Anybus® CompactCom™ 40

CC-Link

NETWORK GUIDE

HMSI-27-280 2.3 en-US ENGLISH

Important User Information

Liability

Every care has been taken in the preparation of this document. Please inform HMS Industrial Networks of any inaccuracies or omissions. The data and illustrations found in this document are not binding. We, HMS Industrial Networks, reserve the right to modify our products in line with our policy of continuous product development. The information in this document is subject to change without notice and should not be considered as a commitment by HMS Industrial Networks. HMS Industrial Networks assumes no responsibility for any errors that may appear in this document.

There are many applications of this product. Those responsible for the use of this device must ensure that all the necessary steps have been taken to verify that the applications meet all performance and safety requirements including any applicable laws, regulations, codes, and standards.

HMS Industrial Networks will under no circumstances assume liability or responsibility for any problems that may arise as a result from the use of undocumented features, timing, or functional side effects found outside the documented scope of this product. The effects caused by any direct or indirect use of such aspects of the product are undefined, and may include e.g. compatibility issues and stability issues.

The examples and illustrations in this document are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular implementation, HMS Industrial Networks cannot assume responsibility for actual use based on these examples and illustrations.

Intellectual Property Rights

HMS Industrial Networks has intellectual property rights relating to technology embodied in the product described in this document. These intellectual property rights may include patents and pending patent applications in the USA and other countries.

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 40 CC-Link. The document describes the features that are specific to Anybus CompactCom 40 CC-Link. For general information regarding Anybus CompactCom, consult the Anybus CompactCom design guides.

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 CC-Link specific functionality is to be used, in-depth knowledge of CC-Link networking internals and/or information from the official CC-Link specifications may be required. In such cases, the persons responsible for the implementation of this product should either obtain the CC-Link specification to gain sufficient knowledge or limit their implementation in such a way that this is not necessary.

For additional related documentation and file downloads, 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 M40 Hardware Design Guide	HMS	HMSI-216–126
Anybus CompactCom B40 Design Guide	HMS	HMSI-27-230
Anybus CompactCom Host Application Implementation Guide	HMS	HMSI-27-334
CC-Link Remote Device Station Conformance Test specification	CLPA	BAP-C0401-012-F
CC-Link Specification (Profile)	CLPA	BAP-05028-H
CC-Link Family System Profile (CSP+) Creation Guideline	CLPA	BCN89000-0756-A

1.3 Document History

Version	Date	Description
1.00	2014-12-19	First official version
2.0	2017-01-31	Moved from FM to DOX
2.1	2017-07-10	Added appendix on backward compatibility
2.2	2019-02-28	Rebranding

1.4 Document Conventions

Ordered lists are used for instructions that must be carried out in sequence:

1. First do this
2. Then do this

Unordered (bulleted) lists are used for:

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

...and for action-result type instructions:

► This action...

→ leads to this result

Bold typeface indicates interactive parts such as connectors and switches on the hardware, or menus and buttons in a graphical user interface.

```
Monospaced text is used to indicate program code and other
kinds of data input/output such as configuration scripts.
```

This is a cross-reference within this document: *Document Conventions, p. 3*

This is an external link (URL): www.hms-networks.com

This is additional information which may facilitate installation and/or operation.

This instruction must be followed to avoid a risk of reduced functionality and/or damage to the equipment, or to avoid a network security risk.

Caution
This instruction must be followed to avoid a risk of personal injury.

WARNING
This instruction must be followed to avoid a risk of death or serious injury.

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.
- The terms “basic” and “extended” are used to classify objects, instances and attributes.

1.6 Trademark Information

Anybus® is a registered trademark of HMS Industrial Networks .

All other trademarks are the property of their respective holders.

2 About the Anybus CompactCom 40 CC-Link

2.1 General

The Anybus CompactCom 40 CC-Link communication module provides instant CC-Link slave functionality via the patented Anybus CompactCom host interface. Any device that supports this standard can take advantage of the features provided by the module, allowing seamless network integration regardless of network type.

This product conforms to all aspects of the host interface for Anybus CompactCom 40 modules defined in the Anybus CompactCom 40 Hardware and Software Design Guides, making it fully interchangeable with any other device following that specification. Generally, no additional network related software support is needed, however in order to be able to take full advantage of advanced network specific functionality, a certain degree of dedicated software support may be necessary.

2.2 Features

- CC-Link Remote Device
- I/O: up to 128 bit points and 16 word points of 16 bit data (CC-Link v.1.10, default)
- I/O: up to 896 bit points and 128 word points of 16 bit data (CC-Link v.2.00)

  To use CC-Link v.2.00, the CC-Link Host Object (F7h) must be implemented.
- Max. read process data: 368 bytes
- Max. write process data: 368 bytes
- Max. process data (read + write, in bytes): 736 bytes
- Automatic CC-Link System Area handshaking

  To turn on the automatic handshaking feature, the CC-Link Host Object (F7h) must be implemented.
- Configuration file (CSP+) provided by HMS Industrial Networks
- Possibility to customize Vendor Code, Model Code and Version via application interface
- Baud Rate and Station Number configuration via application interface
- Galvanically isolated bus

ⓘ *Please note that Anybus CompactCom 40 modular device and sync functionality are not supported in this version of Anybus CompactCom 40 CC-Link.*

2.3 Differences Between 40 and 30 Series

- A new default Process Data (PD) mapping scheme has been implemented
- The specific CC-Link mapping commands have been removed
- A different diagnostics functionality has been implemented

2.4 Fieldbus Conformance Notes

- The Anybus CompactCom 40 CC-Link has been tested standalone and found to comply with the CC-Link Conformance Test specification, publication BAP-C0401-012-F. The end product will however need to be re-certified in order to comply with CC-Link certification policies.

- The application alone is responsible for maintaining compatibility with the profiles defined in the CC-Link Specification (Profile), publication BAP-05028-H. This documentation is available free of charge to all registered CLPA members. To sing up as a member, please contact the CLPA (www.cc-link.org).
- In order to pass the conformance test the application will have to show an error status when an erroneous baud rate or station number according to CC-Link is used. The Anybus CompactCom 40 CC-Link will not allow the setting of the Setup Complete attribute in the Anybus Object in such a case.
- If the host application handles the System area (no automatic handshake), it also must take full responsibility for all parts of the conformance test related to the system area.
- The CC-Link V.2.00 protocol is implemented without return check in the module.

3 Basic Operation

3.1 General Information

3.1.1 Software Requirements

No additional network support code needs to be written in order to support the Anybus CompactCom 40 CC-Link, however due to the nature of theCC-Link networking system certain restrictions must be taken into account:

- No acyclic data exchange.

  ADIs must be mapped as Process Data, in accordance with the CC-Link mapping scheme, in order to be represented on the network.

  ADI Names, types and similar attributes cannot be accessed from the network.

  No support for network reset requests.

  Up to 5 diagnostic instances (see *Diagnostic Object (02h), p. 18*) can be created by the host application during normal operation. An additional 6th instance may be created in event of a major fault.

For in depth information regarding the Anybus CompactCom software interface, consult the general Anybus CompactCom 40 Software Design Guide.

3.1.2 CC-Link Family System Profile (CSP+) File

Each device on CC-Link is associated with a CC-Link Family System Profile (CSP+) file, which holds a description of the device and its functions.

HMS Industrial Networks supplies a generic CSP+ file which can serve as a basis for new implementations. A tool for modifying the file is available at the CLPA web page (www.cc-link.org, membership is required).

3.1.3 Communications Settings

Network related communication settings, which can be accessed by the end user, are grouped in the Network Configuration Object (04h).

In the case of CC-Link, this includes:

Station Number:

On CC-Link, each device on the network must be assigned a unique Station Number. The highest possible Station Number depends on the number of occupied stations.

Baud Rate:

The module supports all common CC-Link baud rates up to 10Mbps.

3.2 Data Exchange

3.2.1 Bit Area vs. Word Area

On CC-Link, data is divided into two categories as follows:

Bit Area:

Data is accessed on a bit-by-bit basis. Data is commonly referred to as RX #nn (Slave->Master) and RY #nn (Master->Slave) where “nn” represents an addressable point (i.e. a single bit) in the Bit Area.

Word Area:

Data is accessed as 16-bit words. Data is commonly referred to as RWr #nn (Slave->Master) and RWw #nn (Master->Slave) where “nn” represents an addressable point (i.e. a word) in the Word Area.

Direction	Bit Area Points (hex)	Word Area Points (dec)
Slave -> Master	RX00 .... RX37F	RWr0... RWr127
Master -> Slave	RY00 ... RY37F	RWw0 ... RWw127

3.2.2 Data Sizes

CC-Link Version 1

By default, the module automatically calculates the required number of occupied stations based on the mapped Process Data. The following data sizes are possible.

