

# Anybus<sup>®</sup> CompactCom<sup>™</sup> B40 Modbus Serial

Modbus-TCP

## NETWORK GUIDE

SCM-1202-160 1.0 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

<b>1</b>	<b>Preface .....</b>	<b>3</b>
1.1	About this Document .....	3
1.2	Related Documents .....	3
1.3	Document History .....	3
1.4	Document Conventions .....	3
1.5	Document Specific Conventions .....	4
1.6	Trademark Information .....	4
<b>2</b>	<b>About the Anybus CompactCom B40 Modbus Serial - Modbus-TCP.....</b>	<b>5</b>
2.1	General Information .....	5
2.2	Features .....	5
<b>3</b>	<b>Basic Operation .....</b>	<b>6</b>
3.1	Software Requirements.....	6
3.2	Startup and Identity Customization .....	7
3.3	Device Customization .....	11
3.4	Communication Settings.....	11
3.5	Network Data Exchange .....	12
<b>4</b>	<b>Modbus-TCP Register Implementation .....</b>	<b>13</b>
4.1	Holding Registers (4x) .....	13
4.2	Input Registers (3x) .....	13
4.3	Coils (0x) .....	13
4.4	Discrete Inputs (1x) .....	13
<b>5</b>	<b>Modbus-TCP Functions .....</b>	<b>14</b>
5.1	Read Coils.....	15
5.2	Read Discrete Inputs.....	15
5.3	Read Holding Registers.....	15
5.4	Read Input Registers.....	16
5.5	Write Single Coil .....	16
5.6	Write Single Register .....	16
5.7	Write Multiple Coils.....	16
5.8	Write Multiple Registers.....	17
5.9	Read/Write Multiple Registers.....	17
5.10	Read Device Identification .....	17
<b>6</b>	<b>FTP Server .....</b>	<b>18</b>
6.1	Session Example .....	18

---

<b>7</b>	<b>Web Server .....</b>	<b>19</b>
7.1	General Information .....	19
7.2	Web Pages.....	19
<b>A</b>	<b>Technical Specification.....</b>	<b>23</b>
A.1	Network Status LED .....	23
A.2	Module Status LED .....	24
A.3	LINK/Activity LED 3/4.....	24
A.4	Fatal Error .....	24
A.5	Ethernet Interface.....	24
<b>B</b>	<b>Implementation Details .....</b>	<b>25</b>
B.1	SUP-Bit Definition .....	25
B.2	Anybus State Machine .....	25
B.3	Application Watchdog Timeout Handling.....	25
<b>C</b>	<b>Secure HICP (Secure Host IP Configuration Protocol) .....</b>	<b>26</b>
C.1	General .....	26
C.2	Operation .....	26

# 1 Preface

## 1.1 About this Document

This document is intended to provide a good understanding of the functionality offered by the Anybus CompactCom B40 Modbus Serial - Modbus-TCP.

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 should normally be sufficient to implement a design. However if advanced Modbus-TCP specific functionality is to be used, in-depth knowledge of Modbus-TCP networking internals and/or information from the official Modbus-TCP specifications may be required. In such cases, the persons responsible for the implementation of this product should either obtain the Modbus-TCP 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](http://www.anybus.com/support).

## 1.2 Related Documents

Document	Author	Document ID
Anybus CompactCom 40 Software Design Guide	HMS	HMSI-216-125
Anybus CompactCom B40 Design Guide	HMS	HMSI-27-230
Anybus CompactCom Host Application Implementation Guide	HMS	HMSI-27-334
Anybus CompactCom 40 Modbus-TCP Network Guide	HMS	SCM-1202-027

## 1.3 Document History

Version	Date	Description
1.0	2020-06-26	First release

## 1.4 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](http://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.*

## 1.5 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  $\pm 0.10$  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.

### 1.5.1 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

## 1.6 Trademark Information

Anybus® is a registered trademark of HMS Industrial Networks AB.

All other trademarks are the property of their respective holders.

---

## 2 About the Anybus CompactCom B40 Modbus Serial - Modbus-TCP

### 2.1 General Information

The Anybus CompactCom B40 Modbus Serial - Modbus-TCP is a communication solution for simple industrial field devices. The host application communicates with the product using the Modbus RTU protocol. The Anybus CompactCom B40 Modbus Serial - Modbus-TCP then communicates the data to the network. Typical applications are basic level I/O blocks, temperature controllers, measuring devices, and sensors.

