Network Guide

Anybus® CompactCom 40 Ethernet POWERLINK

Doc.Id. HMSI-27-219
Rev. 1.70

Connecting Devices™

HALMSTAD • CHICAGO • KARLSRUHE • TOKYO • BEIJING • MILANO • MULHOUSE • COVENTRY • PUNE • COPENHAGEN

HMS Industrial Networks
Mailing address: Box 4126, 300 04 Halmstad, Sweden
Visiting address: Stationsgatan 37, Halmstad, Sweden

E-mail: info@hms-networks.com
Web: www.anybus.com

Important User Information

This document is intended to provide a good understanding of the functionality offered by Ethernet POWERLINK. The document only describes the features that are specific to the Anybus CompactCom 40 Ethernet POWERLINK. For general information regarding the Anybus CompactCom 40, consult the Anybus CompactCom 40 design guides.

The reader of this document is expected to be familiar with high level software design, and communication systems in general. The use of advanced Ethernet POWERLINK-specific functionality may require in-depth knowledge in Ethernet POWERLINK networking internals and/or information from the official Ethernet POWERLINK specifications. In such cases, the people responsible for the implementation of this product should either obtain the Ethernet POWERLINK specification to gain sufficient knowledge or limit their implementation in such a way that this is not necessary.

Liability

Every care has been taken in the preparation of this manual. Please inform HMS Industrial Networks AB of any inaccuracies or omissions. The data and illustrations found in this document are not binding. We, HMS Industrial Networks AB, 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 AB. HMS Industrial Networks AB 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 AB 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 AB cannot assume responsibility for actual use based on these examples and illustrations.

Intellectual Property Rights

HMS Industrial Networks AB 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 US and other countries.

Trademark Acknowledgements

Anybus ® is a registered trademark of HMS Industrial Networks AB. All other trademarks are the property of their respective holders.

Warning:	This is a class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.
ESD Note:	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.

Oct 2015 Doc Id HMSI-27-219

Table of Contents

Preface **About This Document**

Related Documents 5
Document History 5
Conventions & Terminology 6
Abbreviations 6
Support 6

Chapter 1 **About the Anybus CompactCom 40 Ethernet POWERLINK**

General 7
Features 7

Chapter 2 **Basic Operation**

General Information 8
POWERLINK and CANopen Implementation 8
Software Requirements 8
Device Customization 9
Device Identity 9
XML Device Description (XDD) 9
Communication Settings 10
Diagnostics 10
Synchronization 10
Network Data Exchange 11
Application Data (ADIs) 11
Process Data 11
Network Reset Handling 12
Software Reset 12
Reset Node 12
Reset Communication 12
Reset Configuration 12

Chapter 3 **Object Dictionary**

Standard Objects 13
General 13
Object Entries 14
Access Rights 21
Manufacturer Specific Objects 22
General 22
Translation of Status Codes 22
Network Data Format 23
Object Entries 24

Chapter 4 Anybus Module Objects

General Information ... 25
Anybus Object (01h) ... 26
Diagnostic Object (02h) ... 27
Network Object (03h) ... 28
Network Configuration Object (04h) ... 30
File System Interface Object (0Ah) ... 32

Chapter 5 Host Application Objects

General Information ... 33
Sync Object (EEh) ... 34
Ethernet Host Object (F9h) ... 36
POWERLINK Object (E9h) ... 37

Appendix A Categorization of Functionality

Basic ... 38
Extended ... 38

Appendix B Implementation Details

SUP-Bit Definition ... 39
Anybus State Machine ... 39

Appendix C Timing & Performance

General Information ... 40
Event Based Process Data Delay ... 40

Appendix D Technical Specification

Front View ... 41
Protective Earth (PE) Requirements ... 42
Power Supply ... 42
Environmental Specification ... 42
EMC Compliance ... 42

P. About This Document

For more information, documentation etc., please visit the HMS website, 'www.anybus.com'.

P.1 Related Documents

Document	Author
Anybus CompactCom 40 Software Design Guide	HMS
Anybus CompactCom 40 Hardware Design Guide	HMS
Anybus CompactCom 40 Software Driver User Guide	HMS

P.2 Document History

Summary of Recent Changes (1.60 ... 1.70)

Change	Page(s)
Corrected data types for object attributes #3 and #4 in POWERLINK Object (E9h)	37
Changed "network identity" to "device identity", added sentence to important section	9
Changed advanced to extended	
Network Configuration Object (04h) category changed to basic	30

Revision List

Revision	Date	Author(s)	Chapter(s)	Description
1.00	2014-03-25	KeL		First official revision
1.50	2014-07-11	KaD	All	Major update
1.60	2015-05-28	KeL	2, 3, 4, 5, D	Misc updates
1.70	2015-10-16	KeL	2, 5	Minor Updates

P.3 Conventions & Terminology

The following conventions are used throughout this manual:

- Numbered lists provide sequential steps
- Bulleted lists provide information, not procedural steps
- The terms 'Anybus' or 'module' refers to the Anybus CompactCom 40 module, brick or chip.
- The terms 'host' or 'host application' refers to the device that hosts the Anybus module
- Hexadecimal values are either written in the format NNNNh or the format 0xNNNN, where NNNN is the hexadecimal value

P.4 Abbreviations

Abbreviation	Meaning
SCNM	Slot Communication Network Management
CN	Controlled Node
EPL	Ethernet POWERLINK
MN	Managing Node
XDD	XML Device Description
RPDO	Receive Process Data Object
TPDO	Transmit Process Data Object

P.5 Support

For general contact information and where to find support, please refer to the contact and support pages at www.anybus.com.

1. About the Anybus CompactCom 40 Ethernet POWERLINK

1.1 General

The Anybus CompactCom 40 Ethernet POWERLINK communication module provides instant Ethernet POWERLINK connectivity via the patented Anybus CompactCom host interface. Any device that supports this standard can take advantage of the features offered by the module, allowing seamless network integration regardless of network type.

The modular approach of the Anybus CompactCom 40 platform allows the implementation to be extended to fit specific application requirements. Furthermore, the identity can be customized, allowing the end product to appear as a vendor-specific implementation rather than a generic Anybus module.

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.

