Anybus® CompactCom™ 40 Modbus Serial

EtherCAT

NETWORK GUIDE

SCM-1202-157 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

This page intentionally left blank

1 Preface

1.1 About this Document

This document is intended to provide a good understanding of the functionality offered by the Anybus CompactCom 40 Modbus Serial - EtherCAT.

The reader of this document is expected to be familiar with high level software design and communication systems in general. The information in this user manual should normally be sufficient to implement a design. However if advanced EtherCAT specific functionality is to be used, in-depth knowledge of EtherCAT networking internals and/or information from the official EtherCAT specifications may be required. In such cases, the persons responsible for the implementation of this product should either obtain the EtherCAT 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 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 EtherCAT Network Guide	HMS	SCM-1202–034
SEMI Device FW Upgrade, V1.0.0	EtherCAT Technology Group	ETG.5003.2 S (R) V1.0.0
IEC 61158-6	IEC	
EtherCAT Slave Information Specification, V1.0.10	EtherCAT Technology Group	ETG.2000 S (R) V1.0.10
CiA Draft Standard 301 v4.02	CAN in Automation	

1.3 Document History

Version	Date	Description
1.0	2020-04-22	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. 4*

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.

1.5 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 ±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 Trademarks

Anybus® is a registered trademark of HMS Industrial Networks.

EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

All other trademarks are the property of their respective holders.

2 About the Anybus CompactCom 40 Modbus Serial - EtherCAT

2.1 General Information

The Anybus CompactCom 40 Modbus Serial - EtherCAT is a communication solution for simple industrial field devices. The Modbus master communicates with the product using the Modbus RTU protocol. The Anybus CompactCom 40 Modbus Serial - EtherCAT 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 - EtherCAT share footprint and electrical interface with the other members of the product family, independent of fieldbus or network. The application connector provides an interface between the Modbus master (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.

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 (EtherCAT)

- CANopen over EtherCAT (CoE)
  - Complete Access support
- DS301 compliant
- Galvanically isolated bus electronics
- Network Identity customization
- Up to 1486 bytes of fast cyclic I/O in each direction
- Network cycle time down to 100 µs

If TwinCAT 3, or a later version than 2.11, is used, the max amount of process data will be 1473 bytes, due to limitations in the tool.

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 - EtherCAT. The application sets up the Modbus RTU communication, and the Anybus CompactCom maps the process data to the industrial network/fieldbus.

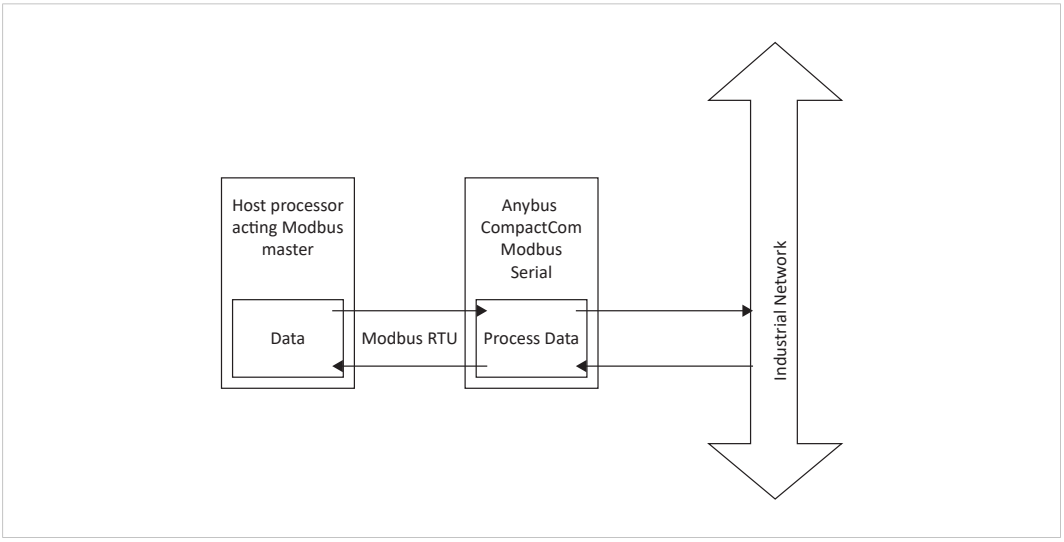


Fig. 1

3 Basic Operation

3.1 General Informaton

3.1.1 EtherCAT Slave Interface (ESI) File

Each device on EtherCAT is associated with a EtherCAT Slave information (ESI) file in XML format, which holds a description of the device and its functions.