The Anybus CompactCom B40 Modbus Serial - Modbus-TCP 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 B40 Modbus Serial - Modbus-TCP 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](http://www.anybus.com/support).



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.

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

- 10/100 Mbit, full/half duplex operation
- Modbus-TCP server/slave (up to 4 simultaneous connections)
- Max. read process data: 1536 bytes
- Max. write process data: 1536 bytes
- Max. process data (read + write, in bytes): 3072 bytes
- Customizable Identity Information
- Galvanic isolation between the host application and the industrial network available if used with the CompactCom B40 connector board

## **3 Basic Operation**

### **3.1 Software Requirements**

No additional network support code needs to be written in order to support the Anybus CompactCom B40 Modbus Serial - Modbus-TCP, however due to the nature of the Modbus-TCP networking system, certain restrictions must be taken into account:

- Modbus-TCP reset requests are not supported.



## 3.2 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.

### 3.2.1 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
Web Server:	Disabled
FTP Server:	Disabled
Vendor URL:	www.anybus.com
Model Name:	Model Name
User App Name:	User App Name
Serial Number:	0x12345678
Vendor Name:	Vendor Name
Product Name:	Product Name
Firmware Version:	1.2.3

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	0x7B	The data size in this example is 123 bytes
	0x00	
Data	0xF9 0x01 0x00 0x03 0x01 0x00 0x00 0xF9 0x01 0x00 0x06 0x01 0x00 0x00 0xFA 0x01 0x00 0x04 0x0E 0x00 0x77 0x77 0x77 0x2E 0x61 0x6E 0x79 0x62 0x75 0x73 0x2E 0x63 0x6F 0x6D 0xFA 0x01 0x00 0x06 0x0A 0x00 0x4D 0x6F 0x64 0x65 0x6C 0x20 0x4E 0x61 0x6D 0x65 0xFA 0x01 0x00 0x07 0x0D 0x00 0x55 0x73 0x65 0x72 0x20 0x41 0x70 0x70 0x20 0x4E 0x61 0x6D 0x65 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	Disable Web Disable FTP Vendor URL  Model Name  User App Name  Serial Number Vendor Name  Product Name  Firmware Ver.
CRC	0xXX	CRC-16
	0xXX	

### Response

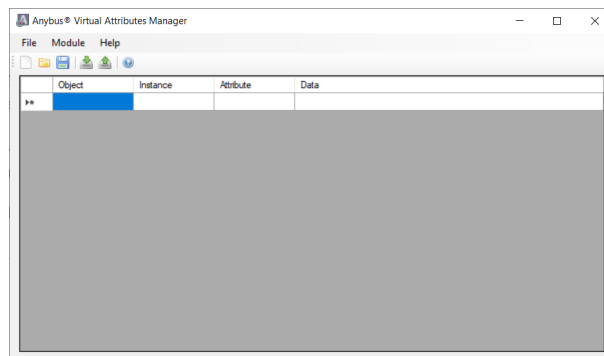
	Value	Note
Modbus Address	0xXX	
Function Code	0x46	FC70
Command	0x02	Set_Attr_Resp
Object	0x01	Anybus Object
Instance	0x01	
	0x00	
Ext0	0x11	Attribute 17
Ext1	0x00	Not used
Data Size	0x00	
	0x00	
CRC	0xXX	CRC-16
	0xXX	



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

### 3.2.2 Virtual Attributes with Anybus Virtual Attributes Manager

1. Start the Anybus Virtual Attributes Manager



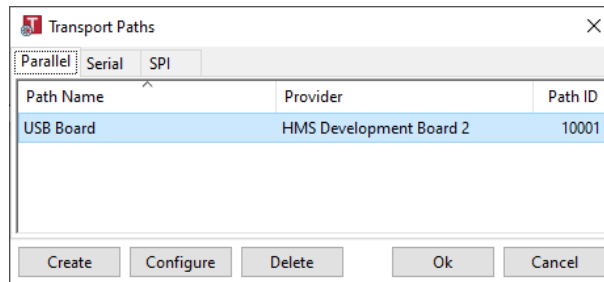
2. Enter the virtual attributes data for the attributes needed. The example below is setting up the attributes with the following values:

