


Fieldbus Appendix Anybus-S EtherCAT

Rev 1.00

HMS Industrial Networks AB


Germany +49 - 721 - 96472 - 0
Japan +81 - 45 - 478 - 5340
Sweden +46 - 35 - 17 29 20
U.S.A +1 - 773 - 404 - 3486


ge-sales@hms-networks.com
jp-sales@hms-networks.com
sales@hms-networks.com
us-sales@hms-networks.com



Table of Contents

Preface	About This Document
	How To Use This Document..... P-1
	Important User Information..... P-1
	Related Documents..... P-2
	Document History P-2
	Conventions & Terminology P-3
	Support..... P-3
 Chapter 1	 About the Anybus-S EtherCAT
	General.....1-1
	Features.....1-1
	Fieldbus Conformance Notes1-1
	Overview.....1-2
	Network Status Indicators1-2
 Chapter 2	 Basic Operation
	General Information.....2-1
	Identity Customization.....2-1
	EtherCAT Implementation Details.....2-2
	<i>General Information</i>2-2
	<i>Sync Managers</i>2-2
	<i>FMMUs</i>2-2
	<i>Addressing Modes</i>2-2
	<i>Watchdog Functionality</i>2-3
	<i>Implemented Services</i>2-3
	CANopen Implementation Details.....2-4
	<i>General Information</i>2-4
	<i>Implemented Services</i>2-4
	Data Exchange.....2-5
	<i>Parameter Data</i>2-5
	<i>I/O Data</i>2-5
 Chapter 3	 Object Dictionary Implementation
	Standard Objects3-1
	<i>General</i>3-1
	<i>Object Entries</i>3-1
	Manufacturer Specific Objects3-2
	<i>General</i>3-2
	<i>Input Buffer</i>3-2
	<i>Output Buffer</i>3-3

Chapter 4 Mailbox Interface

Overview	4-1
Fault Information	4-1
Set Product Code (SET_PRODUCT_CODE)	4-2
Set Product Info (SET_PRODUCT_INFO)	4-3
Set Product Info All (SET_PROD_INFO_ALL)	4-5
Set Hardware Revision (SET_HARDWARE_REV)	4-7
Emergency Message Indication (EMCY_MESSAGE)	4-8

Chapter 5 Fieldbus Specific Area**Appendix A Miscellaneous**

Control Register Area	A-1
-----------------------------	-----

Appendix B Slave Information Interface Area Defaults

General	B-1
Structure Category String	B-2
Structure Category General	B-2
Structure Category FMMU	B-2
Structure Category SyncM	B-3
Structure Category TxPDO and RxPDO for Each PDO	B-3

Appendix C Mechanical Specification**Appendix D Connectors**

Application Connector	D-1
Ethernet Connector (RJ45)	D-1

Appendix E Technical Specification (Preliminary)

Electrical Specification	E-1
<i>Protective Earth (PE) Requirements</i>	E-1
<i>Power Supply</i>	E-1
Environmental Specification	E-1
EMC Compliance (CE)	E-1

About This Document

How To Use This Document

This document is intended to be used as a supplement to the Anybus-S Parallel Design Guide. The reader of this document is expected to have basic knowledge in the EtherCAT fieldbus system, and communication systems in general. Please consult the general Anybus-S Parallel Design Guide for further information about the Anybus-S platform.

IMPORTANT: This is a preliminary document. Information may be missing or incorrect.

Important User Information

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 application meets all performance and safety requirements including any applicable laws, regulations, codes, and standards.

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

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 or liability for actual use based on these examples and illustrations.

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.

Related Documents

Document name	Author
Anybus-S Parallel Design Guide	HMS
RFC 894	RFC
EtherCAT Communication Specification	EtherCAT Technology Group
EtherCAT Indicator Specification	
CiA Draft Standard 301 (Application Layer and Communication Profile)	CiA
-	-

Document History

Summary of Recent Changes (vx.xx... v1.00)

[illegible]

Revision List

[illegible]

Conventions & Terminology

The following conventions are used throughout this document:

- Numbered lists provide sequential steps
- Bulleted lists provide information, not procedural steps
- The term ‘module’ refers to the Anybus module
- The term ‘application’ refers to the device connected to the Anybus application connector
- Hexadecimal values are written in the format NNNNh, where NNNN is the hexadecimal value.