1.2 Features

- Two Ethernet POWERLINK ports (RJ45)
- Supports Ethernet POWERLINK V2.0 Communication Profile Specification version 1.2.0 (Controlled Node)
- Integrated hub
- 100 Mbit/s, half duplex operation
- 200 µs cycle time
- Supports ring redundancy
- Customizable identity information
- Supports 1 TPDO and 1 RPDO (each can hold 1490 bytes)
- Up to 57343 ADIs
- Adaptable XDD file included
- Supports segmented SDO transfer
- PollResponse Chaining
- Multiplexing
- Support for SYNC functionality

2. Basic Operation

2.1 General Information

2.1.1 POWERLINK and CANopen Implementation

Ethernet POWERLINK (EPL) is a deterministic real-time protocol for standard Ethernet. It is an open protocol managed by the Ethernet POWERLINK Standardization Group (EPSG).

Ethernet POWERLINK extends Ethernet according to the IEEE 802.3 standard with mechanisms to transfer data with predictable delivery. The communication meets timing demands typical for high performance automation and motion applications.

The Ethernet POWERLINK communication profile is based on CANopen communication profiles DS301 and DS302. Based on this communication profile, the multitude of CANopen device profiles can be used in a POWERLINK environment without changes.

Ethernet POWERLINK manages the network traffic using dedicated time-slots for isochronous and asynchronous data. Only one networked device at the time gains access to the network media. Thus transmission of data will never interfere and precise communication timing is guaranteed. The mechanism is called Slot Communication Network Management (SCNM). SCNM is managed by one particular networked device - the Managing Node (MN). All other nodes are called Controlled Nodes. The module can participate as a Controlled Node (CN) in Ethernet POWERLINK networks.

2.1.2 Software Requirements

Generally, no additional network support code needs to be written in order to support the Anybus CompactCom 40 Ethernet POWERLINK. However, due to the nature of the Ethernet POWERLINK networking system, certain restrictions must be taken into account:

- One diagnostic instance (See "Diagnostic Object (02h)" on page 27) can be created by the host application in event of a major fault.[1] This event will not show on the POWERLINK network.
- Only ADIs with instance numbers less than or equal to 57343 can be accessed from the network.

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

See also...

- "Diagnostic Object (02h)" on page 27 (Anybus Module Objects)
- Anybus CompactCom 40 Software Design Guide, "Application Data Object (FEh)"

1. This limit is set by the module, not by the network.

2.2 Device Customization

2.2.1 Device Identity

By default, the module uses the following identity settings:

- Vendor ID: 0000001Bh (HMS Industrial Networks)
- Device Type: 00000000h (Generic Device)
- Product Code: 00000028h (Anybus CompactCom 40 Ethernet POWERLINK)
- Manufacturer Device Name: 'Anybus CompactCom 40 Ethernet POWERLINK'
- Manufacturer Name: 'HMS'

Optionally, it is possible to customize the identity of the module by implementing the corresponding instance attributes in the POWERLINK Host Object.

See also...

- "POWERLINK Object (E9h)" on page 37 (Host Application Object)

IMPORTANT: *If the identity settings are changed, recertification of the module is needed. For the end product to pass the EPSG conformace tests and be certified , a separate Vendor ID has to be requested from EPSG (free of charge).*

2.2.2 XML Device Description (XDD)

On Ethernet POWERLINK, the characteristics of a device is stored in an XML file with the suffix XDD. This file is used by configuration tools etc. when setting up the network configuration. HMS supplies a standard (generic) XDD file, which corresponds to the default settings in the module. However, all implementations will add changes to the default settings, making it necessary to create a custom XDD file where the changes are reflected. This invalidates the default identity information and requires recertification of the product.

IMPORTANT: *HMS approves use of the standard XDD file only under the condition that it matches the actual implementation and that the identity information remains unchanged.*

There is support for dynamic generation of the XDD file, including information about the configuration and host application ADIs, among other things. The XDD file can either be read via the network (object 1021h in the Object Dictionary) or via the Anybus File System Object (0Ah).

To be able to automatically generate an XDD file, the host application must support the command Get_Instance_Number_By_Order in the Application Data Object. If this command is not supported, an error code will be returned when trying to read the XDD file.

For more information see

- "Object Entries" on page 14
- "File System Interface Object (0Ah)" on page 32
- Application Object (see Anybus CompactCom 40 Software Design Guide)

2.3 Communication Settings

As with other Anybus CompactCom products, network related communication settings are grouped in the Network Configuration Object (04h).

In this case, this includes...

- **Node ID Setting**

  The Anybus CompactCom module participates as a Controlled Node in the network, with a Node ID in the range 1 - 239.

See also...

- "Network Configuration Object (04h)" on page 30 (Anybus Module Object)

2.4 Diagnostics

A major unrecoverable event will cause the module to enter the EXCEPTION state. This will be recorded in the Diagnostic Object, but not reported to the POWERLINK network. The module will cease communication on the network without notice.

See "Diagnostic Object (02h)" on page 27 (Anybus Module Object).

2.5 Synchronization

The Anybus CompactCom 40 Ethernet POWERLINK module supports synchronization. To utilize SYNC in the application, implement the SYNC object. See "Sync Object (EEh)" on page 34.

If synchronous operation is supported it will be enabled for all POWERLINK NMT states with a fixed cycle length and cyclic data exchange, i.e. the POWERLINK NMT states Pre Operational 2, Ready to Operate and Operational.

If SYNC is supported, the SYNC cycle time will be written to the SYNC object as soon as the value is received from the network. If SYNC is not supported, it will be written after the NMTResetConfiguration command.

The SYNC signal must be synced in the Anybus state WAIT_PROCESS (where the SYNC signal may differ much between the configured and the actual value).

When SYNC is established, the SYNC signal is produced upon the expected reception of SoC frames, in all NMT states that transmit SoC frames.

2.6 Network Data Exchange

2.6.1 Application Data (ADIs)

Application Data Instances (ADIs) can be accessed from the network via dedicated object entries in the Manufacturer Specific Range (2001h - FFFFh), see "Manufacturer Specific Objects" on page 20.

Note: The XDD file must match the actual ADI implementation in the host application.

To be able to automatically generate an XDD file, the host application must support the command Get_Instance_Number_By_Order in the Application Data Object. If this command is not supported, an error code will be returned when trying to read the XDD file.

The generated XDD file contains information about the ADIs present in the host application and specifies e.g. name, type, access rights and the PDO mapping capabilities of each ADI. It is not possible to retrieve the names of individual subindex fields in the ADIs, therefore these have a default generic name, with the subindex number attached to the end.