Occupied Stations	Bit Points	Word Points	Total (Bit + Word, in bytes)
1	32 bits	4 words	12
2	64 bits	8 words	24
3	96 bits	12 words	36
4	128 bits	16 words	48

CC-Link Version 2

By implementing the attribute attribute Network Settings (#4) in the CC-Link Host Object (F7h) it is possible to customize the implementation for CC-Link version 2 and use larger data sizes through extension cycles. In such case, the following sizes are possible:

Occupied Stations	1 Extension Cycle		2 Extension Cycles		4 Extension Cycles		8 Extension Cycles	
	Points	Total	Points	Total	Points	Total	Points	Total
1	32 bits	12 bytes	32 bits	20 bytes	64 bits	40 bytes	128 bits	80 bytes
	4 words		8 words		16 words		32 words	
2	64 bits	24 bytes	96 bits	44 bytes	192 bits	88 bytes	384 bits	176 bytes
	8 words		16 words		32 words		64 words	
3	96 bits	36 bytes	160 bits	68 bytes	320 bits	136 bytes	640 bits	272 bytes
	12 words		24 words		48 words		96 words	
4	128 bits	48 bytes	224 bits	92 bytes	448 bits	184 bytes	896 bits	368 bytes
	16 words		32 words		64 words		128 words	

ℹ *On CC-Link, certain parts of the Bit and Word data may be reserved and should not be used for data exchange. For more information, see CC-Link System Area Implementation, p. 13.*

For conformance test of CC-Link version 2, it is necessary to enable the CC-Link conformance test mode, see Network CC-Link Object (08h), p. 25, command CCL_Conf_Test_Mode..

4 Process Data Mapping

4.1 Default CC-Link Buffer Memory Map

The default profile for Anybus CompactCom 40 CC-Link is “Generic Device”. It is up to the application to realize the implementation, including the mapping of ADIs to the corresponding memory. This makes it possible to comply with any profile buffer data layout.

See also...

- *CC-Link System Area Implementation, p. 13*
- *Network Object (03h), p. 20*

4.1.1 Default Buffer Memory (Bit Points)

The table shows the layout of the bit areas in the default buffer memory. The addresses refer to the unit relative addresses, not the absolute addresses for the complete network.

Slave → Master		**Master → Slave**	
Point	**Signal name**	**Point**	**Signal name**
RX0 to RX Q-11h	User area	RY0 to RX Q-11h	User area
RX Q-10h	Reserved	RY Q-10h	Reserved
RX Q-Fh		RY Q-Fh	
RX Q-Eh		RY Q-Eh	
RX Q-Dh		RY Q-Dh	
RX Q-Ch		RY Q-Ch	
RX Q-Bh		RY Q-Bh	
RX Q-Ah		RY Q-Ah	
RX Q-9h		RY Q-9h	
RX Q-8h	Initial data processing request flag	RY Q-8h	Initial data processing complete flag
RX Q-7h	Initial data setting complete flag	RY Q-7h	Initial data setting request flag
RX Q-6h	Error status flag	RY Q-6h	Error reset request flag
RX Q-5h	Remote READY	RY Q-5h	Reserved
RX Q-4h	Reserved	RY Q-4h	
RX Q-3h		RY Q-3h	
RX Q-2h		RY Q-2h	
RX Q-1h		RY Q-1h	

- CC-Link version 1 or Extension Cycles = 1:

  Q = Number of occupied stations × 32

- CC-Link version 2 and Extension Cycles >= 2:

  Q = (Number of occupied stations × 32 – 16) × Extension Cycles

4.1.2 Default Buffer Memory (Word Points)

The table shows the layout of the word areas in the default buffer memory. The addresses refer to the unit relative addresses, not the absolute addresses for the complete network.

Slave → Master		**Master → Slave**	
Point	**Signal name**	**Point**	**Signal name**
RWr 0	User area	RWw 0	User area
to		to	
RWr Z-1	User area	RWw Z-1	User area

- CC-Link version 1 or Extension Cycles = 1:

  Z = Number of occupied stations × 4
- CC-Link version 2 and Extension Cycles >= 2:

  Z = Number of occupied stations × 4 × Extension Cycles

4.2 Mapping

This scheme is used when the host application uses the mapping commands:

- Map_ADI_Write_Area and Map_ADI_Write_Ext_Area map data to the RX (bit) and RWr (word) areas.
- Map_ADI_Read_Area and Map_ADI_Read_Ext_Area map data from the RY (bit) and RWw (word) areas.
- ADIs are mapped to consecutive locations in the respective areas in the same order as the mapping commands are issued.
- All bit data types (BITSx, BITx or PADx) are mapped to the CC-Link bit area as long as no previous mapping command with a non bit data type has been received. After that, for any following mapping command with a bit data type, the data will be mapped in the CC-Link word area, see the examples on the following pages.
- By default no padding is performed. Any padding or alignment demand to reach the profile layout is the responsibility of the application.
- The Map_ADI_Write_Area and Map_ADI_Read_Area commands are in byte granularity and can only be used for data types of 8, 16, 32 or 64 bit length.
  - Byte alignment during mapping will be enforced when using this command.
  - The use of e.g. the data type BIT3 will generate a NAK (negative acknowledgment) message in answer to the request.
  - The host application may need to perform padding, e.g. if word alignment is needed.

  See mapping example with standard command below.
- The Map_ADI_Write_Ext_Area and Map_ADI_Read_Ext_Area commands are in bit granularity and can be used for all data types.
  - Bit data will be mapped consecutively in the bit area as long as BITSx, BITx or PADx data types are used.
  - Once a command with any other data type has been received, all data will be mapped in the word area. This may disrupt the byte alignment, that is a condition. It is the responsibility of the host application to perform all padding to avoid error messages due to not aligned data mapping.
  - Not all elements in an ARRAY or STRUCT ADI need to be mapped.

  See mapping example with extended mapping commandon below.
- In default mode, with no network settings implemented in the CC-Link Host Object (F7h), the number of stations are increased if the bits or words do not fit the starting configuration.
- ADIs with data type structure are put in the CC-Link word area as a complete chunk/blob, unless covered by any of the rules above.

Implementation of CC-Link profiles might require that the application use some padding in the mapping to get the memory buffer layout as the profile requires. This can be done by using ADI # 0 or PADx for padding in the map directed to the CC-Link bit areas and using ADI# xx (dummy byte) for padding in the CC-Link word areas. These areas should be marked as reserved in the buffer memory description (towards the CC-Link network).

A faulty mapping will generate a NAK (negative acknowledgment). An exception will be generated if the settings in the CC-link Host Object do not fully comply with the selected mapping. Exception information will be available in the Network Object (03h). For more information see:

- *Network Object (03h), p. 20*
- *Exception Information, p. 37*

4.2.1 Mapping example with standard command

This is an example of nine (A-I) Read mapping commands using the Map_ADI_Read_Area command.

Command	Data Type	No. of Elements
A	BITS8	3
B	BITS16	1
C	BOOL	1
D	UINT8	1
E	UINT16	1
F	BOOL	2
G	BITS32	1
H	BITS16	1
I	UINT32	1

This is the resulting RY mapping in the bit area:

Point	Contents (Command number[Element:bit])							
RY #7... 0	A[0:7]	A[0:6]	A[0:5]	A[0:4]	A[0:3]	A[0:2]	A[0:1]	A[0:0]
RY #15... 8	A[1:7]	A[1:6]	A[1:5]	A[1:4]	A[1:3]	A[1:2]	A[1:1]	A[1:0]
RY #23... 16	A[2:7]	A[2:6]	A[2:5]	A[2:4]	A[2:3]	A[2:2]	A[2:1]	A[2:0]
RY #31... 24	B[0:7]	B[0:6]	B[0:5]	B[0:4]	B[0:3]	B[0:2]	B[0:1]	B[0:0]
RY #39... 32	B[0:15]	B[0:14]	B[0:13]	B[0:12]	B[0:11]	B[0:10]	B[0:9]	B[0:8]

This is the resulting RWw mapping in the word area:

	Contents (Command number[bit/byte/word])															
Point	Contents (LSB)								Contents (MSB)							
RWw #0	C								D							
RWw #1	E[0]								E[1]							
RWw #2	F[0]								F[1]							
RWw #3	G[LSW]								G[LSW]							
RWw #4	G[MSW]								G[MSW]							
RWw #5	H[7]	H[6]	H[5]	H[4]	H[3]	H[2]	H[1]	H[0]	H[15]	H[14]	H[13]	H[12]	H[11]	H[10]	H[9]	H[8]
RWw #6	I[LSW]								I[LSW]							
RWw #7	I[MSW]								I[MSW]							

4.2.2 Mapping example with extended mapping command

This is an example of three (A-C) Write mapping commands using the Map_ADI_Write_Ext_Area command.

Please note that not all elements in an ADI need to be mapped. The elements to be mapped, are defined by the settings of “Index first element” and “Nbr of consecutive elements”.

