Anybus® CompactCom™ B40

Modbus Serial - DeviceNet

NETWORK GUIDE

SCM-1202-159 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

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 - DeviceNet.

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 DeviceNet specific functionality is to be used, in-depth knowledge of DeviceNet networking internals and/or information from the official DeviceNet specifications may be required. In such cases, the persons responsible for the implementation of this product should either obtain the DeviceNet 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
DeviceNet Specification	ODVA	-
CIP specification, Volumes 1 (CIP Common)	ODVA	-

1.3 Document History

Version	Date	Description
1.0	2020-06-18	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

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 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 - DeviceNet

2.1 General Information

The Anybus CompactCom B40 Modbus Serial - DeviceNet provides a quick and simple solution to sending process data easy and efficiently between a Modbus RTU network and a DeviceNet network.

The Anybus CompactCom B40 Modbus Serial - DeviceNet share footprint and electrical interface with the other members of the product family. The product has two connectors that provides communication with the host application board. The host application connector provides an interface between the host application (Modbus RTU) and the Anybus CompactCom, while the network connector provides access to DeviceNet. The Anybus CompactCom acts as a Modbus RTU slave on the host application side.

All dimensions expressed in this document are stated in millimeters and have a tolerance of ±0.10 mm unless stated otherwise.

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

- CIP Parameter Object Support
- Explicit messaging
- UCMM Capable
- Bit-strobed I/O
- Change-of-state / Cyclic I/O
- Polled I/O
- Galvanic isolation between the host application and the industrial network available if used with the CompactCom B40 connector board
- Max. read process data: 512 bytes
- Max. write process data: 512 bytes
- Max. process data (read + write, in bytes): 1024 bytes
- Automatic Baud Rate Detection

3 Basic Operation

3.1 Electronic Data Sheet (EDS)

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

See also...

- *Startup and Identity Customization, p. 7*
- *Parameter Object (0Fh), p. 28* (CIP object)

> **!** To comply with CIP specification requirements, custom EDS-implementations require a new Vendor ID and/or Product Code.
>
> To obtain a Vendor ID, contact the ODVA.

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
Vendor ID:	0x005A
Product Code:	0x003F
Serial Number:	0x12345678
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	0x35	The data size in this example is 53 bytes
	0x00	
Data	0xFC 0x01 0x00 0x01 0x02 0x00 0x5A 0x00 0xFC 0x01 0x00 0x03 0x02 0x00 0x3F 0x00 0xFF 0x01 0x00 0x03 0x04 0x00 0x78 0x56 0x34 0x12 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	Vendor ID Product Code Serial Number Product Name Firmware Ver.
CRC	0xXX	CRC-16
	0xXX	

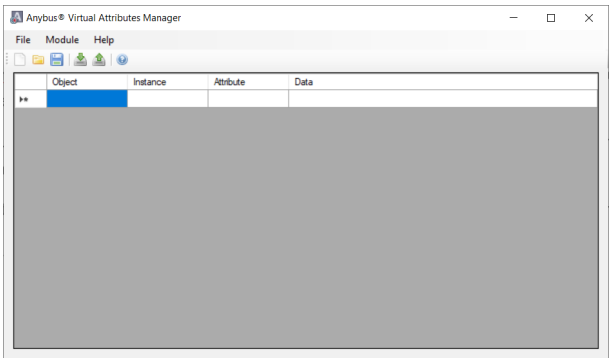
Response

	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.