2.6.2 Process Data

On POWERLINK, ADIs mapped as Process Data can be exchanged on the bus, following the predefined time scheme with slots for sent and received data. Process data is mapped as PDOs, and the module supports 1 TPDO and 1 RPDO.

Process data can be either statically or dynamically mapped. Dynamic mapping is configured by the Managing Node of the network. To support dynamic mapping, the application has to implement the Remap_ADI_Write_Area and Remap_ADI_Read_Area commands in the Application Data Object. See Application Data Object (FEh) in Anybus CompactCom 40 Software Design Guide for more information. If these commands are not implemented, the static mapping, provided by the application using the Map_Adi command of the Network Object, will be used.

There are several object entries in the Object Dictionary that have different access rights depending on whether dynamic mapping or static mapping is used. If dynamic mapping is supported, the object entries can be changed and are marked as "rw" (read/write). If the application uses static mapping, they cannot be changed and are marked as "ro" (read only).

The values in object entries 1400h, 1600h, 1800h and 1A00h cannot be changed in POWERLINK states corresponding to IDLE or PROCESS_ACTIVE (See "Anybus State Machine" on page 39). If the network tries to change any of these entries when the device is in one of those states the command will be aborted by the CompactCom and an SDO abort code will be returned to the network.

See object entries 1400h, 1600h, 1800h and 1A00h in "Object Dictionary" on page 13.

2.7 Network Reset Handling

2.7.1 Software Reset

If a software reset is requested from the network, the module will issue a Reset command to the Application Object (FFh). The module will enter the EXCEPTION state. The module then has to be power-cycled using the reset pin.

2.7.2 Reset Node

If a software reset is requested from the network, the module will issue a Reset command to the Application Object (FFh). The module will enter the EXCEPTION state. The module then has to be power-cycled using the reset pin.

2.7.3 Reset Communication

The module will return to state WAIT_PROCESS, and the parameters in the Communication Profile Area (1000h-1FFFh) will be reset.

2.7.4 Reset Configuration

The module will return to state WAIT_PROCESS and the configuration parameters, set in the object dictionary, will be used to generate the active configuration. COP object entries marked as “valid on reset” will be activated.

3. Object Dictionary

3.1 Standard Objects

3.1.1 General

The standard object dictionary is implemented according to the EPSG (Ethernet POWERLINK Standardization Group) DS301. Note that certain object entries correspond to settings in the POWERLINK Object (E9h).

3.1.2 Object Entries

Note: None of these objects can be mapped as TPDO or RPDO.

Index	Object Name	Sub-index	Description	Type	Access[a]	Value/Notes
1000h	NMT_DeviceType_U32	00h	Device Type	UNSIGNED32	const	0000 0000h (No profile), see "POWERLINK Object (E9h)" on page 37.
1001h	ERR_ErrorRegister_U8	00h	Error register	UNSIGNED8	ro	00h (default)
1006h	NMT_CycleLen_U32	00h	Cycle time of full POWERLINK cycle.	UNSIGNED32	rw, valid on reset	200 (default) Valid range 200 - 2147483 (µs)
1008h	NMT_ManufactDevName_VS	00h	Manufacturer device name	VISIBLE_STRING64	const	This information is determined by the POWERLINK Object, which can optionally be implemented in the host application. See "POWERLINK Object (E9h)" on page 37.
1009h	NMT_ManufactHwVers_VS	00h	Manufacturer hardware version	VISIBLE_STRING64	const	
100Ah	NMT_ManufactSwVers_VS	00h	Manufacturer software version	VISIBLE_STRING64	const	
1018h	NMT_IdentityObject_REC	00h	Number of entries	UNSIGNED8	const	04h
		01h	VendorId_U32: Vendor ID	UNSIGNED32	const	This information is determined by the POWERLINK Object, which can optionally be implemented in the host application. See "POWERLINK Object (E9h)" on page 37.
		02h	ProductCode_U32: Product Code	UNSIGNED32	const	
		03h	RevisionNo_U32: Revision Number	UNSIGNED32	const	
		04h	SerialNo_U32: Serial Number	UNSIGNED32	const	
1020h	CFM_VerifyConfiguration_REC	00h	Number of entries	UNSIGNED8	const	02h
		01h	ConfDate_U32: Configuration Date	UNSIGNED32	rw, valid on reset	0 (default) Valid range 0 - FFFFFFFFh
		02h	ConfTime_U32: Configuration Time	UNSIGNED32	rw, valid on reset	0 (default) Valid range 0 - FFFFFFFFh
1021h	CFM_StoreDevDescrFormat_DOM	00h	The contents of an XDD file.	DOMAIN[b]	ro	If the object 1022h has the value 00h, this object will contain an XDD file adapted to the specific host application. See "XML Device Description (XDD)" on page 9.

Index	Object Name	Sub-index	Description	Type	Access[a]	Value/Notes
1022h	CFM_StoreDevDescrFormat_U16	00h	Description of the format of what is stored in object 1021h.	UNSIGNED16	ro	00h: Object 1021h contains an XDD file. FFh: Object 1021h does not contain an XDD file.
1030h	NMT_InterfaceGroup_00h_REC	00h	Number of entries	UNSIGNED8	const	09h
		01h	InterfaceIndex_U16: Interface index	UNSIGNED16	ro	0001h
		02h	InterfaceDescription_VSTR: Interface description	VISIBLE_STRING194	const	See “POWERLINK Object (E9h)” on page 37 for default value (Manufacturer name, Manufacturer product name, and Manufacturer hardware version).
		03h	InterfaceType_U8: Interface type	UNSIGNED8	const	06h
		04h	InterfaceMtu_U16: Interface maximum transmission unit	UNSIGNED16	const	1500 bytes
		05h	InterfacePhysAddress_OSTR: Interface Physical Address	OCTET_STRING6	const	MAC address, see description of “Ethernet Host Object (F9h)” on page 36..
		06h	InterfaceName_VSTR: Interface Name	VISIBLE_STRING11	ro	“Interface 1”
		07h	InterfaceOperStatus_U8: Interface Operational Status	UNSIGNED8	ro	01h (default)
		08h	InterfaceAdminSate_U8: Interface Admin State	UNSIGNED8	rw	01h (default) Valid range 00h - 01h
		09h	Valid_Bool: Valid	BOOLEAN	rw	01h (default) Valid range 00h - 01h
1300h	SDO_SequLayerTimeout_U32	00h	SDO sequence layer timeout	UNSIGNED32	rw, valid on reset	15000 (default) Valid range 100 - FFFFFFFFh
1400h	PDO_RxCommParam_00h_REC	00h	Number of entries	UNSIGNED8	const	02h
		01h	NodeID_U8: Node ID	UNSIGNED8	rw	00h (default) For static mapping: 00h For dynamic mapping: set by MN
		02h	MappingVersion_U8: Mapping version	UNSIGNED8	rw/ro[c]	00h (default) For static mapping: 00h For dynamic mapping: set by MN

