Anybus® CompactCom™ 40 Modbus Serial

EtherNet/IP

NETWORK GUIDE

SCM-1202-131 1.3 en-US ENGLISH

Important User Information

Disclaimer

The information in this document is for informational purposes only. Please inform HMS Industrial Networks of any inaccuracies or omissions found in this document. HMS Industrial Networks disclaims any responsibility or liability for any errors that may appear in this document.

HMS Industrial Networks reserves the right to modify its products in line with its policy of continuous product development. The information in this document shall therefore not be construed as a commitment on the part of HMS Industrial Networks and is subject to change without notice. HMS Industrial Networks makes no commitment to update or keep current the information in this document.

The data, examples and illustrations found in this document are included for illustrative purposes and are only intended to help improve understanding of the functionality and handling of the product. In view of the wide range of possible applications of the product, and because of the many variables and requirements associated with any particular implementation, HMS Industrial Networks cannot assume responsibility or liability for actual use based on the data, examples or illustrations included in this document nor for any damages incurred during installation of the product. Those responsible for the use of the product must acquire sufficient knowledge in order to ensure that the product is used correctly in their specific application and that the application meets all performance and safety requirements including any applicable laws, regulations, codes and standards. Further, HMS Industrial Networks will under no circumstances assume liability or responsibility for any problems that may arise as a result from the use of undocumented features or functional side effects found outside the documented scope of the 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.

Table of Contents

Page

1 Preface

1.1 About this Document

This network guide is intended to provide a good understanding of the functionality offered by the Anybus CompactCom 40 Modbus Serial - EtherNet/IP .

The reader of this document is expected to be familiar with high level software design and communication systems in general. The information in this network guide, along with the Anybus CompactCom B40 Modbus Serial user manual should normally be sufficient to implement a design. However, if advanced EtherNet/IP specific functionality is required for the network interface of the device, in-depth knowledge of EtherNet/IP networking internals and/or information from the official EtherNet/IP specifications may be required. In such cases, the persons responsible for the implementation of this product should either obtain the EtherNet/IP specification to gain sufficient knowledge or limit their implementation in such a way that this is not necessary.

For additional information, please visit the support website at www.anybus.com/support.

1.2 Document History

Version	Date	Description
1.0	2019-04-30	First release
1.1	2019-05-16	Minor corrections
1.2	2019-12-19	Added SHICP Minor corrections
1.3	2020-05-14	New startup and identity customization section Minor corrections

1.3 Document Conventions

Numbered lists indicate tasks that should be carried out in sequence:

1. First do this
2. Then do this

Bulleted lists are used for:

- Tasks that can be carried out in any order
- Itemized information

► An action

→ and a result

User interaction elements (buttons etc.) are indicated with bold text.

```
Program code and script examples
```

Cross-reference within this document: *Document Conventions, p. 3*

External link (URL): www.hms-networks.com

WARNING
Instruction that must be followed to avoid a risk of death or serious injury.

Caution
Instruction that must be followed to avoid a risk of personal injury.

> **!** Instruction that must be followed to avoid a risk of reduced functionality and/or damage to the equipment, or to avoid a network security risk.

> *Additional information which may facilitate installation and/or operation.*

1.4 Document Specific Conventions

- The terms “Anybus” or “module” refers to the Anybus CompactCom module.
- The terms “host” or “host application” refer to the device that hosts the Anybus.
- Hexadecimal values are written in the format NNNNh or 0xNNNN, where NNNN is the hexadecimal value.
- A byte always consists of 8 bits.
- All dimensions in this document have a tolerance of ±0.10 mm unless otherwise stated.
- Outputs are TTL compliant unless otherwise stated.
- Signals which are “pulled to GND” are connected to GND via a resistor.
- Signals which are “pulled to 3V3” are connected to 3V3 via a resistor.
- Signals which are “tied to GND” are directly connected to GND,
- Signals which are “tied to 3V3” are directly connected to 3V3.

1.4.1 Pin Types

The pin types of the connectors are defined in the table below. The pin type may be different depending on which mode is used.

Pin type	Definition
I	Input
O	Output
I/O	Input/Output (bidirectional)
OD	Open Drain
Power	Pin connected directly to module power supply, GND or 3V3

1.5 Trademarks

Anybus® is a registered trademark of HMS Industrial Networks.

EtherNet/IP is a trademark of ODVA, Inc.

All other trademarks are the property of their respective holders.

2 About the Anybus CompactCom 40 Modbus Serial - EtherNet/IP

2.1 General Information

The Anybus CompactCom 40 Modbus Serial - EtherNet/IP is a communication solution for simple industrial field devices. The host application communicates with the product using the Modbus RTU protocol. The Anybus CompactCom 40 Modbus Serial - EtherNet/IP then communicates the data to the network. Typical applications are basic level I/O blocks, temperature controllers, measuring devices, and sensors.

The Anybus CompactCom 40 Modbus Serial - EtherNet/IP software interface is designed to be network protocol independent, making it possible to support several networking systems using the same application software code/driver.

The Anybus CompactCom 40 Modbus Serial - EtherNet/IP share footprint and electrical interface with the other members of the product family, independent of fieldbus or network. The host application connector provides an interface between the host application (Modbus RTU) and the Anybus CompactCom, while the network connector provides access to the chosen network. The Anybus CompactCom acts as a Modbus RTU slave on the host application side.

The Anybus CompactCom 40 family offers a wide range of functionality. For advanced products and applications, we recommend the standard Anybus CompactCom 40.

For general information about other products using the Anybus CompactCom 40 platform, consult www.anybus.com/support.

This 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.

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.

2.2 Features (EtherNet/IP)

- Two EtherNet/IP ports
- Max. read process data: 1448 bytes
- Max. write process data: 1448 bytes
- Max. process data (read + write, in bytes): 2896 bytes
- Beacon Based DLR (Device Level Ring) and linear network topology supported
- 10/100 Mbit, full/half duplex operation
- Web server
- FTP server
- Customizable identity information
- CIP parameter object support

> **!** All Anybus CompactCom 40 Modbus Serial, where the host is running an example application, will be precertified for network conformance. This is done to ensure that the final product can be certified, but it does not necessarily mean that the final product does not require recertification. Contact HMS Industrial Networks for further information.

2.3 Overview

The picture below shows the data flow in the Anybus CompactCom 40 Modbus Serial - EtherNet/IP. The Modbus master sets up the Modbus RTU communication, and the Anybus CompactCom maps the process data to the industrial network/fieldbus.

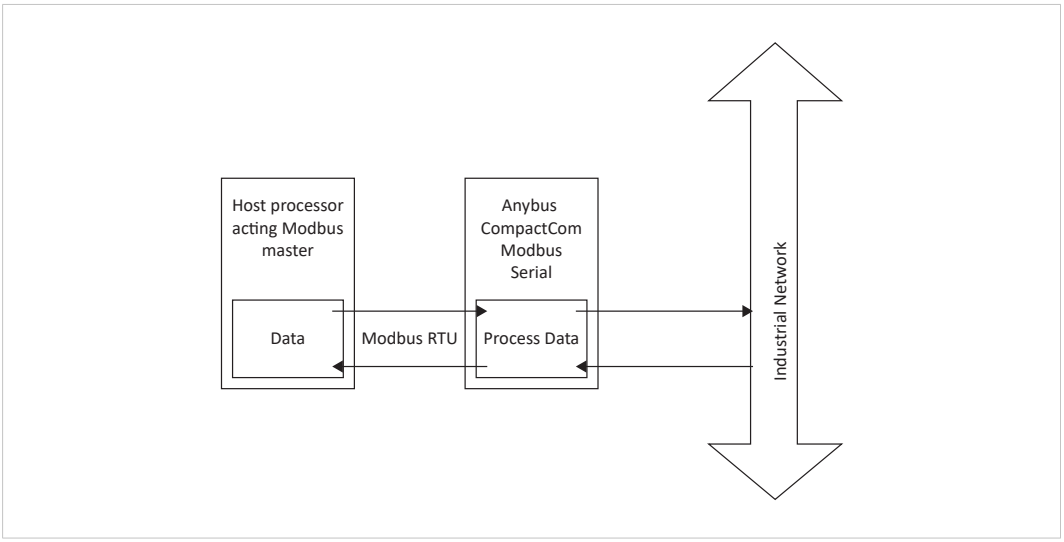


Fig. 1

3 Basic Operation

3.1 Electronic Data Sheet (EDS)

On EtherNet/IP, the characteristics of a device is stored in an ASCII data file with the suffix EDS. This file is used by configuration tools etc. when setting up the network configuration. HMS Industrial Networks supplies a standard (generic) EDS file, which corresponds to the default settings in the module.

Since the module implements the Parameter Object, it is possible for configuration tools such as Rockwell Automation RSNetWorx™ to automatically generate a suitable EDS-file.

See also..

- *Parameter Object (0Fh), p. 30* (CIP object)

3.2 Network Identity

By default, the module uses the following identity settings:

Vendor ID:	005Ah (HMS Industrial Networks)
Device Type:	002Bh (Generic Device)
Product Code:	0037h (Anybus CompactCom 40 EtherNet/IP)
Product Name:	“CompactCom 40 EtherNet/IP(TM)”

Optionally, it is possible to customize the identity of the module by using the Anybus Custom Modbus function code 70.

See also...

- *Identity Object (01h), p. 21* (CIP object)
- *Startup and Identity Customization, p. 8*

3.3 Startup and Identity Customization

To customize the identity of the Anybus CompactCom (e.g. Vendor ID, Product Code, etc.), Virtual Attributes are used.

The most common customizations will be described here. For more detailed information, see the related documents listed in the beginning of this document.

Setting up the virtual attributes in the Anybus CompactCom can be accomplished in two different ways.

- Using the user-defined Modbus function code (Function code 70).

  The use of Function code 70 can be included in the Modbus master. Hence the CompactCom does not need to be preprogrammed before mounting it in the host application.