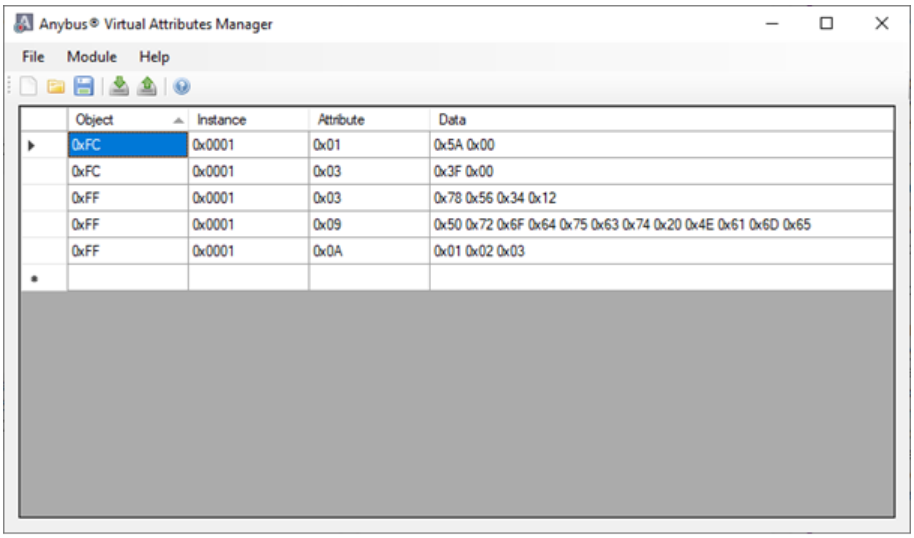
3.2.2 Virtual Attributes with Anybus Virtual Attributes Manager

1. Start the Anybus Virtual Attributes Manager



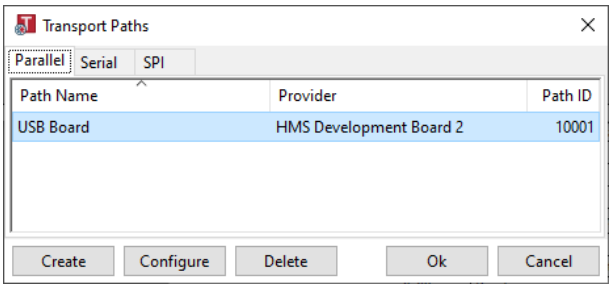
2. Enter the virtual attributes data for the attributes needed. The example below is setting up the attributes with the following values:

Virtual Attribute	Example Value
Vendor ID:	0x005A
Product Code:	0x003F
Serial Number:	0x12345678
Product Name:	Product Name
Firmware Version:	1.2.3



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.

3.3 Communication Settings

The node address is configured using 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 1 value	Used node address settings	Comment
0-63	Node address X	X is the “Application switch 1” value. The resulting node address is stored and will still be used if “Application switch 1” value is set to 64-255.
64-255	Use currently stored node address	Factory default settings: Node address: 63 **Note:** Node address may be set from the network.

The baud rate is configured by the “Application switch 2” 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 value	Used baud rate settings	Comment
0-3	Baud rate: X	X is the “Application switch 2” value. Resulting baud rate is stored and will still be used if “Application switch 2” value is set to 4-255. 0: 125 kbps 1: 250 kbps 2: 500 kbps 3: Auto baud
4-255	Use currently stored baud rate	Factory default settings: Baud rate: 3 (auto baud) **Note:** Baud rate may be set from the network.

The communication settings can also be changed from the DeviceNet network. Note that changing the configuration through DeviceNet will only be possible if “Application switch 1” is set to 64-255 and if “Application switch 2” is set to 4-255 respectively.

3.4 Network Data Exchange

3.4.1 Process Data

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

3.4.2 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

4 CIP Objects

4.1 General Information

This chapter specifies the CIP-objects implementation in the module. The objects described herein can be accessed from the network, but not by the host application.

Mandatory Objects:

Vendor Specific Objects:

4.2 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. 7*.

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 Attributes

#	Name	Access	Type	Value/Description
1	Vendor ID	Get	UINT	005Ah (HMS Industrial Networks)
2	Device Type	Get	UINT	002Bh (Generic Device)
3	Product Code	Get	UINT	003Fh (Anybus CompactCom 40 DeviceNet(TM))
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 Industrial Networks AB)
7	Product Name	Get	SHORT_STRING	“Anybus CompactCom 40 DeviceNet(TM)”
11	Active language	Set	Struct of: USINT USINT USINT	Requests sent to this instance are forwarded to the Application Object. 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 Application Object and translated to CIP standard.