Index	Object Name	Sub-index	Description	Type	Access[a]	Value/Notes
1600h	PDO_RxMappParam_00h_AU64	00h	The number of ADI host application objects mapped as receive PDO	UNSIGNED8	rw	00h (default) For static mapping: decided by application[c] For dynamic mapping: set by MN
		01h - FEh	ObjectMapping	UNSIGNED64	rw/ro[c]	-
1800h	PDO_TxCommParam_00h_REC	00h	Number of entries	UNSIGNED8	const	02h
		01h	NodeID_U8: Node ID	UNSIGNED8	rw	00h (default)
		02h	MappingVersion_U8: Mapping version	UNSIGNED8	rw/ro[c]	00h (default) For static mapping: 00h For dynamic mapping: set by MN
1A00h	PDO_TxMappParam_00h_AU64	00h	The number of ADI host application objects mapped as transmit PDO	UNSIGNED8	rw	00h (default) For static mapping: decided by application[c] For dynamic mapping: set by MN
		01h - FEh	ObjectMapping	UNSIGNED64	rw/ro[c]	-
1C0Bh	DLL_CNLossSoC_REC	00h	Number of entries	UNSIGNED8	const	03h
		01h	CumulativeCnt_U32: Cumulative count	UNSIGNED32	rw	0 (default) Valid range 0 - FFFFFFFFh **Note**: if the module is reset, this attribute will be set to its default value.
		02h	ThresholdCnt_U32: Threshold count	UNSIGNED32	ro	00h
		03h	Threshold_U32: Threshold	UNSIGNED32	rw	15 (default) Valid range 0 - FFFFFFFFh

Index	Object Name	Sub-index	Description	Type	Access[a]	Value/Notes
1C0Fh	DLL_CNCRCError_REC	00h	Number of entries	UNSIGNED8	const	03h
		01h	CumulativeCnt_U32: Cumulative count	UNSIGNED32	rw	0 (default) Valid range 0 - FFFFFFFFh **Note**: if the module is reset, this attribute will be set to its default value.
		02h	ThresholdCnt_U32: Threshold count	UNSIGNED32	ro	00h
		03h	Threshold_U32: Threshold	UNSIGNED32	rw	15 (default) Valid range 0 - FFFFFFFFh
1C14h	DLL_CNLossOfSocTolerance_U32	00h		UNSIGNED32	rw	100 000 (default) Valid range 0 - 2147483000 (ns)
1F50h	PDL_DownloadProgData_ADOM	00h	Number of entries	UNSIGNED8	ro	01h
		01h	Program	DOMAIN[b]	wo	A HIFF (HMS Interchange File Format) file, used to update the firmware of the module.
1F51h	PDL_ProgCtrl_AU8	00h	Number of entries	UNSIGNED8	ro	01h
		01h	ProgCtrl: Program control	UNSIGNED8	rw	01h (Writing a value other than 1 will result in an Abort SDO Transfer message (error code 08000024h.)
1F52h	PDL_LocVerApplSw_REC	00h	Number of entries	UNSIGNED8	const	02h
		01h	ApplSwDate_U32	UNSIGNED32	ro	Number of days between 1984-01-01 and the current software build time.
		02h	ApplSwTime_U32	UNSIGNED32	ro	Milliseconds since midnight of the current software build time.
1F81h	NMT_Node Assignment_AU32	00h	Number of entries	UNSIGNED8	rw, valid on reset	FEh (default) Valid range 01h - FEh
		01h - FEh	NodeAssignment	UNSIGNED32	rw, valid on reset	0 (default) This value is a bit field.

Index	Object Name	Sub-index	Description	Type	Access[a]	Value/Notes
1F82h	NMT_FeatureFlags_U32	00h	Feature flags	UNSIGNED32	const	48245h if dynamic mapping is supported, and the node ID is set by hardware. 48205h if static mapping is supported and the node ID is set by hardware. 48645h if dynamic mapping is supported, and the node ID is set by software. 48605h if static mapping is supported and the node ID is set by software.
1F83h	NMT_EPLVersion_U8	00h	Ethernet POWERLINK version	UNSIGNED8	const	20h
1F8Ch	NMT_CurrNMTState_U8	00h	Current NMT State	UNSIGNED8	ro	-
1F8Dh	NMT_PresPayloadLimitList_AU16	00h	Number of entries	UNSIGNED8	rw, valid on reset	FEh (default) Valid range 01h - FEh
		01h - FEh	PresPayloadLimit	UNSIGNED16	rw, valid on reset	36 (default) Valid range: 0, 36 - 1490, FFFFh[d]
1F93h	NMT_EPLNodeID_REC	00h	Number of entries	UNSIGNED8	const	02h
		01h	NodeID_U8: NodeID	UNSIGNED8	ro	The configured node ID, see Network Configuration Object, Instace #1, POWERLINK Node ID on page 31.
		02h	NodeIByHW_BOOL: NodeID by hardware	BOOLEAN	ro	01h if the application configured the node ID in the Anybus SETUP state, otherwise 00h.