To ensure interoperability and to reduce the complexity for the end user, it is strongly recommended to create a custom ESI file to match the final implementation of the product. To aid with the ESI file creation, HMS provides a tool called HMS EtherCAT ESI Generator, which is freely downloadable from the Anybus CompactCom 40 Modbus Serial - EtherCAT product page on www.anybus.com.

The EtherCAT Technology Group (ETG) requires that the Vendor ID is changed to reflect the vendor of the end product. The following scenarios, among others, may require additional changes to the EtherCAT Slave information file.

- The use of a custom Product Code.
- The use of an own Vendor ID.
- Change of the product revision.
- The use of Ethernet over EtherCAT (EoE).

Note that deviations from the generic ESI file requires the use of custom Product Codes apart from the required custom Vendor ID.

3.1.2 Device Identity

In a generic implementation (i.e. no network specific support is implemented) the module will appear as a generic HMS device with the following identity information:

Object Entry	Value
Vendor ID	E000 001Bh (HMS Industrial Networks Secondary Vendor ID, has to be replaced by the Vendor ID of the end product vendor)
Product Code	0000 0036h
Device Name	Anybus CompactCom 40 EtherCAT
Serial Number	(Assigned during manufacturing)

3.2 EtherCAT Implementation Details

3.2.1 General Information

The module implements a full EtherCAT slave with the following basic properties:

Application Layer:	CANopen over EtherCAT
FMMUs.	4
Sync Managers.	4
RAM Size:	16 kByte

See also...

- *CANopen over EtherCAT Implementation Details, p. 10*

3.2.2 Sync Managers

The module features four Sync Managers:

Sync Manager 0	Used for mailbox write transfers (Master to Slave). The module has a configurable write mailbox size with default size of 276 bytes, corresponding to 255 bytes plus relevant protocol headers and padding.
Sync Manager 1	Used for mailbox read transfers (Slave to Master). The module has a configurable read mailbox size with default size of 276 bytes, corresponding to 255 bytes plus relevant protocol headers and padding.
Sync Manager 2	Contains the RxPDOs (in practice, Sync Manager 2 holds the Read Process Data).
Sync Manager 3	Contains the TxPDOs (in practice, Sync Manager 3 holds the Write Process Data).

3.2.3 FMMUs

There are four FMMUs. The EtherCAT master can use the FMMUs freely for any purpose.

3.2.4 Addressing Modes

As a full EtherCAT slave, the module supports the following addressing modes:

- position addressing
- node addressing
- logical addressing

3.2.5 Watchdog Functionality

Output I/O Sync Manager Watchdog

If enabled, this watchdog monitors the PDO communication towards the Anybus module. If the master doesn't update the Read Process Data within the specified time period, this will trigger a timeout condition in the module, causing it to shift from OPERATIONAL to SAFE-OPERATIONAL.

The sync manager watchdog is enabled by default in the ESI file, with a default time period of 100 ms.

PDI Watchdog

PDI watchdog functionality is supported.

3.3 CANopen over EtherCAT Implementation Details

3.3.1 General Information

The module implements CANopen over EtherCAT. The object implementation is based on the DS301 communication profile.

See also...

- *Data Exchange, p. 15*
- *Object Dictionary (CANopen over EtherCAT), p. 18*

3.3.2 Implemented Services

The module implements the following CANopen services:

Service	Description
SDO Download Expedited	Writes up to four octets to the slave
SDO Download Normal	Writes up to a negotiated number of octets to the slave
Download SDO Segment	Writes additional data if the object size exceeds the negotiated no. of octets
SDO Upload Expedited	Reads up to four octets from the slave
SDO Upload Normal	Reads up to a negotiated number of octets from the slave
Upload SDO Segment	Reads additional data if the object size exceeds the negotiated no. of octets
Abort SDO Transfer	Server abort of service in case of an erroneous condition
Get OD List	Reads a list of available indexes
Get Object Description	Reads details of an index
Get Entry Description	Reads details of a subindex
Emergency	Reports unexpected conditions

3.4 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.