Device Status

bit(s)	Name		
0	Module Owned		
1	(reserved, set to 0)		
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, set to 0)		
4... 7	Extended Device Status:		
	Value:	Meaning:	Priority (higher number means higher priority):
	0010b	Faulted I/O Connection	3
	0011b	No I/O connection established	0
	0101b	Major fault	4
	0110b	Connection in Run mode	1
	0111b	Connection in Idle mode	2
	(other)	(reserved)	
8	Set for minor recoverable faults		These bits represent a combination of network specific faults (see CIP specifications) and faults generated by the module.
9	Set for minor unrecoverable faults		
10	Set for major recoverable faults		
11	Set for major unrecoverable faults		
12... 15	(reserved, set to 0)		

Service Details: Reset

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

There are two types of network reset requests on DeviceNet:

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 an "out of box" configuration and performs a reset, refer to Modbus register 0x0FFF for actions needed to be taken by the application.

4.3 Message Router (02h)

Category

Extended

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

-

4.4 DeviceNet Object (03h)

Category

Extended

Object Description

This object provides means for configuring the DeviceNet interface of the module.

Supported Services

Class:	Get_Attribute_Single
Instance:	Get_Attribute_Single
	Set_Attribute_Single
	Allocate Master/Slave Connection Set (4Bh)
	Release Master/Slave Connection Set (4Ch)

Class Attributes

#	Name	Access	Type	Value
1	Revision	Get	UINT	0002h (Object revision)

Instance Attributes

#	Name	Access	Type	Comments
1	MAC ID	Get/Set	USINT	Currently used node address (0 - 63). Set access right is conditional. For more information, see *Communication Settings, p. 10*.
2	Baud Rate	Get/Set	USINT	Value — Baud Rate 0 — 125 kbps 1 — 250 kbps 2 — 500 kbps Set access right is conditional. For more information, see *Communication Settings, p. 10*. Setting this attribute will also affect attribute #100 (Disable auto baud).
3	BOI	Get/Set	BOOL	Defines CAN controller action in case of a Bus-Off interrupt. A Bus-Off Interrupt is generated from the underlying CAN layer. It indicates that no communication is possible on the bus, e.g. due to a short circuit between lines. Value — Meaning False — The CAN controller is reset, but will not try to restart the communication on the bus True — The CAN controller is reset and will try to restart communication on the bus A Bus-Off Interrupt is generated from the underlying CAN layer. It indicates that no communication is possible on the bus, e.g. due to a short circuit between lines.
4	Bus-Off Counter	Get/Set	USINT	00h
5	Allocation Information	Get	Struct of: BYTE USINT	Allocation choice byte MAC ID (node address) of master

#	Name	Access	Type	Comments
6	MAC ID Switch changed	Get	BOOL	Indicates if the MAC ID (node address) has changed since startup. The attribute is implemented only if the node address (MAC ID) is set from the Network Configuration Object at startup. Value — Meaning True — Changed False — No change
7	Baud rate Switch changed	Get	BOOL	Indicates if the baud rate has changed since startup. The attribute is implemented only if the baud rate is set from the Network Configuration Object at startup. Value — Meaning True — Changed False — No change
8	MAC ID Switch value	Get	USINT	Actual value of node address switches. The attribute is implemented only if the node address (MAC ID) is set from the Network Configuration Object at startup.
9	Baud rate Switch value	Get	USINT	Actual value of baud rate switches The attribute is implemented only if the baud rate is set from the Network Configuration Object at startup.
10	Quick Connect	Get/Set	BOOL	Enables/Disables the Quick Connect feature. Disabled by default. Value — Meaning True — Enable False — Disable
100	Disable auto baud	Get/Set	BOOL	Enables/Disables auto baud. Stored in non-volatile memory Value — Meaning True — Enable False — Disable

4.5 Assembly Object (04h)

Category

Extended

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 default assembly instance IDs used are in the vendor specific range.

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. 11*

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 64h Attributes (Producing Instance)

#	Name	Access	Type	Value/Description
3	Produced Data	Get	Array of BYTE	Process data, written from the application and sent to the CIP network. This data corresponds to the Write Process Data.
4	Size	Get	UINT	Number of bytes in attribute 3

Instance 96h Attributes (Consuming Instance)

#	Name	Access	Type	Value/Description
3	Produced Data	Get	Array of BYTE	Process data, received from the CIP network master and read by the application. Corresponds to the Read Process data.
4	Size	Get	UINT	Number of bytes in attribute #3

4.6 Connection Object (05h)

Category

Extended

Object Description

This object allocates and manages the internal resources associated with both I/O and Explicit Messaging Connections. It is used to model the communication specific characteristics of an application-to-application(s) relationship.