- Using the Anybus Virtual Attributes Manager.

  The Virtual Attributes Manager is recommended for use during development and for low volume production, since manual user operations are needed for every Anybus CompactCom that shall be programmed.

Once the virtual attributes are written to the Anybus CompactCom, they are saved in non-volatile memory. It is not necessary to write the virtual attributes at each startup.

3.3.1 Virtual Attributes with Specific Modbus Function Code 70

With Modbus function code 70, the Modbus master has access to the Anybus CompactCom internal messaging protocol. This means that all attributes within the Anybus CompactCom are potentially accessible.

When writing the virtual attributes to the Anybus CompactCom, the Anybus object, Object 01h, Instance 1, Attribute 17 is used. All information relevant for the basic virtual attributes will be covered here. For more information, refer to the related documents section in this document.

The example shows example values to the basic virtual attributes:

Virtual Attribute	Example Value
Vendor ID:	0x005A
Product Code:	0x0037
Web Server:	Disabled
FTP Server:	Disabled
Serial Number:	0x12345678
Vendor Name:	Vendor Name
Product Name:	Product Name
Firmware Version:	1.2.3
Hardware Version:	3

To set the virtual attributes in the Anybus CompactCom to these values, using the Modbus function 70, create the request below:

Modbus function 70 Request

	Value	Note
Modbus Address	0xXX	
Function Code	0x46	FC70
Command	0x42	Set_Attribute
Object	0x01	Anybus Object
Instance	0x01	
	0x00	
Ext0	0x11	Attribute 17
Ext1	0x00	Not used
Data Size	0x5C	The data size in this example is 92 bytes
	0x00	
Data	0xF8 0x01 0x00 0x01 0x02 0x00 0x5A 0x00 0xF8 0x01 0x00 0x03 0x02 0x00 0x37 0x00 0xF9 0x01 0x00 0x03 0x01 0x00 0x00 0xF9 0x01 0x00 0x06 0x01 0x00 0x00 0xFF 0x01 0x00 0x03 0x04 0x00 0x78 0x56 0x34 0x12 0xFF 0x01 0x00 0x08 0x0B 0x00 0x56 0x65 0x6E 0x64 0x6F 0x72 0x20 0x4E 0x61 0x6D 0x65 0xFF 0x01 0x00 0x09 0x0C 0x00 0x50 0x72 0x6F 0x64 0x75 0x63 0x74 0x20 0x4E 0x61 0x6D 0x65 0xFF 0x01 0x00 0x0A 0x03 0x00 0x01 0x02 0x03 0xFF 0x01 0x00 0x0B 0x02 0x00 0x03 0x00	Vendor ID Product Code Disable Web Disable FTP Serial Number Vendor Name Product Name Firmware Ver. Hardware Ver.
CRC	0xXX	CRC-16
	0xXX	

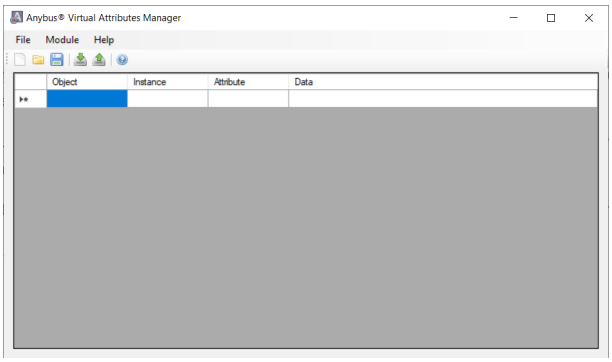
Response

	Value	Note
Modbus Address	0xXX	
Function Code	0x46	FC70
Command	0x02	Set_Attr_Resp
Object	0x01	Anybus Object
Instance	0x01	
	0x00	
Ext0	0x11	Attribute 17
Ext1	0x00	Not used
Data Size	0x00	
	0x00	
CRC	0xXX	CRC-16
	0xXX	

Requests with a size larger than 244 bytes will return Modbus exception code ILLEGAL DATA VALUE.

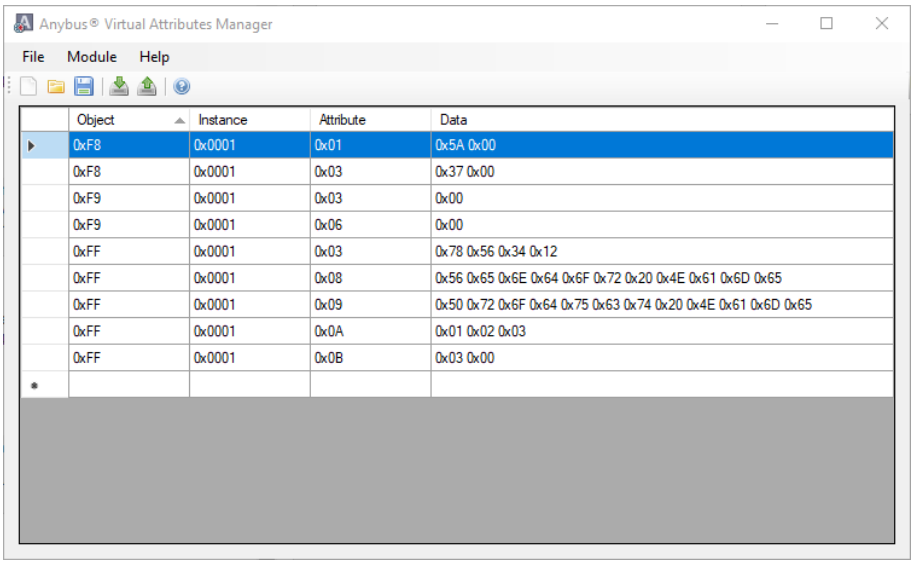
3.3.2 Virtual Attributes with Anybus Virtual Attributes Manager

1. Start the Anybus Virtual Attributes Manager



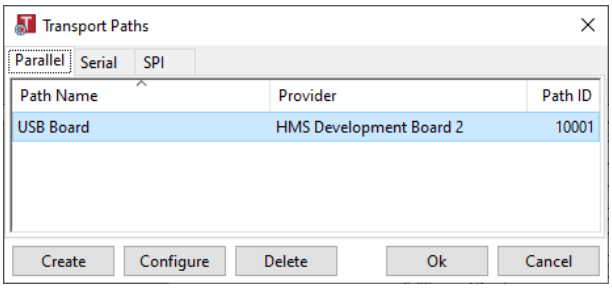
2. Enter the virtual attributes data for the attributes needed. The example below is setting up the attributes with the following values:

Virtual Attribute	Example Value
Vendor ID:	0x005A
Product Code:	0x0037
Web Server:	Disabled
FTP Server:	Disabled
Serial Number:	0x12345678
Vendor Name:	Vendor Name
Product Name:	Product Name
Firmware Version:	1.2.3
Hardware Version:	3



3. Mount the Anybus CompactCom to the USB starterkit board.

4. Select Module->Download and select the correct Transport Path to your USB board.



5. The virtual attributes will be programmed and saved in non-volatile memory.

3.4 Communication Settings

IP address communication settings are configured by the “Application switch 1” register. An application may select to write the value from a physical DIP switch, rotary switch or similar, to this register or it can assign it by other means, see below.

“Application switch 2” is not used for EtherNet/IP network configuration.

Application switch 1 value	User communication settings	Comment
0	Use currently stored communication settings	Factory default stored communication settings: IP address: 0.0.0.0 Subnet mask: 0.0.0.0 Gateway address: 0.0.0.0 DHCP: ON **Note**: Communication settings may be set by external software, see below.
1-254	IP address: 192.168.0.X Subnet mask: 255.255.255.0 Gateway address: 0.0.0.0 DHCP: OFF	Where X in the IP address is the “Application switch 1” value. Resulting communication settings are stored and can later be used if “Application switch 1” value is set to 0.
255	IP address: N/A Subnet mask: N/A Gateway address: N/A DHCP: ON	Communication settings received by DHCP are stored and can later be used if “Application switch 1” value is set to 0.

The communication settings can also be changed from the internal web page, using the IPConfig tool, available at www.anybus.com/support, or through the EtherNet/IP network. Note that changing the configuration through any of these interfaces will affect the currently used and/or stored configuration, but will only be used after the next restart if “Application switch 1” is set to 0.

3.4.1 IPConfig

The IP address can also be configured using the IPConfig tool that is available for download at the product page at www.anybus.com/support.

When the application is started, the network is automatically scanned for Anybus products. The network can be rescanned at any time.

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

Validate the new settings by clicking **Apply**, or close the pane to cancel all changes.

3.5 Beacon Based DLR (Device Level Ring)

Device Level Ring (DLR) is a network technology for industrial applications that uses embedded switch functionality in automation end devices, such as programmable automation controllers and I/O modules, to enable Ethernet ring network topologies at the device level. DLR technology adds network resilience to optimize machine operation. Beacon based DLR networks consist of a ring supervisor and a number of ring nodes, and use “beacons” to detect breaks in the ring. When a DLR network detects a break in the ring, it provides ways to alternatively route the data to recover the network. Diagnostics built into DLR products can identify the point of failure, thus helping to speed maintenance and reduce repair time. The Anybus CompactCom 40 Modbus Serial - EtherNet/IP implements the DLR protocol, which is always enabled. The device is able to process and act on beacon frames sent by ring supervisors, and supports beacon rates down to 100 µs.

3.6 Network Data Exchange

3.6.1 Process Data

Process Data is represented as dedicated instances in the Assembly Object (CIP).

See also...

- *Assembly Object (04h), p. 24* (CIP object)
- *Connection Manager (06h), p. 26* (CIP object)

3.6.2 Translation of Data Types

The Anybus data types are translated to CIP-standard and vice versa as follows:

Anybus Data Type	CIP Data Type	Comments
UINT8	USINT	One byte
UINT16	UINT	Two bytes

3.6.3 File System

Overview

Anybus CompactCom 40 Modbus Serial - EtherNet/IP has an in-built file system that can be accessed from the network. Three directories are predefined:

VFS	The virtual file system that e.g. holds the web pages of the module.
Application	Reserved (not used by the Anybus CompactCom 40 Modbus Serial)
Firmware	Firmware updates are copied here in order to update the Anybus CompactCom the next power cycle.

