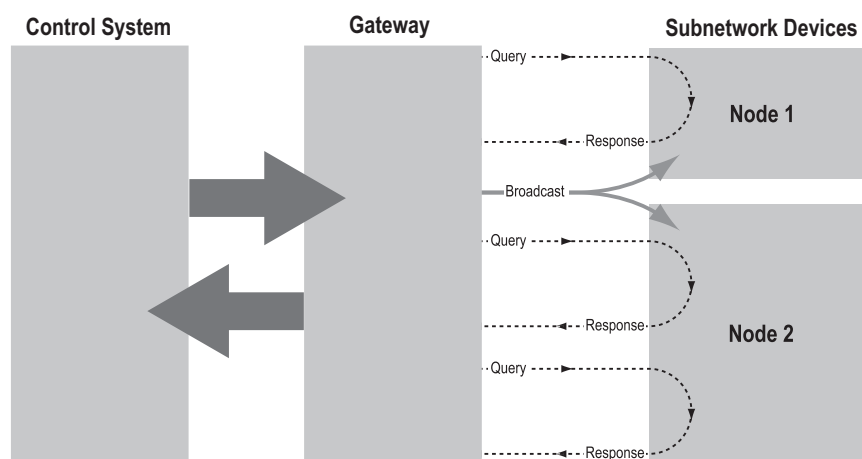
- **Frame Object**

“Frame objects” are low level entities used to compose a transaction (see above). A frame object can represent a fixed value (a constant), a range of values (limit objects), a block of data or a calculated checksum.

2.3.3 Master Mode

In this mode, the communication is based on a query-response scheme; when the Anybus Communicator issues a query on the subnetwork, the addressed node is expected to issue a response to that query. Nodes are not permitted to issue responses spontaneously, i.e. without first receiving a query.

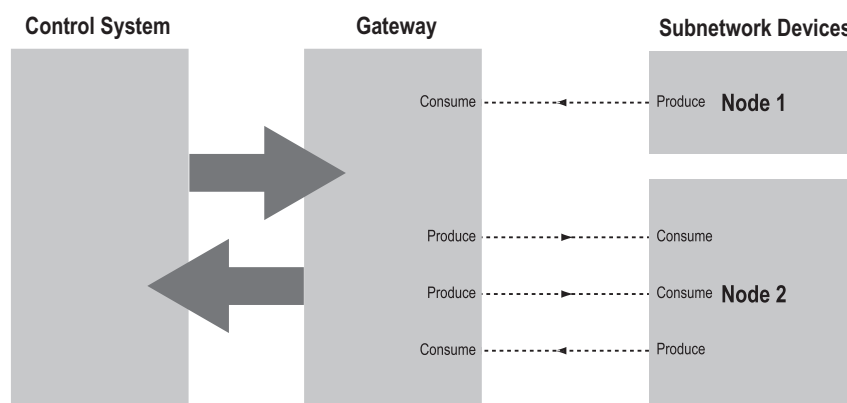
There is one exception to this rule; the broadcaster. Most protocols offer some way of broadcasting messages to all nodes on the network, without expecting them to respond to the broadcasted message. This is also reflected in the Anybus Communicator, which features a dedicated broadcaster node.



In Master Mode, Anybus Configuration Manager comes preloaded with most commonly used Modbus RTU commands, which can conveniently be reached by right-clicking on a node in the Anybus Configuration Manager and selecting “Insert New Command”. Note however that this does not in any way prevent other protocols based on the same query-response message-scheme to be implemented.

2.3.4 Generic Data Mode

In this mode, there is no master-slave relationship between the nodes on the subnetwork and the Anybus Communicator. Any node, including the module, may spontaneously produce or consume a message. Nodes do not have to respond to messages, nor do they have to wait for a query in order to send one.



In the figure above, the Anybus Communicator “consumes” data that is “produced” by a node on the subnetwork. This “consumed” data can then be accessed from the higher level network. This also works the other way around; the data received from the higher level network is used to “produce” a message on the subnetwork to be “consumed” by a node.

2.3.5 DF1 Master Mode

Please refer to “DF1 Protocol Mode” on page 76.

2.4 PROFINET IO

2.4.1 General

The PROFINET IO interface provides PROFINET IO Soft Real-Time Communication. PROFINET is the open Industrial Ethernet standard for Automation from PROFIBUS International.

