Anybus® X-gateway Network Interface Addendum

EtherNet/IP™ Adapter
Modbus®-TCP Server

Doc: HMSI-27-251
Rev: 2.10

HMS 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
www.anybus.com

Important User Information

This document is intended to provide a good understanding of the functionality offered by the network interface described here.

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

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. Modbus® is a registered trademark of Modbus Organization, Inc. ODVA™ and EtherNet/IP™ are trademarks of ODVA, 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 X-gateway Network Interface Addendum
EtherNet/IP Adapter - Modbus-TCP Server

HMSI-27-251

Table of Contents

Chapter 4 Web Interface

Chapter 5 Modbus-TCP Server

Chapter 6 Ethernet/IP

Chapter 7 FTP Server

Chapter 8 Telnet Server

Appendix A System Files

Appendix B Server Side Include (SSI) Functionality

Appendix C Ethernet Transport Provider

Appendix D Technical Specification

P. About This Document

P.1 How To Use This Document

This document describes network specific features and procedures needed when operating the Ethernet interface for the Anybus X-gateway. For general information and operating instructions for the Anybus X-gateway, please consult the Anybus X-gateway User Manual.

The reader of this document is expected to be familiar with local area networks and communication systems in general.

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

P.2 Related Documents

Document	Author
Anybus X-gateway User Manual	HMS
Anybus-S EtherNet/IP Fieldbus Appendix	HMS
Anybus X-gateway EtherNet/IP Adapter Interface Network Installation Sheet	HMS
Anybus X-gateway Modbus-TCP Server Interface Network Installation Sheet	HMS
Open Modbus-TCP Specification, v 1.0	Schneider Electric

P.3 Document History

Summary of Recent Changes (2.00... 2.10)

Change	Page(s)
New document title	All
Added trademark acknowledgments	2
Modified list of related documents	7
Added connector assignments, replaced graphics	10
Added description of fieldbus-specific ACM settings	15
Added compliance info	56

Revision List

Revision	Date	Author	Chapter	Description
1.00	2004-09-28	PeP	All	First release
1.01	2004-10-05	PeP	All	Minor adjustments & corrections
1.02	2007-03-05	PeP	C	Added info about transport provider + misc. minor updates
1.03	2007-05-07	PeP	A	Minor update
1.04	2007-11-23	PeP	13, 18	Added missing information + misc. adjustments
1.05	2008-05-21	PeP	B	Minor update
1.06	2010-01-22	KeL	3, 5	Minor update
1.07	2010-05-16	KeL	1, 3	Minor update
1.08	2011-07-26	KeL	P, 5	Minor update
2.00	May 2014	SDa	Multiple	New hardware and ACM, new Doc. ID
2.10	March 2015	ThN	All	Misc. changes and additions, new document title

P.4 Conventions & Terminology

The following conventions are used throughout this manual:

- Numbered lists provide sequential steps
- Bulleted lists provide information, not procedural steps
- The term 'gateway' is used when referring to the Anybus X-gateway
- The term 'Ethernet Interface' refers to the Anybus X-gateway Ethernet 2-port slave interface
- The term 'user manual' refers to the Anybus X-gateway User Manual.
- Hexadecimal values are written in the format NNNNh, where NNNN is the hexadecimal value.
- 16/32 bit values are generally stored in Motorola (big endian) format unless otherwise stated.

P.5 Sales and Support

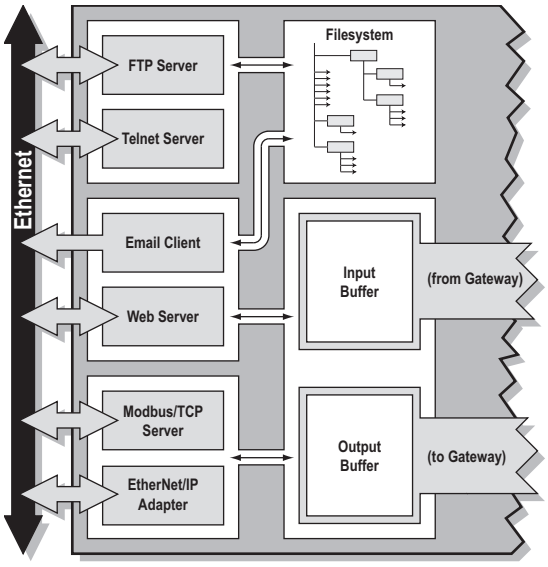
For general contact information and global support, please visit www.anybus.com.

1. About the Ethernet 2-Port Slave Interface

1.1 General Description

The 2-port Ethernet Slave interface implements a multi purpose communication solution for the Anybus X-gateway platform, offering industrial protocol support, as well as web and email capabilities.

The interface exchanges data via two data buffers a.k.a. the Input and Output Buffers. These buffers can be accessed via a built in web server, or via industrial protocols such as Modbus-TCP and EtherNet/IP. A built in email client can send email messages triggered by a data event.



Dynamic content capabilities allows data from the input/output buffers to be monitored on web pages, or included in email messages.

- **Input Buffer**

  This buffer holds data *from* the gateway, i.e. data *from* another network.

- **Output Buffer**

  This buffer holds data that shall be sent *through* the gateway *to* another network.

1.2 Features

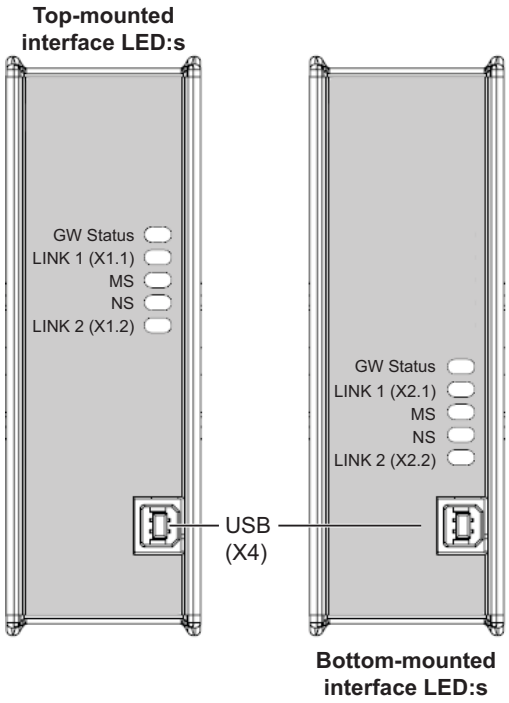
- 2 EtherNet/IP ports available simultaneously. 10/100 Mbit operation in full or half-duplex.
- Supports shielded (FTP) and unshielded (UTP) cables
- Flexible file system providing both volatile and non-volatile storage areas
- Per-user security framework
- Quality of Service (QoS) available
- Device Level Ring (DLR) functionality available (announce-based ring participant)
- Integrated FTP server provides easy file management using standard FTP clients.
- Server Side Include (SSI) capability
- Web server with dynamic data capability
- Email client with dynamic data capability
- Event-triggered email handling
- DNS Capability
- DHCP/ARP/Anybus IPconfig (HICP) support

1.3 External View

1.3.1 Ethernet Interface Status LEDs

LED	Colour	Indication
GW Status	See the X-gateway User Manual	
LINK 1	Red (flashing)	10 Mbit/s
	Green (flashing)	100 Mbit/s
	Off	Link not detected
MS[a]	Green	Normal operation
	Green (flashing)	Standby, not yet configured
	Red	Major fault, unrecoverable
	Red (flashing)	Minor fault, recoverable
	Alt. Red/Green	Hardware self test
	Off	No power
NS[a]	Green	EtherNet/IP connection(s) established
	Green (flashing)	No EtherNet/IP connections
	Red	Duplicate IP address
	Red (flashing)	Connection timeout
	Alt. Red/Green	Hardware self test
	Off	No power or no IP address set
LINK 2	Red (flashing)	10 Mbit/s
	Green (flash)	100 Mit/s
	Off	Link not detected

a. Not used for Modbus-TCP



1.3.2 Connectors and Switches

Ethernet Connectors (X1.1+1.2, X2.1+2.2)

See "Ethernet Connectors Pinout (RJ45)" on page 56.

Gateway Power Connector (X3)

See the X-gateway User Manual.

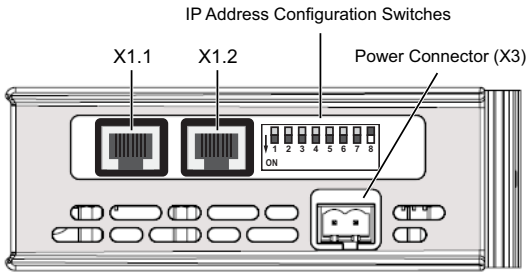
USB Gateway Config Connector (X4)

See the X-gateway User Manual.

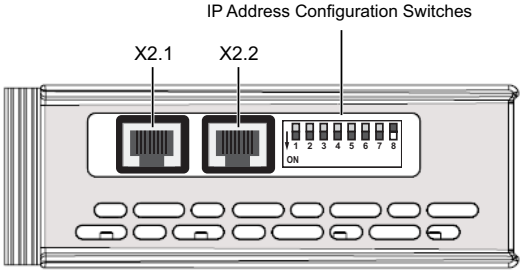
IP Address Configuration Switches

See "Configuration Switches" on page 13.