Virtual Attribute	Example Value
Web Server:	Disabled
FTP Server:	Disabled
Vendor URL:	www.anybus.com
Model Name:	Model Name
User App Name:	User App Name
Serial Number:	0x12345678
Vendor Name:	Vendor Name
Product Name:	Product Name
Firmware Version:	1.2.3

Object	Instance	Attribute	Data
0xF9	0x0001	0x03	0x00
0xF9	0x0001	0x06	0x00
0xFA	0x0001	0x04	0x77 0x77 0x77 0x2E 0x61 0x6E 0x79 0x62 0x75 0x73 0x2E 0x63 0x6F 0x6D
0xFA	0x0001	0x06	0x4D 0x6F 0x64 0x65 0x6C 0x20 0x4E 0x61 0x6D 0x65
0xFA	0x0001	0x07	0x55 0x73 0x65 0x72 0x20 0x41 0x70 0x70 0x20 0x4E 0x61 0x6D 0x65
0xFF	0x0001	0x03	0x78 0x56 0x34 0x12
0xFF	0x0001	0x08	0x56 0x65 0x6E 0x64 0x6F 0x72 0x20 0x4E 0x61 0x6D 0x65
0xFF	0x0001	0x09	0x50 0x72 0x6F 0x64 0x75 0x63 0x74 0x20 0x4E 0x61 0x6D 0x65
0xFF	0x0001	0x0A	0x01 0x02 0x03

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.

### 3.3 Device Customization

#### 3.3.1 Modbus-TCP Implementation

By default, a “Read Device Identification” request returns the following information:

<b>Vendor Name</b>	“HMS”
<b>Product Code:</b>	“Anybus CompactCom B40 Modbus Serial - Modbus-TCP”
<b>Major Minor Rev.:</b>	The current firmware version of the product
<b>Vendor URL:</b>	(no information returned by default)
<b>Product Name:</b>	(no information returned by default)
<b>Model Name:</b>	(no information returned by default)
<b>User Application Name:</b>	(no information returned by default)

It is possible to customize this information. See [Startup and Identity Customization, p. 7](#) for more information.

### 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 Modbus-TCP 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 <b>Note:</b> 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](http://www.anybus.com/support), or through the Modbus-TCP 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](http://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 Network Data Exchange

### 3.5.1 Process Data

Modbus does not feature a dedicated cyclic data channel in the same sense as many other networks. Process data can, however, still be accessed from the network via dedicated entries in the Modbus register map.

Process data can be accessed on a bit by bit basis (as Coils & Discrete Inputs) - or - as 16 bit entities (Holding Registers & Input Registers).



*For natural reasons, writing to the write process data register area has no effect, and reading unused register locations will return zeroes.*

#### Example

Each 16-bit Modbus register contains 2 bytes from the process data at the corresponding address, i.e. Modbus register N holds process data byte (N\*2) in the low byte and (N\*2 + 1) in the high byte.

Process Data			Modbus Register			
Byte	Type	Value	Register	Value	Comment	
0	UINT16	1234h	0	1234h	-	
1						
2	UINT8	00h	1	FF00h	Two bytes from the process data in one register.	
3	UINT8	FFh				
4	UINT32	11223344h	2	3344h	LSB * A 32-bit type occupies two Modbus * registers. MSB	
5			3	1122h		
6						
7						
8	BOOL[3]	01h	4	0001h	-	
9		00h				
10		01h	5	3401h	-	
11	UINT16	1234h	6	0012h	High byte from unmapped process data is set to zero.	
12						

## 4 Modbus-TCP Register Implementation

### 4.1 Holding Registers (4x)

Range	Contents	Notes
0000h...02FFh	Read Process Data (1536 bytes)	-
0300h...07FFh	Reserved	-
0800h...0AFFh	Write Process Data (1536 bytes)	-
0B00h...0FFFh	Reserved	-
1000h...1002h	Reserved	-
1003h	Process Active Timeout	This value specifies how long the module shall stay in the PROCESS_ACTIVE state, after receiving a Modbus-TCP request. For more information, see <a href="#">Anybus State Machine, p. 25</a> .
1004h	Enter/Exit Idle Mode	0: Not Idle, >0: Idle
1005h...100Fh	Reserved	-
1010h...101Fh	Parameter Number 1	-
1020h...102Fh	Parameter Number 2	
...	...	
7000h...700Fh	Parameter Number 1536	