General Information

The built-in file system hosts 28 MByte of non volatile storage, which can be accessed by the HTTP and FTP servers.

The maximum number of directories and files that can be stored in the root directory is 511 if only short filenames are used (8 bytes name + 3 bytes extension). The number of files that can be stored in other directories, other than the root directory, is unlimited.

The file system uses the following conventions:

- \ (backslash) is used as a path separator
- Names may contain spaces, but must not begin or end with one.
- Valid characters in names are ASCII character numbers less than 127, excluding the following characters: \ / : * ? “ < > |
- Names cannot be longer than 48 characters
- A path cannot be longer than 126 characters (filename included)

See also ...

- *FTP Server, p. 15*
- *Web Server, p. 16*

> *The file system is located in flash memory. Due to technical reasons, each flash segment can be erased approximately 100000 times before failure, making it unsuitable for random access storage.*

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

4 FTP Server

The built-in FTP server makes it easy to manage the file system using a standard FTP client. It can be disabled during start-up, see *Startup and Identity Customization, p. 8*.

If the FTP server is disabled, it is not possible to update the firmware from the network. If firmware updates from the network are desired, a method to re-enable the FTP server must be implemented.

The following port numbers are used for FTP communication:

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

The FTP server supports up to two concurrent clients.

4.1 Session Example

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

1. Open the Windows Explorer.
2. In the address field, type FTP://<address>
   – Substitute <address> with the IP address of the Anybus module
3. Press **Enter**. The Explorer will now attempt to connect to the Anybus module using the specified settings. If successful, the file system will be displayed in the Explorer window.

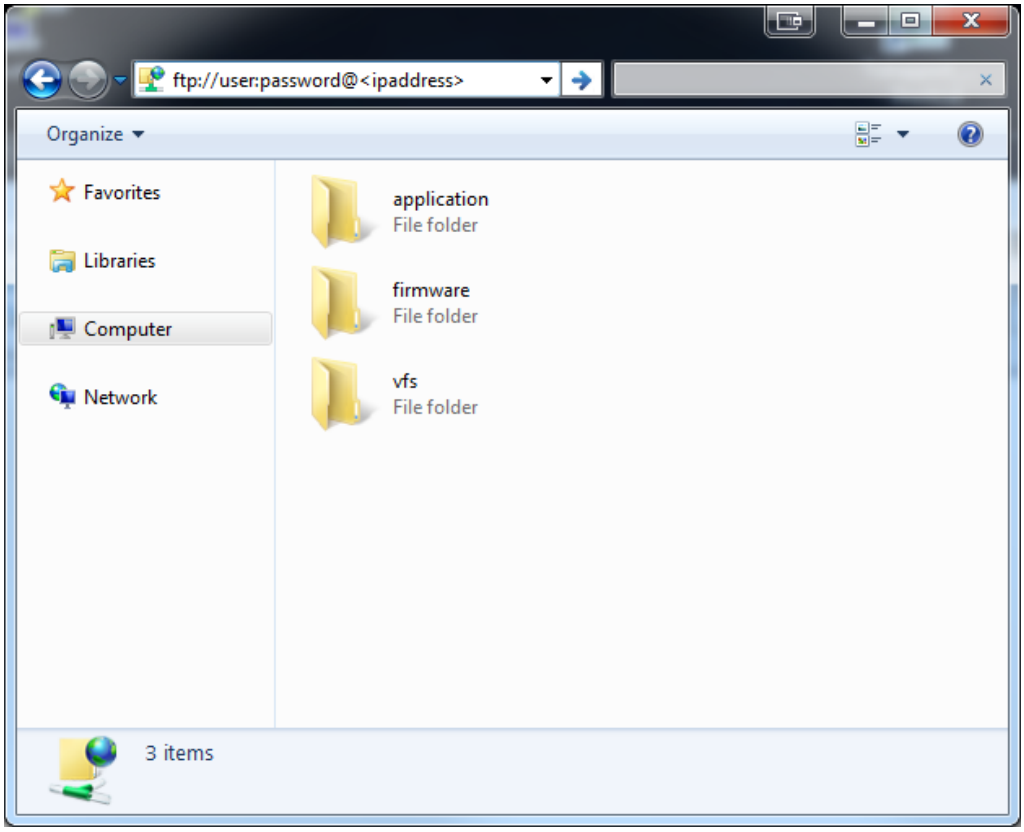


Fig. 2

5 Web Server

5.1 General Information

The built-in web server provides a flexible environment for end-user interaction and configuration purposes.

The web interfaces are stored in the file system, which can be accessed through the FTP server. If necessary, the web server can be completely disabled during start-up, see *Startup and Identity Customization, p. 8*.

See also...

- *FTP Server, p. 15*

5.2 Web Pages

The web pages provide access to:

- Network configuration parameters
- Network status information
- Access to the host application parameters.

5.2.1 Network Configuration

The network configuration page provides interfaces for changing TCP/IP settings.

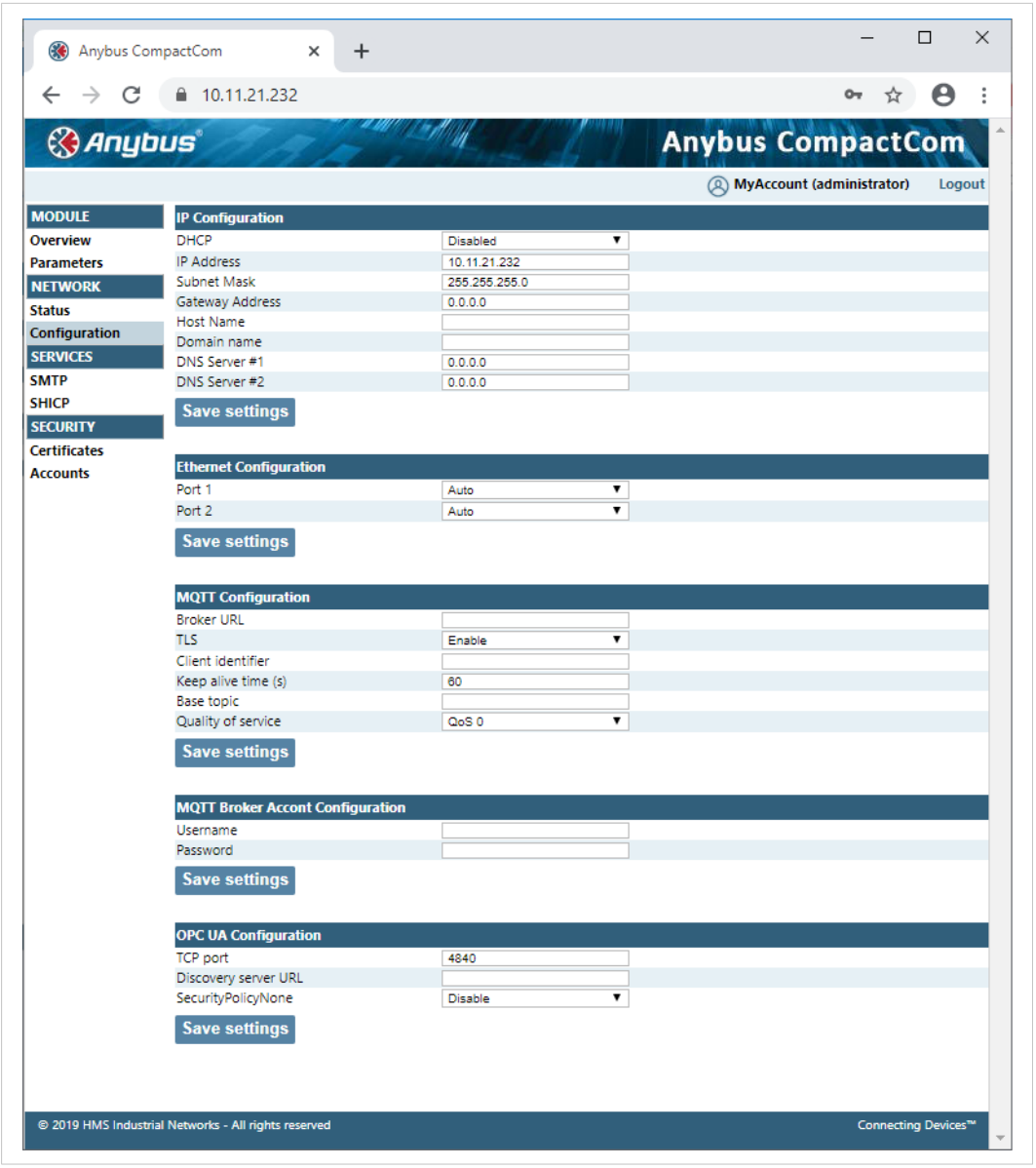


Fig. 3

The module needs to be reset for the TCP/IP settings to take effect. The Ethernet configuration settings will take effect immediately.

IP Configuration

The module needs a reset for any changes to take effect. The settings will only be used if application switch 1 is set to 0.

Name	Description
DHCP	Enable or disable DHCP Default value: enabled
IP address	The TCP/IP settings of the module Default values: 0.0.0.0 Value ranges: 0.0.0.0 - 255.255.255.255
Subnet mask	
Gateway	
Host name	IP address or name Max 64 characters
Domain name	IP address or name Max 48 characters
DNS 1	Primary and secondary DNS server, used to resolve host name Default values: 0.0.0.0 Value ranges: 0.0.0.0 - 255.255.255.255
DNS 2	

Ethernet Configuration

Changes will take effect immediately.