Command	# mapping items	ADI #	Total Nbr of elements	Index first element	Nbr of consecu-tive elements	Nbr of type descrip-tors	Data type specifiers
A	2	3	1	0	1	1	65(BIT1)
	-	10	1	0	1	1	47(PAD15)

B	1	20	5	0	3	5	70(BIT6)
							34(PAD2)
							9(BITS8)
							36(PAD4)
							68(BIT4)

C	1	7	4	1	3	4	9(BITS8)
							5(UINT16)
							10(BITS16)
							0(BOOL)

The mapping will start in the bit area as the first ADI to be mapped is of a bit data type (BIT1). Once an ADI of a word data type is to be mapped, the mapping will be moved to the word area. All remaining data, whether of a bit data type or a word data type will from then on be mapped in the word area.

This is the resulting RX mapping in the bit area:

Point	Command[mapping Item]:ADI[IndexElement]:(bit/byte/word)							
RX #7... 0	A[1]:10[0]: (b6)	A[1]:10[0]: (b5)	A[1]:10[0]: (b4)	A[1]:10[0]: (b3)	A[1]:10[0]: (b2)	A[1]:10[0]: (b1)	A[1]:10[0]: (b0)	A[0]:3[0]: (b0)
RX #15... 8	A[1]:10[0]: (b14)	A[1]:10[0]: (b13)	A[1]:10[0]: (b12)	A[1]:10[0]: (b11)	A[1]:10[0]: (b10)	A[1]:10[0]: (b9)	A[1]:10[0]: (b8)	A[1]:10[0]: (b7)
RX #23... 16	B[0]:20[1]: (b1)	B[0]:20[1]: (b0)	B[0]:20[0]: (b5)	B[0]:20[0]: (b4)	B[0]:20[0]: (b3)	B[0]:20[0]: (b2)	B[0]:20[0]: (b1)	B[0]:20[0]: (b0)
RX #31... 24	B[0]:20[2]: (b7)	B[0]:20[2]: (b6)	B[0]:20[2]: (b5)	B[0]:20[2]: (b4)	B[0]:20[2]: (b3)	B[0]:20[2]: (b2)	B[0]:20[2]: (b1)	B[0]:20[2]: (b0)

This is the resulting RWr mapping in the word area:

Point		Command[mapping Item]:ADI[IndexElement]:(bit/byte/word)							
RWr #0	LSB	C[0]:7[1]:(LSB)							
	MSB	C[0]:7[1]:(MSB)							
RWr #1	LSB	C[0]:7[2]: (b7)	C[0]:7[2]: (b6)	C[0]:7[2]: (b5)	C[0]:7[2]: (b4)	C[0]:7[2]: (b3)	C[0]:7[2]: (b2)	C[0]:7[2]: (b1)	C[0]:7[2]: (b0)
	MSB	C[0]:7[2]: (b15)	C[0]:7[2]: (b14)	C[0]:7[2]: (b13)	C[0]:7[2]: (b12)	C[0]:7[2]: (b11)	C[0]:7[2]: (b10)	C[0]:7[2]: (b9)	C[0]:7[2]: (b8)
RWr #2	LSB	C[0]:7[3]:(byte)							
	MSB	-							

5 CC-Link System Area Implementation

5.1 System Area Modes

An essential part of the CC-Link communication is the CC-Link System Area. This area holds various status- and diagnostic flags, and can either be handled automatically by the Anybus device (default) or by the host application.

5.1.1 System Area Handled by Anybus (Default, Basic)

All flags in the System Area are handled automatically by the module, unless this functionality has been explicitly disabled by not implementing the attribute System Area Handler (#5) in the CC-Link Host Object (F7h).

The error indication flag can only be set by a diagnostic latching event. Please refer to *Diagnostics, p. 14* for more information.

The commands Initial_Data_Setting_Notification and Inital_Data_Processing_Notification in the CC-Link Host Object have to be supported. If any of these commands is not accepted by the host application, the host application must respond with “Unsupported Object” or “Unsupported Command”, thus not acknowledging the request.

5.1.2 System Area Handled by Host Application (Advanced)

If the attribute “System Area Handler” (#5) has been set to -1 (disabled), the host application alone is responsible for handling the CC -Link status flags in accordance with one of the profiles defined in the CC-Link specification. To achieve this, the host application must map one or several ADIs to the corresponding location(s) in the CC-Link memory map. Please refer to *System Area Location, p. 14* for more information.

5.2 System Area Layout

This section is only relevant when the System Area is handled automatically by the module.

Slave -> Master		**Master -> Slave**	
Bit Offset	**Contents**	**Bit Offset**	**Contents**
0	(reserved)	0	(reserved)
1		1	
2		2	
3		3	
4		4	
5		5	
6		6	
7		7	
8	Initial Data Processing Request	8	Initial Data Processing Complete
9	Initial Data Setting Complete	9	Initial Data Setting Request
10	Error Status	10	Error Reset Request
11	Remote READY	11	(reserved)
12	(reserved)	12	
13		13	
14		14	
15		15	

The various flags listed in the table above are handled as described in *CC-Link Handshaking Implementation, p. 35*.

See also...

- *System Area Location, p. 14*
- *Diagnostics, p. 14*

5.3 System Area Location

This section is only relevant when the System Area is handled automatically by the module.

The default location of the System Area is at the very end of the Bit Area as follows:

Point	Contents	Point	Contents
RX #0	User area (holds Write Process Data)	RY #0	User area (holds Read Process Data)
RX #1		RY #1	
...		...	
RX #Q-18		RY #Q-18	
RX #Q-17		RY #Q-17	
RX #Q-16	(Reserved for CC-Link System Area)	RY #Q-16	(Reserved for CC-Link System Area)
RX #Q-15		RY #Q-15	
...		...	
RX #Q-2		RY #Q-2	
RX #Q-1		RY #Q-1	

(The table above illustrates how data is represented as seen from the CC-Link master. “Q” represents the number of addressable points in the Bit Area of the Anybus CompactCom.)

It is possible to change the location of the System Area by implementing the attribute System Area Handler (#5) in the CC-Link Host Object (F7h). It is also possible to disable it altogether by setting this attribute to -1. In such case, the host application is responsible for handling the CC-Link communication in consistency with one of the profiles defined in the CC-Link Specification (profile).

See also...

- *Mapping, p. 10*
- *CC-Link Handshaking Implementation, p. 35*

5.4 Diagnostics

This section is only relevant when the System Area is handled automatically by the module.

The module supports up to 5 diagnostic entries during normal conditions, plus an additional 6th entry in case of a major unrecoverable event.

Diagnostics are represented through the Error Status and Remote READY flags.

No information on the actual cause of a diagnostic event is forwarded to the network, apart from the severity, that is given implicitly by monitoring the Remote READY flag. A major event will remove this flag (if set).

Latching event functionality is supported and a latching event will set the Error Status flag. If the event has Major severity, the Remote READY flag will be removed. Please note that only latching diagnostic events will set the Error Status flag.

Error Status flag

This flag reflects the state of the Diagnostic Object as follow:

1:	Latching diagnostic events exist The flag stays high until the master has acknowledged the event through the Error Reset Request flag.
0:	No diagnostic event exists - or - Error Reset Request flag high

Remote READY flag

1:	Normal operation
0:	Diagnostic event with ‘Major’ severity exists. Normal behavior is resumed when the event has been resolved (i. e. when the host application removes the corresponding diagnostic instance) - and - the master has acknowledged the event through the Error Reset Request flag (see also Error Status and Error Code above) - or - Initial Data Setting Request flag high

See also...

- *System Area Modes, p. 13*
- *CC-Link Handshaking Implementation, p. 35*
- *Diagnostic Object (02h), p. 18*

6 Anybus Module Objects

6.1 General Information

This chapter specifies the Anybus module object implementation and how they correspond to the functionality in the Anybus CompactCom 40 CC-Link.

- *Anybus Object (01h), p. 17*
- *Diagnostic Object (02h), p. 18*
- *Network Object (03h), p. 20*
- *Network Configuration Object (04h), p. 22*
- *Network CC-Link Object (08h), p. 25*
- *Anybus File System Interface Object (0Ah), p. 27*

6.2 Anybus Object (01h)

Category

Basic

Object Description

This object groups common Anybus information, and is described thoroughly in the general Anybus CompactCom 40 Software Design Guide.

Supported Commands

Object:	Get_Attribute
Instance:	Get_Attribute Set_Attribute Get_Enum_String

Object Attributes (Instance #0)

Consult the general Anybus CompactCom Software 40 Design Guide for further information.

Instance Attributes (Instance #1)

Basic

#	Name	Access	Type	Value	
1	Module type	Get	UINT16	0403h (Anybus CompactCom 40)	
2... 11	-	-	-	Consult the general Anybus CompactCom Software Design Guide for further information.	
12	LED colors	Get	struct of: UINT8 (LED1A) UINT8 (LED1B) UINT8 (LED2A) UINT8 (LED2B)	Value: 01h 02h 01h 02h	Color: Green Red Green Red
13... 18	-	-	-	Consult the general Anybus CompactCom Software Design Guide for further information.	
19	Network time	Get	UINT64	0 (not supported by CC-Link)	

6.3 Diagnostic Object (02h)

Category

Basic

Object Description

This object provides a standardized way of handling host application events & diagnostics, and is thoroughly described in the general Anybus CompactCom Software 40 Design Guide.