Top-mounted interface



Bottom-mounted interface



2. File System

2.1 General

Web content, email messages, and configuration data are stored in files in a built-in file system. The file system is a fixed-size storage area with a hierarchical directory structure. Files can be grouped in directories for increased readability, and a security framework protects important files from unauthorized access.

Conventions & Limitations

- **Case Sensitivity**

  The file system is case sensitive, i.e.'Anybus.txt' does not equal 'Anybus.TXT'.

- **File name / Path name length**

  File names can be a maximum of 48 characters long. Path names can be 256 characters in total, file name included.

- **Free Space**

  Approximately 1.4 MB non-volatile (FLASH), and 1MB volatile.

- **Data Corruption**

  Each FLASH segment can be rewritten approximately 1,000,000 times before data corruption. One or more flash segments will be erased upon deleting, renaming or moving a file, appending data to an existing file, or formatting the file system.

2.2 Security Framework

The file system features two security levels; *Admin* and *Normal*. The security level is set per user, unless running in Global Admin Mode (see below).

- **Admin Users**

  Admin users have unrestricted access to the file system through FTP and Telnet. User accounts for Admin users are defined in the file 'ad_pswd.cfg'.

- **Normal User**

  Normal users are restricted to the \user\ directory and it's subdirectories, i.e. when a normal user connects via FTP or telnet, this will be their root directory. User accounts for normal users are defined in the file 'sys_pswd.cfg'.

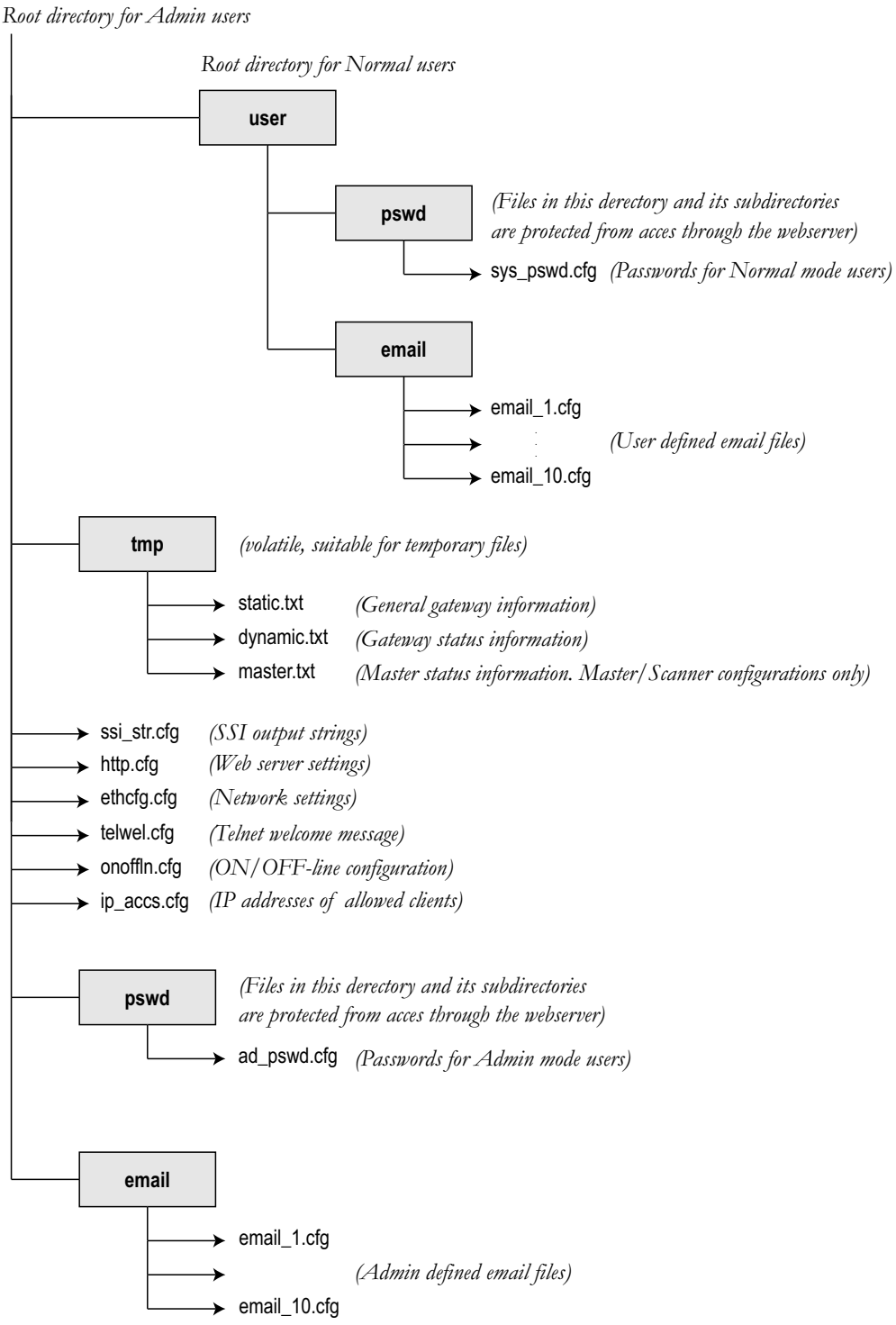
Files in the file system can be protected from web access through username/password authorization, see "Password Files" on page 40 and "web_accs.cfg" on page 40. It is also possible to configure which IP addresses and what protocols that are allowed to connect to the ethernet interface, see "ip_accs.cfg" on page 36.

Global Admin Mode

If no admin password file (See "Password Files" on page 40) is found, the interface will run in Global Admin Mode; i.e. all users will have Admin access rights. No login is needed for Telnet, and the FTP server accepts any username/password combination. Global Admin Mode is primarily intended for product configuration and development.

2.3 Structure

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



For more information about the configuration files shown above, see "System Files" on page 35.

3. Installation and Configuration

3.1 General Information

The Ethernet interface offers several ways of configuring the network settings:

- Configuration Switches (see below).
- Configuration File (See "ethcfg.cfg" on page 35).
- Anybus IPconfig (HICP client) (See "Anybus IPconfig (HICP)" on page 14).
- Via the Web Interface (See "Web Interface" on page 16).
- ARP (See "Address Resolution Protocol (ARP)" on page 14).
- EtherNet/IP (see "TCP/IP Interface Object, Class F5h" on page 27).

If the configuration switches are set to a value other than 0 (zero), the network configuration will be determined by the value of the configuration switches, see below.

If the switches are set to 0 (zero), the network configuration will be loaded from the file 'etccfg.cfg'. If this file cannot be found, the Ethernet interface will attempt to retrieve its network settings from a DHCP/BootP server. If no server is found, the interface will indicate an error on the Network Status LED. (Note that the network configuration can still be set using ARP or HICP). The interface supports DHCP Reboot, i.e. it will ask the DHCP/BootP for the IP address stored in the configuration file. If the address is free to use, it will be assigned to the Ethernet interface. If not, it will be assigned a new address.

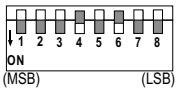
3.2 Configuration Switches

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

If the switch is set to a value between 1-254[1], the interface will use the settings described below.

```
IP address:       192.168.0.n (n reflects the binary value of the switches)
Subnet mask:      255.255.255.0 (fixed)
Gateway address:  0.0.0.0 (fixed)
```

Example:



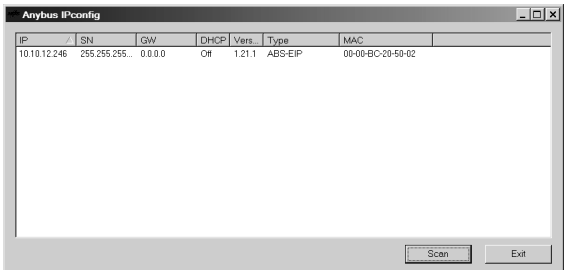
The switches are set to 00010100 (20 decimal). The IP address of the Ethernet interface will be 192.168.0.20

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

1. 255 is used for local broadcasts.

3.3 Anybus IPconfig (HICP)

HMS IP Configuration Protocol, or HICP, provides a simple way to configure Anybus-based Ethernet interfaces using a user-friendly software utility from HMS called Anybus IPconfig.



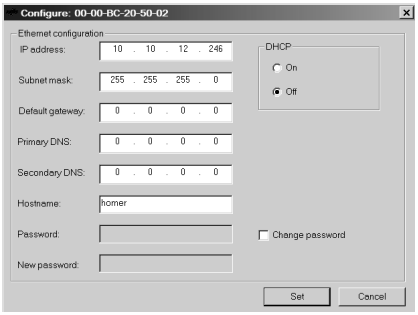
During startup, the utility scans the network for Anybus-based Ethernet interfaces. If needed, the network can be rescanned by clicking **Scan**.

To edit the settings of the Ethernet interface, double click on its entry in the list.

A window will appear, containing the IP configuration and password settings. The IP configuration can be edited directly.

To enter a password, check the **Change password** box, and enter the password under **New password**. (If the interface has already been assigned a password, first enter the old password under **Password**.)

When finished, click **Set**. The new IP configuration will now be stored in the file 'ethcfg.cfg'.