### 4.2 Input Registers (3x)

Range	Contents	Notes
0000h...02FFh	Write Process Data	-
0300h...07FFh	Reserved	-

### 4.3 Coils (0x)

Range	Contents	Notes
0000h...2FFFh	Read Process Data	-
3000h...7FFFh	Reserved	-

### 4.4 Discrete Inputs (1x)

Range	Contents	Notes
0000h...2FFFh	Write Process Data	-
3000h...07FFFh	Reserved	-

## 5 Modbus-TCP Functions

The following Modbus-TCP functions are implemented in the module:

#	Function
1	Read Coils
2	Read Discrete Inputs
3	Read Holding Registers
4	Read Input Registers
5	Write Single Coil
6	Write Single Register
15	Write Multiple Coils
16	Write Multiple Registers
23	Read/Write Multiple Registers
43/14	Read Device Identification

Exception Codes:

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



## 5.1 Read Coils

**Function Code:** 1  
**Register Type:** 0x (Coils)

### Details

This function is mapped to the Read Process data as follows:

Coil #	Process Data Byte #	Bit #
0000h	0000h	0
0001h		1
0002h		2
0003h		3
...		...
0007h		7
0008h	0001h	0
0009h		1
000Ah		2
000Bh		3
...		...
000Fh		7
...	...	...
2FF8h	05FFh	0
2FF9h		1
2FFAh		2
2FFBh		3
...		...
2FFFh		7

## 5.2 Read Discrete Inputs

**Function Code:** 2  
**Register Type:** 1x (Discrete Inputs)

### Details

This function is mapped to the Write Process data; the mapping is otherwise identical to that of the “Read Coils” function described above.

## 5.3 Read Holding Registers

**Function Code:** 3  
**Register Type:** 4x (Holding Registers)

### Details

Mapped to Read- and Write Process Data, ADIs, and configuration registers. It is allowed to read parts of a larger Anybus CompactCom data type; it is also allowed to read multiple ADIs using a single request.

## 5.4 Read Input Registers

**Function Code:** 4  
**Register Type:** 3x (Input Registers)

### Details

Mapped to Write Process Data and diagnostic events.

## 5.5 Write Single Coil

**Function Code:** 5  
**Register Type:** 0x (Coils)

### Details

This function is mapped to the Read Process data, and the mapping is identical to that of the “Read Coils” function described above.

## 5.6 Write Single Register

**Function Code:** 6  
**Register Type:** 4x (Holding Registers)

### Details

Mapped to Read- and Write Process Data, ADIs and configuration registers. ADIs must be written as a whole, however the Process Data area accepts writes of any size.

## 5.7 Write Multiple Coils

**Function Code:** 15  
**Register Type:** 0x (Coils)

### Details

This function is mapped to the Read Process data, and the mapping is identical to that of the “Read Coils” function described above.

## 5.8 Write Multiple Registers

Function Code:	16
Register Type:	4x (Holding Registers)

### Details

Mapped to Read- and Write Process Data, ADIs and configuration registers.



*ADIs must be written as a whole, but the Process Data area accepts writes of any size.*

---

## 5.9 Read/Write Multiple Registers

Function Code:	23
Register Type:	4x (Holding Registers)

### Details

Mapped to Read- and Write Process Data, ADIs and configuration registers.



*ADIs must be written as a whole, but the Process Data area accepts writes of any size.*

*It is allowed to read parts of larger data types, and to read multiple ADIs using a single request.*

*The write operation is performed before the read. If there is an overlap in the read and write ranges, the newly written data will be returned by the read operation.*

---

## 5.10 Read Device Identification

Function Code:	43 (subcode 14)
Register Type:	-

### Details

Basic and regular device identification objects are supported according to the Modbus specification. Extended device identification objects are not supported.

## 6 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. 7](#).



*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.