Index	Object Name	Sub-index	Description	Type	Access[a]	Value/Notes
1F98h	NMT_CycleTiming_REC	00h	Number of entries	UNSIGNED8	const	08h
		01h	IsochrTxMaxPayload_U16	UNSIGNED16	const	1490
		02h	IsochrRxMaxPayload_U16	UNSIGNED16	const	1490
		03h	PresMaxLatency_U32	UNSIGNED32	const	1000
		04h	PReqActPayloadLimit_U16	UNSIGNED16	rw, valid on reset	The mapped sized is measured in number of bytes. Valid range 36 - 1490 For dynamic mapping: 36 (default) For static mapping: The default value will match the size of the process data the module is configured to receive from the network.
		05h	PResActPayloadLimit_U16	UNSIGNED16	rw, valid on reset	The mapped sized is measured in number of bytes. Valid range 36 - 1490 For dynamic mapping: 36 (default) For static mapping: The default value will match the size of the process data the module is configured to send to the network.
		06h	AsndMaxLatency_U32	UNSIGNED32	const	1000
		07h	MultiplCycleCnt_U8	UNSIGNED8	rw, valid on reset	0 (default) Valid range 0 - 255
		08h	AsyncMTU_U16	UNSIGNED16	rw, valid on reset	300 default) Valid range 300 - 1500

Index	Object Name	Sub-index	Description	Type	Access[a]	Value/Notes
1F99h	NMT_CNBasicEthernetTime-out_U32	00h	After booting, this is the maximum time in microseconds the module silently listens for POWERLINK traffic on the network, before it decides if it should go to the Basic Ethernet state (no EPL traffic) or Pre-operational 1 state (with EPL traffic).	UNSIGNED32	rw, valid on reset	5000000 (default)
1F9Bh	NMT_MultiplCycleAssign_AU8	00h	Number of entries	UNSIGNED8	rw, valid on reset	FEh (default) Valid range 01h - FEh
		01h - FEh	Cycle number	UNSIGNED8	rw, valid on reset	0 (default) Valid range 0 - NMT_Cycletiming_REC.MultiplCycleCnt_U8 (the value of object 1F98h subindex 7).
1F9Eh	NMT_ResetCmd_U8	00h	Reset command	UNSIGNED8	rw	FFh

a. Access rights defined as described in “Access Rights” on page 21.

b. The type DOMAIN is used to transfer an arbitrarily large block of data between a client an a server. The contents of the data block is application specific.

c. See Dynamic and Static mapping in “Network Data Exchange” on page 11.

d. In object 1F8Dh, subindices 1..254, there are two special values: If the value of subindex *index* is 0, the node shall not listen to PRes frames from the node with node ID *index*. If the value is FFFFh, the PRes payload limit of the node with node ID *index* is equal to the value of 1F98/2h. Otherwise the value at sub-index *index* defines the PRes payload limit of the node with node ID *index*.

3.1.3 Access Rights

The view point is from the network into the device.

Access	Description
rw	Read and write, the value is not stored in flash memory.
wo	Write only, the value is not stored in flash memory.
ro	Read only.
const	Read only, constant value.
valid on reset	Any changes to this object will be valid after the reception of an NMTResetConfiguration command.

3.2 Manufacturer Specific Objects

3.2.1 General

Each object entry in the manufacturer specific range (2001h...FFFFh) corresponds to an instance (a.k.a. ADI) within the Application Data Object (FEh), i.e. network accesses to these objects results in object requests towards the host application. In case of an error, the status (or error) code returned in the response from the host application will be translated into the corresponding CANopen abort code.

Note: If ADI has > 1 elements: COP subindex = (ADI element - 1).
If ADI has 1 element: COP subindex = (ADI element (only 0 is valid)).

Important: As any access to these object entries will result in an object access towards the host application, the time spent communicating on the host interface must be taken into account when calculating the SDO timeout value.

3.2.2 Translation of Status Codes

Status (or error codes) are translated to POWERLINK abort codes as follows:

Anybus CompactCom Status Code	Anybus CompactCom Status Code No.	POWERLINK SDO Abort Code #[a]	POWERLINK Abort Code Description
Invalid message format	02h	0604 0047h	General internal incompatibility in the device.
Unsupported object	03h	0602 0000h	Object does not exist in the object dictionary.
Unsupported instance	04h	0602 0000h	Object does not exist in the object dictionary.
Unsupported command	05h	0604 0043h	General parameter incompatibility.
Invalid CmdExt[0]	06h	0602 0000h	Object does not exist in the object dictionary. (ADI access).
Invalid CmdExt[1]	07h	0609 0011h	Subindex does not exist. (ADI access).
Attribute not settable	08h	0601 0002h	Attempt to write a read only object.
Attribute not gettable	09h	0601 0001h	Attempt to read a write only object.
Too Much Data	0Ah	0607 0012h	Data type does not match, length of service parameter too high.
Not Enough Data	0Bh	0607 0013h	Data type does not match, length of service parameter too low.
Out of range	0Ch	0609 0030h	Value range of parameter exceeded (only for write access). **Note**: Use only when Anybus CompactCom status codes 11h (Value too high) or 12h (Value too low) cannot be used.
Invalid state	0Dh	0800 0022h	Data cannot be transferred or stored to the application because of the present device state.
Out of resources	0Eh	0504 0005h	Out of memory.
Segmentation failure	0Fh	0800 0000h	General error.
Segmentation buffer overflow	10h	0800 0000h	General error.
Value too high	11h	0609 0031h	Value of parameter written is too high.
Value too low	12h	0609 0032h	Value of parameter written is too low.
Attribute controlled from another channel	13h	0800 0021h	Data cannot be transferred or stored to the application because of local control.
Object Specific Error	FFh	0800 0000h	General error.

a. The default error code will be the 'General error' code (0800 0000h) if no corresponding error meets the error definition.

3.2.3 Network Data Format

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

Anybus Data Type	Native POWERLINK Data Type	Notes
BOOL	UNSIGNED8	An Anybus BOOL is always 8 bits, unlike a POWERLINK BOOLEAN that can be mapped as a single bit.
SINT8	INTEGER8	
SINT16	INTEGER16	
SINT32	INTEGER32	
UINT8	UNSIGNED8	
UINT16	UNSIGNED16	
UINT32	UNSIGNED32	
CHAR or array of CHAR	VISIBLE_STRING*n*	An array of n characters is encoded as a VISIBLE_STRING*n*, e.g. VISIBLE_STRING32 when n is 32. The "n" in VISIBLE_STRINGn specifies the maximum allowed length: the actual length may be shorter. It is not necessary to NULL terminate the string. In the XDD file, the type is always specified as VISIBLE_STRING, without a specified maximum length.
OCTET or array of OCTET	OCTET_STRING*n*	An array of n octets are encoded as a OCTET_STRINGn, e.g. OCTET_STRING6 when n is 6. A single OCTET is treated as an array with one OCTET. In the XDD file, the type is always specified as OCTET_STRING, without a specified maximum length.
ENUM	UNSIGNED8	
BIT1	BOOLEAN or BIT1[a]	It is recommended to map BIT1 as BOOLEAN
BITn (2 ≤ n ≤ 7)	BITn[a]	
BITS8	UNSIGNED8 or BIT8[a]	It is recommended to map BITS8 as UNSIGNED 8
BITS16	UNSIGNED16 or BIT16[a]	It is recommended to map BITS16 as UNSIGNED 16
BITS32	UNSIGNED32 or BIT32[a]	It is recommended to map BITS32 as UNSIGNED 32
SINT64	INTEGER64	
UINT64	UNSIGNED64	
FLOAT	REAL32	

a. This POWERLINK data type does not have a data type number to be specified in the device XDD file.