Name	Description
Port 1	Ethernet speed/duplex settings Default value: auto
Port 2	

5.2.2 Ethernet Statistics Page

The Ethernet statistics web page contains the following information:

Ethernet Link		Description
Port 1	Speed:	The current link speed
	Duplex:	The current duplex configuration
Port 2	Speed:	The current link speed
	Duplex:	The current duplex configuration

Ethernet/IP Statistics	Description
Established Class1 Connections	Current number of established class1 connections
Established Class3 Connections	Current number of established class3 connections
Connection Open Requests	Number of received connection open requests
Connection Open Format Rejects	Connection open requests rejected due to request format error
Connection Open Resource Rejects	Connection open requests rejected due to lack of resources
Connection Open Other Rejects	Connection open requests rejected due to other reasons
Connection Close Requests	Number of received connection open requests
Connection Close Format Rejects	Connection close requests rejected due to request format error
Connection Close Other Rejects	Connection close requests rejected due to other reasons
Connection Timeouts	Number of connection timeouts

Interface Counters	Description
In Octets:	Received bytes
In Ucast Packets:	Received unicast packets
In NUcast packets:	Received non unicast packets (broadcast and multicast)
In Discards:	Received packets discarded due to no available memory buffers
In Errors:	Received packets discarded due to reception error

Interface Counters	Description
In Unknown Protos:	Received packets with unsupported protocol type
Out Octets:	Sent bytes
Out Ucast packets:	Sent unicast packets
Out NUcast packets:	Sent non unicast packets (broadcast and multicast)
Out Discards:	Outgoing packets discarded due to no available memory buffers
Out Errors:	Transmission errors

Media Counters	Description
Alignment Errors	Frames received that are not an integral number of octets in length
FCS Errors	Frames received that do not pass the FCS check
Single Collisions	Successfully transmitted frames which experienced exactly one collision
Multiple Collisions	Successfully transmitted frames which experienced more than one collision
SQE Test Errors	Number of times SQE test error messages are generated (Not provided with current PHY interface)
Deferred Transmissions	Frames for which first transmission attempt is delayed because the medium is busy
Late Collisions	Number of times a collision is detected later than 512 bit-times into the transmission of a packet
Excessive Collisions	Frames for which a transmission fails due to excessive collisions
MAC Receive Errors	Frames for which reception of an interface fails due to an internal MAC sublayer receive error
MAC Transmit Errors	Frames for which transmission fails due to an internal MAC sublayer receive error
Carrier Sense Errors	Times that the carrier sense condition was lost or never asserted when attempted to transmit a frame
Frame Size Too Long	Frames received that exceed the maximum permitted frame size
Frame Size Too Short	Frames received that are shorter than lowest permitted frame size

6 CIP Objects

6.1 General Information

This chapter specifies the CIP-object implementation in the module. These objects can be accessed from the network, but not directly by the host application.

Mandatory objects

Vendor Specific Objects:

6.2 Identity Object (01h)

Object Description

The Identity Object provides identification of and general information about the module.

Instance attributes 1, 3, 4, 6 and 7 can be customized during start-up, see *Startup and Identity Customization, p. 8*.

Supported Services

Class:	Get_Attribute_Single
	Get_Attributes_All
Instance:	Get_Attribute_Single
	Set_Attribute_Single
	Get_Attributes_All
	Reset

Class Attributes

#	Name	Access	Type	Value
1	Revision	Get	UINT	0001h (Object revision)
2	Max instance	Get	UINT	Maximum instance number
3	Number of instances	Get	UINT	Number of instances

Instance 01h Attributes

Attributes #1, 3, 4, 6 and 7 can be customized by implementing the Anybus Modbus command FC70.

#	Name	Access	Type	Value/Description
1	Vendor ID	Get	UINT	005Ah (HMS Industrial Networks AB)
2	Device Type	Get	UINT	002Bh (Generic Device)
3	Product Code	Get	UINT	0037h (Anybus CompactCom 40 EtherNet/IP)
4	Revision	Get	Struct of: USINT USINT	Major and minor firmware revision
5	Status	Get	WORD	See Device Status table below
6	Serial Number	Get	UDINT	Unique serial number (assigned by HMS)
7	Product Name	Get	SHORT_STRING	“CompactCom 40 EtherNet/IP (TM)”
11	Active language	Set	Struct of: USINT USINT USINT	Requests sent to this instance are forwarded to the module. If the request is accepted, the module will update the language accordingly.
12	Supported Language List	Get	Array of: Struct of: USINT USINT USINT	List of languages supported by the host application. The list is read from the module and translated to CIP standard. By default the only supported language is English. The application has to implement the corresponding attributes in the application object to enable more languages.

Device Status

bit(s)	Name
0	Module Owned
1	(reserved)
2	Configured This bit shows if the product has other settings than "out-of-box". The value is set to true if the module's NV storage is changed from default.
3	(reserved)
4... 7	Extended Device Status: Value: Meaning: 0000b Unknown 0010b Faulted I/O Connection 0011b No I/O connection established 0100b Non volatile configuration bad 0101b Major fault 0110b Connection in Run mode 0111b Connection in Idle mode (other) (reserved)
8	Set for minor recoverable faults.
9	Set for minor unrecoverable faults.
10	Set for major recoverable faults.
11	Set for major unrecoverable faults.
12... 15	(reserved)

Service Details: Reset

The module forwards reset requests from the network to the host application.

There are two types of network reset requests on EtherNet/IP:

Type 0: Power Cycling Reset	This service emulates a power cycling of the module, refer to Modbus register 0x0FFF for actions needed to be taken by the application.
Type 1: Out of box reset	This service sets a "out of box"configuration and performs a reset, refer to Modbus register 0x0FFF for actions needed to be taken by the application.

6.3 Message Router (02h)

Object Description

The Message Router Object provides a messaging connection point through which a client may address a service to any object class or instance residing in the physical module.

In the Anybus CompactCom module it is used internally to direct object requests.

Supported Services

Class:	-
Instance:	-

Class Attributes

-

Instance Attributes

-

6.4 Assembly Object (04h)

Object Description

The Assembly object uses static assemblies and holds the process data sent/received by the host application. It allows data to and from each object to be sent or received over a single connection.

The terms “input” and “output” are defined from the network’s point of view. An input will produce data on the network and an output will consume data from the network.

See also ....

- *Network Data Exchange, p. 13*

Supported Services

Class:	Get_Attribute_Single
Instance:	Get_Attribute_Single
	Set_Attribute_Single

Class Attributes

#	Name	Access	Type	Value
1	Revision	Get	UINT	0002h (Object revision)
2	Max instance	Get	UINT	Maximum instance number

Instance 03h Attributes (Heartbeat, Input-Only)

This instance is used as heartbeat for Input-Only connections. The data size of the Heartbeat instance in the Forward_Open request should be 0 bytes, however other values are also permitted.

#	Name	Access	Type	Value/Description
3	Data	Set	N/A	- (The data size of this attribute is zero)
4	Size	Get	UINT	0 (Number of bytes in attribute 3)

Instance 04h Attributes (Heartbeat, Listen-Only)

This instance is used as heartbeat for listen-only connections. The data size of the Heartbeat instance in the Forward_Open request should be 0 bytes, however other values are also permitted.

#	Name	Access	Type	Value/Description
3	Data	Set	N/A	- (The data size of this attribute is zero)
4	Size	Get	UINT	0 (Number of bytes in attribute 3)

Instance 06h Attributes (Heartbeat, Input-Only Extended)

This instance is used as heartbeat for input-only extended connections, and does not carry any attributes. The state of connections made to this instance does not affect the state of the Anybus CompactCom module, i.e. if the connection times out, the module does not switch to the Error state. The data size of the Heartbeat instance in the Forward_Open request should be 0 bytes, however other values are also permitted.

#	Name	Access	Type	Value/Description
3	Data	Set	N/A	- (The data size of this attribute is zero)
4	Size	Get	UINT	0 (Number of bytes in attribute 3)

Instance 07h Attributes (Heartbeat, Listen-Only Extended)

This instance is used as heartbeat for listen-only extended connections, and does not carry any attributes. The state of connections made to this instance does not affect the state of the Anybus CompactCom 40 module, i.e. if the connection times out, the module does not switch to the Error state. The data size of the Heartbeat instance in the Forward_Open request should be 0 bytes, however other values are also permitted.

#	Name	Access	Type	Value/Description
3	Data	Set	N/A	- (The data size of this attribute is zero)
4	Size	Get	UINT	0 (Number of bytes in attribute 3)

Instance 64h Attributes (Producing Instance)

#	Name	Access	Type	Value/Description
3	Produced Data	Get	Array of BYTE	This data corresponds to the Write Process Data.
4	Size	Get	UINT	Number of bytes in attribute 3

See also...

Network Data Exchange, p. 13

Instance 96h Attributes (Consuming Instance)

#	Name	Access	Type	Value/Description
3	Produced Data	Set	Array of BYTE	This data corresponds to the Read Process Data.
4	Size	Get	UINT	Number of bytes in attribute 3

See also...

Network Data Exchange, p. 13

6.5 Connection Manager (06h)

Object Description

This object is used for connection and connectionless communications, including establishing connections across multiple subnets.

Supported Services

Class:	-
Instance:	Get Attribute All
	Get Attribute Single
	Set Attribute Single
	Large_Forward_Open
	Forward_Open
	Forward_Close

Class Attributes

(No supported class attributes)

Instance Attributes

#	Name	Access	Type	Value/Description
1	Open Requests	Set	UINT	Number of Forward_Open service requests received.
2	Open Format Rejects	Set	UINT	Number of Forward_Open service requests which were rejected due to bad format.
3	Open Resource Rejects	Set	UINT	Number of Forward_Open service requests which were rejected due to lack of resources.
4	Open Other Rejects	Set	UINT	Number of Forward_Open service requests which were rejected for reasons other than bad format or lack of resources.
5	Close Requests	Set	UINT	Number of Forward_Close service requests received.
6	Close Format Rejects	Set	UINT	Number of Forward_Close service requests which were rejected due to bad format.
7	Close Other Rejects	Set	UINT	Number of Forward_Close service requests which were rejected for reasons other than bad format.
8	Connection Timeouts	Set	UINT	Total number of connection timeouts that have occurred in connections controlled by this Connection Manager.