Support

HMS Sweden (Head Office)

E-mail:	support@hms-networks.com
Phone:	+46 (0) 35 - 17 29 20
Fax:	+46 (0) 35 - 17 29 09
Online:	www.anybus.com

HMS America

E-mail:	us-support@hms-networks.com
Phone:	+1-773-404-2271
Toll Free:	888-8-Anybus
Fax:	+1-773-404-1797
Online:	www.anybus.com

HMS Germany

E-mail:	ge-support@hms-networks.com
Phone:	+49-721-96472-0
Fax:	+49-721-964-7210
Online:	www.anybus.com

HMS Japan

E-mail:	jp-support@hms-networks.com
Phone:	+81-45-478-5340
Fax:	+81-45-476-0315
Online:	www.anybus.com

About the Anybus-S EtherCAT

General

The Anybus-S EtherCAT communication module provides instant EtherCAT connectivity via the patented Anybus-S application interface. Any device that supports this standard can take advantage of the features offered by the module, providing seamless network integration regardless of network type.

This product conforms to all aspects of the application interface defined in the Anybus-S Parallel Design Guide, making it fully interchangeable with any other device following that specification.

Features

- CANopen over EtherCAT
- DS301 v4.02 compliant
- Galvanically isolated network electronics
- Customizable Identity Information
- EMCY support
- Supports segmented SDO access
- Up to 512 bytes of cyclic data in each direction
- Up to 2048 bytes of acyclic data in each direction
- Generic EDS-file provided

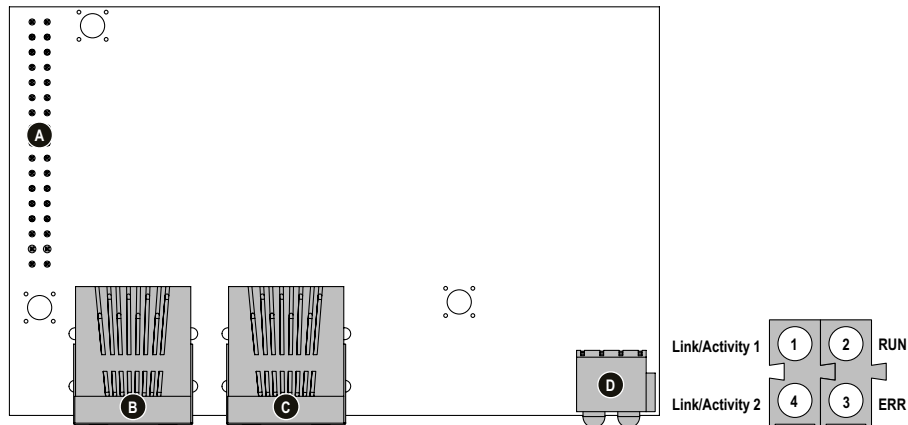
Fieldbus Conformance Notes

Important Note: The following information is preliminary/pending:

This product is pre-certified for network compliance. While this is done to ensure that the final product *can* be certified, it does not necessarily mean that the final product doesn't need recertification.

Contact HMS for further information.

Overview



#	Description	Notes
A	Application Connector	Consult the general Anybus-S Parallel Design Guide for further information
B	Ethernet Port 1 (In)	See D-1 "Ethernet Connector (RJ45)"
C	Ethernet Port 2 (Out)	
D	Network Status Indicators	(see below)

Network Status Indicators

Led	State	Description
Link/Activity 1	Off	No link sensed on Ethernet port 1
	Green	Link sensed on Ethernet port 1
	Green, flickering	Exchanging packets on Ethernet port 1
RUN	Off	The device is in INIT state
	Green, blinking	The device is in PRE-OPERATIONAL state
	Green, single flash	The device is in SAFE-OPERATIONAL state
	Green	The device is in OPERATIONAL state
ERR	Off	EtherCAT communication is in working condition
	Red, blinking	General configuration error
	Red, single flash	Slave device application has changed the EtherCAT state autonomously: Parameter 'Change' in the AL status register is set to 01 (change/error).
	Red, double flash	Application watchdog timeout
	Red	Syncmanager watchdog timeout
Link/Activity 2	Off	No link sensed on Ethernet port 2
	Green	Link sensed on Ethernet port 2
	Green, flickering	Exchanging packets on Ethernet port 2

Basic Operation

General Information

Software Requirements

Generally, no additional network support code needs to be written in order to support the Anybus-S EtherCAT; however in order to take advantage of advanced network specific functionality, a certain degree of dedicated software support may be necessary.