3.4.1 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:	0xE000001B
Product Code:	0x00000036
Major Revision:	1
Minor Revision:	2
EoE:	Disabled
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	0x75	The data size in this example is 117 bytes
	0x00	
Data	0xF5 0x01 0x00 0x01 0x04 0x00 0x1B 0x00 0x00 0xE0 0xF5 0x01 0x00 0x02 0x04 0x00 0x36 0x00 0x00 0x00 0xF5 0x01 0x00 0x03 0x02 0x00 0x01 0x00 0xF5 0x01 0x00 0x04 0x02 0x00 0x02 0x00 0xF5 0x01 0x00 0x11 0x01 0x00 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 Major Rev. Minor Rev. Disable EoE Disable Web Disable FTP Serial Number Vendor Name Product Name Firmware Ver. Hardware Ver.
CRC	0xXX	CRC-16
	0xXX	

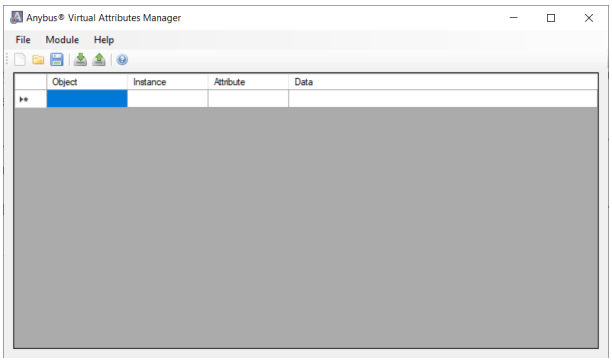
Response

	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.

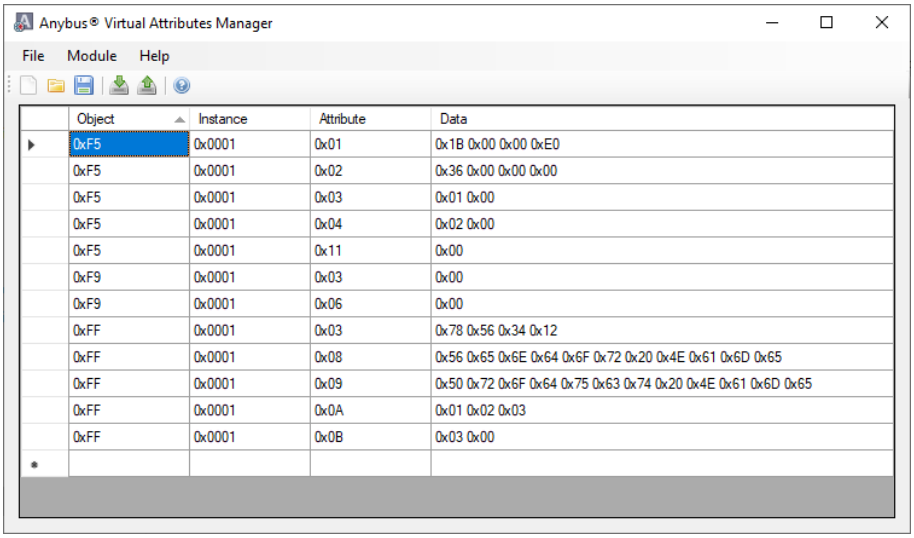
3.4.2 Virtual Attributes with Anybus Virtual Attributes Manager

1. Start the Anybus Virtual Attributes Manager



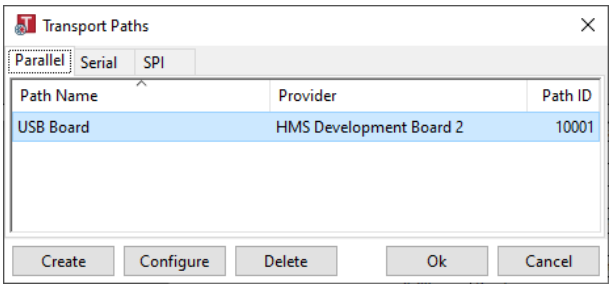
2. Enter the virtual attributes data for the attributes needed. The example below is setting up the attributes with the following values:

Virtual Attribute	Example Value
Vendor ID:	0xE000001B
Product Code:	0x00000036
Major Revision:	1
Minor Revision:	2
EoE:	Disabled
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.