In the case of CC-Link, diagnostics is represented through the CC-Link System Area flags. If the module handles the System Areas you have to create/remove one Diagnostics instance to pass the CC-Link conformance test. If the host application handles the System Area (advanced) the Diagnostic Object does not have to be implemented.

The module supports acknowledgment of latching diagnostic events if the System area handler functionality is enabled in the CC_Link Host Object (F7h), instance #1, attribute #5.. If a latching event cannot be created, a network specific error response will be returned.

The module supports six instances of this object, where one is reserved for a major unrecoverable diagnostic event. No information about the actual cause of a diagnostic event will be forwarded to the network, only the severity.

See also...

- *System Area Modes, p. 13*
- *System Area Layout, p. 13*
- *Diagnostics, p. 14*
- *Network Configuration Object (04h), p. 22*

Supported Commands

Object:	Get_Attribute Create Delete
Instance:	Get_Attribute

Object Attributes (Instance #0)

#	Name	Access	Data Type	Value
1	Name	Get	Array of CHAR	“Diagnostic”
2	Revision	Get	UINT8	01h
3	Number of instances	Get	UINT16	See general Anybus CompactCom 40 Software Design Guide
4	Highest instance no.	Get	UINT16	
11	Max no. of instances	Get	UINT16	5+1
12	Supported functionality	Get	BITS32	0000 0001h (Latching event supported)

Instance Attributes (Instance #1)

#	Name	Access	Type	Value
1	Severity	Get	UINT8	See general Anybus CompactCom 40 Software Design Guide
2	Event Code	Get	UINT8	
3-7	(reserved)			

Command Details: Create

Category

Basic

Details

Command Code:	03h
Valid for:	Object Instance

Description

This command creates a new instance, in this case representing a new diagnostic event in the host application.

When a latching event can not be created, a network specific extended error response, as described below, will be returned.

- Command details:

  Please consult the Anybus CompactCom 40 Software Design Guide for more information.

- Response details (Success):

Field	Contents
MsgData[0–1]	The number of the created instance

- Response details (Error):

Field	Contents
MsgData[0]	FFh (Object specific error)
MsgData[1]	FFh (Network specific error)
MsgData[2]	CC-Link specific error code, see below

CC-Link Specific Error Code	Description
01h	A latching event could not be created due to invalid state (the module is not in state WAIT_ PROCESS).
02h	A latching event could not be created as the System Area functionality is not enabled.

6.4 Network Object (03h)

Category

Basic

Object Description

For more information regarding this object, consult the general Anybus CompactCom 40 Software Design Guide.

See also....

- *Network CC-Link Object (08h), p. 25*

Supported Commands

Object:	Get_Attribute
Instance:	Get_Attribute Set_Attribute Get_Enum_String Map_ADI_Write_Area Map_ADI_Read_Area Map_ADI_Write_Ext_Area Map_ADI_Read_Ext_Area

Object Attributes (Instance #0)

#	Name	Access	Data Type	Value
1	Name	Get	Array of CHAR	“Network”
2	Revision	Get	UINT8	02h
3	Number of instances	Get	UINT16	0001h
4	Highest instance no.	Get	UINT16	0001h

Instance Attributes (Instance #1)

Basic

#	Name	Access	Data Type	Value
1	Network type	Get	UINT16	0090h
2	Network type string	Get	Array of CHAR	“CC-Link”
3	Data format	Get	ENUM	00h (LSB first)
4	Parameter data support	Get	BOOL	False
5	Write process data size	Get	UINT16	Current write process data size (in bytes). Updated on each successful Map_ADI_Write_Area or Map_ADI_Write_Ext_Area. Consult the general Anybus CompactCom 40 Software Design Guide for further information.
6	Read process data size	Get	UINT16	Current read process data size (in bytes). Updated on each successful Map_ADI_Read_Area or Map_ADI_Read_Ext_Area. Consult the general Anybus CompactCom 40 Software Design Guide for further information.
7	Exception information	Get	UINT8	

6.5 Network Configuration Object (04h)

Category

Basic

Object Description

This object holds network specific configuration parameters that may be set by the end user. A reset command (factory default) issued towards this object will result in all instances being set to their default values, including resetting the NVS storage.

The Station Number is set to “Not set” and the Baud Rate to 2 (2.5 Mbps).

See also ..

- *Communications Settings, p. 7*
- *CC-Link Host Object (F7h), p. 29*

Instances #1 and #2 have to be implemented if the end product is going to be recertified according to CC-Link Certification Policies.

Supported Commands

Object:	Get_Attribute Reset
Instance:	Get_Attribute Set_Attribute Get_Enum_String

Object Attributes (Instance #0)

#	Name	Access	Data Type	Value
1	Name	Get	Array of CHAR	“Network configuration”
2	Revision	Get	UINT8	01h
3	Number of instances	Get	UINT16	0002h
4	Highest instance no.	Get	UINT16	0003h

Instance Attributes (Instance #1, Station Number)

This instance holds the actual CC-Link station number.

Basic

#	Name	Access	Data Type	Value
1	Name	Get	Array of CHAR	"Station number" Multilingual, see *Multilingual Strings, p. 24*
2	Data type	Get	UINT8	04h (UINT8)
3	Number of elements	Get	UINT8	01h (one element)
4	Descriptor	Get	UINT8	07h (get/set/shared access)
5	Value	Get/Set	UINT8	**Value** **Setting** 0 Not set (default) 1 — 64 Station number >64 (not valid) Notes: Default is 0. If the station number is not set, the Anybus CompactCom 40 CC-Link may enter the state EXCEPTION. The sum of the Station Number and the Number of Occupied Stations may not exceed 65. A value that is not valid will result in the Setup Complete command being not acknowledged.
6	Configured value	Get	UINT8	Holds the configured value. **Value** **Setting** 0 Not set 1 — 64 Station number >64 (not valid)

Any value written to attribute #5 after setup is finished, will be saved in attribute #6 instead of in attribute #5. After a power cycle has been performed, attribute #5 will be updated with the value saved in attribute #6. If no valid station number is entered the module will enter Anybus state EXCEPTION, when leaving Anybus state NW_INIT.

Instance Attributes (Instance #2, Baud Rate)

Basic

#	Name	Access	Data Type	Value
1	Name	Get	Array of CHAR	“Baud rate” Multilingual, see *Multilingual Strings, p. 24*
2	Data type	Get	UINT8	08h (ENUM)
3	Number of elements	Get	UINT8	01h (one element)
4	Descriptor	Get	UINT8	07h (get/set/shared access)
5	Value	Get/Set	ENUM	Value — Setting 0 — 156 kbps 1 — 625 kbps 2 — 2.5 Mbps (default) 3 — 5 Mbps 4 — 10 Mbps other — (not valid)
6	Configured value	Get	ENUM	Holds the configured value Value — **Speed/String** 0 — 156 kbps 1 — 625 kbps 2 — 2.5 Mbps 3 — 5 Mbps 4 — 10 Mbps other — (not valid)

Any value written to attribute #5 after setup is finished, will be saved in attribute #6 instead of in attribute #5. After a power cycle has been performed, attribute #5 will be updated with the value saved in attribute #6. If no valid baud rate is entered the module will enter Anybus state EXCEPTION, when leaving Anybus state NW_ INIT.

Multilingual Strings

The instance names in this object are multilingual and are translated based on the current language settings as follows:

Instance	English	German	Spanish	Italian	French
1	Station No.	Geräteadresse	Direcc nodo	Indirizzo	Adresse
2	Baud rate	Datenrate	Veloc transf	Velocità dati	Vitesse

6.6 Network CC-Link Object (08h)

Category

Basic, extended

Object Description

-

Supported Commands

Object:	Get_Attribute
Instance:	Get_Attribute CCL_Conf_Test_Mode

Object Attributes (Instance #0)

#	Name	Access	Data Type	Value
1	Name	Get	Array of CHAR	“Network CC-Link”
2	Revision	Get	UINT8	02h
3	Number of instances	Get	UINT16	0001h
4	Highest instance no.	Get	UINT16	0001h

Instance Attributes (Instance #1)

The values of attributes #1 and #2 are calculated automatically by the module unless the host application has specified other values in the CC-Link Host Object (F7h). The attribute values are valid from the first transition to WAIT_PROCESS, and can be used to establish the location of the CC-Link System Area during development etc.