For further information about the Anybus-S software interface, consult the general Anybus-S Parallel Design Guide.

Electronic Data Sheet (EDS)

Each device on EtherCAT is associated with an Electronic Data Sheet (a.k.a 'EDS'-file), which holds a description of the device and its functions. Most importantly, the file describes the object dictionary implementation in the module.

HMS supplies a generic EDS-file which can be used as a basis for new implementations. Note however that this file must be adjusted to fit the end product.

Note: If using a modified EDS-file, it is required to customize the Vendor ID and Product Code.

See also...

- 1-1 "Fieldbus Conformance Notes"

Identity Customization

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

Information	Default Value
Vendor ID	0000 001Bh (HMS Networks)
Product Code	0000 0032h (Anybus-S EtherCAT)
Revision Number	(assigned at production)
Manufacturer Device Name	'Anybus-S'
Manufacturer Hardware Revision	-

It is possible to customize this information so that the module appears as a vendor specific implementation rather than a generic Anybus product. Note that this invalidates the generic EDS-file supplied by HMS.

See also...

- 1-1 "Fieldbus Conformance Notes"
- 2-1 "Electronic Data Sheet (EDS)"
- 4-2 "Set Product Code (SET_PRODUCT_CODE)"
- 4-3 "Set Product Info (SET_PRODUCT_INFO)"
- 4-5 "Set Product Info All (SET_PROD_INFO_ALL)"
- 4-7 "Set Hardware Revision (SET_HARDWARE_REV)"

EtherCAT Implementation Details

General Information

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

- Application Layer: CANopen
- FMMU Channels: 4
- SM Channels: 4
- RAM Size: 4kByte
- Bit-oriented FMMU operation

See also...

- 2-4 “CANopen Implementation Details”

Sync Managers

The module features four Sync Managers:

- **Sync Manager 0**
Used for mailbox write transfers (Master to Slave).
The module supports mailbox sizes of 50... 256 bytes (default = 192).
- **Sync Manager 1**
Used for mailbox read transfers (Slave to Master).
The module supports mailbox sizes of 50... 256 bytes (default = 192).
- **Sync Manager 2**
Contains the RxPDOs specified by the PDO assignment.
In practice, Sync Manager 2 holds the unmodified Output I/O Data.
- **Sync Manager 3**
Contains the TxPDOs specified by the PDO assignment.
In practice, Sync Manager 3 holds the unmodified Input I/O Data.

Note: In this context, the term ‘mailbox’ must not be confused with Anybus mailbox messaging.

FMMUs

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

Addressing Modes

There are a number of different addressing modes which can be applied when communicating with EtherCAT slaves. As a full EtherCAT slave device, the module supports position addressing, node addressing and logical addressing.

Watchdog Functionality

Apart from the standard watchdog functionality, the following additional watchdogs are implemented:

- **PDI Watchdog**

This watchdog monitors the CPU in the module. Each access from the CPU to the ESC resets this watchdog. To a degree this watchdog also monitors the application CPU, since each update of the Input I/O Data from the application triggers a new access to the ESC.

Note: This watchdog is configured and enabled by the EtherCAT master.

- **Output I/O Sync Manager Watchdog**

If the Output I/O Data is not updated by the master within the configured time, the watchdog will timeout and cause a shift from 'OPERATIONAL' to 'SAFE-OPERATIONAL'. The resolution of this watchdog is 1ms.

Note: This watchdog is configured by the EtherCAT master, and is disabled by default.

See also...