Important: Every ADI must be specified in the XDD file with their corresponding POWERLINK data type.

Note 1: Single element ADIs are represented as simple variables.

Note 2: ADIs with multiple elements of the same type are represented as arrays.

Note 3: ADIs with multiple elements of different types are represented as RECORD.

Note 4: ADIs of data type CHAR will be represented as VISIBLE_STRING. ADIs of data type OCTET will be represented as OCTET_STRING.

3.2.4 Object Entries

Requests for objects in the range 2001h to FFFFh will be forwarded to the host Application Data Object. The object indices are linearly mapped with the offset 2000h to Anybus ADIs.

Index	Object Name	Notes
2001h	ADI 0001h	Object index 2001h corresponds to ADI 0001h
2002h	ADI 0002h	Object index 2002h corresponds to ADI 0002h
...	...	...
FFFFh	ADI DFFFh	Object index FFFFh corresponds to ADI DFFFh (57343)

4. Anybus Module Objects

4.1 General Information

This chapter specifies the Anybus Module Object implementation and how they correspond to the functionality in the Anybus CompactCom 40 Ethernet POWERLINK.

Standard Objects:

- "Anybus Object (01h)" on page 26
- "Diagnostic Object (02h)" on page 27
- "Network Object (03h)" on page 28
- "Network Configuration Object (04h)" on page 30

Network Specific Objects:

- "File System Interface Object (0Ah)" on page 32

4.2 Anybus Object (01h)

Category

Basic

Object Description

This object assembles all common Anybus data, 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 40 Software Design Guide for further information.)

Instance Attributes (Instance #1)

Basic

#	Name	Access	Type	Value
1	Module type	Get	UINT16	0401h (Standard Anybus CompactCom 40)
2... 11	-	-	-	Consult the general Anybus CompactCom 40 Software Design Guide for further information.
12	LED colors	Get	struct of: UINT8 (LED1A) UINT8 (LED1B) UINT8 (LED2A) UINT8 (LED2B)	Value:Color: 01h Green 02h Red 01h Green 02h Red
13... 16	-	-	-	Consult the general Anybus CompactCom 40 Software Design Guide for further information.

Extended

#	Name	Access	Type	Value
17	Virtual attributes	Get/Set	-	Consult the general Anybus CompactCom 40 Software Design Guide for further information.
18	Black list/White list	Get/Set		
19	Network time	Get		

4.3 Diagnostic Object (02h)

Category

Basic

Object Description

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

The module supports one instance of this object, reserved for a major unrecoverable diagnostic event. For POWERLINK, a major unrecoverable diagnostic event always causes the module to enter the state EXCEPTION. Thus these events will not be visible on the POWERLINK network.

Supported Commands

Object:	Get_Attribute Create Delete
Instance:	Get_Attribute

Object Attributes (Instance #0)

#	Name	Access	Data Type	Value
1... 4	-	-	-	Consult the general Anybus CompactCom 40 Software Design Guide for further information.
11	Max no. of instances	Get	UINT16	1 (Major unrecoverable diagnostic event)
12	Supported Functionality	Get	BITS32	0 (Latching events not supported)

Instance Attributes (Instance #1)

Basic

#	Name	Access	Type	Value
1	Severity	Get	UINT8	Consult the general Anybus CompactCom 40 Software Design Guide for further information.
2	Event Code	Get	UINT8	
3... 7	-	-	-	Consult the general Anybus CompactCom 40 Software Design Guide for further information.

In this implementation, only unrecoverable diagnostic events are detected, and no information is forwarded to the network.

See also...

- "Diagnostics" on page 10

4.4 Network Object (03h)

Category

Basic

Object Description

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

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	UINT16	2
3	Number of instances	Get	UINT16	01h
4	Highest instance number	Get	UINT16	01h

Instance Attributes (Instance #1)

Basic

#	Name	Access	Type	Description
1	Network type	Get	UINT16	009Fh
2	Network type string	Get	Array of CHAR	‘POWERLINK’
3	Data format	Get	ENUM	00h (LSB first; POWERLINK is a Little Endian network)
4	Parameter data support	Get	BOOL	True
5	Write process data size	Get	UINT16	Current write process data size (in bytes) Updated on every successful Map_ADI_Write_Area, Remap_ADI_Write_Area and Map_ADI_Write_Ext_Area.[a]
6	Read process data size	Get	UINT16	Current read process data size (in bytes) Updated on every successful Map_ADI_Read_Area, Remap_ADI_Read_Area and Map_ADI_Read_Ext_Area.
7	Exception Information	Get	UINT8	00h

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

4.5 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 the Node ID being set to its default value.

See also...

- "Communication Settings" on page 10

Supported Commands

Object:	Get_Attribute Reset
Instance:	Get_Attribute Set_Attribute

Object Attributes (Instance #0)

#	Name	Access	Data Type	Value
1	Name	Get	Array of CHAR	'Network Configuration'
2	Revision	Get	UINT16	1
3	Number of instances	Get	UINT16	01h
4	Highest instance number	Get	UINT16	01h

Instance Attributes (Instance #1, POWERLINK Node ID)

Basic

#	Name	Access	Type	Description
1	Name[a]	Get	Array of CHAR	'Node ID'
2	Data type	Get	UINT8	04h (= UINT8)
3	Number of elements	Get	UINT8	01h (single element)
4	Descriptor	Get	UINT8	07h (read/write/shared access)
5	Value	Get/Set	UINT8	Valid range: 1 - 239 (Default = 0) As the default value is not in the valid range, the host application must specify a valid node ID before the module can communicate on the network. If no valid node ID is entered the module will go into the Anybus state EXCEPTION instead of into the Anybus state WAIT_PROCESS when leaving the Anybus state NW_INIT. A value written in the Anybus state SETUP will take effect immediately. A value written in a later Anybus state will take effect the next time the module is booted. Writing to this attribute will always update attribute 6, "Configured Value", immediately.
6	Configured Value	Get	UINT8	The configured value is equal to the last value written to attribute 5, "Value".

a. Multilingual, see "Multilingual Strings" on page 31.