A specific Connection Object Instance manages the communication specific aspects related to an end-point.

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)

Instances #1, #10... #14 Attributes (Explicit messaging)

#	Name	Access	Type	Comments
1	State	Get	USINT	Value: State 0: Non existent 1: Configuring 2: Waiting for connection ID 3: Established 4: Time out 5: Deferred delete
2	Instance Type	Get	USINT	0000h (Explicit messaging connection)
3	Transport Class trigger	Get	BYTE	83h (Server, Transport class 3)
4	Produced connection ID	Get	UINT	CAN ID for transmission
5	Consumed connection ID	Get	UINT	CAN ID for reception
6	Initial Comm Characteristics	Get	BYTE	The message group over which the communication occurs: Value: Message Group 21: Instance #1 33: Instances #10... #14
7	Produced Connection Size	Get	UINT	512 bytes
8	Consumed Connection Size	Get	UINT	512 bytes
9	Expected Packet Rate	Get/Set	UINT	2500 ms (timing associated with this connection)
12	Watchdog timeout action	Get/Set	USINT	Value: Action 0001h: Auto delete (default) 0003h: Deferred delete
13	Produced Connection path length	Get	UINT	0000h (No connection path)
14	Produced Connection path	Get	EPATH	-
15	Consumed Connection path length	Get	UINT	0000h (No connection path)
16	Consumed Connection path	Get	EPATH	-
17	Production Inhibit Time	Get	UINT	0000h
18	Connection Timeout Multiplier	Get/Set	BOOL	Specifies the multiplier applied to the expected packet rate value to derive the value for the Inactivity/Watchdog Timer. Value: Meaning 0: x4 (default) 1: x8 2: x16 3: x32 4: x64 5: x128 6: x256 7: x512 8-255: Reserved

Instance #2 Attributes (Poll or “COS/Cyclic consuming”)

When attribute #9 (EPR, Expected Packet Rate) is set from the network, an IO connection is triggered.

#	Name	Access	Type	Comments
1	State	Get	USINT	Value: State 0: Non existent 1: Configuring 2: Waiting for connection ID 3: Established 4: Time out
2	Instance type	Get	USINT	0001h (I/O Connection)
3	Transport Class trigger	Get	BYTE	Value: Meaning 82h: Server, Polled, Class 2 80h: Server, COS/Cyclic, Class 0, No Ack. 82h: Server, COS/Cyclic, Class 2, Ack.
4	Produced connection ID	Get	UINT	Value: Meaning FFFFh: Not consuming (COS/Cyclic) Other: CAN ID for transmission
5	Consumed connection ID	Get	UINT	CAN ID for reception (Polled)
6	Initial Comm Characteristics	Get	BYTE	Value: Message Group 01h: Polled • Produces over message group 1 • Consumes over message group 2 F1h: COS/Cyclic, No Ack • Consumes only over message group 2 01h: COS/Cyclic, Ack • Produces over message group 1 (Ack) • Consumes over message group 2
7	Produced Connection Size	Get	UINT	Value: Meaning FFFFh: COS/Cyclic Other: Size of Write Process Data (Polled)
8	Consumed Connection Size	Get	UINT	Size of Read Process Data
9	Expected Packet Rate	Get/Set	UINT	Timing associated with this connection
12	Watchdog timeout action	Get	USINT	0000h (Transition to the timed out state)
13	Produced Connection path length	Get	UINT	0000h (COS/Cyclic) 0007h (Polled)
14	Produced Connection path	Get (COS/ Cylic) Get/Set (Polled)	EPATH	No value (COS/Cyclic) 20 04 25 nn nn 30 03h (Polled, nn = producing instance number in assembly object) Contents will be stored in non volatile storage
15	Consumed Connection path length	Get/Set	UINT	0007h
16	Consumed Connection path	Get/Set	EPATH	20 04 25 nn nn 30 03h (nn = consuming instance number in assembly object) Contents will be stored in non volatile storage