- A-1 "Watchdog Counter Input (7D2h... 7D3h)"

Implemented Services

The module implements the following EtherCAT services:

Service	Supported	Description
Auto increment physical read (APRD)	Yes	-
Auto increment physical write (APWR)	Yes	-
Auto increment Read Write (APRW)	Yes	-
Configured address read (FPRD)	Yes	-
Configured address write (FPWR)	Yes	-
Configured address Read Write (FPRW)	Yes	-
Broadcast Read (BRD)	Yes	-
Broadcast Write (BWR)	Yes	-
Logical Read (LRD)	Yes	-
Logical Write (LWR)	Yes	-
Logical Read Write (LRW)	Yes	-
Auto increment physical read multiple write (ARMW)	Yes	-
Configured read multiple write (FRMW)	Yes	-

CANopen Implementation Details

General Information

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

The Input and Output Data is mapped to dedicated objects in the manufacturer-specific range. I/O Data is exchanged by means of Process Data Objects (PDOs), while Parameter Data is exchanged using Service Data Objects (SDOs).

See also...

- 3-1 “Object Dictionary Implementation”

Implemented Services

The module implements the following CANopen services:

Service	Supported	Description
SDO Download Expedited	Yes	Writes up to four octets to the slave
SDO Download Normal	Yes	Writes up to a negotiated number of octets to the slave
Download SDO Segment	Yes	Writes additional data if the object size exceeds the negotiated no. of octets.
SDO Upload Expedited	Yes	Reads up to four octets from the slave
SDO Upload Normal	Yes	Reads up to a negotiated number of octets from the slave
Upload SDO Segment	Yes	Reads additional data if the object size exceeds the negotiated no. of octets
Abort SDO Transfer	Yes	Server abort of service in case of an erroneous condition
Get OD List	Yes	Reads a list of available indices
Get Object Description	Yes	Reads details of an index
Get Entry Description	Yes	Reads details of a sub-index
Emergency	Yes	Reports unexpected conditions. (see 4-8 “Emergency Message Indication (EMCY_MESSAGE)”)

Data Exchange

Parameter Data

The Anybus Input- and Output Data buffers are represented as object entries in the manufacturer specific range (2000h...5FFFh).

See also...

- 3-2 “Manufacturer Specific Objects”

I/O Data

General Information

I/O Data is exchanged as Process Data Objects (PDO). The module will only map as many PDOs as required to hold the specified amount of I/O Data.

Example:

When using 200 bytes of Input I/O Data, TxPDO 1A00h will hold the first 128 bytes, and TxPDO 1A01 will hold the remaining 72 bytes. TxPDOs 1A02h and 1A03h will not be mapped.

RxPDO Mapping

RxPDO no.	Mapped to Object...	Output Data Range
1A00h	2000h, sub-index 1... 128	0... 127
1A01h	2001h, sub-index 1... 128	128...255
1A02h	2002h, sub-index 1... 128	256... 383
1A03h	2003h, sub-index 1... 128	384... 511

Note: The RxPDO mapping is static and cannot be changed.

See also...

- 3-3 “Output Buffer”

TxPDO Mapping

TxPDO no.	Mapped to Object...	Input Data Range
1600h	2100h, sub-index 1... 128	0... 127
1601h	2101h, sub-index 1... 128	128...255
1602h	2102h, sub-index 1... 128	256... 383
1603h	2103h, sub-index 1... 128	384... 511

Note: The TxPDO mapping is static and cannot be changed.

See also...

- 3-2 “Input Buffer”

Object Dictionary Implementation

Standard Objects

General

The standard object dictionary is implemented in accordance with the DS301 specification (v4.02).