3.4 Address Resolution Protocol (ARP)

The IP address can be changed during runtime using the ARP command from a PC. This method can be used to re-configure Ethernet interfaces that have already been configured, or even to reconfigure interfaces outside the host's subnet.

Below is an example of how to change the IP address from a Microsoft Windows™ command prompt::

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

The 'arp -s' command will store the IP and MAC (printed on the back of the Anybus-X gateway) addresses in the PC's ARP table. When the 'ping' command is executed, the PC sends this information to the interface using the MAC address. The interface detects that it was addressed with the correct MAC address and adopts the IP settings sent by the PC. The new settings are then stored in the file 'ethcfg.cfg'.

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

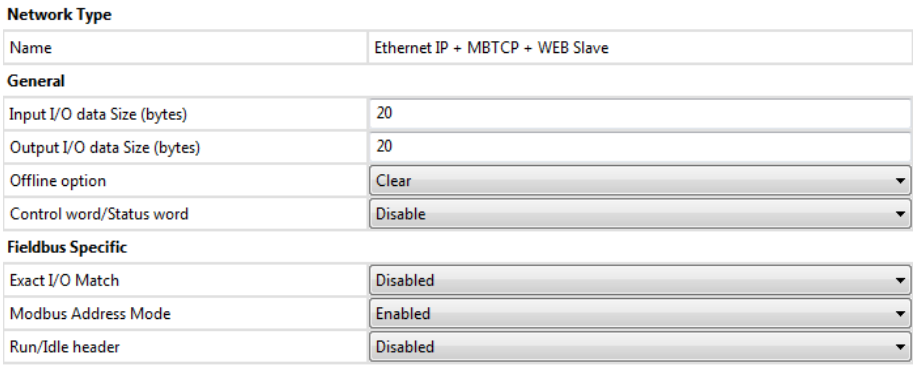
Example:

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

3.5 Gateway Config Interface

The gateway and the Ethernet slave interface may be configured by using the software tool **Anybus Configuration Manager** (ACM), which is available from www.anybus.com.

When ACM is connected to the gateway via the USB configuration connector, the following settings are available:



Setting	Description
Input I/O data Size (bytes)	Specifies the amount of Input I/O Data to exchange on Ethernet.
Output I/O data Size (bytes)	Specifies the amount of Output I/O Data to exchange on Ethernet.
Offline option	Determines what action to perform if the network goes offline. The gateway can either freeze (keep the current value) or clear (set to zero) the data from the network that has gone offline.
Control word/Status word	Enables/disables representation of the Control/Status word on Ethernet.
Exact I/O Match	If enabled, the X-gateway only accepts EtherNet/IP IO connection requests which exactly matches the configured I/O sizes.
Modbus Address Mode	Specifies the layout of the Modbus register map.
Run/Idle header	Enables/disables the use of a run/idle header. If the header is enabled and indicating *Idle*, the X-gateway will stop copying data and use the 'Offline options' settings instead.

See also...

- The Anybus X-gateway User Manual, for full details on using ACM.
- The online help in ACM, for further help on the available settings.
- "Modbus-TCP Server" on page 18
- "Ethernet/IP" on page 22
- "Assembly Object, Class 04h" on page 24

4. Web Interface

4.1 General

The Ethernet interface features a fast full featured web server with SSI capabilities. A default web interface provides access to most common options via any standard web browser. The web interface is, however, fully customizable and can be adapted to fit a particular product.

The default web interface differs slightly depending on network type (i.e. slave, master, fieldbus type, etc). Some basic functions are however, essentially the same.

- **Device Diagnostics**[1]

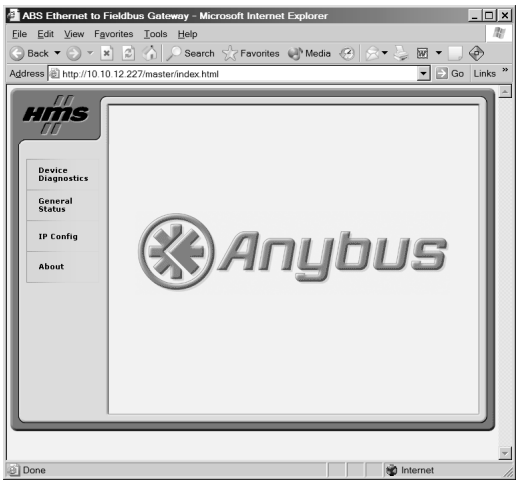
  These pages are different depending on network type, and holds diagnostic information about the network interface. For more information, see 4-17 "Device Diagnostics (Masters/Scanners only)".

- **General Status**

  This page provides an overview of the gateway initialisation parameters and general gateway diagnostics (these values corresponds to the values set using the gateway config interface).

- **IP Config**

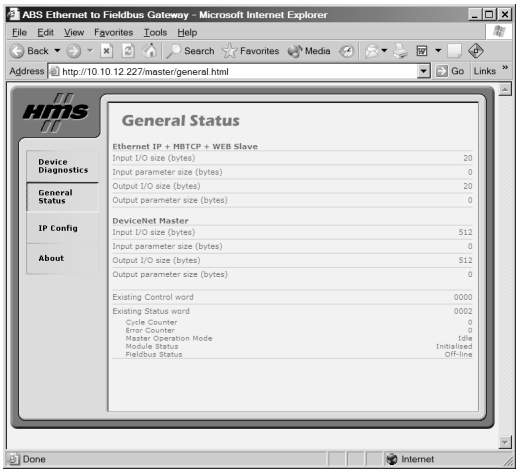
  This page holds the current TCP/IP settings, DNS configuration, and SMTP server settings.



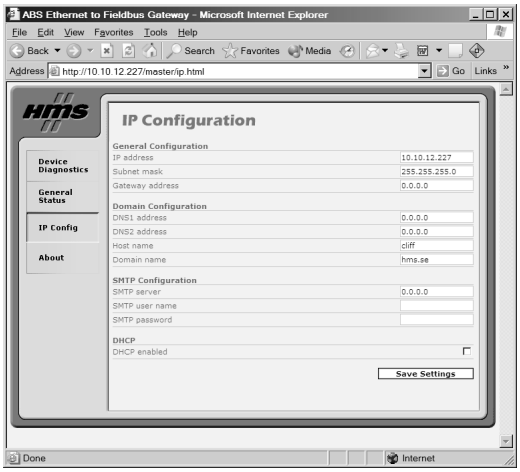
Start Page

- **About**

  This page holds the software version numbers and serial numbers of the different components of the gateway. This page also holds the MAC-ID of the ethernet interface.



General Status Page



IP Configuration Page

1. The Device Diagnostic pages are only available on network master/scanner interfaces.

4.2 Device Diagnostics (Masters/Scanners only)

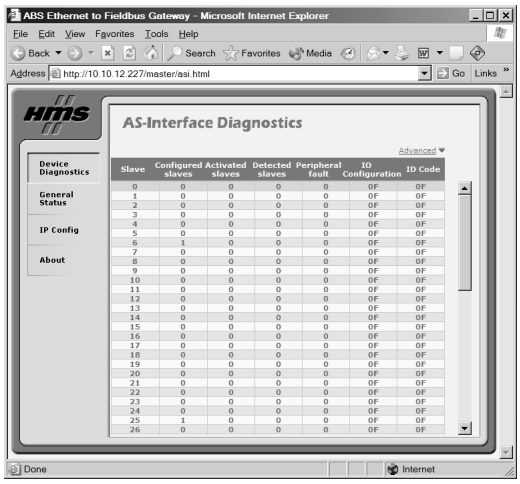
These pages are unique to each network type, and is only available on network master interfaces. The diagnostic pages for each supported network are described below.

4.2.1 AS-I Master Interface

The diagnostic page for this device holds the list of configured slaves, activated slaves, detected slaves, peripheral fault, and the I/O configuration and ID code for each AS-Interface slave.

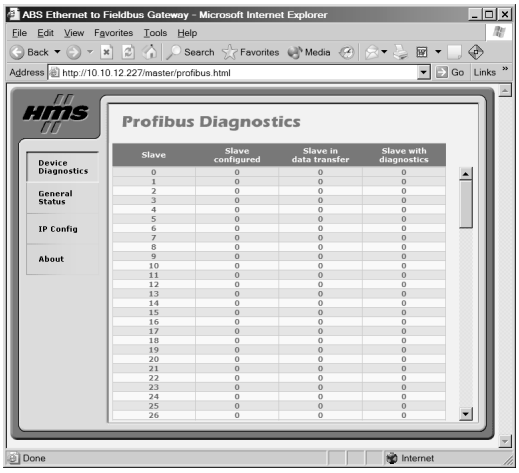
Additional advanced diagnostics can be viewed by clicking 'Advanced' link in the upper right corner.

For more information about how these diagnostic figures shall be interpreted, consult the Anybus-X AS-I Network Interface Addendum.



AS-Interface Diagnostic (Standard)

4.2.2 Profibus Master Interface



Profibus Master Diagnostic Page

The diagnostic page for this device holds the current status of each slave on the Profibus network.