#	Name	Access	Type	Comments
17	Production Inhibit Time	Get	UINT	0000h
18	Connection Timeout Multiplier	Get/Set	BOOL	Specifies the multiplier applied to the expected packet rate value to derive the value for the Inactivity/Watchdog Timer. Value — Meaning 0 — x4 (default) 1 — x8 2 — x16 3 — x32 4 — x64 5 — x128 6 — x256 7 — x512 8-255 — Reserved

Instance #3 Attributes (Bit-strobe)

When attribute #9 (EPR, Expected Packet Rate) is set from the network, an IO connection is triggered.

#	Name	Access	Type	Comments
1	State	Get	USINT	Value: State 0: Non existent 1: Configuring 2: Waiting for connection ID 3: Established 4: Time out
2	Instance Type	Get	USINT	0001h (I/O Connection)
3	Transport Class trigger	Get	BYTE	82h (Transport class & Trigger Server, Cyclic, Class 2)
4	Produced connection ID	Get	UINT	CAN ID for transmission
5	Consumed connection ID	Get	UINT	CAN ID for reception
6	Initial Comm Characteristics	Get	BYTE	Produces over message group 1 Consumes over message group 2
7	Produced Connection Size	Get	UINT	Size of produced data on this connection. Max of: 8 bytes, Mapped Process data
8	Consumed Connection Size	Get	UINT	0008h
9	Expected Packet Rate	Get/Set	UINT	Timing associated with this connection
12	Watchdog timeout action	Get/Set	USINT	0000h (Transition to the timed out state)
13	Produced Connection path length	Get	UINT	0007h
14	Produced Connection path	Get/Set	EPATH	20 04 25 nn nn 30 03h (nn = producing instance number in assembly object) Contents will be stored in non volatile storage
15	Consumed Connection path length	Get	UINT	0007h
16	Consumed Connection path	Get/Set	EPATH	20 04 25 nn nn 30 03h (nn = consuming instance number in assembly object) Contents will be stored in non volatile storage
17	Production Inhibit Time	Get	UINT	0000h
18	Connection Timeout Multiplier	Get/Set	BOOL	Specifies the multiplier applied to the expected packet rate value to derive the value for the Inactivity/Watchdog Timer. Value: Meaning 0: x4 (default) 1: x8 2: x16 3: x32 4: x64 5: x128 6: x256 7: x512 8-255: Reserved

Instance #4 Attributes (COS/Cyclic producing)

When attribute #9 (EPR, Expected Packet Rate) is set from the network, an IO connection is triggered.

#	Name	Access	Type	Comments
1	State	Get	USINT	Value: State 0: Non existent 1: Configuring 2: Waiting for connection ID 3: Established 4: Time out
2	Instance type	Get	USINT	0001h (I/O Connection)
3	Transport Class trigger	Get	BYTE	Value: Meaning 00h: Client, Cyclic, Class 0 (No Ack.) 10h: Client, COS, Class 0 (No Ack.) 02h: Client, Cyclic, Class 2 (Ack.) 12h: Client, COS, Class 2 (Ack.)
4	Produced connection ID	Get	UINT	CAN ID for transmission
5	Consumed connection ID	Get	UINT	Value: Meaning FFFFh: Not acknowledged Other: CAN ID for reception (Ack.)
6	Initial Comm Characteristics	Get	BYTE	Value: Message Group 0Fh: Producing only over message group 1 (No Ack.) 01hh: Produces over message group 1 Consumes over message group 2 (Ack.)
7	Produced Connection Size	Get	UINT	Size of produced data on this connection.
8	Consumed Connection Size	Get	UINT	0000h (Consumes 0 bytes on this connection)
9	Expected Packet Rate	Get/Set	UINT	Timing associated with this connection.
12	Watchdog timeout action	Get	USINT	0000h (Transition to the timed out state)
13	Produced Connection path length	Get	UINT	0007h
14	Produced Connection path	Get/Set	EPATH	20 04 25 nn nn 30 03h (nn = producing instance number in assembly object)
15	Consumed Connection path length	Get	UINT	0000h (No ack.) 0005h (Acknowledged) Contents will be stored in non volatile storage
16	Consumed Connection path	Get/Set	EPATH	No value (No ack.) 20 2B 25 01 00h (Acknowledged) Contents will be stored in non volatile storage
17	Production Inhibit Time	Get/Set	UINT	0000h