#	Name	Access	Type	Value
1	Network Settings	Get	Struct of:	Current network settings:
			UINT8	CC-Link Version; 01h = v1.10, 02h = v2.00
			UINT8	Number of occupied stations
			UINT8	Number of extension cycles
2	System Area Handler	Get	SINT16	System Area location (or -1 in case the system area is handled by the host application)
3-4	(reserved)			
5	CC-Link V.2 Conformance test mode	Get	BOOL	TRUE if the special CC-Link version 2.00 conformance test mode is active (activated by the command CCL_Conf_Mode).

Command Details: CCL_Conf_Test_Mode

Category

Extended

Details

Command Code:	12h
Valid for:	Object Instance

Description

This command enables/disables the special CC-Link version 2 conformance test mode. For conformance test of CC-link version 2, it is necessary that a special version of the slave can loop all accepted data from RY/RWw to RX/RWx. This command can activate the functionality temporarily or permanently, or deactivate the functionality permanently. The command may only be issued during SETUP.

- Command details:

Field	Contents	
CmdExt[0]	Conformance test mode	
	Value:	Description:
	00h	Deactivate conformance test mode permanently.
	01h	Activate conformance test mode temporarily. Conformance test mode will not be activated after a reset/power cycle.
	02h	Activate conformance test mode permanently.
CmdExt[1]	(reserved)	

- Response details (Success):

Field	Contents
Data[0–n]	(not used)

- Response details (Error):

Error	Contents
Standard error codes	See Anybus CompactCom 40 Software Design Guide

6.7 Anybus File System Interface Object (0Ah)

Category

Advanced

Object Description

This object provides an interface to the built-in file system. In an Anybus CompactCom 40 CC-Link, the file system consists of one folder, called “Firmware”. This folder is used to save a firmware file to upgrade the module. After a reset the firmware in the module will be upgraded and the file erased.

Please consult the Anybus CompactCom 40 Software Design Guide for more information.

Supported Commands

(Consult the Anybus CompactCom 40 Software Design Guide for more information)

Object Attributes (Instance #0)

(Consult the Anybus CompactCom 40 Software Design Guide for more information)

Instance Attributes (Instance #1)

(Consult the Anybus CompactCom 40 Software Design Guide for more information)

7 Host Application Objects

7.1 General Information

This chapter specifies the host application object implementation in the module. The objects listed here may optionally be implemented within the host application firmware to expand the CC-Link implementation.

Network Specific Object:

- *CC-Link Host Object (F7h), p. 29*

7.2 CC-Link Host Object (F7h)

Category

Basic, extended

Object Description

This object implements CC-Link specific features in the host application.

The implementation of this object is optional; the host application can support none, some, or all of the attributes specified below. The module will attempt to retrieve the values of these attributes during startup; if an attribute is not implemented in the host application, simply respond with an error message (“06h, Invalid CmdExt[0]”). In such case, the module will use its default value.

Supported Commands

Object:	Get_Attribute Initial_Data_Setting_Notification Initial_Data_Processing_Notification
Instance:	Get_Attribute

Object Attributes (Instance #0)

#	Name	Access	Data Type	Value
1	Name	Get	Array of CHAR	“CC-Link”
2	Revision	Get	UINT8	01h
3	Number of instances	Get	UINT16	0001h
4	Highest instance no.	Get	UINT16	0001h

Instance Attributes (Instance #1)

Basic

#	Name	Access	Type	Default Value	Comment
1	Vendor Code	Get	UINT16	0212h (“HMS”)	Assigned by the CLPA
2	SW Version	Get	UINT8	(Anybus firmware release no.)	Valid settings range from 1 to 3Fh.

Extended

#	Name	Access	Type	Default Value	Comment
3	Model Code	Get	UINT8	7Fh (“Generic Device”)	For possible settings, consult the CC-Link Specification (Profile) for device type codes.
4	Network Settings	Get	Struct of:	Current network settings:	
			UINT8	01h	CC-Link Version; 01h = v1.10, 02h = v2.00
			UINT8	(calculated automatically)	Number of occupied stations Valid settings: 01h, 02h, 03h, 04h
			UINT8	1	Number of extension cycles. Valid settings, CC-Link v1: 01h Valid settings, CC-Link v2: 01h, 02h, 04h, 08h
5	System Area Handler	Get	SINT16	(calculated automatically; last 16 bits in the bit area)	System Area location • -1: host application • 0...880: Module. Value specifies the offset of the system area in the bit area. Must be located on an even 16-bit boundary.
6	Output Hold/Clear	Get	UINT8	1 (Hold)	Output action: • 0: Clear • 1: Hold This attribute indicates to the CC-Link master how the application handles the output data when there is an active error in the master (the master station application), that prevents the master to send an output value to the slave. This condition is signaled to the application by the Anybus State ERROR.

Initial_Data_Setting_Notification

Category

Basic

Details

Command Code:	10h
Valid for:	Object Instance

Description

ⓘ *This section is only relevant when the System Area is handled automatically by the module.*

This command will be issued when the master initiates the Initial Data Setting cycle, i.e. on the rising edge of the flag Initial Data Setting Request.

The host application may either accept or reject (i.e. by responding with “Unsupported Object” or “Unsupported Command”) the command; in either case, the module will continue the Initial Data Setting Cycle by setting the flag Initial Data Setting Complete once the response has been received.

- Command details:

  (No data)

- Response details:

  (No data)

Initial_Data_Processing_Notification

Category

Basic

Details

Command Code:	11h
Valid for:	Object Instance

Description

ⓘ *This section is only relevant when the System Area is handled automatically by the module.*

This command informs the application when the state transition WAIT_PROCESS to PROCESS_ACTIVE has occurred during automatic handshaking. If the application has disabled the automatic handshake using the attribute System area handler (#5) of the CC-Link object, or if the CC-Link object is not present, this command will never be issued. No extra information is transmitted with the command, and no information is expected in the response message.

The host application may either accept or reject (i.e. by responding with “Unsupported Object” or “Unsupported Command”) the command; in either case, the module will continue the Initial Data Setting Cycle by setting the flag Initial Data Processing Complete-flag once the response has been received.

- Command details:

  (No data)

- Response details:

  (No data)

A Categorization of Functionality

The objects, including attributes and services, of the Anybus CompactCom and the application are divided into two categories: basic and extended.

A.1 Basic

This category includes objects, attributes and services that are mandatory to implement or to use. They will be enough for starting up the Anybus CompactCom and sending/receiving data with the chosen network protocol. The basic functions of the industrial network are used.

Additional objects etc, that will make it possible to certify the product also belong to this category.

A.2 Extended

Use of the objects in this category extends the functionality of the application. Access is given to the more specific characteristics of the industrial network, not only the basic moving of data to and from the network. Extra value is given to the application.

Some of the functionality offered may be specialized and/or seldom used. As most of the available network functionality is enabled and accessible, access to the specification of the industrial network may be required.

B Implementation Details

B.1 SUP-Bit Definition

The Supervised-bit (SUP) indicates that the module is exchanging data with the CC-Link master.

B.2 Anybus State Machine

The table below describes how the Anybus State Machine relates to the CC-Link network.

Anybus State	Implementation (System Area handled by Anybus module)	Implementation (System Area handled by Host Application)
WAIT_PROCESS	Automatic CC-Link handshaking in progress	Waiting for the first refresh message
ERROR	Possible causes: • Timeout error (ERR21) • 0 channel carrier detection status (ERR22) • insufficient number of data for the host (ERR30...32) When the problem disappears, the module returns to the previous state.	
PROCESS_ACTIVE	The module enters this state on the rising edge of 'Initial Data Processing Complete'	The module enters this state when the first refresh message has been received
IDLE	PLC/Master in STOP mode	
	When the PLC/Master returns to RUN mode, the module returns to the previous state.	
EXCEPTION	Possible causes: • Configuration error • other error that cannot be indicated to the application in any other way Examples: • Illegal combinations of attribute settings in the CC-Link Host Object (F7h) • Illegal combinations of mapping commands and attribute settings in the CC-Link Host Object (F7h)	

B.3 Application Watchdog Time Out Handling

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

B.4 CC-Link Handshaking Implementation

ⓘ *This section is only relevant when the System Area is handled automatically by the module.*

When the System Area is handled by the Anybus module, all CC-Link handshaking is performed automatically as described in the table below.

Flag	Set when...	Cleared when...
Initial Data Processing Request	State changes from NW_INIT to WAIT_ PROCESS	The following sequence has finished: 1. State transition from WAIT_ PROCESS to PROCESS_ACTIVE 2. The host application has responded to the Initial_Data_Processing_ Notification command, see *CC-Link Host Object (F7h), p. 29*. This purpose of this procedure is to ensure that the host application has detected that the module has shifted to the PROCESS_ACTIVE state.
Initial Data Setting Complete	The host application responds to the Initial_Data_Setting_Notification command, see *CC-Link Host Object (F7h), p. 29*.	At negative flank of “Initial Data Setting Request”.
Error Status (see *Diagnostics, p. 14*, for more informaton	When there is at least one latching event, and “Error Reset Request” is false	“Error Reset Request” is set.
Remote READY (If both the Set and Clear conditions are true, the Clear functionality is given priority)	(initial setting) - At the rising edge of “Initial Data Processing Complete”	Either “Initial Data Setting Request” is set or at least one latching event with major severity exist
	(runtime) - When “Error Status” and “Error Reset Request” is false	

ⓘ *The initial value of the System Area is false, i.e. all flags are cleared during startup.*

C Certification Information

C.1 Basics

The following steps are necessary to perform to obtain a certification:

1. Change Vendor Code:

   Replace the HMS Vendor ID with a unique Vendor Code. This is done by implementing the CC-Link Host Object (F7h), instance #1, attribute #1 and returning the Vendor Code when receiving a Get_Attribute request.