Supported Features

- Soft Real-Time (RT) communication
- Cyclic data exchange (10 ms cycle time)
- Acyclic Data exchange (Record Data Requests)
- Up to 64 slots / 1 subslot
- Up to 512 bytes of I/O in each direction
- TCP/IP Configuration via DCP (Discovery and Configuration Protocol)

2.4.2 I/O Configuration

PROFINET makes a distinction between fast cyclical data, a.k.a. “IO data”, and acyclical data, called “Record Data”. By default, all data in the input and output data areas are exchanged as IO data. It is however possible to specify how much data to exchange as IO data, and how much data to exchange using acyclic Record Data read/write requests.

On PROFINET, the IO data is built up by I/O modules. In the case of the ABC, the actual I/O module configuration is adopted from the I/O controller/supervisor, provided that the total I/O sizes specified by the IO controller does not exceed the sizes specified in the Anybus Configuration Manager.

For information about how the IO and Record Data relates to the input and output data areas, see “Data Representation (IO Data & Record Data)” on page 22.

2.4.3 GSDML File

On PROFINET, all devices are associated with a GSDML file. The GSDML file is the equivalent of the PROFIBUS GSD file, and is based on the EXtensible Markup Language (XML).

This file holds information about the device (in this case the Anybus Communicator), its features, and possible I/O configurations. The latest version of the GSDML file for the Anybus Communicator can be downloaded from the HMS website, “www.anybus.com”.

2.4.4 Data Representation (IO Data & Record Data)

As mentioned previously, the actual I/O configuration is determined by the IO Controller. The modules are mapped to the input and output data areas in the order of their slot number.

Example:

In this example, the I/O Sizes for the ABC has been set to the following values:

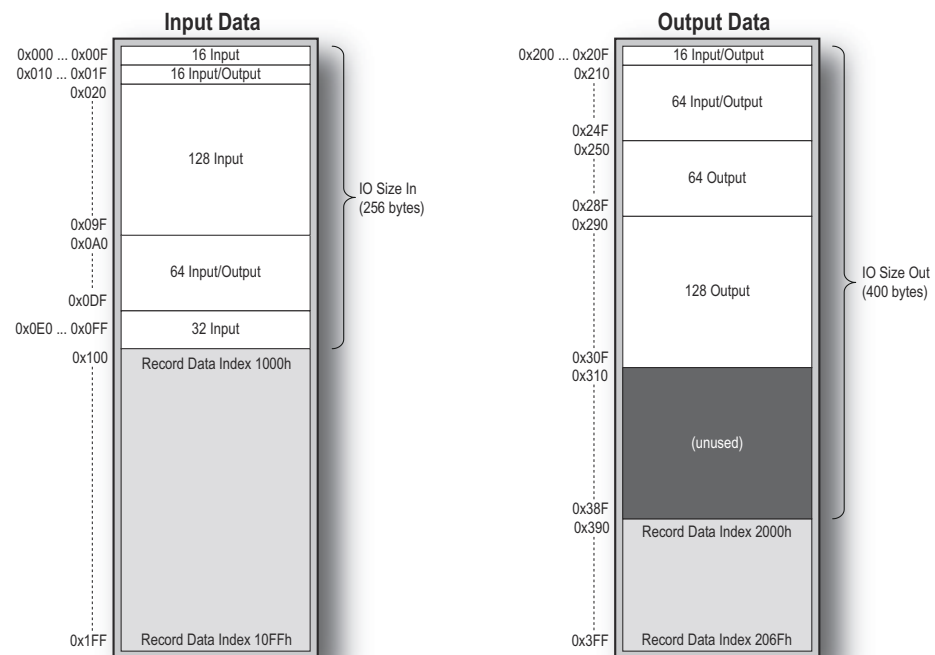
IO Size In= 256 bytes (0x0100)

IO Size Out= 400 bytes (0x0200)

The following modules are specified in the IO Controller:

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

Resulting memory layout:



Note the “unused” part of the output data area. The reason for this is that only 272 bytes (128+64+64+16) are actually used in the I/O module configuration, although IO Size Out is set to 400 bytes.