### 6.1 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.

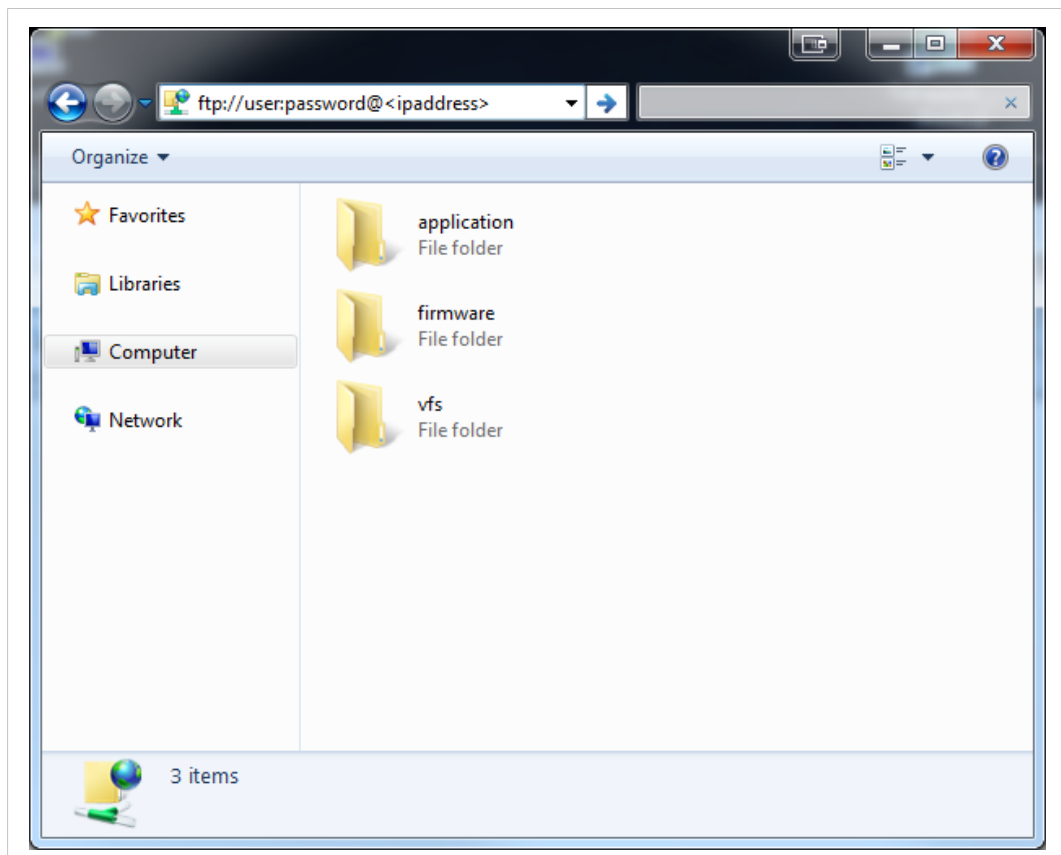


Fig. 1

## 7 Web Server

### 7.1 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. 7](#).

See also...

- [FTP Server, p. 18](#)

### 7.2 Web Pages

The web pages provide access to:

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

#### 7.2.1 Network Configuration

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

The screenshot displays the Anybus CompactCom web interface in a browser window. The address bar shows the URL 192.168.1.2. The interface has a dark blue header with the Anybus logo and the product name. A left sidebar contains a 'MODULE' menu with options: Overview, Parameters, NETWORK, Status, Configuration, SERVICES, and SMTP. The main content area is titled 'IP Configuration' and includes a table for network settings. Below this is an 'Ethernet Configuration' section. Both sections have a 'Save settings' button. The footer contains copyright information for HMS Industrial Networks and the slogan 'Connecting Devices™'.

IP Configuration	
DHCP	Enabled
IP Address	192.168.1.2
Subnet Mask	255.255.255.0
Gateway Address	0.0.0.0
Host Name	
Domain name	
DNS Server #1	0.0.0.0
DNS Server #2	0.0.0.0

Save settings

Ethernet Configuration	
Port 1	Auto
Port 2	Auto

Save settings

© 2013 HMS Industrial Networks - All rights reserved Connecting Devices™

Fig. 2

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
Port 2	Default value: auto

## 7.2.2 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



# A Technical Specification

## A.1 Network Status LED

LED State	Description
Off	No IP address or in state EXCEPTION
Green	At least one Modbus message received
Green, flashing	Waiting for first Modbus message
Red	IP address conflict detected, FATAL ERROR
Red, flashing	Connection timeout. No Modbus message has been received within the configured “process active timeout” time

A test sequence is performed on this LED during startup.

## A.2 Module Status LED

LED State	Description
Off	No power
Green	Normal operation
Red	Major fault (including Anybus exception), FATAL
Red, flashing	Minor fault
Alternating red/green	Firmware update from file system in progress

A test sequence is performed on this LED during startup.