Class 1 Connection Details

General

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

Total number of supported class 1 connections:	4
Max input connection size:	1448 bytes with Large_Forward_Open, 509 bytes with Forward_Open
Max output connection size:	1448 bytes with Large_Forward_Open, 505 bytes with Forward_Open
Supported RPI (Requested Packet Interval):	1... 3200ms
T→O Connection type:	Point-to-point, Multicast, Null
O→-T Connection type:	Point-to-point, Null
Supported trigger types:	Cyclic, CoS (Change of State)
Supported priorities:	Low, High, Scheduled, Urgent

T	Target, in this case the module
O	Origin, in this case the master

Connection Types

- **Exclusive-Owner connection**

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

  Max. no. of Exclusive-Owner connections: 1

  Connection point O →T: Assembly Object, instance 96h

  Connection point T →O: Assembly Object, instance 64h

- **Input-Only connection**

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

  Max. no. of Input-Only connections: Up to 4
  (Shared with Exclusive-Owner and Listen-Only connections)

  Connection point O →T: Assembly Object, instance 03h

  Connection point T →O: Assembly Object, instance 64h

  Please not that if an Exclusive-Owner connection has been opened towards the module and times out, the Input-Only connection times out as well. If the Exclusive-Owner connection is properly closed, the Input-Only connection remains unaffected.

- **Input-Only Extended connection**

  This connections functionality is the same as the standard Input-Only connection. However when this connection times out it does not affect the state of the application.

  Connection point O →T: Assembly Object, instance 06h

  Connection point T →O: Assembly Object, instance 64h

- **Listen-Only connection**

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

  Max. no. of Input-Only connections: Up to 4
  (Shared with Exclusive-Owner and Input-Only connections)

  Connection point O →T: Assembly Object, instance 04h

  Connection point T →O: Assembly Object, instance 64h

- **Listen-Only Extended connection**

  This connections functionality is the same as the standard Listen-Only connection. However when this connection times out it does not affect the state of the application.

  Connection point O →T: Assembly Object, instance 07h

  Connection point T →O: Assembly Object, instance 64h

- **Redundant-Owner connection**

  This connection type is not supported by the module.

Class 3 Connection Details

General

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

No. of simultaneous Class 3 connections:	6
Supported RPI (Requested Packet Interval):	100... 10000 ms
T→O Connection type:	Point-to-point
O→-T Connection type:	Point-to-point
Supported trigger type:	Application
Supported connection size:	1526 bytes

6.6 Parameter Object (0Fh)

Object Description

The Parameter Object provides an interface to the configuration data of the module. It can provide all the information necessary to define and describe each of the module configuration parameters, as well as a full description of each parameter, including minimum and maximum values.

Each parameter is represented by one instance. Instance numbers start at 1, and are incremented by one, with no gaps in the list.

See also ....

- *ADI Object (A2h), p. 34* (CIP Object)

Supported Services

Class:	Get_Attribute_Single
Instance:	Get_Attribute_Single
	Set_Attribute_Single
	Get_Attributes_All
	Get_Enum_String

Class Attributes

#	Name	Access	Type	Value
1	Revision	Get	UINT	0001h (Object revision)
2	Max instance	Get	UINT	Maximum created instance number = class attribute 3 in the Application Data Object (see Anybus CompactCom 40 Software Design Guide)
8	Parameter Class Descriptor	Get	WORD	Default: 0000 0000 0000 1011b Bit: — Contents: 0 — Supports parameter instances 1 — Supports full attributes 2 — Must do non-volatile storage save command 3 — Parameters are stored in non-volatile storage
9	Configuration Assembly Instance	Get	UINT	0000h

Instance Attributes

#	Name	Access	Type	Value/Description
1	Parameter Value	Get/Set	Specified in attributes 4, 5 & 6.	Actual value of parameter This attribute is read-only if bit 4 of Attribute #4 is true
2	Link Path Size	Get	USINT	0007h (Size of link path in bytes)
3	Link Path	Get	Packed EPATH	20 A2 25 nn nn 30 05h (Path to the object from where this parameter's value is retrieved, in this case the ADI Object)
4	Descriptor	Get	WORD	Bit: Contents: 0 Supports Settable Path (N/A) 1 Supports Enumerated Strings (N/A) 2 Supports Scaling (N/A) 3 Supports Scaling Links (N/A) 4 Read only Parameter (always set) 5 Monitor Parameter (N/A) 6 Supports Extended Precision Scaling (N/A)
5	Data Type	Get	USINT	Data type code
6	Data Size	Get	USINT	Number of bytes in parameter value
7	Parameter Name String	Get	SHORT_STRING	Name of the parameter, truncated to 16 chars
8	Units String	Get	SHORT_STRING	"" (default string)
9	Help String	Get	SHORT_STRING	
10	Minimum Value	Get	(Data type)	Minimum value of parameter The Data Type is defined in attribute 5.
11	Maximum Value	Get	(Data type)	Maximum value of parameter The Data Type is defined in attribute 5.
12	Default Value	Get	(Data type)	Default value of parameter The Data Type is defined in attribute 5.
13	Scaling Multiplier	Get	UINT	0001h
14	Scaling Divisor	Get	UINT	
15	Scaling Base	Get	UINT	
16	Scaling Offset	Get	INT	0000h
17	Multiplier Link	Get	UINT	
18	Divisor Link	Get	UINT	
19	Base Link	Get	UINT	
20	Offset Link	Get	UINT	
21	Decimal Precision	Get	USINT	00h

6.7 DLR Object (47h)

Object Description

The Device Level Ring (DLR) Object provides the status information interface for the DLR protocol. This protocol enables the use of an Ethernet ring topology, and the DLR Object provides the CIP application-level interface to the protocol.

Supported Services

Class:	Get_Attribute_Single
	Get_Attributes_All
Instance:	Get_Attribute_Single

Class Attributes

#	Name	Access	Type	Value
1	Revision	Get	UINT	0003h (Object revision)

Instance Attributes

#	Name	Access	Type	Value/Description
1	Network Topology	Get	USINT	Bit: Contents: 0 "Linear" 1 "Ring"
2	Network Status	Get	USINT	Bit: Contents: 0 "Normal" (N/A) 1 "Ring Fault" 2 "Unexpected Loop Detected" 3 "Partial Network Fault" 4 "Rapid Fault/Restore Cycle"
10	Active Supervisor Address	Get	Struct of: UDINT Array of: 6 USINTs	This attribute holds the IP address (IPv4) and/or the Ethernet Mac address of the active ring supervisor.
12	Capability Flags	Get	DWORD	82h (Beacon-based ring node, Flush_Table frame capable)

6.8 QoS Object (48h)

Object Description

Quality of Service (QoS) is a general term that is applied to mechanisms used to treat traffic streams with different relative priorities or other delivery characteristics. Standard QoS mechanisms include IEEE 802.1D/Q (Ethernet frame priority) and Differentiated Services (DiffServ) in the TCP/IP protocol suite.

The QoS Object provides a means to configure certain QoS related behaviors in EtherNet/IP devices.

The object is required for devices that support sending EtherNet/IP messages with nonzero DiffServ code points (DSCP), or sending EtherNet/IP messages in 802.1Q tagged frames.

Supported Services

Class:	Get_Attribute_Single
Instance:	Get_Attribute_Single
	Set_Attribute_Single

Class Attributes

#	Name	Access	Type	Value
1	Revision	Get	UINT	0001h (Object revision)

Instance Attributes

#	Name	Access	Type	Value/Description
1	802.1Q Tag Enable	Set	USINT	Enables or disables sending 802.1Q frames. Value: Contents: 0 Disabled (Default) 1 Enabled
4	DSCP Urgent	Set	USINT	CIP transport class 1 messages with priority Urgent Default: 55
5	DSCP Scheduled	Set	USINT	CIP transport class 1 messages with priority Scheduled Default: 47
6	DSCP High	Set	USINT	CIP transport class 1 messages with priority High Default: 43
7	DSCP Low	Set	USINT	CIP transport class 1 messages with priority Low Default: 31
8	DSCP Explicit	Set	USINT	CIP UCMM and CIP class 3 Default: 27

6.9 ADI Object (A2h)

Object Description

Anybus representation of process data mapped parameters.

See also ..

- *Parameter Object (0Fh), p. 30* (CIP Object)

Supported Services

Class:	Get_Attribute_Single
Instance:	Get_Attribute_Single
	Set_Attribute_Single

Class Attributes

#	Name	Access	Type	Value
1	Revision	Get	UINT	0002h (Object revision)
2	Max Instance	Get	UINT	Equals number of parameters
3	Number of instances	Get	UINT	Equals number of parameters

Instance (1..n) Attributes

#	Name	Access	Type	Value/Description
1	Name	Get	SHORT_STRING	Parameter name (Including length)
2	ABCC Data type	Get	Array of USINT	Data type of instance value
3	No. of Elements	Get	USINT	1 (Number of elements of the specified data type)
4	Descriptor	Get	Array of USINT	Bit field describing the access rights for this instance Bit: Meaning: 0 1 = Get Access 1 1 = Set Access 2 (reserved, set to 0) 3 1 = Write process data mapping possible 4 1 = Read process data mapping possible 5 1 = NVS parameter 6 1 = Data notification enabled
5	Value	Get/Set	Determined by attributes #2, #3 and #9	Instance value
6	Max Value	Get		The maximum permitted parameter value.
7	Min Value	Get		The minimum permitted parameter value.
8	Default Value	Get		The default parameter value.
9	Number of subelements	Get	Array of UINT	Always 0

Attributes #5–8 are converted to/from CIP standard by the module

6.10 TCP/IP Interface Object (F5h)

Object Description

This object provides the mechanism to configure the TCP/IP network interface of the module. It groups the TCP/IP-related settings in one instance for each TCP/IP capable communications interface.