   Membership in the CLPA organization is necessary to obtain a vendor code. The vendor code consists of digits 5 - 8 in the CLPA ID number, issued when you join.

2. Increment SW Version:

   Increment the SW version in the CC-Link Host Object (F7h) if you want to keep track of the versions. This number should be incremented at each change in functionality, leading to a new recertification. Implement the CC-Link Host Object (F7h), instance #1, attribute #2 and return the software version when receiving a Get_Attribute request.

3. Check Hold/Clear Attribute Setting

   The setting of the CC-Link Host Object (F7h), instance 1, attribute 6 (Output Hold/Clear), must comply with the corresponding output failstate on error, applied by the host (PLC).

4. Generate a new CSP+ file

   A CSP+ file, that complies with the customized implementation, must be generated.

These steps are the smallest possible amount of actions that you need to perform to obtain a certification.

C.2 Model Code

When the module is delivered, the Model Code (CC-Link Host Object (F7h), Instance #1, Attribute #3) is set to 7Fh (Generic device). If the host application is similar to an existing CC-Link profile, this code should be changed to reflect that profile.

C.3 CC-Link Version 2.00

By implementing the attribute Network Settings (#4) in the CC-Link Host Object (F7h) it is possible to customize the implementation for CC-Link version 2.00 and use larger data sizes through extension cycles.

For conformance test of CC-Link version 2.00, it is necessary to enable the CC-Link conformance test mode, see *Network CC-Link Object (08h), p. 25*, command CCL_Conf_Test_Mode.

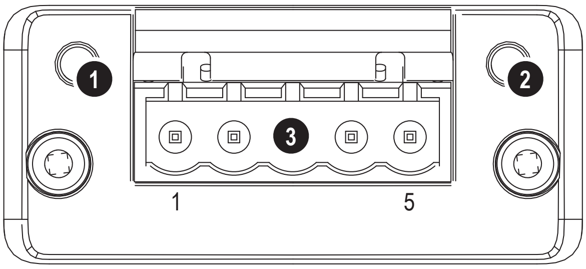
D Exception Information

When the module has entered the EXCEPTION state, further CC-Link specific details about the problem can be read from the attribyte Exception Information' (#7) in the Network Object (03h)

Value	Meaning
00h	No information.
01h	Value out of range for SW Version attribute of CC-Link object.
02h	Value out of range for CC-Link version in the Network Settings attribute of the CC-Link object.
03h	Value out of range for Number of stations in the Network Settings attribute of the CC-Link object.
04h	Invalid value for Extension cycles in the Network Settings attribute of the CC-Link object.
05h	Invalid value for System area handler attribute of the CC-Link object.
06h	CC-Link version and extension cycles are incompatible.
07h	Data mapped at too high offset in CC-Link map for CC-Link version 1.
08h	Data mapped at offset where System area is automatically located.
09h	No room for automatically located System area in CC-Link version 1.
0Ah	Data mapped at too high offset in CC-Link map for specified network settings.
0Bh	System area location out of range specified in Network settings.
0Ch	System area located at too high offset in CC-Link map for CC-Link version 1.
0Dh	Data mapped at System area location.
0Eh	(not used)
0Fh	(not used)
10h	The set Device address is too high for the used number of occupied stations.
11h	Value out of range for the Hold/Clear status attribute (#6) of the CC-Link Host Object

E Technical Specification

E.1 Front View

#	Item	
1	RUN LED	
2	ERROR LED	
3	CC-Link connector	

E.1.1 RUN LED

LED State	Indication/Description
Off	No network participation, timeout status (no power)
Green	Participating, normal operation
Red	Major fault (FATAL error)

E.1.2 ERROR LED

LED State	Indication/Description
Off	No error detected (no power)
Red	Major fault (Exception or FATAL event)
Red, flickering	CRC error (temporary flickering)
Red, flashing	Station Number or Baud rate has changed since startup (flashing)

E.1.3 CC-Link Connector

Pin	Signal	Comment
1	DA	Positive RS485 RxD/TxD
2	DB	Negative RS485 RxD/TxD
3	DG	Signal Ground
4	SLD	Cable Shield
5	FG	Functional Earth

E.2 Functional Earth (FE) Requirements

In order to ensure proper EMC behavior, the Anybus CompactCom 40 CC-Link must be properly connected to functional earth via the FE pad / FE mechanism described in the general *Anybus CompactCom M40 Hardware Design Guide*. If the brick version is used, please make sure that the hardware is properly connected to FE.

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

E.3 Power Supply

E.3.1 Supply Voltage

The module requires a regulated 3.3V power source as specified in the general *Anybus CompactCom M40 Hardware Design Guide*.

E.3.2 Power Consumption

The Anybus CompactCom 40 CC-Link 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 general *Anybus CompactCom M40 Hardware Design Guide*.

The current hardware design consumes up to 280 mA (RMS), with full Gigabit communication on both ports.

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 M40 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.

E.4 Environmental Specification

Consult the Anybus CompactCom Hardware M40 Design Guide for further information.

E.5 EMC Compliance

Consult the Anybus CompactCom Hardware M40 Design Guide for further information.

F Timing & Performance

F.1 General Information

This chapter specifies timing and performance parameters that are verified and documented for the Anybus CompactCom 40 CC-Link.

Category	Parameters
Startup Delay	T1, T2
NW_INIT Handling	T100
Event Based WrMsg Busy Time	T103
Event Based Process Data Delay	T101, T102

For further information, please consult the Anybus CompactCom 40 Software Design Guide.

F.2 Internal Timing

F.2.1 Startup Delay

The following parameters are defined as the time measured from the point where /RESET is released to the point where the specified event occurs.

Parameter	Description	Max.	Unit.
T1	The Anybus CompactCom 40 CC-Link module generates the first application interrupt (parallel mode)	11	ms
T2	The Anybus CompactCom 40 CC-Link module is able to receive and handle the first application telegram (serial mode)	11	ms

F.2.2 NW_INIT Handling

This test measures the time required by the Anybus CompactCom 40 CC-Link module to perform the necessary actions in the NW_INIT-state.

Parameter	Conditions
No. of network specific commands	Max.
No. of ADIs (single UINT8) mapped to Process Data in each direction. (If the network specific maximum is less than the value given here, the network specific value will be used.)	32
Event based application message response time	> 1 ms
Ping-pong application response time	> 10 ms
No. of simultaneously outstanding Anybus commands that the application can handle	1

Parameter	Description	Communication	Max.	Unit.
T100	NW_INIT handling	Event based modes	14	ms

F.2.3 Event Based WrMsg Busy Time

The Event based WrMsg busy time is defined as the time it takes for the module to return the H_ WRMSG area to the application after the application has posted a message.

Parameter	Description	Max.	Unit.
T103	H_WRMSG area busy time	16	µs

F.2.4 Event Based Process Data Delay

“Read process data delay” is defined as the time from when the last bit of the network frame has been received by the network interface, to when the RDPDI interrupt is asserted to the application.

“Write process data delay” is defined as the time from when the application exchanges write process data buffers, to when the first bit of the new process data frame is sent out on the network.

The tests were run in 16-bit parallel event mode, with interrupts triggered only for new process data events. The delay was measured in CC-Link Version 1.10 only. CC-Link Version 2.00 employs several CC-Link cycles(2, 4, 8) to transfer the data to/from the module, which means that the cycle time is a greater factor than the latency caused by the module.

The delay added by the transceiver and other hardware has not been included, as this delay is insignificant compared to the total process data delay.