2.5 Modbus-TCP (Read-Only)

2.5.1 General

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

2.5.2 Data Representation (Modbus-TCP Register Map)

The following function codes are implemented:

Modbus Function	Function Code	Associated with	No. of I/O:s or data points per command
Read Input Registers	4	Input Data area (0x000....0x1FF)	1 - 125 registers
Read Multiple Registers	3	Output Data area (0x200...0x3FF)	1 - 125 registers

The Input & Output Data areas are mapped to Modbus registers as follows:

Register Type	Register #	Memory Location	Area	Comments
Input Registers (3xxxx)	0x0000	0x000...0x001	Input Data area	(Status Register)
	0x0001	0x002...0x003		-
	0x0002	0x004...0x005		-
	0x0003	0x006...0x007		-
		-
	0x00FF	0x1FE...0x1FF		-
Output Registers (4xxxx)	0x0000	0x200...0x201	Output Data area	(Control Register)
	0x0001	0x202...0x203		-
	0x0002	0x204...0x205		-
	0x0003	0x206...0x207		-
		-
	0x00FF	0x3FE...0x3FF		-

Note: If enabled, the control and status registers occupies input register 0x0000 and output register 0x0000.

2.5.3 Supported Exception codes

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

3. File System

3.1 General

General

The Anybus Communicator features a built-in file system, which is used to store information such as web files, network communication settings, e-mail messages etc.

Storage Areas

The file system consists of the different storage areas:

- **Non-volatile area (approx. 2 Mb)**
This section is intended for static files such as web files, configuration files etc.
- **Volatile area (approx. 1 Mb)**
This area is intended for temporary storage; data placed here will be lost in case of power loss or reset.

Important Note:

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

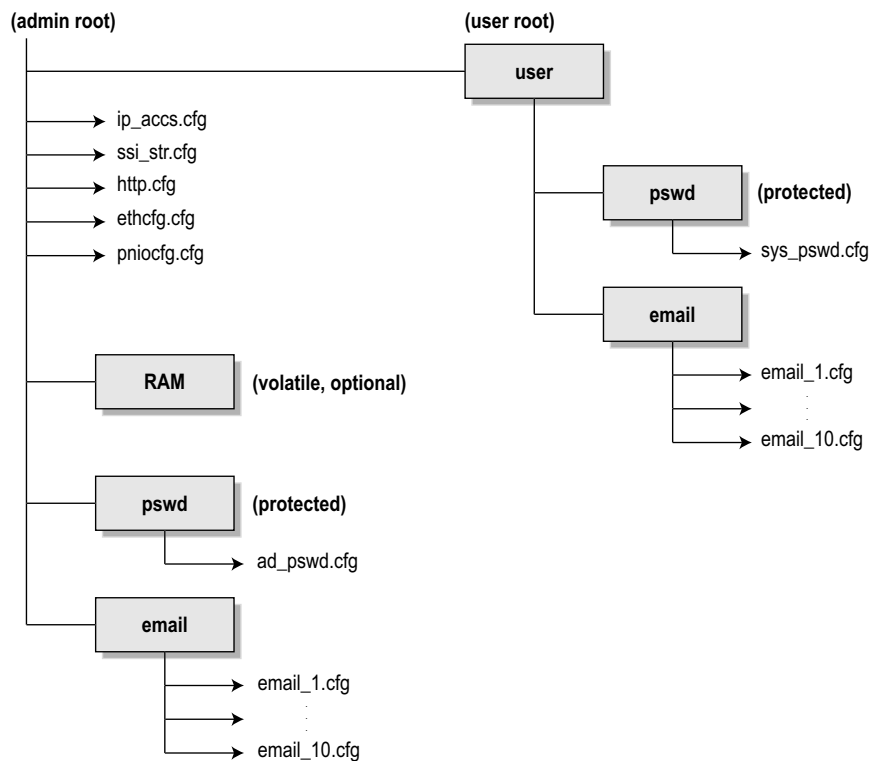
The following operations will erase one or more FLASH segments:

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

Conventions

- \ (backslash) is used as a path separator
- File paths originate from the system root, and as such must begin with \
- File paths must not end with \
- 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)
- File paths 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

3.2 File System Overview



3.3 System Files

The file system 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. Note that some of these files may also be altered by the Anybus Communicator itself, e.g. when using SSI (see “Server Side Include (SSI)” on page 35).

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 later in this document.

The contents of the above files can be redirected:

Example:

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

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

Note: Any directory in the file system can be protected from web access by placing the file web_accs.cfg in the directory, see “Authorization” on page 33.