See also ..

- *Communication Settings, p. 12*

Supported Services

Class:	Get_Attribute_All
	Get_Attribute_Single
Instance:	Get_Attribute_All
	Get_Attribute_Single
	Set_Attribute_Single

Class Attributes

#	Name	Access	Type	Value
1	Revision	Get	UINT	0004h (Object revision)
2	Max instance	Get	UINT	1 (Maximum instance number)
3	Number of instances	Get	UINT	1 (Number of instances)
6	Maximum ID Number Class Attributes	Get	UINT	7 (The attribute number of the last implemented class attribute)
7	Maximum ID Number Instance Attributes	Get	UINT	13 (The attribute number of the last implemented instance attribute)

Instance Attributes

#	Name	Access	Type	Value	Comments
1	Status	Get	DWORD	-	Bit: Meaning: (reserved, set to 0) 0–3 When set to 1h, attribute #5 contains valid configuration from DHCP or non-volatile storage. When set to 2h, attribute #5 contains valid configuration from hardware settings. Remaining values are reserved for future use. 4 Multicast pending if set to 1. 5 Interface configuration pending if set to 1. A new configuration will be loaded at the next reset. 6 AcdStatus. Set to 1 if an address conflict is detected. Address conflict detection is enabled/ disabled in attribute #10. 7 AcdFault 8–31 (reserved, set to 0)
2	Configuration Capability	Get	DWORD	-	Bit: Meaning: 0-1: Always 0. For more information, consult the CIP specifications. 2: Always 1. For more information, consult the CIP specifications. 3: Always 0. For more information, consult the CIP specifications. 4: Always 1. For more information, consult the CIP specifications. 5: The module is hardware configurable when this bit is set to 1. The bit will be set if Modbus register 0x5200 or 0x5201 are set during initialization. 6: Always 0. For more information, consult the CIP specifications. 7: Always 1. For more information, consult the CIP specifications. 8 - 31: (reserved, set to 0)
3	Configuration Control	Get/Set	DWORD	-	Value: Meaning 0: Configuration from non-volatile memory 2: Configuration from DHCP
4	Physical Link Object	Get	Struct of: UINT (Path size) Padded EPATH	- 0002h 20 F6 24 03h	- - Path to Ethernet Link Object, Instance #3
5	Interface Configuration	Get/Set	Struct of: UDINT (IP) UDINT (Mask) UDINT (GW) UDINT (DNS1) UDINT (DNS2) STRING (Domain)		- IP address Subnet mask Default gateway Primary DNS Secondary DNS Default domain
6	Host Name	Get/Set	STRING	-	Host name of Anybus module
8	TTL Value	Get/Set	USINT	1	TTL value for EtherNet/IP multicast packets

#	Name	Access	Type	Value	Comments
9	Mcast Config	Set	Struct of:		IP multicast configuration.
	Alloc Control		USINT	0	Value: Meaning: 0: Use default allocation algorithm to generate multicast addresses 1: Allocate multicast addresses according to the values in the “Num Mcast”- and “Mcast Start Addr”-fields.
	(reserved)		USINT	0	Set to zero. Do not change.
	Num Mcast		UINT	-1	Number of multicast addresses to allocate for EtherNet/IP
	Mcast Start Addr		UDINT	-	Starting multicast address from which to begin allocation
10	SelectAcd	Set	Bool	1	Value: Meaning: 0: Disable ACD 1: Enable ACD (Default). If ACD (address conflict detection) is enabled, bit 6 in attribute #1 will be set if an ACD conflict is detected. The Network Status LED will also indicate a detected conflict, see *LED Indications, p. 45* .
11	LastConflictDetected	Set	Struct of:		ACD Diagnostic parameters Related to the last conflict detected.
	AcdActiviity		USINT	-	State of ACD activity when last conflict detected.
	RemoteMAC		ARRAY of 6 USINT	-	MAC address of remote node form the ARP PDU in which a conflict was detected.
	ArpPdu		ARRAY of 28 USINT	-	Copy of the raw ARP PDU in which a conflict was detected.
13	Encapsulation inactivity timeout	Set	UINT	0 - 3600	Number of seconds of inactivity before a TCP connection is closed. 0: Disabled

6.11 Ethernet Link Object (F6h)

Object Description

This object maintains link specific counters and status information for an IEEE 802.3 communications interface. Exactly one instance for each communications interface on the module is supported. Instances for internally accessible interfaces can also be supported.

See also ..

- *Communication Settings, p. 12*

Supported Services

Class:	Get_Attributes_All
	Get_Attribute_Single
Instance:	Get_Attributes_All
	Get_Attribute_Single
	Set_Attribute_Single
	Get_And_Clear

Class Attributes

Three instances (port 1, port 2 and the internal port) are implemented.

#	Name	Access	Type	Value
1	Revision	Get	UINT	0004h (Object revision)
2	Max Instance	Get	UINT	3 (Maximum instance number)
3	Number of Instances	Get	UINT	3 (Number of instances)
6	Maximum ID Number Class Attributes	Get	UINT	7 (The attribute number of the last implemented class attribute.)
7	Maximum ID Number Instance Attributes	Get	UINT	11 (The attribute number of the last implemented instance attribute.)

Instance Attributes

#	Name	Access	Type	Value	Comments
1	Interface Speed	Get	UDINT	10 or 100	Actual Ethernet interface speed.
2	Interface Flags	Get	DWORD	-	See table “Interface Flags” below.
3	Physical Address	Get	Array of 6 USINTs	(MAC ID)	Physical network address, i.e. assigned MAC address.
4	Interface Counters	Get	Struct of:		
	In Octets		UDINT	N/A	Octets received on the interface
	In Ucast Packets		UDINT	N/A	Unicast packets received on the interface
	In NUcast Packets		UDINT	N/A	Nonunicast packets received on the interface
	In Discards		UDINT	N/A	Inbound packets with unknown protocol
	In Errors		UDINT	N/A	Inbound packets that contain errors (does not include In discards)
	In Unknown Protos		UDINT	N/A	Inbound packets with unknown protocol
	Out Octets		UDINT	N/A	Octets sent on the interface
	Out Ucast Packets		UDINT	N/A	Unicast packets sent on the interface
	Out NUcast Packets		UDINT	N/A	Nonunicast packets sent on the interface
	Out Discards		UDINT	N/A	Outbound packets with unknown protocol
	Out Errors		UDINT	N/A	Outbound packets that contain errors (does not include Out discards)
5	Media Counters	Get	Struct of:		Media specific counters
	Alignment Errors		UDINT	N/A	Frames received that are not an integral number of octets in length
	FCS Errors		UDINT	N/A	Frames received that do not pass the FCS check
	Single Collisions		UDINT	N/A	Successfully transmitted frames that have experienced exactly one collision
	Multiple Collisions		UDINT	N/A	Successfully transmitted frames that have experienced more than one collision
	SQE Test Errors		UDINT	0	The number of times the SQE test error message is generated(Counter not provided with current PHY interface)
	Deferred Transmissions		UDINT	N/A	Frames for which the first transmission attempt is delayed because the medium is busy
	Late Collisions		UDINT	N/A	The number of times a collision is detected later than 512 bit-times into the transmission of a packet
	Excessive Collisions		UDINT	N/A	Frames for which a transmission fails due to excessive collisions
	MAC Transmit Errors		UDINT	N/A	Frames for which a transmission fails due to an internal MAC sublayer receive error
	Carrier Sense Errors		UDINT	N/A	The number of times that the carrier sense condition was lost or never asserted when attempting to transmit a frame
	Frame Too Long		UDINT	N/A	Frames received that exceed the maximum permitted frame size
	MAC Receive Errors		UDINT	N/A	Frames for which reception on an interface fails due to an internal MAC sublayer receive error

#	Name	Access	Type	Value	Comments
6	Interface Control	Get/Set	Struct of:		
	Control Bits		WORD	-	Interface control bits
	Forced Interface Speed		UINT	-	Speed at which the interface shall be forced to operate. Returns 'Object state Conflict' if auto-negotiation is enabled
7	Interface Type	Get	USINT	-	See table "Interface State" below.
8	Interface State	Get	USINT	-	See table "Interface Type" below.
9	Admin State	Get/Set	USINT	-	See table "Admin State" below.
10	Interface Label	Get	SHORT_STRING	—	See table "Interface Label" below.
11	Interface Capability	Get	Struct of:	-	Indication of the capabilities of the interface
	Capability Bits		DWORD	-	Interface capabilities, other than speed/duplex See table "Interface Capability" below.
	Speed/Duplex Options		Struct of:	-	Indicates speed/duplex pairs supported in the Interface Control Attribute
			USINT	-	Speed/duplex array count
			Array of Struct of:	-	Speed/duplex array
			UINT	-	Interface speed
			USINT	-	Interface Duplex Mode 0 = half duplex 1 = full duplex 2 - 255 = Reserved

Interface Flags

Bit	Name	Description
0	Link status	Indicates whether or not the Ethernet 802.3 communications interface is connected to an active network. Value: Meaning: 0 Inactive link 1 Active link
1	Half/full duplex	Indicates the duplex mode currently in use. Value: Meaning: 0 Half duplex 1 Full duplex
2 - 4	Negotiation Status	Indicates the status of link auto-negotiation. Value: Meaning: 0 Auto-negotiation in progress. 1 Auto-negotiation and speed detection failed (using default values) (Recommended default values are 10 Mbps, half duplex) 2 Auto negotiation failed but detected speed (using default duplex value) 3 Successfully negotiated speed and duplex. 4 Auto-negotiation not attempted. Forced speed and duplex.
5	Manual Setting requires Reset	Value: Meaning: 0 Interface can activate changes to link parameters during runtime 1 Reset is required in order for changes to have effect
6	Local Hardware Fault	Value: Meaning: 0 No local hardware fault detected 1 Local hardware fault detected
7-31	(reserved)	Set to 0.