3.5 Data Exchange

3.5.1 Parameter Data

Parameter Data can be accessed from the network via dedicated object entries in the Manufacturer Specific range and the Profile range (2001h - FFFFh). The SDO information protocol allows nodes to retrieve the name and data type of the parameter.

Manufacturer and Profile Specific Objects

General

Each object entry in the manufacturer specific range (2001h...FFFFh) corresponds to a Modbus register, i.e. network accesses to these objects result in access towards the corresponding Modbus register. In case of an error, a descriptive abort code will be returned.

Network Data Format

Data is translated between the native network format and the Anybus CompactCom 40 data format as follows:

Anybus Data Type	Network Data Type	
UINT8	UNSIGNED8	
UINT16	UNSIGNED16	

3.5.2 Process Data

Modbus register	Content	Comment
0x5100	Data Type	0x0004 (UINT8) will result in objects of EtherCAT data type 0x0005 (UNSIGNED8, USINT). 0x0005 (UINT16) will result in objects of EtherCAT data type 0x0006 (UNSIGNED16, UINT).
0x5102	No of Write Parameters	Maximum 1486 parameters when using data type UINT8. Maximum 743 parameters when using data type UINT16.
0x5103	No of Read Parameters	Maximum 1486 parameters when using data type UINT8. Maximum 743 parameters when using data type UINT16.
0x0000 – (Depending on Data type and No of Parameters)	Write Process Data	The first write parameter is represented in object entry 0x2001, sub-index 0, the second write parameter is represented in object entry 0x2002, sub-index 0 and so on. This is valid regardless of the used data type.
0x1000 – (Depending on Data type and No of Parameters)	Read Process Data	The first read parameter is represented in object entry (0x2001 + NumberOfWriteParameters), sub-index 0, the second read parameter is represented in object entry (0x2002 + NumberOfWriteParameters), sub-index 0 and so on. This is valid regardless of the used data type.
0x5004	Network Type	0x0087 (EtherCAT) This register is used to identify the connected module.

The module supports up to 6 TxPDOs and up to 6 RxPDOs, each supporting up to 254 SDO mappings. Each SDO equals one Process Data mapped parameter.

Preferably, the EtherCAT Slave Information file should be altered to match the actual Process Data implementation. This is not a general requirement, but it has a positive impact on compatibility with 3rd party masters.

See also...

- *Standard Objects, p. 18*
- *Manufacturer and Profile Specific Objects, p. 15*

Example 1

No of Write parameters: 3 (0x5102)

No of Read parameters: 4 (0x5103)

Data type: (0x0004, UINT8)

Write parameter No	Modbus register	(CANopen over EtherCAT Object entry)	TxPDO data byte offset
1	0x0000, LSB	0x2001:0	0x00
2	0x0000, HSB	0x2002:0	0x01
3	0x0001, LSB	0x2003:0	0x02

Total TxPDO length: 3 bytes

Read parameter No	Modbus register	XXX Object entry	RxPDO data byte offset
1	0x1000, LSB	0x2004:0	0x00
2	0x1000, HSB	0x2005:0	0x01
3	0x1001, LSB	0x2006:0	0x02
4	0x1001, HSB	0x2007:0	0x03

Total RxPDO length: 4 bytes

Example 2

No of Write parameters: 3 (0x5102)

No of Read parameters: 4 (0x5103)

Data type: (0x0005, UINT16)

Write parameter No	Modbus register	Object entry	TxPDO data byte offset
1	0x0000	0x2001:0	0x00
2	0x0001	0x2002:0	0x02
3	0x0002	0x2003:0	0x04

Total TxPDO length: 6 bytes.