Object Entries

Index	Object Name	Sub-Index	Description	Type	Access	Notes
1000h	Device Type	00h	Device Type	U32	RO	0000 0000h (No profile)
1008h	Manufacturer device name	00h	Manufacturer device name	Visible string	RO	See 2-1 "Identity Customization"
1009h ^a	Manufacturer hardware version	00h	Manufacturer hardware version	Visible string	RO	
1018h	Identity object	00h	Number of entries	U8	RO	04h
		01h	Vendor ID	U32	RO	See 2-1 "Identity Customization"
		02h	Product Code	U32	RO	
		03h	Revision Number	U32	RO	
		04h	Serial Number	U32	RO	
1600h ... 1603h	Receive PDO mapping	00h	Number of entries	U8	R	See 2-5 "RxPDO Mapping"
		01h... N	Mapped object #N	U32	R	
1A00h ... 1A03h	Transmit PDO mapping	00h	Number of entries	U8	R	See 2-5 "TxPDO Mapping"
		01h... N	Mapped object #N	U32	R	
1C00h	Sync Manager Communication Type	00h	Number of entries	U8	R	04h
		01h	Mailbox wr	U8	R	01h
		02h	Mailbox rd	U8	R	02h
		03h	Process Data out	U8	R	03h
		04h	Process Data in	U8	R	04h
1C12h	Sync Manager Rx PDO Assign	00h	Number of entries ^b	U8	R	No. of assigned RxPDOs (0... 4)
		01h	Assigned RxPDO	U8	R	Assigned to RxPDO 1600h
		02h	Assigned RxPDO	U8	R	Assigned to RxPDO 1601h
		03h	Assigned RxPDO	U8	R	Assigned to RxPDO 1602h
		04h	Assigned RxPDO	U8	R	Assigned to RxPDO 1603h
1C13h	Sync Manager Tx PDO Assign	00h	Number of entries ^b	U8	R	No. of assigned TxPDOs (0... 4)
		01h	Assigned TxPDO	U8	R	Assigned to TxPDO 1A00h
		02h	Assigned TxPDO	U8	R	Assigned to TxPDO 1A01h
		03h	Assigned TxPDO	U8	R	Assigned to TxPDO 1A02h
		04h	Assigned TxPDO	U8	R	Assigned to TxPDO 1A03h

a. This object is not enabled by default.

For more information, see 4-7 "Set Hardware Revision (SET_HARDWARE_REV)".

b. The number of entries equals the number of mapped PDOs, see 2-5 "I/O Data".

Manufacturer Specific Objects

General

The Anybus Input- and Output Data buffers can be accessed as object entries in the manufacturer specific range (2001h...5FFFh).

Input Buffer

Index	Object Name	Sub-Index	Description	Type	Access	Notes
2000h	Inputs	00h	No. of entries	U8	RO	-
		01h	Input byte 0000	U8	RO	-
		02h	Input byte 0001			
				
		80h	Input byte 0127			
2001h	Inputs	00h	No. of entries	U8	RO	-
		01h	Input byte 0128	U8	RO	-
		02h	Input byte 0129			
				
		80h	Input byte 0255			
2002h	Inputs	00h	No. of entries	U8	RO	-
		01h	Input byte 0256	U8	RO	-
		02h	Input byte 0257			
				
		80h	Input byte 0511			
...
200Eh	Inputs	00h	No. of entries	U8	RO	-
		01h	Input byte 1792	U8	RO	-
		02h	Input byte 1793			
				
		80h	Input byte 1919			
200Fh	Inputs	00h	No. of entries	U8	RO	-
		01h	Input byte 1920	U8	RO	-
		02h	Input byte 1921			
				
		80h	Input byte 2047			

Note: The module will only create as many object as required to hold the specified amount of data.

Example:

When using 200 bytes of input data, object 2000h will hold the first 128 bytes, and object 2001h will hold the remaining 72 bytes. Objects 2002h... 200Fh will not be available.

See also...

- 2-5 “Data Exchange”

Output Buffer

Index	Object Name	Sub-Index	Description	Type	Access	Notes
2100h	Outputs	00h	No. of entries	U8	RO	-
		01h	Output byte 0000	U8	R(W)	-
		02h	Output byte 0001			
				
		80h	Output byte 0127			
2101h	Outputs	00h	No. of entries	U8	RO	-
		01h	Output byte 0128	U8	R(W)	-
		02h	Output byte 0129			
				
		80h	Output byte 0255			
2102h	Outputs	00h	No. of entries	U8	RO	-
		01h	Output byte 0256	U8	R(W)	-
		02h	Output byte 0257			
				
		80h	Output byte 0511			
2103h	Outputs	00h	No. of entries	U8	RO	-
		01h	Output byte 0512	U8	R(W)	-
		02h	Output byte 0513			
				
		80h	Output byte 0639			
...
210Eh	Outputs	00h	No. of entries	U8	RO	-
		01h	Output byte 1792	U8	R(W)	-
		02h	Output byte 1793			
				
		80h	Output byte 1919			
210Fh	Outputs	00h	No. of entries	U8	RO	-
		01h	Output byte 1920	U8	R(W)	-
		02h	Output byte 1921			
				
		80h	Output byte 2047			

Note 1: For consistency reasons, data declared as I/O data will be read-only, while Parameter Data can be accessed freely.