Multilingual Strings

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

Instance	English	German	Spanish	Italian	French
1	Node ID	Knoten ID	ID Nodo	ID Node	ID Neoud

4.6 File System Interface Object (0Ah)

Category

Extended

Object Description

This object provides an interface to the built-in file system. Each instance represents a handle to a file stream and contains services for file system operations. Available directories are the "firmware" directory and the "XDD" directory.

A dynamically generated XDD file can be read via this object.

See the Anybus CompactCom 40 Software Design Guide for more information.

Supported Commands

See the Anybus CompactCom 40 Software Design Guide for information about object and instance commands.

Object Attributes (Instance #0)

#	Name	Access	Data Type	Value
1	Name	Get	Array of CHAR	'Anybus File System Interface'
2	Revision	Get	UINT8	01h
3	Number of instances	Get	UINT16	-
4	Highest instance no.	Get	UINT16	-
11	Max. no. of instances	Get	UINT16	0001h
12	Disable virtual file system	Get	BOOL	False
13	Total disc size	Get	Array of UINT32	-
14	Free space	Get	Array of UINT32	-
15	Disc CRC	Get	Array of UINT32	-

Instance Attributes

Extended

#	Name	Access	Type	Description
1	Instance type	Get	UINT8	Value: Type: 00h Reserved 01h File instance 02h Directory instance
2	File size	Get	UINT32	File size in bytes (zero for directories)
3	Path	Get	Array of CHAR	Path where the instance operates

5. Host Application Objects

5.1 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 Ethernet POWERLINK implementation.

Standard Objects:

- Application Object (see Anybus CompactCom 40 Software Design Guide)
- Application Data Object (see Anybus CompactCom 40 Software Design Guide)

Network Specific Objects:

- "Ethernet Host Object (F9h)" on page 36
- "Sync Object (EEh)" on page 34
- "POWERLINK Object (E9h)" on page 37

5.2 Sync Object (EEh)

Category

Extended

Object Description

This object contains the host application sync settings. The module will generate a sync pulse to the application if the following conditions are fulfilled: This object is implemented, sync mode is supported and the NMT state is NMT_CS_READY_TO_OPERATE, NMT_CS_PRE_OPERATIONAL_2, NMT_CS_OPERATIONAL or NMT_CS_STOPPED.

See also...

- "Synchronization" on page 10
- Anybus CompactCom 40 Software Design Guide, "SYNC"
- Anybus CompactCom 40 Software Design Guide, "Error Codes"

Supported Commands

Object:	Get_Attribute
Instance:	Get_Attribute Set_Attribute

Object Attributes (Instance #0)

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

Instance Attributes (Instance #1)

#	Name	Access	Type	Value If Not Implemented	Comment
1	Cycle time	Get/Set	UINT32	-	This attribute contains the cycle time on the Ethernet POWERLINK network. It corresponds to the value of the Object Dictionary Object #1006h (NMT Cycle Length Object), see page 14.
2 - 6	-	-	-	-	Not used
7	Sync mode	Get/Set	UINT16	-	Set to 1 (synchronous operation) when the POWERLINK network cycle is isochronous and bit 1 in attribute #8 is set; otherwise, set to 0 (nonsynchronous operation).
8	Supported sync modes	Get	UINT16	-	Bit 0: Nonsynchronous operation (default value if nonsynchronous operation is supported) Bit 1: Synchronous operation is supported Bit 2 - 15: Reserved. Set to zero.

5.3 Ethernet Host Object (F9h)

Category

Basic, extended

Object Description

This object implements Ethernet features in the host application. If the host application specifies a MAC address (instance #1, attribute #1) the module will use this address as the device MAC address. If no MAC address is specified by the application, the MAC address assigned at production will be used.

Supported Commands

Object:	Get_Attribute
Instance:	Get_Attribute Set_Attribute

Object Attributes (Instance #0)

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

Instance Attribute (Instance #1)

Basic

#	Name	Access	Type	Default[a]	Comment
1	MAC address[b]	Get	Array of UINT8	-	6 byte physical address value; overrides the pre-programmed Mac address. Note that the new Mac address value must be obtained from the IEEE.

a. If an attribute is not implemented, the module will use this value instead

b. The module is pre-programmed with a valid Mac address. To use that address, do *not* implement this attribute.

5.4 POWERLINK Object (E9h)

Category

Extended

Object Description

This object implements POWERLINK features in the host application. The application can support none, some, or all attributes specified.

Supported Commands

Object: Get_Attribute

Instance: Get_Attribute
Set_Attribute

Object Attributes (Instance #0)

#	Name	Access	Data Type	Value
1	Name	Get	Array of CHAR	'POWERLINK'
2	Revision	Get	UINT8	01h
3	Number of instances	Get	UINT16	0001h
4	Highest instance no.	Get	UINT16	0001h

Instance Attributes (Instance #1)

#	Name	Access	Type	Value If Not Implemented	Comment
1	Vendor ID	Get	UINT32	0000001Bh	These values replace the default values for CANopen identity object (1018h, see page 14) **Note:** Changing any of these attributes requires a new Vendor ID.
2	Product Code	Get	UINT32	00000028h	
3	Revision high word	Get	UINT16	0000h	
4	Revision low word	Get	UINT16	(hardware revision)	
5	Serial Number	Get	UINT32	(set at production)	
6	Manufacturer device name	Get	Array of CHAR (max 64 bytes)	"Anybus CompactCom 40 Ethernet POWERLINK"	Corresponds to Manufacturer Device Name object (1008h, see page 14).
7	Manufacturer hardware version	Get	Array of CHAR (max 64 bytes)	Hardware version	Corresponds to Manufacturer hardware version object (1009h, see page 14).
8	Manufacturer software version	Get	Array of CHAR (max 64 bytes)	Anybus CompactCom Software Version	Corresponds to Manufacturer software version object (100Ah, see page 14).
10	Device type	Get	UINT32	00000000h	Corresponds to Device type object (1000h, see page 14).
14	Manufacturer name	Get	Array of CHAR (max 64 bytes)	"HMS"	Will be used as part of the Interface Description string in the Interface Group object (1030h, see page 15).