The data written to the Modbus registers is swapped to little endian before it is sent on EtherCAT.

Read parameter No	Modbus register	Object entry	RxPDO data byte offset
1	0x1000	0x2004:0	0x00
2	0x1001	0x2005:0	0x02
3	0x1002	0x2006:0	0x04
4	0x1003	0x2007:0	0x06

Total RxPDO length: 8 bytes.

The data sent from the master on EtherCAT is swapped to big endian when written to the Modbus registers.

3.6 Network Reset Handling

3.6.1 Restore Manufacturer Parameters to Default

Upon receiving a “Restore Manufacturer Parameters to Default” request from the network, the module will set factory default reset bit in the status register.

A factory default reset can only be performed in the EtherCAT state PREOPERATIONAL. Performing a reset in another state than PREOPERATIONAL will generate SDO abort code 08000020h (invalid state).

See also...

- *Standard Objects, p. 18*, entry 1011h (‘Restore Parameters’)

3.7 Device ID

The Device ID is used by the EtherCAT master to explicitly identify a slave. This is useful when changing a faulty device during runtime, a so called HotConnect application. A preconfigured device can be entered into the network, and its Device ID can be set to the same Device ID as the faulty device was appointed.

It is also useful to prevent cable swapping when there are two or more identical devices on the network.

The Device ID range is 1... 65535. Address 0 indicates that the device has yet to be configured. The value can be set using Application switch 1, register 0x5200.

4 Object Dictionary (CANopen over EtherCAT)

4.1 Standard Objects

4.1.1 General

The standard object dictionary is implemented according to the DS301 communication profile.

4.1.2 Object Entries

Index	Object Name	Subin-dex	Description	Type	Access	Notes
1000h	Device Type	00h	Device Type	U32	RO	Default 0000 0000h (No profile)
1001h	Error register	00h	Error register	U8	RO	
1003h	Pre-defined error field	00h	Number of errors	U8	RW	
		01h...0-5h	Error field	U32	RO	
1008h	Manufacturer device name	00h	Manufacturer device name	Visible string	RO	These entries are managed through virtual attributes, see *Startup and Identity Customization, p. 11*
1009h	Manufacturer hardware version	00h	Manufacturer hardware version	Visible string	RO	
100Ah	Manufacturer software version	00h	Manufacturer Software version	Visible string	RO	
1011h	Restore parameters	00h	Largest sub index supported	U8	RO	01h
		01h	Restore all default parameters	U32	RW	See *Restore Manufacturer Parameters to Default, p. 17* for more information
1018h	Identity object	00h	Number of entries	U8	RO	Number of entries
		01h	Vendor ID	U32	RO	These entries are managed through virtual attributes, see *Startup and Identity Customization, p. 11*
		02h	Product Code	U32	RO	
		03h	Revision Number	U32	RO	
		04h	Serial Number	U32	RO	

Index	Object Name	Subin-dex	Description	Type	Access	Notes
1600h - 1xxxh	Receive PDO mapping	00h	No. of mapped application objects in PDO	U8	RO	No. of mapped objects (0.. 254), see *Process Data, p. 15* for more information.
		01h	Mapped object #1	U32	RO	-
		02h	Mapped object #2	U32	RO	-
		...	...	...	...	-
		NNh	Mapped object #NN	U32	RO	-
1A00h - 1xxxh	Transmit PDO mapping	00h	No. of mapped application objects in PDO	U8	RO	No. of mapped objects (0.. 254), see *Process Data, p. 15* for more information.
		01h	Mapped object #1	U32	RO	-
		02h	Mapped object #2	U32	RO	-
		...	...	...	...	-
		NNh	Mapped object #NN	U32	RO	-
1C00h	Sync Manager Communication Type	00h	Number of entries	U8	RO	4
		01h	Mailbox wr	U8	RO	1
		02h	Mailbox rd	U8	RO	2
		03h	Process Data out	U8	RO	3
		04h	Process Data in	U8	RO	4
1C12h	Sync Manager Rx PDO Assign	00h	No. of assigned PDOs	U8	RO	
		01h - NNh	Assigned PDO	U16	RO	
1C13h	Sync Manager Tx PDO Assign	00h	No. of assigned PDOs	U8	RO	
		01h - NNh	Assigned PDO	U16	RO	