## A.3 LINK/Activity LED 3/4

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)

## A.4 Fatal Error

If both the Network Status LED and the Module Status LED are red, a fatal error has occurred.

## A.5 Ethernet Interface

The Ethernet interface 10/100Mbit, full or half duplex operation.

## B Implementation Details

### B.1 SUP-Bit Definition

The supervised bit (SUP) indicates that the network participation is supervised by another network device.

This bit is set when in PROCESS\_ACTIVE, and only if the Process active timeout value is greater than zero (0).

### B.2 Anybus State Machine

The table below describes how the Anybus state machine relates to the Modbus-TCP network

Anybus State	Implementation	Comment
WAIT_PROCESS	Waiting for Modbus requests. The module shifts to PROCESS_ACTIVE when a Modbus request is received.	-
ERROR	IP address conflict.	-
PROCESS_ACTIVE	The module shifts to WAIT_PROCESS if no requests are received within the time stated by Process Active Timeout.	-
IDLE	The IDLE state can be entered/exited by writing to the Modbus Enter/Exit idle state register at address 1004h.	
EXCEPTION	Any Modbus requests will be ignored.	

### B.3 Application Watchdog Timeout Handling

Upon detection of an application watchdog timeout, the module will cease network participation and shift to state EXCEPTION. No other network specific actions are performed.

## C Secure HICP (Secure Host IP Configuration Protocol)

### C.1 General

The Anybus CompactCom B40 Modbus Serial - Modbus-TCP 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](http://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).

### C.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.

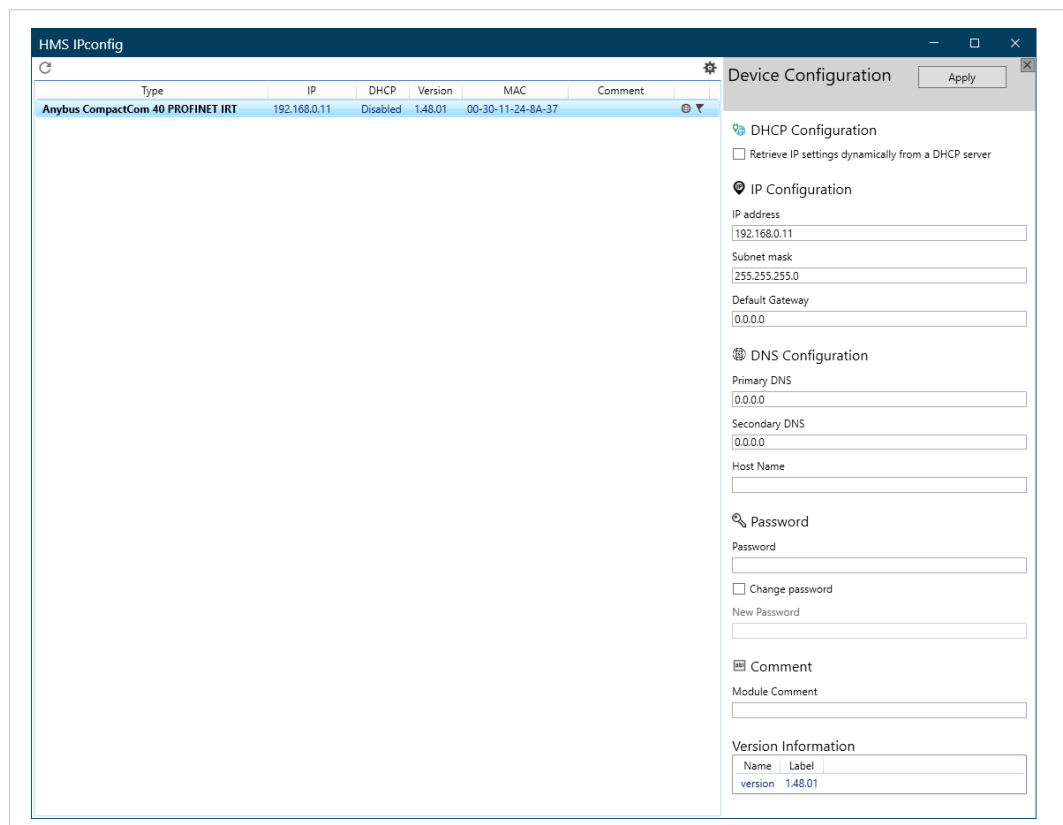


Fig. 3

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.

**This page intentionally left blank**