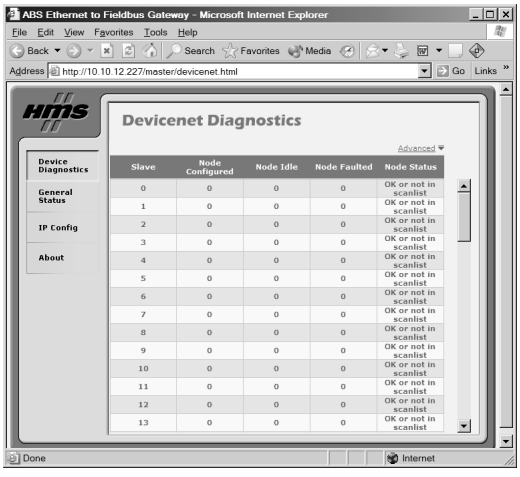
For more information about how these diagnostic figures shall be interpreted, consult the Anybus-X Profibus Master Network Interface Addendum.

4.2.3 DeviceNet Scanner Interface

The diagnostic page for this device holds the current status of each node on the DeviceNet network.

Additional advanced diagnostics can be viewed by clicking 'Advanced' link in the upper right corner.

For more information regarding how to interpret these diagnostic figures, consult the Anybus-X DeviceNet Scanner Network Interface Addendum.



DeviceNet Scanner Diagnostic Page

4.2.4 EtherNet/IP Scanner Interface

The diagnostic page for this interface displays the number of Configured, Active and Erroneous connections.

5. Modbus-TCP Server

5.1 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 access to the Input- and Output Buffers via a subet of the functions defined in the Modbus-TCP specification. While this means that it can communicate with devices that comply with this protocol, it does not mean that all services available in the Modbus-TCP protocol specification are supported.

Modbus-TCP messages are exchanged through TCP port No. 502, and the Modbus TCP server can handle a maximum of 8 simultaneous connections. For detailed information regarding the Modbus-TCP protocol, consult the Open Modbus Specification.[1]

5.2 Addressing Modes

The Input- and Output Buffers are mapped to 16-bit Modbus registers on the Modbus-TCP side. The Ethernet interface features two modes of representation, which affects the Modbus register map.

- **Modbus Address Mode (Default)**

  In this mode, the Input- and Output Buffers are mapped to different Modbus register types.

  Note that coil addressing is not possible in this mode.

  See also...

  - "Data Representation in Modbus Address Mode" on page 20

- **Anybus Address Mode**

  Compared to Modbus Addressing Mode, this mode allows data to be addressed in a more flexible manner. Note however that several function codes can be used to access the very same data. While this may initially appear confusing, it does allow data to be manipulated in ways not possible in Modbus Addressing Mode (e.g. it is possible to manipulate individual bits of a register by accessing coils associated with the same memory location).

  See also...

  - "Data Representation in Anybus Address Mode" on page 21

Note: Modbus Addressing Mode is enabled by setting the 'Modbus Address Mode'-setting in the Gateway Config Interface to 'Enable'. Anybus Addressinge Mode is enabled by setting this parameter to 'Disable'.

See also...

- "Gateway Config Interface" on page 15

1. Open Modbus-TCP Specification, v. 1.0, Schneider Electric

5.3 Modbus Exception Codes

With the exception of broadcast messages, the Modbus Master expects a normal response upon sending a Query. If no response is returned (e.g. due to a transmission error), this will eventually trigger a timeout condition in the Master.

If the server receives a Query but is for some reason unable to process it (i.e. the Modbus master tries to access a non-existent Register), an exception response is returned to inform the master of the nature of the error.

The following exception codes may be returned by the server:

#	Meaning	Description
01h	Illegal Function	The Query contains an illegal or unsupported function call
02h	Illegal Data Address	The Query contains an illegal data address
03h	Illegal Data Value	The Query contains invalid data

5.4 Data Representation in Modbus Address Mode

5.4.1 Supported Function Codes

The following function codes can be used in this mode:

Modbus Function	Function Code	Direction	Associated with Buffer
Read Holding Registers	3	Gateway to Modbus	Output Buffer
Read Input Registers	4		Input Buffer
Write Single Register	6	Modbus to Gateway	Output Buffer
Force Multiple Registers	16		
Mask Write Register	22		

5.4.2 Input Register Map

Each Input Register corresponds to two bytes in the Input Buffer as follows:

Register #	Buffer	Location in Buffer	Comments
1	Input Buffer	0x000... 0x001	Applicable Modbus functions: - Read Input Registers
2		0x002... 0x003	
3		0x004... 0x005	
4		0x006... 0x007	
5		0x008... 0x009	
6		0x00A... 0x00B	
...		...	
255		0x1FC... 0x1FD	
256		0x1FE... 0x1FF	

5.4.3 Holding Register Map

Each Holding Register corresponds to two bytes in the Output Buffer as follows:

Register #	Buffer	Location in Buffer	Comments
1	Output Buffer	0x000... 0x001	Applicable Modbus functions: - Read Holding Registers - Write Single Register - Force Multiple Registers - Mask Write Register
2		0x002... 0x003	
3		0x004... 0x005	
4		0x006... 0x007	
5		0x008... 0x009	
6		0x00A... 0x00B	
...		...	
255		0x1FC... 0x1FD	
256		0x1FE... 0x1FF	

5.5 Data Representation in Anybus Address Mode

5.5.1 Supported Function Codes

The following function codes can be used in this mode:

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

5.5.2 Coil & Register Map

The Input & Output Buffers are mapped to coils and registers as follows:

Register #	Coil #	Buffer	Location in Buffer	Comments
1	1... 16	Input Buffer	0x000... 0x001	Applicable Modbus functions: - Read Coil - Read Input Discretes - Read Holding Registers - Read Input Registers - Read/Write Registers
2	17... 32		0x002... 0x003	
3	33... 48		0x004... 0x005	
4	49... 64		0x006... 0x007	
...	...		...	
255	4065... 4080		0x1FC... 0x1FD	
256	4081... 4096		0x1FE... 0x1FF	
257 ... 1024	4097... 4112 ... 16369... 16384	-	-	(reserved)
1025	16385... 16400	Output Buffer	0x000... 0x001	Applicable Modbus functions: - Read Coil - Read Input Discretes - Read Holding Registers - Read Input Registers - Write Coil - Write Single Register - Force Multiple Coils - Force Multiple Registers - Mask Write Register - Read/Write Registers
1026	16401... 16416		0x002... 0x003	
1027	16417... 16432		0x004... 0x005	
1028	16433... 16448		0x006... 0x007	
1029	16449... 16464		0x008... 0x009	
1030	16465... 16480		0x00A... 0x00B	
1031	16481... 16496		0x00C... 0x00D	
...	...		...	
1279	20449... 20464		0x1FC... 0x1FD	
1280	20465... 20480		0x1FE... 0x1FF	

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

Note 2: Coils are mapped MSB first, i.e. coil 0 corresponds to bit 15 of register 0.

6. Ethernet/IP

6.1 General

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

Implemented Objects

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

The following vendor specific objects are implemented:

Class	Name	Contents
01h	Identity Object	Hods general information and status about the interface.
04h	Assembly Object	Holds the Input and Output data buffers.
AAh	Diagnostic Object	Contains diagnostic information about the ethernet interface.
F5h	TCP/IP Interface Object	Holds the IP settings of the interface.
F6h	Ethernet Link Object	Holds the low level communication properties of the interface.

I/O Data Representation

The Input & Output buffers can be accessed from EtherNet/IP via the Assembly Object, instances 64h (Input) and 96h (Output). For more information, see "Assembly Object, Class 04h" on page 24.

This data can also be accessed via Modbus-TCP, the email client, or the built in web server.

6.2 Identity Object, Class 01h

Services

Class services: Get Attribute All
Get Attribute Single

Instance services: Get Attribute All
Get Attribute Single
Reset

Class Attributes

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

Instance Attributes, Instance 01h

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

Status AttributeExtended Device Status

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

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

Reset Service

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

Reset Type	Action
Power Cycling Reset (Type 0)	This will cause the interface to emulate a power cycling reset.
Out of box reset (Type 1)	This will cause the interface to delete the configuration file ‘ethcfg.cfg’ and reset.

6.3 Assembly Object, Class 04h

Services

Class services: Get Attribute Single

Instance services: Get Attribute Single
Set Attribute Single

Description

The Assembly Object holds all I/O data used for I/O connections.

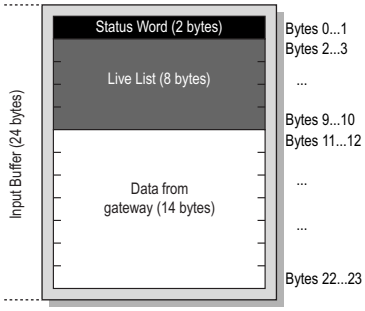
Class Attributes

#	Access	Name	Type	Value	Description
1	Get	Revision	UINT	0002h	Revision 2
2	Get	Max Instance	UINT	0096h	The highest instance no.