Note 2: The module will only create as many objects as required to hold the specified amount of data.

Example:

When using 200 bytes of output data, object 2100h will hold the first 128 bytes, and object 2101h will hold the remaining 72 bytes. Objects 2102h... 210Fh will not be available.

See also...

- 2-5 “Data Exchange”

Mailbox Interface

This chapter describes the fieldbus specific mailbox commands in the module. Consult the Anybus-S Design Guide for more information regarding the mailbox functionality.

Overview

The following fieldbus-specific mailbox commands are implemented in the module:

Command	Description	Page
Set Product Code (SET_PRODUCT_CODE)	These commands are used to customize the identity information of the module.	4-2
Set Product Info (SET_PRODUCT_INFO)		4-3
Set Product Info All (SET_PROD_INFO_ALL)		4-5
Set Hardware Revision (SET_HARDWARE_REV)		4-7
Emergency Message Indication (EMCY_MESSAGE)	This command issues an EMCY message on the bus.	4-8

Fault Information

When a mailbox command cannot be processed, the Message Information register in the header of the response will indicate that an error occurred (consult the Anybus-S Design Guide for more information). If the error code is 'Invalid Other' (Fh), extended error information is available in the Fault Information register (Extended word 7 or 8, depending on the command).

Fault Value	Meaning
0001h	FBU_ERR_ESC_NOT_INITIALISED
0002h	FBU_ERR_OUT_OF_MEMORY
0003h	FBU_ERR_WRONG_BUS_STATE

Set Product Code (SET_PRODUCT_CODE)

This command is used to customize the Product Code in the Identity Object (object entry 1018h). This enables configuration tools etc. to identify the module as a custom implementation rather than a generic Anybus-S module. Please note that the EDS-file must be updated accordingly.

Note: This command may only be issued during initialisation.

Command initiator	Application
Command number	0002h
Extended Header data	Fault information
Message data	Product code
Response message	-
Firmware Revision	All

Command and response layout:

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	
Command	0002h	0002h	SET_PRODUCT_CODE
Data size	0004h	0004h	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	-	
		Fault Information	
Message data byte 1	Product Code (high byte)	Product Code (high byte)	
Message data byte 2	Product Code (mid high byte)	Product Code (mid high byte)	
Message data byte 3	Product Code (mid low byte)	Product Code (mid low byte)	
Message data byte 4	Product Code (low byte)	Product Code (low byte)	

- **Fault Information**
(see 4-1 “Fault Information”)
- **Product Code**
4-byte Product Code.

Set Product Info (SET_PRODUCT_INFO)

This command customizes the following information:

- Vendor ID (Object entry 1018h, sub-index 01h)
- Product Code (Object entry 1018h, sub-index 02h)
- Manufacturer Device Name (Object entry 1008h)

This enables configuration tools etc. to identify the module as a custom implementation rather than a generic Anybus-S module. Please note that the EDS-file must be updated accordingly.

Note: This command may only be issued during initialisation.

Command initiator	Application
Command number	0003h
Extended Header data	Fault information
Message data	Misc. customization info
Response message	-
Firmware Revision	All

Command and response layout:

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	
Command	0003h	0003h	<i>SET_PRODUCT_INFO</i>
Data size	(size)	(size)	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault Information	
Message data byte 1	Vendor ID (high byte)	Vendor ID (high byte)	
Message data byte 2	Vendor ID (mid high byte)	Vendor ID (mid high byte)	
Message data byte 1	Vendor ID (mid low byte)	Vendor ID (mid low byte)	
Message data byte 2	Vendor ID (low byte)	Vendor ID (low byte)	
Message data byte 3	Product Code (high byte)	Product Code (high byte)	
Message data byte 4	Product Code (mid high byte)	Product Code (mid high byte)	
Message data byte 3	Product Code (mid low byte)	Product Code (mid low byte)	
Message data byte 4	Product Code (low byte)	Product Code (low byte)	
Message data byte 5	Device Name Length	Device Name Length	<i>String length (32 ch. max)</i>
Message data byte 6	Device Name (1st char)	Device Name (1st char)	<i>Manufacturer Device Name</i>
Message data byte 7	Device Name (2nd char)	Device Name (2nd char)	
Message data byte 8	Device Name (3rd char)	Device Name (3rd char)	
...	
Message data byte n	Device Name (last char)	Device Name (last char)	...