Interface State

This attribute indicates the current operational state of the interface.

Value	Description
0	Unknown interface state.
1	The interface is enabled and is ready to send and receive data.
2	The interface is disabled.
3	The interface is testing.

Admin State

This attribute controls the administrative setting of the interface state.

Value	Description
0	(reserved)
1	Enable the interface.
2	Disable the interface.
3-255	(reserved)

Interface Label

Instance	Value
1	Port 1
2	Port 2
3	Internal

Interface Type

Instance	Value	Description
1	2	Twisted-pair
2	2	Twisted-pair
3	1	Internal interface

Interface Capability

Bit	Name	Description	Implementation
0	Manual setting requires reset	Indicates whether or not the device requires a reset to apply changes made to the Interface Control attribute (#6). 0 Indicates that the device automatically applies changes made to the Interface Control attribute (#6) and, therefore, does not require a reset in order for changes to take effect. This bit shall have this value when the Interface Control attribute (#6) is not implemented. 1 1 = Indicates that the device does not automatically apply changes made to the Interface Control attribute (#6) and, therefore, will require a reset in order for changes to take effect. Note: this bit shall also be replicated in the Interface Flags attribute (#2), in order to retain backwards compatibility with previous object revisions.	Return 0
1	Auto-negotiate	0 Indicates that the interface does not support link auto-negotiation 1 Indicates that the interface supports link auto-negotiation	0 for internal interface, 1 for external interfaces
2	Auto-MDIX	0 Indicates that the interface does not support auto MDIX operation 1 Indicates that the interface supports auto MDIX operation	0 for internal interface, 1 for external interfaces
3	Manual speed/ duplex	0 Indicates that the interface does not support manual setting of speed/duplex. The Interface Control attribute (#6) shall not be supported. 1 Indicates that the interface supports manual setting of speed/duplex via the Interface Control attribute (#6)	0 for internal interface, 1 for external interfaces
4 - 31	Reserved	Shall be set to 0	Return 0

This page intentionally left blank

A LED Indications

See Anybus CompactCom B40 Modbus Serial User Manual for more information.

A.1 Network Status LED

LED State	Description
Off	No power or no IP address
Green	Online, one or more connections established (CIP Class 1 or 3)
Green, flashing	Online, no connections established
Red	Duplicate IP address, FATAL error
Red, flashing	One or more connections timed out (CIP Class 1 or 3)

A.2 Module Status LED

LED State	Description
Off	No power
Green	Controlled by a Scanner in Run state
Green, flashing	Not configured, or Scanner in Idle state
Red	Major fault (EXCEPTION-state, FATAL error etc.)
Red, flashing	Recoverable fault(s). Module is configured, but stored parameters differ from currently used parameters.

A.3 LINK/Activity LEDs

LED State	Description
Off	No link, no activity
Green	Link (100 Mbit/s) established
Green, flickering	Activity (100 Mbit/s)
Yellow	Link (10 Mbit/s) established
Yellow, flickering	Activity (10 Mbit/s)

B Secure HICP (Secure Host IP Configuration Protocol)

B.1 General

The Anybus CompactCom 40 Modbus Serial - EtherNet/IP supports the Secure HICP protocol used by the Anybus IPconfig utility for changing settings, e.g. IP address, Subnet mask, and enable/disable DHCP. Anybus IPconfig can be downloaded free of charge from the HMS website, www.anybus.com. This utility may be used to access the network settings of any Anybus product connected to the network via UDP port 3250.

The protocol offers secure authentication and the ability to restart/reboot the device(s).

B.2 Operation

When the application is started, the network is automatically scanned for Anybus products. The network can be rescanned at any time by clicking **Scan**.

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

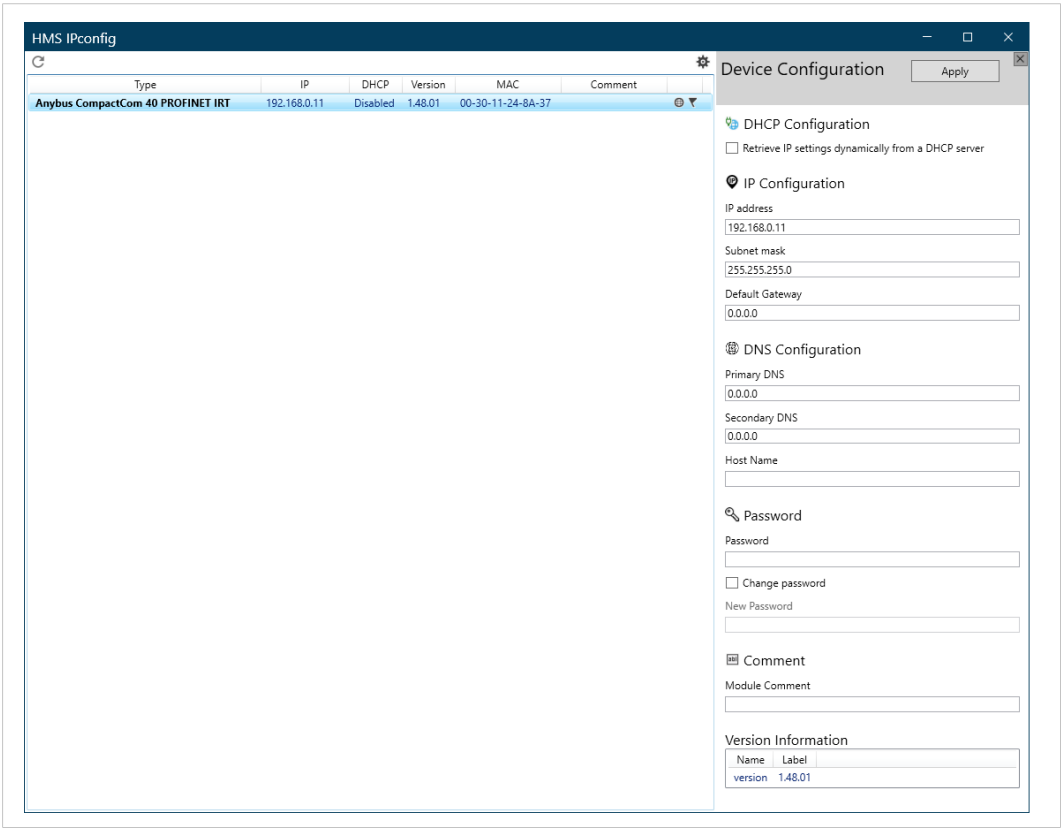


Fig. 4

Validate the new settings by clicking **Set**, or click **Cancel** to cancel all changes. Optionally, the configuration can be protected from unauthorized access by a password.

C Conformance Test Guide

C.1 General

When using the default settings of all parameters, the Anybus CompactCom 40 Modbus Serial - EtherNet/IP is precertified for network compliance. This precertification is done to ensure that the end product *can* be certified.

To be allowed to use EtherNet/IP in a product the vendor is required to be a licensed EtherNet/IP vendor, with a vendor ID of its own. Please contact www.odva.org to obtain a vendor ID.

Changes in the parameters in the example EDS file supplied by HMS Industrial Networks will require a certification. This chapter provides a guide for successfully conformance testing your device, featuring the Anybus CompactCom 40 Modbus Serial - EtherNet/IP, in order to comply with the demands for network certification set by the ODVA.

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

> **!** This appendix provides guidelines and examples of what is needed for conformance testing and certification. Depending on the functionality of your application, there may be additional steps to take.

All screenshots within this document are taken from the ODVA Conformance Test Software Tool for EtherNet/IP CT14, © ODVA Inc. This software is available for order through the ODVA website. It is required to perform pre-testing with this software prior to submitting the product for conformance testing.

Also, a Statement of Conformance file (STC file), describing the EtherNet/IP application, has to be prepared prior to submitting the product for conformance testing.

C.2 Suggested Test Tools

C.2.1 Wireshark

This free, open source tool is the de facto standard for network capture and analysis. It is heavily used by ODVA Test Service Providers (TSPs), HMS Industrial Networks, and the greater EtherNet/IP user base. Wireshark (www.wireshark.org) captures Ethernet traffic using your computer's network interface card, and displays the contents in an intuitive fashion that allows for detailed analysis of the packets. Developers from HMS Industrial Networks have contributed to the EtherNet/IP dissectors (the analysis engine), and it is possible for users to create their own dissectors for their application data. The use of Wireshark is well documented, but there are a few good tips for EtherNet/IP testing that will help users filter out the crucial information.

- Use viewing filters "CIP" to see only EtherNet/IP traffic.
- It is possible to filter by the HMS MAC ID. This will only show Ethernet messages with HMS devices as the source or destination "eth.addr[0:3] == 00:30:11".
- There are many other useful filters available on the Wireshark webpage.

C.2.2 NMAP

NMAP is a free, open source tool for network discovery and security testing. NMAP will discover which TCP and UDP ports are open or responding. It will also determine which layer 3 services are supported by your device. ODVA has strict guidelines for open ports, and mandatory layer 3 services. For the NMAP procedure used by TSPs please see the Sample Test Report that comes with Conformance Test Software from ODVA.

C.2.3 ODVA Conformance Test Software

This automated test software is designed to query, provoke, and detect software flaws in your device. ODVA sells yearly subscriptions of this software to vendors so that they can prepare for conformance testing. This software is also the best way to modify or create the Statement of Conformance (STC) file. Pressing CTRL+D will bring up a GUI for the Data section of the STC file.

Getting Started

After completing the install, a webpage is brought up in the default browser. This page gives an overview of the test software and lists the relevant documentation with a brief summary. The setup for testing is covered in the Conformance Test Software User Manual.