Index	Object Name	Subin-dex	Description	Type	Access	Notes
1C32h	Output SyncManager Parameter	00h	Max subindex supported	U8	RO	12 (0Bh)
		01h	Sync mode	U16	RO/RW	00h: Free Run
		02h	Cycle time	U32	RW	Cycle time in nanoseconds
		03h	Shift time	U32	RW	Shift time in nanoseconds
		04h	Synchronization Types supported	U16	RO	Bit 0 set: FREE_RUN supported
		05h	Minimum cycle time	U32	RO	Minimum cycle time in nanoseconds.
		06h	Output Calc and Copy Time	U32	RO	Output Calc and Copy Time in nanoseconds.
		09h	Delay time	U32	RO	Delay time in nanonseconds. Always set to 0.
		0Ch	Cycle Time Too Small	U16	RO	Cycle time to small
1C33h	Input SyncManager Parameter	00h	Max subindex supported	U8	RO	12 (0Bh)
		01h	Sync mode	U16	RO/RW	00h: Free Run
		02h	Cycle time	U32	RW	Cycle time in nanoseconds, same value as 1C32h, subindex 2.
		03h	Shift time	U32	RW	Shift time in nanoseconds.
		04h	Synchronization Types supported	U16	RO	Bit 0 set: FREE_RUN supported
		05h	Minimum cycle time	U32	RO	Minimum cycle time in nanoseconds, same value as 1C32h, subindex 5.
		06h	Input Calc and Copy Time	U32	RO	Input Calc and Copy Time in nanoseconds.
		0Ch	Cycle Time Too Small	U16	RO	Cycle time to small, same value as 1C32h, subindex 12 (0Bh).

This page intentionally left blank

A Technical Specification

A.1 LED Indications

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

A.1.1 RUN LED

This LED reflects the status of the EtherCAT device.

LED State	Indication	Description
Off	INIT	EtherCAT device in 'INIT'-state (or no power)
Green	OPERATIONAL	EtherCAT device in 'OPERATIONAL'-state
Green, blinking	PRE-OPERATIONAL	EtherCAT device in 'PRE-OPERATIONAL'-state
Green, single flash	SAFE-OPERATIONAL	EtherCAT device in 'SAFE-OPERATIONAL'-state
Flickering	BOOT	The EtherCAT device is in 'BOOT' state
Red	(Fatal Event)	If RUN and ERR turn red, this indicates a fatal event, forcing the bus interface to a physically passive state. Contact HMS technical support.

A.1.2 ERR LED

This LED indicates EtherCAT communication errors etc.

LED State	Indication	Description
Off	No error	No error (or no power)
Red, blinking	Invalid configuration	State change received from master is not possible due to invalid register or object settings.
Red, single flash	Unsolicited state change	Slave device application has changed the EtherCAT state autonomously.
Red, double flash	Sync Manager watchdog timeout	See *Watchdog Functionality, p. 10* for more information.
Red	Application controller failure	Anybus module in EXCEPTION. If RUN and ERR turn red, this indicates a fatal event, forcing the bus interface to a physically passive state. Contact HMS technical support.
Flickering	Booting error detected	E.g. due to firmware download failure.

A.1.3 Link/Activity

These LEDs indicate the EtherCAT link status and activity.

LED State	Indication	Description
Off	No link	Link not sensed (or no power)
Green	Link sensed, no activity	Link sensed, no traffic detected
Green, flickering	Link sensed, activity	Link sensed, traffic detected

A.2 Functional Earth (FE) Requirements

In order to ensure proper EMC behavior, the module must be properly connected to functional earth via the FE pad / FE mechanism described in the general Anybus CompactCom B40 User Manual.

HMS Industrial Networks does not guarantee proper EMC behaviour unless these FE requirements are fulfilled.

A.3 Power Supply

A.3.1 Supply Voltage

The module requires a regulated 3.3V power source as specified in the general Anybus CompactCom B40 User Manual.