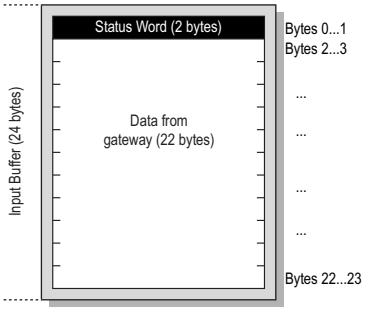
Instance Attributes, Instance 64h

#	Access	Name	Type	Value	Description
3	Get	Input Buffer	Array of BYTE	-	Mapped to Input Buffer

Depending on the type of gateway and how it has been set up to operate, up to 10 bytes (bytes 0...9) may be occupied by the Status Word and the Live List, see below. (For further information about the Status Word and the Live List, consult the main user manual)



Example A:
I/O Data Size = 24 bytes
Live List = Enabled
Control & Status Word = Enabled

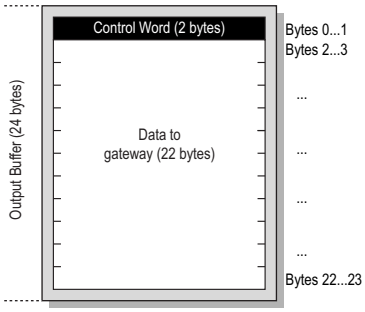


Example B:
I/O Data Size = 24 bytes
Live List = Disabled
Control & Status Word = Enabled

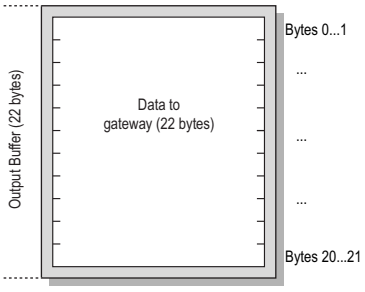
Instance Attributes, Instance 96h

#	Access	Name	Type	Value	Description
3	Set	Output Buffer	Array of BYTE	-	Mapped to Output Buffer

Depending on how the gateway is set to operate, the first 2 bytes (bytes 0...1) may be occupied by the Control Word, see below. (For more information about the Control Word, consult the X-gateway User Manual).



Example A:
I/O Data Size = 24 bytes
Control Word = Enabled



Example B:
I/O Data Size = 22 bytes
Control Word = Disabled

6.4 Diagnostic Object, Class AAh

Services

Class services: Get Attribute All

Instance services: Get Attribute Single

Description

This vendor specific object provides diagnostic information from the Ethernet interface.

Class Attribute

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

Instance Attributes, Instance 01h

#	Access	Name	Type	Description
01h	Get	Serial number	UDINT	Serial number
02h	Get	Vendor ID	UINT	Manufacturer Vendor ID
03h	Get	Fieldbus Type	UINT	Fieldbus Type
04h	Get	Software version	UINT	Interface software revision
07h	Get	Watchdog counter out	UINT	Counter, incremented every 1ms
0Ah	Get	Interface Type	UINT	Interface Type (0083h = Ethernet)
0Fh	Get	Input buffer size	UINT	Size of Input buffer (in bytes)
12h	Get	Output buffer size	UINT	Size of output buffer (in bytes)
18h	Get	MAC ID	Array of USINT	Ethernet MAC address of the interface (6 bytes)
19h	Get	IP Address	UDINT	Current IP address
1Ah	Get	Subnet mask	UDINT	Current subnet mask
1Bh	Get	Gateway address	UDINT	Current gateway address
1Ch	Get	SMTP server address	UDINT	SMTP server address
1Dh	Get	DHCP state	UDINT	0=DHCP enabled, 1=DHCP disabled
1Eh	Get	Bootloader version	UDINT	Bootloader firmware version
1Fh	Get	Application interface version	UINT	Application interface software version
20h	Get	Fieldbus software version	UINT	Fieldbus interface software version

6.5 TCP/IP Interface Object, Class F5h

Services

Class services: Get Attribute All
Get Attribute Single

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

Description

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

Class Attributes

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

Instance Attributes, Instance 01h

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

6.6 Ethernet Link Object, Class F6h

Services

Class services: Get Attribute All
Get Attribute Single

Instance services: Get Attribute All
Get Attribute Single

Description

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

Class Attributes

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

Instance Attributes, Instance 01h

#	Access	Name	Type	Value	Description
1	Get	Interface Speed	UDINT	10 or 100 (Mbps)	Actual communication speed
2	Get	Interface Flags	DWORD	-	-
3	Get	Physical Address	Array of 6 USINTS	MAC address	Ethernet MAC address

7. FTP Server

7.1 General

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

- **Normal users**

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

- **Admin users**

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

- **Global Admin Mode**

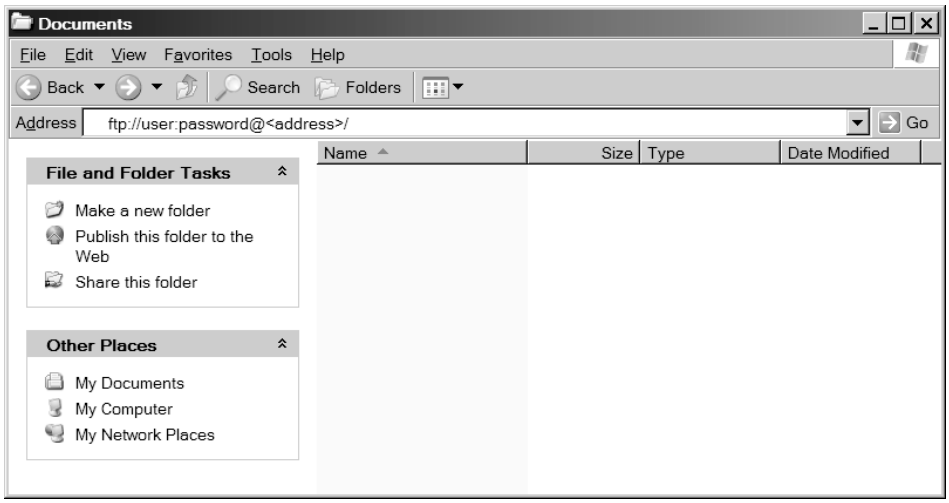
  Any username/password combination will be accepted. All users has unrestricted access to the file system, i.e. the root directory will be '\'.

For more information about the security framework, see 2-11 "Security Framework".

7.2 Accessing the FTP Server from Windows Explorer

To access the FTP server from Windows Explorer, open an explorer window, and type the following in the Address following:

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



Substitute 'user' and 'password' with a valid username and password combination, and replace <address> with the hostname or IP address of the ethernet interface.

Note: The default admin login is username: 'ABX' and password: 'FTPAccess'

8. Telnet Server

8.1 General

Using a Telnet client, the user can access the file system in the ethernet interface using a command line interface similar to the Microsoft Windows command prompt. Depending on security settings, different parts of the file system can be accessed by the user:

- **Normal users**

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

- **Admin users**

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

- **Global Admin Mode**

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

8.2 Accessing the Telnet Server from Windows

To access the telnet server using the telnet client provided with Windows, type the following in the command prompt:

```
telnet <address>
```

(Substitute <address> with the hostname or IP address of the ethernet interface)

The server will now prompt for a valid username password combination:

```
HMS Anybus-S Ethernet module
Login: <user>
Password: <password>
```

(Substitute <user> and <password> with a valid user/password combination)

If successful, you are now logged in and can enter commands using the built in command line interface.

```
Login OK (Admin mode)

\>
```

Note: The default admin login is username: 'ABX' (username) and password: 'FTPAccess'.

8.3 General Commands

8.3.1 admin

Syntax:

```
admin
```

Provided that the user can supply a valid admin username/password combination, this command enables admin access rights to normal users. (This command has no effect in Global Admin Mode).

8.3.2 help

Syntax:

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

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

```
General commands:

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

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

8.3.3 version

Syntax:

```
version
```

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

8.3.4 exit

Syntax:

```
exit
```

This command closes the Telnet session.

8.4 File System Operations

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

8.4.1 dir

Syntax:

```
dir [path]
```

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

8.4.2 md

Syntax:

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

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

8.4.3 rd

Syntax:

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

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

8.4.4 cd

Syntax:

```
cd [path]
```

Changes current directory.

8.4.5 format

Syntax:

```
format
```

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

8.4.6 del

Syntax:

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

Deletes a file.

8.4.7 ren

Syntax:

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

Renames a file or directory.

8.4.8 move

Syntax:

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

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

8.4.9 copy

Syntax:

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

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

8.4.10 type

Syntax:

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

Types the contents of a file.

8.4.11 mkfile

Syntax:

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

Creates an empty file.

8.4.12 append

Syntax:

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

Appends a line to a file.

8.4.13 df

Syntax:

```
df
```

Displays file system info.

8.5 Diagnostic Commands

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

8.5.1 arps

Syntax:

```
arps
```

Display ARP stats and table

8.5.2 iface

Syntax:

```
iface
```

Display net interface stats

8.5.3 sockets

Syntax:

```
sockets
```

Display socket list

8.5.4 routes

Syntax:

```
routes
```

Display IP route table

A. System Files

System files typically contain parameters that control the general behavior of the interface, or provide status information from the gateway. These files are regular ASCII files that can be edited with any text editor. Note that depending on security settings, some of these files may be inaccessible for normal users. Generally, the interface must be restarted in order for any changes in these files to take effect.

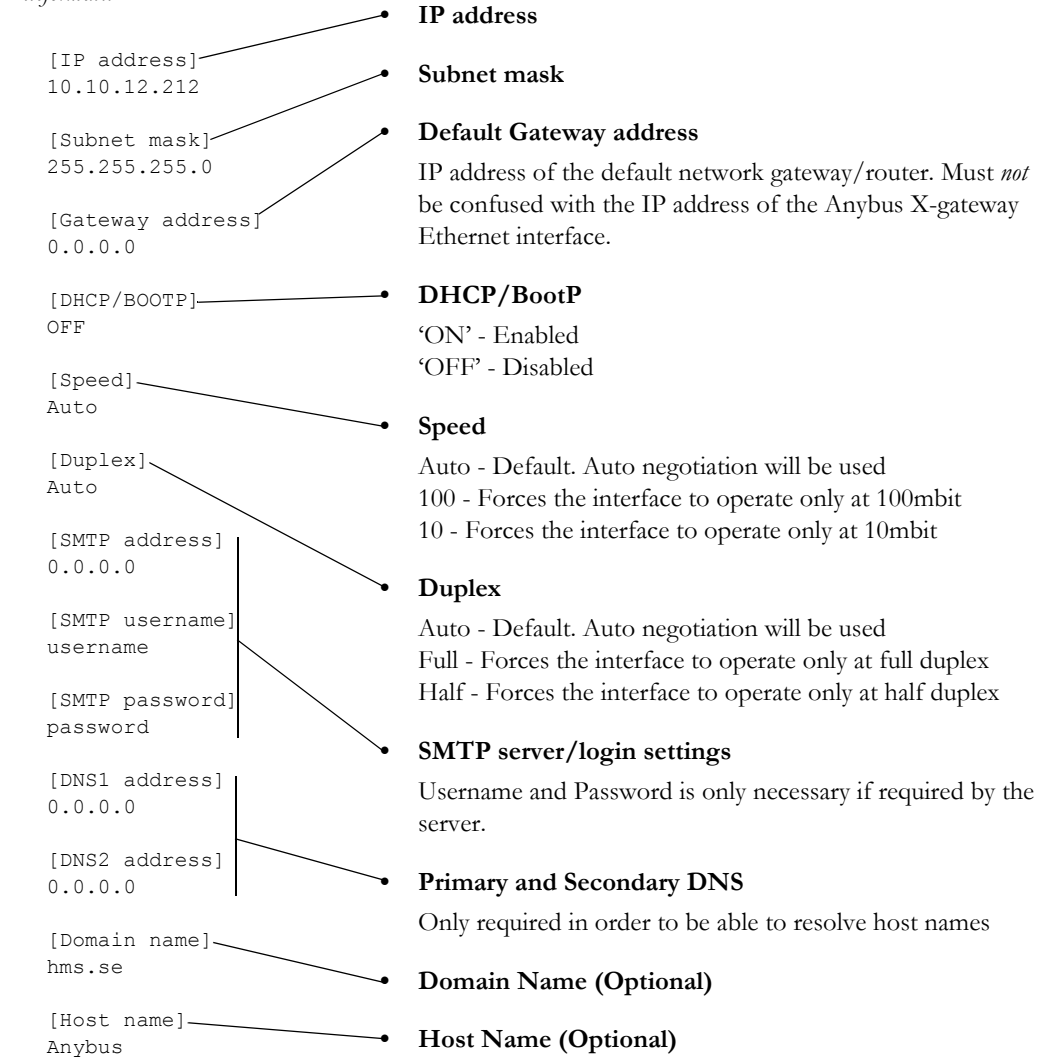
Note: It is very important to follow the exact syntax specifications for each configuration file, otherwise the interface may have problems interpreting it, which can result in faulty or unexpected behaviour.

A.1 Configuration files

A.1.1 ethcfg.cfg

This file contains the network configuration and is read by the interface at start up. The settings in this file may be affected by several mailbox- and SSI commands. For more information about network configuration see "Installation and Configuration" on page 13.

Fileformat::



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

Example:

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

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

A.1.2 ip_accs.cfg

This file specifies which IP addresses and protocols that are allowed to connect to the interface, and is structured as follows:

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

The allowed IP addresses are specified separately for each protocol under the corresponding header. The wildcard '*' can be used to allow series of IP addresses. If a protocol header is not given, the system will use the settings specified under the header 'All'. If the 'All' header is not given, the protocol will not accept any connections.

Example:

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

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

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

Example:

```
[File path]
\my_settings\ip_access_rights.cfg
```

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

IMPORTANT: *Do not under any circumstances enter address '0.0.0.0' for [All], since this will in effect prevent any external access to the scanner interface altogether. Failure to observe this will render the product unusable and require service from the HMS support office.*

A.1.3 onoffln.cfg

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

Example:

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

[Timeout]
10

[Commands]
3, 16, 23
```

ON/OFF-line trigger

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

Timeout

Value: Timeout value. A value of 10 equals 1000ms.

Commands (Optional)

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

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

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

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

Example:

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

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

A.1.4 telwel.cfg

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

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

Example:

```
[File path]
\my_settings\telnet_welcome_message.txt
```

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

A.1.5 http.cfg

By default, the built in web server recognizes the following file types based on their filename extension:

Content Type	File Extension(s)	Scanned for SSI
text/html	*.htm; *.html; *.shtm	Yes
image/gif	*.gif	No
image/jpeg	*.jpeg; *.jpg; *.jpe	No
image/x-png	*.png	No
application/x-javascript	*.js	No
text/plain	*.bat; *.txt; *.c; *.h; *.cpp	No
application/x-zip-compressed	*.zip	No
application/octet-stream	*.exe; *.com	No
text/vnd.wap.wml	*.wml	No
application/vnd.wap.wmlc	*.wmlc	No
image/vnd.wap.wbmp	*.wbmp	No
text/vnd.wap.wmlscript	*.wmls	Yes
application/vnd.wap.wmlscript	*.wmlsc	No
text/xml	*.xml	No
application/pdf	*.pdf	No

It is possible to add additional content types by adding them in 'http.cfg' under the heading [FileTypes]. When a file is requested through the web server it will first search for the file types specified in this file. If it's not found in this file it will search for it among the default content types. This means that adding file type in this file will replace it's predefined type. If a file extension is not recognized, the content type is set to binary data "/".

It is also possible to reconfigure which filetypes that shall be scanned for SSI by adding them under the heading [SSIFileTypes]. Up to 50 additional SSI file types can be defined.

File Format

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

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

Example

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

[SSIFileTypes]
htm
html
xml
```

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

Example:

```
[File path]
\user\config\http_configuration.cfg
```

A.1.6 Email files (email_1.cfg, email_2.cfg ... email_10.cfg)

These files contain predefined email messages and information on how and when to send them. Each email message is triggered by an event in the Input/Output buffers[1]. The Input and Output buffers are scanned once every 0.5 second, i.e. an event must be present longer than 0.5 seconds in order to be detected. It is possible to define up 10 user defined, and 10 admin defined messages, each triggered by different events. These shall be placed in the directories "\user\email\" for normal user configurable messages and "\email" for administrator-configurable messages. The files must be named 'email_1.cfg', 'email_2.cfg' ... 'email_10.cfg'.

Fileformat:

```
[Register]
Buffer, Offset, Type

[Register match]
Match Value, Mask, Match operand

[To]
Recipient(s)

[From]
Sender

[Subject]
Subject line

[Headers]
Extra Headers

[Message]
Message body
```

Parameter	Description
Buffer[a]	Trigger source buffer. Possible values are 'IN' or 'OUT'
Offset[a]	Trigger source offset from the start of the buffer, shall be written in decimal or hexadecimal.
Type	Trigger source data type. Possible values are 'byte', 'word', and 'long'
Match Value[a]	Value to compare with the source data. Shall be written in decimal or hexadecimal.
Mask[a]	The interface performs a logical 'and' on the source data and this Mask before the value is compared with the Match Value. The value shall be written in decimal or hexadecimal.
Match Operand	Specifies how the data shall be compared with the Match Value. Possible values: '<', '=', '>'
Recipient(s)	Destination email addresses, semicolon separated
Sender	Sender email address
Subject line	Email subject (One line only)
Extra Headers	Optional. May be useful for advanced users when for example sending HTML emails etc.
Message Body	Actual message. Can contain SSI (See "Server Side Include (SSI) Functionality" on page 46).

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

The trigger data is read from the location specified by the parameters Buffer, and Offset. A logical AND is performed between the read data and the Mask value, and the result is compared with the Match Value parameter according to the Match Operand If the result is true, the email message is sent to the specified recipient.

1. In order to be able to use this feature, a valid SMTP server account must have been set in the system file 'ethcfg.cfg' (See "ethcfg.cfg" on page 35).

A.2 Password Files

A.2.1 ad_pswd.cfg & sys_pswd.cfg

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

The file format is the following:

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

Example:

```
User:Password
```

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

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

Example:

```
Username
```

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

A.2.2 web_accs.cfg

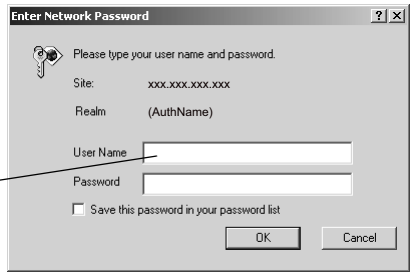
To protect a directory from web access, a file called 'web_accs.cfg' must be placed in the directory to protect. This file shall contain a list of users that are allowed to browse the protected directory and its subdirectories. Multiple of these password files may be present in the system, allowing different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

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

File format:

```
User:Password

[AuthName]
(Message goes here)
```



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

Example:

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

[AuthName]
(Message goes here)
```

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

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

A.3 Virtual Files

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

```
\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
```

A.4 Gateway Status Files

A.4.1 General

These files are made up of lists of keys and their values, in the following format:

Fileformat:

```
<Key Name>=<Value>
```

- <Key Name> is a unique identifier.
- <Value> is a value associated with the <Key Name>. It can either be a single hexadecimal value, a list of hexadecimal values, or a binary list:

Type	Meaning
Hex8	Single hexadecimal value, 8 bit
Hex16	Single hexadecimal value, 16 bit
Hex32	Single hexadecimal value, 32 bit
Hex48	Single hexadecimal value, 48 bit
Hex8 List	Multiple hexadecimal values, separated by dots (.) Example: 48.4F.4D.45.52.20.53.49.4D.53.4F.4E
Binary list	Multiple boolean values (no separation). Example: 11010110110110010011

A.4.2 dynamic.txt

This file contains the value of the Control and Status words, and must be refreshed before use (See "Refreshing Dynamic Gateway Status Information" on page 54).