Chapter	Contents
1	System requirements and installation
2	How to select a device and how to modify the Statement of Conformance file
3	How to set up the network to prepare for testing
4	How to run the test software

The User Manual - Critical Points

Users are strongly encouraged to read through the Conformance Test Software User Manual to fully understand the testing software. The following points are meant to recapture the critical sections of this document.

- The Network Interface that will be used for testing needs to be selected from the available network interface cards in the Setup menu.

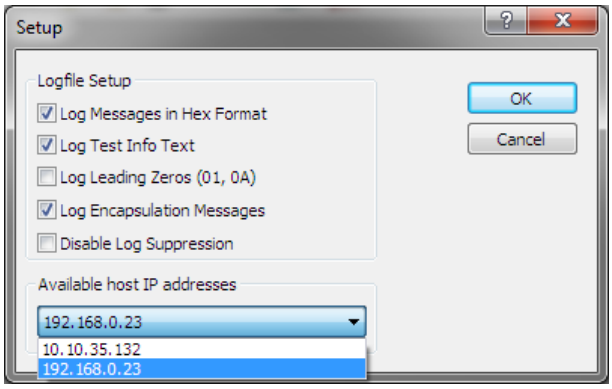


Fig. 5

- Most devices will comply with the default timeout settings, but some require more relaxed standards for responses. This can be set in the Set Message Wait Timers menu.
- The latest version of the CT Test software requires users to allocate a second IP address for their network interface card.
- Enabling the Encapsulation Logging feature of the CT test will allow users to efficiently work with Wireshark captures and Conformance Test logs.

C.2.4 EZ-EDS

EZ-EDS is a free utility made available by ODVA. This tool is very helpful for editing and testing Electronic Data Sheets. Electronic Data Sheets are ASCII formatted files that describe data organization, configuration, and performance capabilities. They are commonly called EDS files, and have the extension .eds. EDS files can be built and modified using a text editor, but EZ-EDS provides a graphical user interface that brings attention to major fields. EZ-EDS also tests EDS

files for correct formatting. Much of the possible content of EDS files is optional, and ODVA tests stress correct formatting and not content.

C.2.5 Anybus EDS Generator

The Anybus EDS Generator tool automatically generates an EDS file by scanning a device using the Anybus CompactCom 40 EtherNet/IP. This tool is easy to use and will provide a correctly configured EDS file that matches your product. It is still required to validate the EDS file via EZ-EDS.

The tool is available from the HMS Industrial Networks web site.

C.2.6 Sample Test Reports

The subscription to the conformance test software includes the EtherNet/IP Sample Test Report document. This document outlines the manual procedure that testers will perform in addition to running the automated test software.

Manual Test Procedure

Some features of EtherNet/IP cannot be properly verified by automated test software or the development of a fully automated test may be impractical. For these features, a manual test procedure, as well as passing criteria, is listed in the Sample Test Report. This is the exact procedure and criteria used by the Test Service Providers. The majority of functionality that needs to be tested manually is provided by the systems of the Anybus CompactCom, and has no interaction with the host application. Therefore, developers using the Anybus CompactCom may omit this lengthy procedure, but they must check the following:

- HMS recommends everyone to complete the Physical layer and EDS test sections of the sample test report. This ensures that produce labeling of LED's is correct and that the EDS file is verified prior to submitting the product to the TSP for conformance test
- If DLR is enabled in the product, it is required to be able to configure the speed and the duplex of all Ethernet ports in some way. The host application may elect to disable the standard means of configuring the speed and duplex in the Anybus CompactCom by:
  - Disabling set access to the Ethernet Link object by setting instance attribute #9 (Enable Communication Settings from NET) in the Host EtherNet/IP object to False.
  - Disabling the web server. On the standard web pages of the Anybus CompactCom it is possible to configure speed and duplex of the ethernet ports. For applications using transparent ethernet functionality the web server is always disabled.

  If none of the these ways of configuring the speed and duplex is possible, the host application must provide some other way to configure them. For example the application can have a keypad interface which can be used for configuration.
- If the host application includes hardware switches (for example DIP switches or rotary switches) for configuring the IP address, HMS recommends to perform the TCP/IP Interface manual test cases (Section 4) in the sample test report.

C.3 Statement of Conformance (STC)

> **!** This document is not a comprehensive guide. Following the steps below will not absolutely guarantee that a device will complete conformance testing.

The goal of this section is to explain the relation of the contents in the data filed of the Anybus Modbus command FC70 to the conformance test and the Statement of Conformance (STC).

It is recommended to read the CIP Protocol Test Specification and the EtherNet/IP Test Specification prior to testing. In these documents the expected response and/or the acceptable behavior are stated, which is useful to be able to avoid a lot of initial errors. Modifications can be made to the Statement of Conformance and to the host application at an early stage, reducing time and effort.

C.3.1 Implementation Using the Anybus Custom Modbus Function Code 70

The implementation may have to be adapted, to make sure that the end product will pass a conformance test. Using the CT Software, follow the instructions below.

The implementation has impact on the following CIP object: Identity Object (01h). It also has impact on how the STC is configured. The following need to be considered.

Vendor ID

The Vendor ID must match the Vendor name in the CT software and the STC.

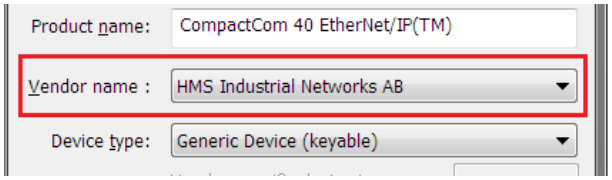


Fig. 6

First time EtherNet/IP vendors may not find their name available from the drop down menu, as it's not certain that the test software has been updated. It is possible to pre-test with any Vendor ID in the list, to reduce the number of errors reported due to Vendor ID mismatch, as long as the Vendor ID is changed in both the device and in the STC before actual conformance testing.

Alternatively, vendors can add the vendor information to the VID.dat file.

Product Code

The Product Code must match the Product Code in the drop down list:

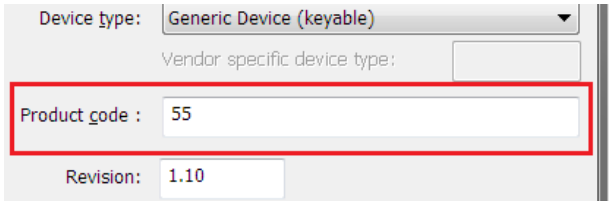


Fig. 7

Revision

The Revision must match the revision field <major>. <minor>.

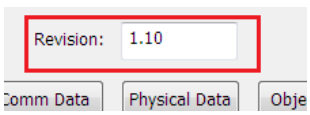


Fig. 8

Serial Number

The current version of CT test does not check serial number.

According to the CIP specification, the combination of Vendor ID and serial number must be unique. It is not permitted to use a custom serial number in combination with the HMS Vendor ID (005Ah).

Product Name

The Product Name must match the Product Name field.

Fig. 9

MAC Address

The MAC address should be listed in the Statement of Conformance. This can be done in the Physical Data section.

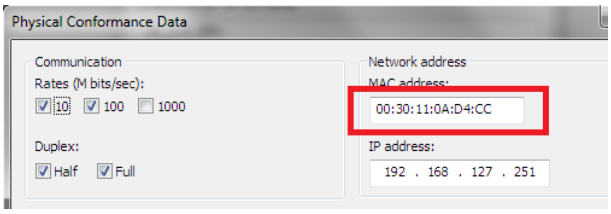


Fig. 10

D Copyright Notices

lwIP is licenced under the BSD licence:

Copyright (c) 2001-2004 Swedish Institute of Computer Science.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Print formatting routines

Copyright (C) 2002 Michael Ringgaard. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the project nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS”AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2002 Florian Schulze.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the authors nor the names of the contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

ftpd.c - This file is part of the FTP daemon for lwIP

FatFs - FAT file system module R0.09b (C)ChaN, 2013

FatFs module is a generic FAT file system module for small embedded systems. This is a free software that opened for education, research and commercial developments under license policy of following trems.

Copyright (C) 2013, ChaN, all right reserved.

The FatFs module is a free software and there is NO WARRANTY. No restriction on use. You can use, modify and redistribute it for personal, non-profit or commercial products UNDER YOUR RESPONSIBILITY. Redistributions of source code must retain the above copyright notice.

Copyright (c) 2016 The MINIX 3 Project.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT

SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Author: David van Moolenbroek <david@minix3.org>

MD5 routines

Copyright (C) 1999, 2000, 2002 Aladdin Enterprises. All rights reserved.

This software is provided “as-is”, without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software. Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

L. Peter Deutsch

ghost@aladdin.com

Copyright 2013 jQuery Foundation and other contributors

http://jquery.com/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE ANDNONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

---

rsvp.js

Copyright (c) 2013 Yehuda Katz, Tom Dale, and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense,

and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

---

libb (big.js)
The MIT Expat Licence.
Copyright (c) 2012 Michael Mclaughlin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The “inih” library is distributed under the New BSD license:
Copyright (c) 2009, Ben Hoyt
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Ben Hoyt nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY BEN HOYT “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL BEN HOYT BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

****
*

open62541 is licensed under the Mozilla Public License v2.0

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/.

To obtain customized changes please contact foss@anybus.com.

****
*

musl as a whole is licensed under the following standard MIT license:

Copyright © 2005-2014 Rich Felker, et al.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

****
*

PCG Random Number Generation for C.

Copyright 2014 Melissa O'Neill <oneill@pcg-random.org>

Licensed under the Apache License, Version 2.0 (“the License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

For additional information about the PCG random number generation scheme, including its license and other licensing options, visit

http://www.pcg-random.org

****
*

queue.h

Copyright (c) 1991, 1993

The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

@(#)queue.h 8.5 (Berkeley) 8/20/94

Format - lightweight string formatting library.
Copyright (C) 2010-2013, Neil Johnson
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

© 2020 HMS Industrial Networks
Box 4126
300 04 Halmstad, Sweden

info@hms.se