Parameter	Description	Delay (min.)	Delay (typ.)	Delay (max.)	Unit
T101	Read process data delay	-	-	33	µs
T102	Write process data delay	-	-	35	µs

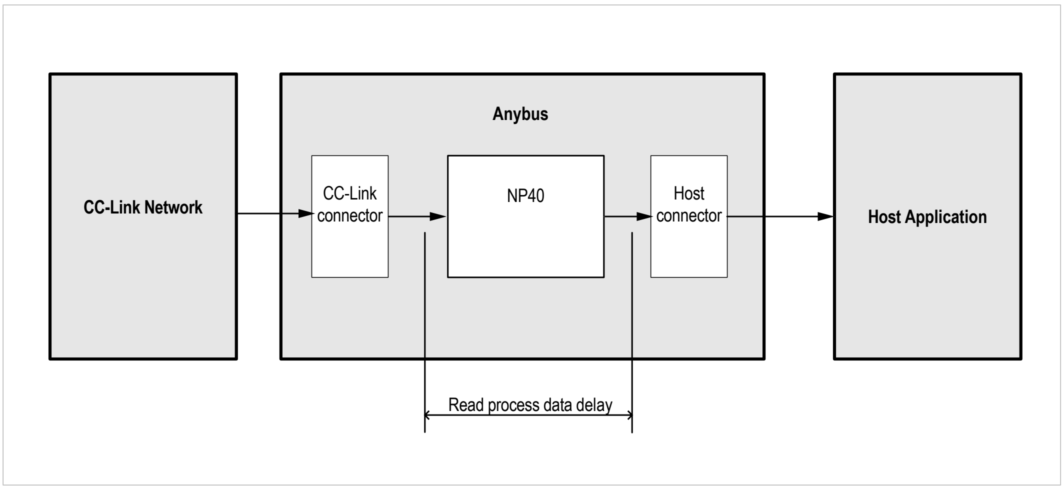


Fig. 1

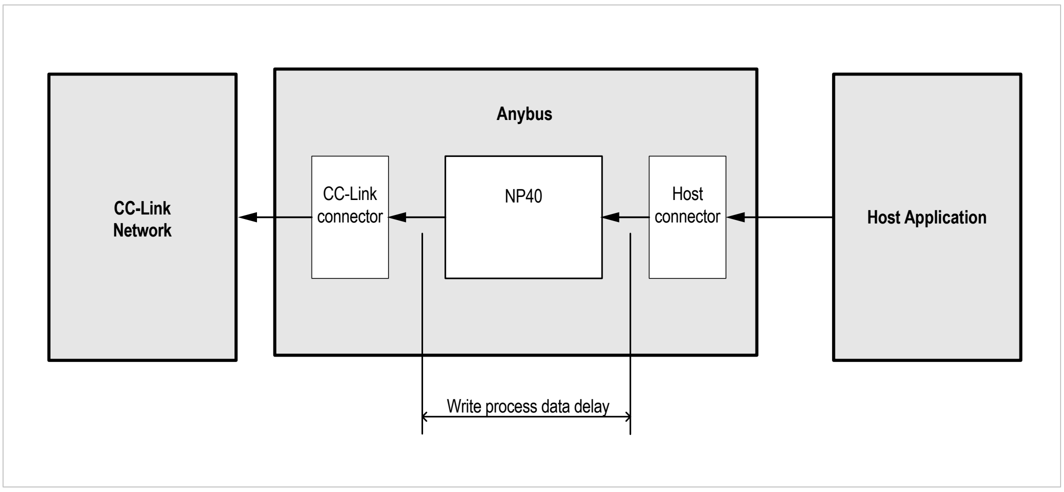


Fig. 2

G Backward Compatibility

The Anybus CompactCom M40 series of industrial network modules have significantly better performance and include more functionality than the modules in the Anybus CompactCom 30 series. The 40 series is backward compatible with the 30 series in that an application developed for the 30 series should be possible to use with the 40 series, without any major changes. Also it is possible to mix 30 and 40 series modules in the same application.

This appendix presents the backwards compatibility issues that have to be considered for Anybus CompactCom 40 CC-Link, when designing with both series in one application, or when adapting a 30 series application for the 40 series.

G.1 Initial Considerations

There are two options to consider when starting the work to modify a host application developed for Anybus CompactCom 30-series modules to also be compatible with the 40-series modules:

- Add support with as little work as possible i.e. reuse as much as possible of the current design.
  - This is the fastest and easiest solution but with the drawback that many of the new features available in the 40-series will not be enabled (e.g. enhanced and faster communication interfaces, larger memory areas, and faster communication protocols).
  - You have to check the hardware and software differences below to make sure the host application is compatible with the 40-series modules. Small modifications to your current design may be needed.
- Make a redesign and take advantage of all new features presented in the 40-series.
  - A new driver and host application example code are available at www.anybus.com/starterkit40 to support the new communication protocol. This driver supports both 30-series and 40-series modules.
  - You have to check the hardware differences below and make sure the host application is compatible with the 40-series modules.

ⓘ *This information only deals with differences between the 30-series and the 40-series.*

Link to support page: www.anybus.com/support.

G.2 Hardware Compatibility

Anybus CompactCom is available in three hardware formats; Module, Chip, and Brick.

G.2.1 Module

The modules in the 30-series and the 40-series share physical characteristics, like dimensions, outline, connectors, LED indicators, mounting parts etc. They are also available as modules without housing.

Fig. 3 Anybus CompactCom M30/M40

G.2.2 Chip

The chip (C30/C40) versions of the Anybus CompactCom differ completely when it comes to physical dimensions.

> ! There is no way to migrate a chip solution from the 30-series to the 40-series without a major hardware update.

G.2.3 Brick

The Anybus CompactCom B40-1 does not share dimensions with the Anybus CompactCom B30. The B40-1 is thus not suitable for migration. However HMS Industrial Networks has developed a separate brick version in the 40-series, that can be used for migration. This product, B40-2, shares dimensions etc. with the B30. Please contact HMS Industrial Networks for more information on the Anybus CompactCom B40-2.

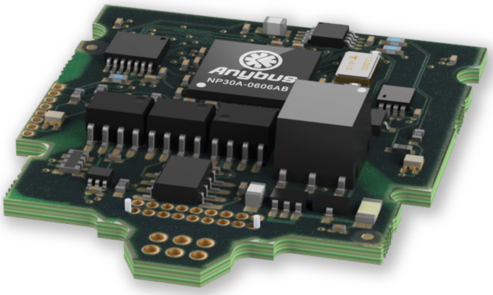


Fig. 4 Anybus CompactCom B30

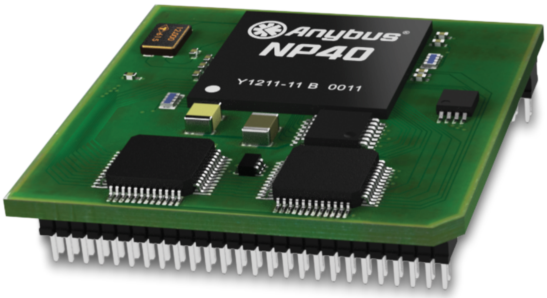


Fig. 5 Anybus CompactCom B40–1 (not for migration)

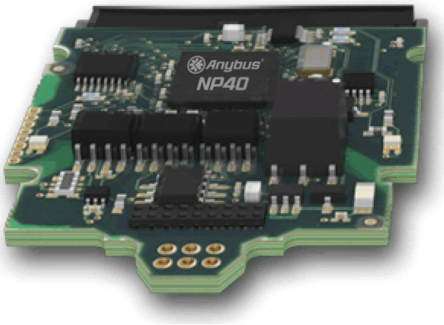


Fig. 6 Anybus CompactCom B40–2

G.2.4 Host Application Interface

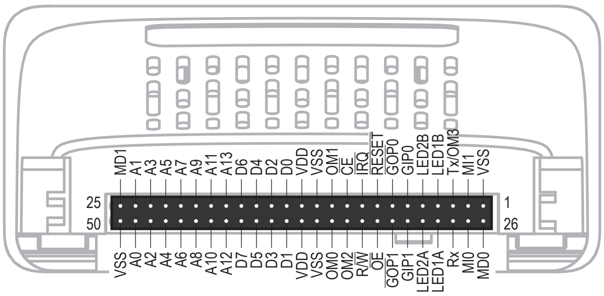


Fig. 7

Some signals in the host application interface have modified functionality and/or functions which must be checked for compatibility. See the following sections.

Tx/OM3

In the 30-series, this pin is only used for Tx. It is tri-stated during power up, and driven by the Anybus CompactCom UART after initialization. In the 40-series this pin is used as a fourth operating mode setting pin (OM3). During startup after releasing the reset, this pin is read to determine the operating mode to use. The pin is then changed to a Tx output.

In the 40-series, this pin has a built-in weak pull-up. If this pin, on a 30-series module or brick is unconnected, pulled high, or connected to a high-Z digital input on the host processor, it will be compatible with the 40-series. An external pull-up is recommended, but not required.

! If this pin is pulled low by the host during startup in a 30-series application, any 40-series module or brick, substituted in the application, will not enter the expected operating mode.

Related Information: Anybus CompactCom M40 Hardware Design Guide (HMSI-216-126), Section “Application Connector Pin Overview”

Module Identification (MI[0..1])

These pins are used by the host application (i.e. your product) to identify what type of Anybus CompactCom that is mounted. The identification differs between the 30-series and the 40-series.

If your software use this identification you need to handle the new identification value.