#	Name	Access	Type	Comments
18	Connection Timeout Multiplier	Get/Set	BOOL	Specifies the multiplier applied to the expected packet rate value to derive the value for the Inactivity/Watchdog Timer. Value – Meaning 0 – x4 (default) 1 – x8 2 – x16 3 – x32 4 – x64 5 – x128 6 – x256 7 – x512 8-255 – Reserved

4.7 Parameter Object (0Fh)

Category

Extended

Object Description

The Parameter Object provides an interface to the parameters of the module. It can provide a full description of each parameter, including minimum and maximum values and a text string describing the parameter.

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

Configuration tools, such as RSNetworx, can extract information about the parameters and present them with their actual name and range to the user.

See also...

- *ADI Object (A2h), p. 33* (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 the general Anybus CompatCom 40 Software Design Guide for further information.
8	Parameter class descriptor	Get	WORD	Default: 0000 0000 0000 01011b 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 (Configuration assembly not supported)

Instance Attributes

#	Name	Access	Type	Comments
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
3	Link Path	Get	Packed EPATH	21 mm mm 25 nn nn 30 05h (Path to the object from where this parameter's value is retrieved, in this case the ADI Object. "mm mm" is A2 00h by default, but can be customized using the Anybus DeviceNet Host Object to change the ABCC ADI Class Object number)
4	Descriptor	Get	WORD	Bit — Contents 0 — Supports Settable Path (N/A) 1 — Supports Enumerated Strings 2 — Supports Scaling (N/A) 3 — Supports Scaling Links (N/A) 4 — Read only Parameter 5 — Monitor Parameter (N/A) 6 — Supports Extended Precision Scaling (N/A)
5	Data type	Get	EPATH	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	(not supported)
9	Help String	Get	SHORT_STRING	
10	Minimum value	Get	(Data Type)	Minimum value of parameter
11	Maximum value	Get	(Data Type)	Maximum value of parameter
12	Default value	Get	(Data Type)	Default value of parameter
13	Scaling Multiplier	Get	UINT	0001h (not supported)
14	Scaling Divisor	Get	UINT	
15	Scaling Base	Get	UINT	
16	Scaling Offset	Get	INT	0000h (not supported)
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	

4.8 Acknowledge Handler Object (2Bh)

Category

Extended

Object Description

This object notifies the producing application of acknowledge reception, acknowledge timeouts, and production retry limit.

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 (01h)

#	Name	Access	Type	Value
1	Acknowledge Timer	Get/Set	UINT	16 ms (Time to wait for acknowledge, in ms, before resending)
2	Retry Limit	Get/Set	USINT	01h (number of ack timeouts before retry limit reached event)
3	Producing Connection Instance	Get	UINT	04h (Connection instance, which contains the path of the producing I/O application object, which will be notified of Ack Handler events)

4.9 ADI Object (A2h)

Object Description

Anybus representation of process data mapped parameters.

See also ..

- *Parameter Object (0Fh), p. 28* (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

A Technical Specification

A.1 LED Indications

A.1.1 Network Status

LED State	Indication
Off	Not online / No network power
Green	On-line, one or more connections are established
Flashing Green (1 Hz)	On-line, no connections established
Red	Critical link failure, fatal event
Flashing Red (1 Hz)	One or more connections timed-out
Alternating Red/Green	Executing self test

A.1.2 Module Status

LED State	Indication
Off	Not operating
Green	Operating in normal condition
Flashing Green (1 Hz)	Missing, incorrect or incomplete configuration, device needs commissioning.
Red	Unrecoverable Fault(s)
Flashing Red (1 Hz)	Recoverable Fault(s)
Alternating Red/Green	Executing self test

A.2 Functional Earth (FE) Requirements

In order to ensure proper EMC behavior, the module must be properly connected to protective earth via the FE pad / FE mechanism described in the general Anybus CompactCom B40 Hardware Design Guide.

tHMS Industrial Networks AB does not guarantee proper EMC behavior unless these FE requirements are fulfilled.