A.3.2 Power Consumption

The Anybus CompactCom 40 Modbus Serial - EtherCAT is designed to fulfil the requirements of a Class B module. For more information about the power consumption classification used on the Anybus CompactCom platform, consult the Anybus CompactCom B40 User Manual.

The current hardware design consumes up to 430 mA.

It is strongly advised to design the power supply in the host application based on the power consumption classifications described in the general Anybus CompactCom Hardware Design Guide, and not on the exact power requirements of a single product.

In line with HMS policy of continuous product development, we reserve the right to change the exact power requirements of this product without prior notification. Note however that in any case, the Anybus CompactCom 40 Modbus Serial - EtherCAT will remain as a Class B module.

A.4 Environmental Specification

Consult the Anybus CompactCom B40 User Manual for further information.

A.5 EMC Compliance

Consult theAnybus CompactCom B40 User Manual for further information.

B Secure HICP (Secure Host IP Configuration Protocol)

B.1 General

The Anybus CompactCom 40 Modbus Serial - EtherCAT 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.

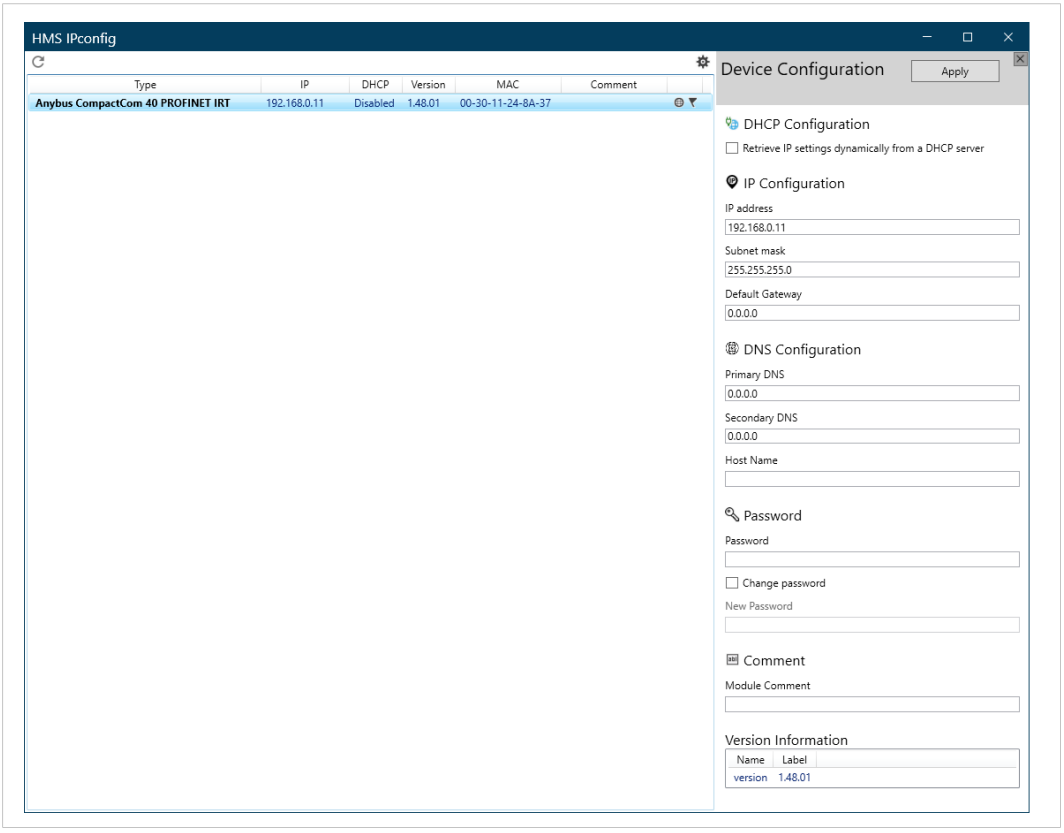


Fig. 2

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 Copyright Notices

Print formatting routines

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.

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.

lwIP is licenced under the BSD licence:

rsvp.js

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:

MD5 routines

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

This page intentionally left blank

Box 4126
300 04 Halmstad, Sweden

info@hms.se