MI1	MI0	Module Type
LOW	LOW	Active Anybus CompactCom 30
HIGH	LOW	Active Anybus CompactCom 40

MI[0..1] shall only be sampled by the application during the time period from power up to the end of SETUP state. The pins are low at power up and before reset release.

Related Information: *Anybus CompactCom M40 Hardware Design Guide (HMSI-216-126)*, Section “Settings/Sync”.

GIP[0..1]/LED3[A..B]

These pins are tri-stated inputs by default in the 30-series. In the 40-series, these pins are tri-stated until the state NW_INIT. After that they become open-drain, active low LED outputs (LED3A/LED3B).

No modification of the hardware is needed, if your current design has

- tied these pins to GND
- pulled up the pins
- pulled down the pins
- left the pins unconnected

However, if the application drive the pins high, a short circuit will occur.

If you connect the pins to LEDs, a pull-up is required.

In the 40-series, there is a possibility to set the GIP[0..1] and GOP[0..1] in high impedance state (tri-state) by using attribute #16 (GPIO configuration) in the Anybus object (01h). I.e. if it is not possible to change the host application hardware, this attribute can be configured for high impedance state of GIP and GOP before leaving NW_INIT state.

Related Information: *Anybus CompactCom M40 Hardware Design Guide (HMSI-216-126)*, Section “LED Interface/D8-D15 (Data Bus)”.

GOP[0..1]/LED4[A..B]

These pins are outputs (high state) by default in the 30-series. In the 40-series, these pins are tri-stated until the state NW_INIT, and after that they become push-pull, active low LED outputs (LED4A/LED4B).

This change should not affect your product.

Related Information: *Anybus CompactCom M40 Hardware Design Guide (HMSI-216-126)*, Section 3.2.3, “LED Interface/D8-D15 (Data Bus)”.

Address Pins A[11..13]

The address pins 11, 12, and 13 are ignored by the 30-series. These pins must be high when accessing the 40-series module in backwards compatible 8-bit parallel mode. If you have left these pins unconnected or connected to GND, you need to make a hardware modification to tie them high.

Max Input Signal Level (V_{IH})

The max input signal level for the 30-series is specified as V_{IH}=V_{DD}+0,2 V, and for the 40-series as V_{IH}=3.45 V. Make sure that you do not exceed 3.45 V for a logic high level.

RMII Compatibility

If the RMII mode is being used on an Anybus CompactCom 40 module and it is desired to remain compatible with the 30 series, it is important to disable this connection when switching to an Anybus CompactCom 30 module due to pin conflicts. The RMII port of the host processor should be set to tristate by default, and only be enabled if an RMII capable Anybus CompactCom 40 is detected. In case the RMII connection cannot be disabled through an internal hardware control on the host processor, it will be necessary to design in external hardware (i.e. a FET bus switch) to prevent short circuits

Related Information: *Anybus CompactCom M40 Hardware Design Guide (HMSI-216-126)*, Section 3.2.5, “RMII — Reduced Media-Independent Interface”.

G.3 General Software

G.3.1 Extended Memory Areas

The memory areas have been extended in the 40-series, and it is now possible to access larger sizes of process data (up to 4096 bytes instead of former maximum 256 bytes) and message data (up to 1524 bytes instead of former maximum 255 bytes). The 30-series has reserved memory ranges that the application should not use. The 40-series implements new functionality in some of these memory areas.

To use the extended memory areas you need to implement a new communication protocol which is not part of this document.

Memory areas not supported by the specific network cannot be used. Make sure you do not access these areas, e.g. for doing read/write memory tests.

Related Information: *Anybus CompactCom 40 Software Design Guide (HMSI-216-125)*, Section “Memory Map”

G.3.2 Faster Ping-Pong Protocol

The ping-pong protocol (the protocol used in the 30-series) is faster in the 40-series. A 30-series module typically responds to a so called ping within 10-100 µs. The 40-series typically responds to a ping within 2 µs.

Interrupt-driven applications (parallel operating mode) may see increased CPU load due to the increased speed.

G.3.3 Requests from Anybus CompactCom to Host Application During Startup

All requests to software objects in the host application must be handled and responded to (even if the object does not exist). This applies for both the 30-series and the 40-series. The 40-series introduces additional objects for new functionality.

There may also be additional commands in existing objects added to the 40-series that must be responded to (even if it is not supported).

If your implementation already responds to all commands it cannot process, which is the expected behavior, you do not need to change anything.

G.3.4 Anybus Object (01h)

Attribute	30-series	40-series	Change/Action/Comment
#1, Module Type	0401h	0403h	Make sure the host application accepts the new module type value for the 40-series.
#15, Auxiliary Bit	Available	Removed	It is not possible to turn off the “Changed Data Indication” in the 40-series. Also see “Control Register CTRL_AUX-bit” and “Status Register STAT_AUX-bit” below.
#16, GPIO Configuration	Default: General input and output pins	Default: LED3 and LED4 outputs	See also .. • *GIP[0..1]/LED3[A..B], p. 46* • *GOP[0..1]/LED4[A..B], p. 46*

G.3.5 Control Register CTRL_AUX-bit

30-series The CTRL_AUX bit in the control register indicates to the Anybus CompactCom if the process data in the current telegram has changed compared to the previous one.

40-series The value of the CTRL_AUX bit is always ignored. Process data is always accepted.

All released Anybus CompactCom 30 example drivers from Anybus CompactCom comply with this difference.

Related Information: *Anybus CompactCom 40 Software Design Guide (HMSI-216-125)*, section “Control Register”.

G.3.6 Status Register STAT_AUX-bit

30-series The STAT_AUX bit in the status register indicates if the output process data in the current telegram has changed compared to the previous one. This functionality must be enabled in the Anybus object (01h), Attribute #15. By default, the STAT_AUX bit functionality is disabled.

40-series The STAT_AUX bit indicates updated output process data (not necessarily changed data) from the network compared to the previous telegram. The functionality is always enabled.

All released Anybus CompactCom 30 example drivers from HMS Industrial Networks comply with this difference.

Related Information: *Anybus CompactCom 40 Software Design Guide (HMSI-216-125)*, section “Status Register”.

G.3.7 Control Register CTRL_R-bit

30-series The application may change this bit at any time.

40-series For the 8-bit parallel operating mode, the bit is only allowed to transition from 1 to 0 when the STAT_M-bit is set in the status register. When using the serial operating modes, it is also allowed to transition from 1 to 0 in the telegram immediately after the finalizing empty fragment.

All released Anybus CompactCom 30 example drivers from HMS Industrial Networks comply with this difference.

Related Information: *Anybus CompactCom 40 Software Design Guide (HMSI-216-125)*, section “Control Register”.

G.3.8 Modifications of Status Register, Process Data Read Area, and Message Data Read Area

In the 40-series, the Status Register, the Process Data Read Area, and the Message Data Read Area are write protected in hardware (parallel interface). If the software for some reason writes to any of those areas, a change is needed.

All releasedAnybus CompactCom 30 example drivers from HMS Industrial Networks comply with this difference.

G.4 Network Specific — CC-Link

G.4.1 Network CC-Link Object (08h)

The specific CC-Link mapping commands: Map_ADI_Specified_Write_Area, Map_ADI_Specified_Read_Area have been removed. With these two previously used commands it was possible to freely specify the location of the mapped data in the CC-Link address map; this is not possible anymore.

The mapping is now handled with the commands: Map_ADI_Write_Area, Map_ADI_Read_Area or Map_ADI_Write_Ext_Area and Map_ADI_Read_Ext_Area. The location of the mapped data in the CC-Link address map can now only be managed by using these commands in conjunction with the chronological order the commands are sent.

See *Anybus CompactCom 40 CC-Link Network Guide* for the new Process Data mapping scheme details.

G.4.2 Network Object (03h)

Process Data

A new default Process Data mapping scheme has been implemented in the 40-series. Data type BOOL is now mapped to the Word-area. In the 40-series, use the new BITx data types instead to map the data to the Bit-area.

All bit data must be mapped before all other data types for the data to be mapped to the Bit-area. If mapping bit data after having mapped other “non-bit-data” the data will be mapped to the Word-area.

The change was made to get a more logical and faster mapping.

You need to make sure the process data is mapped according to the above in the host application.

G.4.3 Diagnostic Object (02h)

To use the Diagnostic Events in conjunction with the automatic System Area Handler (CC-Link Host Object (F7h), Attribute #5, System Area Handler) in the Anybus CompactCom 40 CC-Link, there is one modification required. The application is now required to use the Diagnostic Events with the severity representing Latching Events or handle the system area completely in the application. For details on Diagnostic Event with latching severity, see the *Anybus CompactCom 40 Software Design Guide*, Diagnostic Object.

If the Diagnostic Event created is not of the latching severity, the system area will not work according the CC-Link specification.

If the automatic System Area Handler functionality is used previously, add the use of Diagnostic Events with a latching event severity or let the host application handle the system area completely.

This page intentionally left blank

Box 4126
300 04 Halmstad, Sweden

info@hms.se