Key Name	Meaning	Type
ModuleInfoAbs1.Controlword	Ethernet interface controlword value	Hex16
ModuleInfoAbs1.Statusword	Ethernet interface statusword value	Hex16
ModuleInfoAbs2.Controlword	Other network interface controlword value	Hex16
ModuleInfoAbs2.Statusword	Other network interface controlword value	Hex16

A.4.3 master.txt

This file is only present in master/scanner configuration and contains master/scanner related status information. This file must be refreshed before use in order to provide accurate information (see "Refreshing Dynamic Gateway Status Information" on page 54).

Profibus Master Specific Contents

Key Name	Meaning	Type
ModuleInfoAbs2.Configured	List of configured slaves (128 entries) 1: Configured, 0: Not Configured	Binary List
ModuleInfoAbs2.DataTransfer	List of slaves in data transfer (128 entries) 1: Slave in Data Transfer, 0: Slave not in Data Transfer	Binary List
ModuleInfoAbs2.Diagnostics	List of slaves with diagnostics (128 entries) 1: Slave Diagnostics available, 0: Slave Diagnostics not available	Binary List

DeviceNet Scanner Specific Contents

Key Name	Meaning	Type
ModuleInfoAbs2.NodeActive	List of active nodes (64 entries). 1: Node Active, 0: Node not Active	Binary List
ModuleInfoAbs2.NodeIdle	List of idle nodes (64 entries). 1: Node Idle, 0: Node not Idle	Binary List
ModuleInfoAbs2.NodeFaulted	List of faulted nodes (64 entries). 1: Faulted, 0: Not faulted	Binary List
ModuleInfoAbs2.NodeStatus	List containing the status of each node 00: No error, Node is not in the scanlist 46: Duplicate MACID failure 47: Scanner configuration error 48: Device communication error 49: Incorrect device identity 4A: Data overrun error 4B: No network traffic detected 4C: No network traffic to the scanner detected 4D: Incorrect I/O data size 4E: Device does not exist 4F: Transmit failure 50: Device is in 'Idle' mode 51: Device is in 'Fault' mode 52: Fragmentation error 53: Unable to initialise device 54: Node not yet initialised 55: Receive buffer overflow 56: Node changed to 'Idle' mode 57: Shared master error 58: Shared choice error 59: ADR failed 5A: CAN port disabled by application 5B: Bus-off condition detected 5C: No bus power detected 5F: Updating flash 60: In test mode 61: Scanner halted by application 62: Unrecoverable firmware failure 63: Unrecoverable hardware failure	Hex8 List
ModuleInfoAbs2.Connections	Current number of established connections towards other nodes.	Hex8
ModuleInfoAbs2.PacketRate	Current expected packet rate.	Hex16
ModuleInfoAbs2.Baudrate	Currently used baudrate. 00: 125kbps, 01: 250kbps, 02: 500kbps.	Hex8
ModuleInfoAbs2.MacId	DeviceNet MacID	Hex8

AS-I Master Specific Contents

Key Name	Meaning	Type
ModuleInfoAbs2.Configured	List of configured slaves (64 entries) 1: Slave Configured, 0: Slave not Configured	Binary List
ModuleInfoAbs2.Active	List of Active Slaves (64 entries) 1: Slave Active, 0: Slave not Active	Binary List
ModuleInfoAbs2.Detected	List of Detected Slaves (64 entries) 1: Slave Detected, 0: Slave not Detected	Binary List
ModuleInfoAbs2.Fault	List of Peripheral Fault (64 entries) 1: Peripheral Fault, 0: No fault	Binary List
ModuleInfoAbs2.IOConfig	List w. I/O Configuration of each slave (64 entries).	Hex8 List
ModuleInfoAbs2.IDCode	List w. ID Codes of each slave (64 entries).	Hex8 List
ModuleInfoAbs2.Status	Bit field containing misc. status information: Bit 0: Offline mode Bit 1: (reserved) Bit 2: EEPROM OK Bit 3 Automatic addressing enabled (set by the user) Bit 4: Periphery fault Bit 5: (reserved) Bit 6: (reserved) Bit 7: (reserved) Bit 8: Offline phase active Bit 9: Voltage on AS-Interface too low Bit 10: Normal operation Bit 11: 1: Configuration Mode Bit 12: Automatic programming possible Bit 13: (reserved) Bit 14: Slave with address 0 exists Bit 15: Actual configuration matches configured configuration	Hex16
ModuleInfoAbs2.ConfigMode	Current operation mode 1: Configuration Mode, 0. Protected Mode	Hex8
ModuleInfoAbs2.NibbleMode	Data Format 1: Nibble Mode, 0: Byte Mode	Hex8

A.4.4 static.txt

This file is updated during startup and contains the current I/O configuration and various information about the onboard network interfaces.

Key Name	Meaning	Type
ModuleInfoAbs1.InputIoSize	Ethernet Input buffer size.	Hex16
ModuleInfoAbs1.InputParSize	(not used)	-
ModuleInfoAbs1.OutputIoSize	Ethernet Output buffer size.	Hex16
ModuleInfoAbs1.OutputParSize	(not used)	-
ModuleInfoAbs1.FieldbusType	Fixed value of 0083.	Hex16
ModuleInfoAbs1.ModuleType	Fixed value of 0101.	Hex16
ModuleInfoAbs1.SerialNumber	Ethernet interface serial number.	Hex32
ModuleInfoAbs1.BootloaderVersion	Ethernet interface bootloader revision.	Hex16
ModuleInfoAbs1.SoftwareVersion	Ethernet interface software revision.	Hex16
ModuleInfoAbs1.EthernetMacId	Ethernet interface MacID.	Hex48
ModuleInfoAbs2.InputIoSize	Other network interface Input I/O buffer size.	Hex16
ModuleInfoAbs2.InputParSize	Other network interface Input Parameter Data buffer size.	Hex16
ModuleInfoAbs2.OutputIoSize	Other network interface Output I/O buffer size.	Hex16
ModuleInfoAbs2.OutputParSize	Other network interface Output Parameter Data buffer size.	Hex16
ModuleInfoAbs2.FieldbusType	Other network interface type: 0001 = Profibus DP 0005 = Profibus DPV1 0011 = Interbus 0015 = LonWorks 0020 = CANOpen 0025 = DeviceNet 0035 = FIP IO 0040 = Modbus Plus 0045 = Modbus RTU 0065 = ControlNet 0083 = Ethernet 0090 = CC-Link 0091 = AS-Interface	Hex16
ModuleInfoAbs2.ModuleType	Other network interface class: 0101 = Slave interface 0201 = Master interface.	Hex16
ModuleInfoAbs2.SerialNumber	Other network interface serial number.	Hex32
ModuleInfoAbs2.BootloaderVersion	Other network interface bootloader revision.	Hex16
ModuleInfoAbs2.SoftwareVersion	Other network interface software revision.	Hex16
ModuleInfoAbs2.EthernetMacId	Other interface MacID (Ethernet<>Ethernet configurations only)	Hex48
Gateway.SerialNumber	Generic gateway serial number	Hex32
Gateway.BootloaderVersion	Generic gateway bootloader revision	Hex16
Gateway.LibraryVersion	Generic gateway library revision.	Hex16
Gateway.ApplicationVersion	Generic gateway application revision.	Hex16
Gateway.ProductVersion	Generic gateway product revision.	Hex16

B. Server Side Include (SSI) Functionality

The SSI functionality makes it possible to display or alter I/O data and configuration settings on a web page. It is also possible to use SSI functions in email messages, however due to natural reasons some SSI functions cannot be used in email messages.

B.1 Functions

B.1.1 DisplayIP

This function returns the currently used IP address.

Syntax:

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

B.1.2 DisplaySubnet

This function returns the currently used Subnet mask.

Syntax:

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

B.1.3 DisplayGateway

This function returns the currently used Gateway address.

Syntax:

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

B.1.4 DisplayDNS1

This function returns the address of the primary DNS server.