A.3 Power Supplies

A.3.1 Supply Voltage

The module/brick requires a regulated 3.3 V power source as specified in the general Anybus CompactCom B40 Hardware Design Guide.

A.3.2 DeviceNet Power Supply

The total number of units that can be connected to the DeviceNet bus is limited by the maximum current that the power supply can deliver to the bus. Maximum current consumption per unit is specified in the DeviceNet specification to 750 mA. If e.g. the supply can deliver 9 A and all units consume maximum current, the maximum numbers of units allowed on the bus are 12 (12x750 mA = 9A).

The Anybus CompactCom 40 DeviceNet module accepts 11 - 25 V on the industrial network side of the module.

A.4 Power Consumption

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 B40 Hardware Design Guide, and not on the exact power requirements of a single product.

Note that 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 will remain as a Class B module.

The brick alone consumes up to 115 mA. The connector board will add up to 3.5 mA to the power consumption. A complete solution, including a brick, a connector board and LEDs with maximum allowed current consumption, will consume up to 147 mA.

Maximum current consumption on the network side at 11 - 25 V is 39 mA/brick.

A.5 Environmental Specification

Consult the Anybus CompactCom B40 Hardware Design Guide for further information.

A.6 EMC Compliance

A.6.1 Environmental Specification

Consult the Anybus CompactCom B40 Hardware Design Guide for further information.

B Implementation Details

B.1 DeviceNet Implementation

B.1.1 Predefined Connection Set

The module acts as a Group 2 server and supports the Predefined Master/Slave Connection Set.

COS Connection	When the master allocates this connection type, the module transmits the all Process Data at a change of state. An inhibit time can be set to prevent the module from sending too often. The module supports up to 512 bytes in each direction for this type of connection. The size of the connection is checked against the number of bytes mapped as Process Data.
Cyclic Connection	When the master allocates this connection type, the module cyclically transmits the Process Data at the configured interval. The module supports up to 512 bytes in each direction for this type of connection.
Bit Strobe Connection	When the master allocates this connection type, the module transmits data when the bit strobe message is received, and produces up to 512 bytes.
Polled Connection	When the master allocates this connection type, the module transmits the Process Data data when a poll command is received. The module supports up to 512 bytes in each direction for this type of connection.
Explicit Connection	The predefined explicit connection has a buffer of 512 bytes.
Idle/Running	The module is considered to be in Idle mode when not receiving any DeviceNet telegrams, or when receiving DeviceNet telegrams with no data. In other cases, the module is considered to be in Run mode. This affects the Anybus State machine as describe in *Anybus State Machine, p. 38*.

B.1.2 Unconnected Message Server (UCMM)

The module is a UCMM capable device, and supports the Unconnected Explicit Message Request port, Group3, Message ID=6.

Explicit Message Server	The module supports up to 5 simultaneous explicit message connections.

B.2 SUP-Bit Definition

The supervised bit (SUP) indicates that the network participation is supervised by another network device. For DeviceNet this bit is set when the connection object has a connection.

B.3 Anybus State Machine

The table below describes how the Anybus State Machine relates to the DeviceNet network status.

State	DeviceNet Specific Meaning	Notes
WAIT_PROCESS	The module will stay in this state until a Class 0 connection is opened.	(Not set for explicit connections.)
ERROR	Class 0 connection error, Bus-Off event detected or dup-MAC-fail	If the error is fatal, such, such as dup-MAC-fail or Bus-Off, the module will stay in this state until a HW reset is done. (A Bus-Off occurs when it is impossible to communicate on the underlying CAN layer, e.g. if the lines are short circuited.)
PROCESS_ACTIVE	Error free Class 0 connection active	-
IDLE	Class 0 connection idle	Can only be set for connections consuming data.
EXCEPTION	Some kind of unexpected behavior, e.g. watchdog timeout.	The Module Status LED will turn red to indicate a major fault, and turn the Network Status LED off.

This page intentionally left blank

Box 4126
300 04 Halmstad, Sweden

info@hms.se
SCM-1202-159 1.0 en-US / 2020-06-18 / 18988