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) in the status register indicates that the network participation is supervised by another network device.

On POWERLINK, a slave (CN) has no real knowledge if it is supervised by another network device. It is considered supervised if it enters the NMT_CS_OPERATIONAL state where exchange of valid PDO data takes place. If the Managing Node (MN) of the network is lost, the device will leave the operational state and the supervised bit will be cleared.

Supervised Bit State	Description
0	The module is not in NMT_CS_OPERATIONAL
1	The module is in NMT_CS_OPERATIONAL

B.2 Anybus State Machine

The table below describes how the Anybus state machine relates to the Ethernet POWERLINK network.

Anybus State	Implementation	Comment
WAIT_PROCESS	The module stays in this state until the setup is finalized and the controlled node and the managing node together initiates a transition to POWERLINK state NMT_CS_READY_TO_OPERATE, which equals Anybus state IDLE. If the managing node orders the controlled node to stop, the POWERLINK state will switch to NMT_CS_STOPPED and the Anybus state will switch to WAIT_PROCESS.	In case synchronous mode is supported, sync signals will start being produced in this state.
ERROR	There is no designated ERROR state on POWERLINK.	Errors result in a transition to POWERLINK state NMT_CS_PRE_OPERATIONAL_1. Any transition from this state will cause the Anybus state to switch to WAIT_PROCESS.
PROCESS_ACTIVE	Corresponds to POWERLINK state NMT_CS_-OPERATIONAL	Received PDO data is valid only in this state.
IDLE	POWERLINK state NMT_CS_READY_TO_OPERATE and NMT_CS_OPERATIONAL before receiving the first valid PDO data. Transition to PROCESS_ACTIVE can only be initiated by MN.	-
EXCEPTION	POWERLINK state NMT_GS_OFF. No network traffic is sent or received, though the hub continues to operate. Unexpected errors, e.g. watchdog timeouts, causes a transition to this state.	ERROR and STATUS LEDs turn red (to indicate a major fault). The module has to be reset to leave this state.

C. Timing & Performance

C.1 General Information

This chapter specifies timing and performance parameters that are verified and documented for the Anybus CompactCom 40 Ethernet POWERLINK.

The following timing aspects are measured:

Category	Parameters	Page
Event Based Process Data Delay	T101, T102	40

For general timing information, see the Anybus CompactCom 40 Software Design Guide.

C.2 Event Based Process Data Delay

"Read process data delay" is defined as the time from when the last bit of the network frame enters the module, to when the RDPDI interrupt is asserted to the application.

"Write process data delay" is defined in two different ways, depending on network type.

- For software stack based cyclic/polled networks, it is defined as the time from when the module exchanges write process data buffers, to when the first bit of the new process data frame is sent out on the network.
- For COS (Change Of State) networks, it 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.

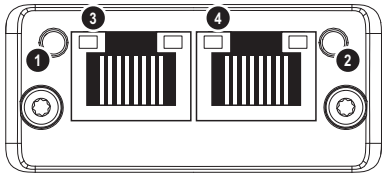
Parameter	Description	Measured Time	Comment	Unit
T101	Read process data delay	617 ns	Time from last input bit received until nIRQ goes active on the chip	ns
T102	Write process data delay	2 µs	Time from when the CompactCom module exchanges write process data buffers, to when the first bit of the new process data frame is sent out on the network interface towards the PHY.	µs

D. Technical Specification

D.1 Front View

Ethernet Connector

#	Item
1	STATUS LED
2	ERROR LED
3	Link/Activity LED (port 1)
4	Link/Activity LED (port 2)



STATUS LED (Module Status)

LED State	Description
Off	Module is off, initializing, or not active.
Green, fast flashing[a]	NMT_CS_BASIC_ETHERNET Basic Ethernet state: no POWERLINK traffic has been detected.
Green, single flash	NMT_CS_PRE_OPERATIONAL_1. Only asynchronous data.
Green, double flash	NMT_CS_PRE_OPERATIONAL_2. Asynchronous and synchronous data. No PDO data.[b]
Green, triple flash	NMT_CS_READY_TO_OPERATE. Ready to operate. Asynchronous and synchronous data. No PDO data.[b]
Green	NMT_CS_OPERATIONAL. Fully operational. Asynchronous and synchronous data. PDO data is sent and received.
Green, slow flashing[c]	NMT_CS_STOPPED Module stopped (for controlled shutdown, for example). Asynchronous and synchronous data. No PDO data.[b]
Red	If the ERROR LED also is red, a fatal event was encountered.

a. On 50 ms, off 50 ms.
b. Any process data sent is declared not valid and received process data must be ignored in this state.
c. On 200 ms, off 200 ms.

ERROR LED (Network Status)

LED State	Description
Off	No error
Red	If the STATUS LED is not red, a non-fatal error has been detected. If the STATUS LED is red, a fatal event was encountered.

Link/Activity LED 3 & 4

LED State	Description
Off	No link.
Green	Link, no traffic.
Green, flashing	Link and traffic.

Ethernet Interface

The Ethernet interface supports 100 Mbit/s, half duplex operation.

D.2 Protective Earth (PE) Requirements

In order to ensure proper EMC behavior, the module must be properly connected to protective earth via the PE pad / PE mechanism described in the general Anybus CompactCom 40 Hardware Design Guide.

HMS Industrial Networks does not guarantee proper EMC behavior unless these PE requirements are fulfilled.

D.3 Power Supply

Supply Voltage

The module requires a regulated 3.3V power source as specified in the general Anybus CompactCom 40 Hardware Design Guide.

Power Consumption

The Anybus CompactCom 40 Ethernet POWERLINK is designed to fulfil the requirements of a Class B module. For more information about the power consumption classification used on the Anybus CompactCom 40 platform, consult the general Anybus CompactCom 40 Hardware Design Guide.

D.4 Environmental Specification

Consult the Anybus CompactCom 40 Hardware Design Guide for further information.

D.5 EMC Compliance

Consult the Anybus CompactCom 40 Hardware Design Guide for further information.