Syntax:

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

B.1.5 DisplayDNS2

This function returns the address of the secondary DNS server.

Syntax:

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

B.1.6 DisplayHostName

This function returns the hostname.

Syntax:

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

B.1.7 DisplayDomainName

This function returns the default domain name.

Syntax:

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

B.1.8 DisplayDhcpState

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

Syntax:

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

B.1.9 DisplayDhcpSupport

This function returns 'Arg1' if DHCP is enabled and 'Arg2' if it's disabled.

Syntax:

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

B.1.10 DisplayEmailServer

This function returns the currently used SMTP server address.

Syntax:

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

B.1.11 DisplaySMTPUser

This function returns the username used for SMTP authentication.

Syntax:

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

B.1.12 DisplaySMTPPswd

This function returns the password used for SMTP authentication.

Syntax:

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

B.1.13 StoreEtnConfig[1]

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

Syntax:

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

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

Accepted fields in form:

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

Default output:

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

For information about how to change the SSI output, please see "Changing the SSI Output" on page 53.

1. This function cannot be used within email messages

B.1.14 printf

This SSI function includes a formatted string which may contain data from the Input/Output buffers, 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.dddddE+-xx, where the number of d's specified by the precision. The default precision is 6; a precision of 0 suppresses the decimal point.
g, G	float; %e or %E is used if the exponent is less than -4 or greater than or equal to the precision; otherwise %f is used. Trailing zeros and trailing decimal point are not printed.
%	no argument is converted; print a %

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

Argument	Description
InReadSByte(*offset*)	Reads a signed byte from position *offset* in the input buffer
InReadUByte(*offset*)	Reads an unsigned byte from position *offset* in the input buffer
InReadSWord(*offset*)	Reads a signed word (short) from position *offset* in the input buffer
InReadUWord(*offset*)	Reads an unsigned word (short) from position *offset* in the input buffer
InReadSLong(*offset*)	Reads a signed longword (long) from position *offset* in the input buffer
InReadULong(*offset*)	Reads an unsigned longword (long) from position *offset* in the input buffer
InReadString(*offset*)	Reads a string (char*) from position *offset* in the input buffer
InReadFloat(*offset*)	Reads a floating point (float) value from position *offset* in the input buffer
OutReadSByte(*offset*)	Reads a signed byte from position *offset* in the output buffer
OutReadUByte(*offset*)	Reads an unsigned byte from position *offset* in the output buffer
OutReadSWord(*offset*)	Reads a signed word (short) from position *offset* in the output buffer
OutReadUWord(*offset*)	Reads an unsigned word (short) from position *offset* in the output buffer
OutReadSLong(*offset*)	Reads a signed longword (long) from position *offset* in the output buffer
OutReadULong(*offset*)	Reads an unsigned longword (long) from position *offset* in the output buffer
OutReadString(*offset*)	Reads a NULL terminated string (char*) from position *offset* in the output buffer
OutReadFloat(*offset*)	Reads a floating point (float) value from position *offset* in the output buffer
MbReadSWord(*id*)	Used to control the gateway, see B-54 "Gateway Control".

B.1.15 scanf[1]

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

Syntax:

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

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

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

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

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

Argument	Description
OutWriteByte(*offset*)	Writes a byte to position *offset* in the output buffer
OutWriteWord(*offset*)	Writes a word (short) to position *offset* in the output buffer
OutWriteLong(*offset*)	Writes a long to position *offset* in the output buffer
OutWriteString(*offset*)	Writes a string to position *offset* in the output buffer
OutWriteFloat(offset)	Writes a floating point (float) value to position *offset* in the output buffer

Default output:

```
Write succeeded
Write failed
```

For information about changing the default SSI output, see "Changing the SSI Output" on page 53.

1. This function cannot be used within email messages

B.1.16 GetText[1]

This SSI function gets the text from an object and stores it in the Output buffer.

Syntax:

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

ObjName- Name of object.

offset - Specifies the offset from the beginning of the output buffer.

n - Specifies maximum number of characters to read (Optional)

Default output:

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

For information about changing the default SSI output, see "Changing the SSI Output" on page 53.

B.1.17 SaveToFile[1]

This SSI function saves the contents of a passed form to a file. The passed name/value pair will be written to the file "File name" separated by the "Separator" string. The contents can either be Appended to the file or overwrite the current content of the file.

Syntax:

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

Default output:

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

For information about changing the default SSI output, see "Changing the SSI Output" on page 53.

B.1.18 IncludeFile

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

Syntax:

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

Default output:

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

For information about changing the default SSI output, see "Changing the SSI Output" on page 53.

1. This function cannot be used within email messages

B.2 Changing the 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()".

B.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"
Invalid DHCP state: "String to use when the DHCP state is invalid"

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

[IncludeFile]
Failure: "String to use when failure" To include filename %s can be included
to the string once

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

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

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

Example:

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

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

B.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) )'-->
```

B.3 Gateway Control

B.3.1 Refreshing Dynamic Gateway Status Information

The system files 'dynamic.txt' and 'master.txt' (master/scanner configurations only) holds dynamic status information from the gateway and the onboard network interfaces. In order to provide accurate information, these files must be refreshed prior to displaying their contents.

The following SSI command sequence will instruct the gateway to refresh it's status files:

Syntax:

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

(Substitute 'ID' with a value from the table below)

ID	Action
20	Refresh 'master.txt'
21	Refresh 'dynamic.txt'

B.3.2 Restarting the Gateway

It is possible to reset the gateway using the following SSI command sequence:

Syntax:

```
<?--#exec cmd_argument='printf( "Data: %u", MbReadSWord( 1 ) )'-->
```

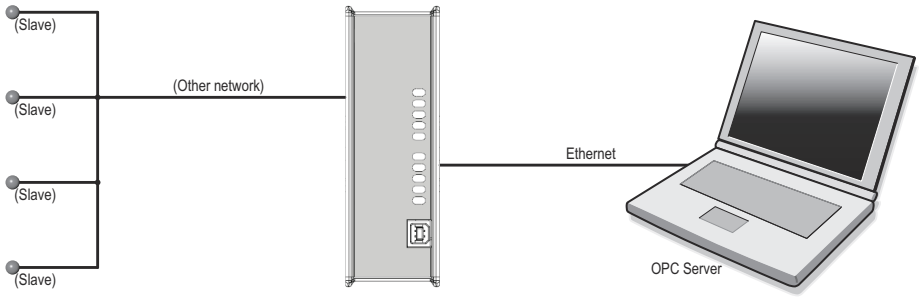
C. Ethernet Transport Provider

C.1 General Information

The Ethernet interface supports the Transport Provider protocol, which allows a host to control the network interface on the other side of the gateway using the standardized Anybus-S API.

This includes...

- Anybus OPC Server
- Anybus NetTool for DeviceNet
- Anybus NetTool for PROFIBUS
- Custom applications based on the Anybus-S API

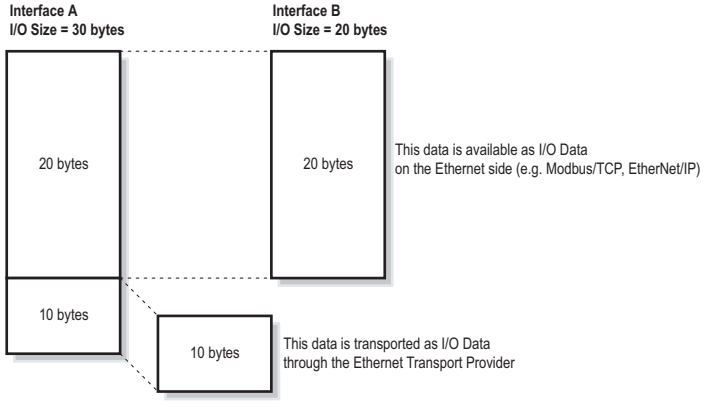


C.2 Allocation of I/O Data

The Transport Provider uses parts of the Input- and Output Buffers to transfer I/O data. The amount of data allocated for the Transport Provider is defined as the difference in I/O sizes between the two network interfaces.

Example:

Transport Provider I/O Size =(I/O Size, Interface A) - (I/O Size, Interface B)
I/O Size, Interface A =30 bytes
I/O Size, Interface B =20 bytes
Transport Provider I/O Size =10 bytes



Note: In the Transport Provider shall handle the complete I/O image towards the other network, set the I/O size to 0 (zero) on the Ethernet interface.

D. Technical Specification

D.1 Network Interface Details

General

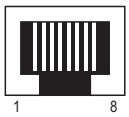
- 2 RJ-45 Ethernet Ports
- 10 and 100 Mbit operation, full or half duplex
- Twisted-pair cables
- Flexible file system providing both volatile and non-volatile storage areas
- Security framework
- Integrated FTP server provides easy file management using standard FTP clients.
- Telnet server featuring a command line interface
- Server Side Include (SSI) capability
- Web server
- Email client (Messages can be triggered by I/O data events)
- IP Access Control
- DHCP/ARP/HICP support
- DNS support

Control Protocols

- **Modbus-TCP**
  Conforms to Modbus-TCP v1.0.
- **Ethernet/IP**
  Group 2 and 3 server capable.

D.2 Ethernet Connectors Pinout (RJ45)

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



D.3 Compliance information

CIP Product Compliance