- **Fault Information**
(see 4-1 “Fault Information”)
- **Vendor ID**
4-byte Vendor ID.
- **Product Code**
4-byte Product Code.
- **Device Name Length**
Length of the Manufacturer Device Name in bytes.
- **Device Name**
Manufacturer Device Name as ASCII string.

Set Product Info All (SET_PROD_INFO_ALL)

This command is similar to SET_PRODUCT_INFO, except that it also specifies the Revision Number of the product. Please note that the EDS-file must be updated accordingly.

Note: This command may only be issued during initialisation.

Command initiator	Application
Command number	0004h
Extended Header data	Fault information
Message data	Misc. customization info
Response message	-
Firmware Revision	All

Command and response layout:

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	
Command	0004h	0004h	SET_PROD_INFO_ALL
Data size	(size)	(size)	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	-	
		Fault Information	
Message data byte 1	Vendor ID (high byte)	Vendor ID (high byte)	
Message data byte 2	Vendor ID (mid high byte)	Vendor ID (mid high byte)	
Message data byte 3	Vendor ID (mid low byte)	Vendor ID (mid low byte)	
Message data byte 4	Vendor ID (low byte)	Vendor ID (low byte)	
Message data byte 5	Product Code (high byte)	Product Code (high byte)	
Message data byte 6	Product Code (mid high byte)	Product Code (mid high byte)	
Message data byte 7	Product Code (mid low byte)	Product Code (mid low byte)	
Message data byte 8	Product Code (low byte)	Product Code (low byte)	
Message data byte 9	Revision No.(high byte)	Revision No. (high byte)	
Message data byte 10	Revision No. (mid high byte)	Revision No. (mid high byte)	
Message data byte 11	Revision No. (mid low byte)	Revision No. (mid low byte)	
Message data byte 12	Revision No. (low byte)	Revision No. (low byte)	
Message data byte 13	Device Name Length	Device Name Length	String length (32 ch. max)
Message data byte 14	Device Name (1st char)	Device Name (1st char)	Manufacturer Device Name
Message data byte 15	Device Name (2nd char)	Device Name (2nd char)	...
Message data byte 16	Device Name (3rd char)	Device Name (3rd char)	...
...
Message data byte n	Device Name (last char)	Device Name (last char)	...

- **Fault Information**
(see 4-1 “Fault Information”)
- **Vendor ID**
4-byte Vendor ID.
- **Product Code**
4-byte Product Code.
- **Revision No.**
4-byte Revision Number.
- **Device Name Length**
Length of the Manufacturer Device Name in bytes.
- **Device Name**
Manufacturer Device Name as ASCII string.

Set Hardware Revision (SET_HARDWARE_REV)

This command implements the Manufacturer Hardware Revision (object 1009h).

Note: This command may only be issued during initialisation.

Command initiator	Application
Command number	0006h
Extended Header data	Fault information
Message data	Manufacturer hardware revision value
Response message	-
Firmware Revision	All

Command and response layout:

	Command	Expected response
Message ID	(ID)	(ID)
Message information	4002h	0002h
Command	0006h	0006h
Data size	0004h	0004h
Frame count	0001h	0001h
Frame number	0001h	0001h
Offset high	0000h	0000h
Offset low	0000h	0000h
Extended word 1	-	-
Extended word 2	-	-
Extended word 3	-	-
Extended word 4	-	-
Extended word 5	-	-
Extended word 6	-	-
Extended word 7	-	-
Extended word 8	-	-
		Fault Information
Message data byte 1	Revision (high byte)	Revision (high byte)
Message data byte 2	Revision (mid high byte)	Revision (mid high byte)
Message data byte 3	Revision (mid low byte)	Revision (mid low byte)
Message data byte 4	Revision (low byte)	Revision (low byte)

SET_HARDWARE_REV

- **Fault Information**
(see 4-1 “Fault Information”)
- **Revision**
4-byte Manufacturer Hardware Revision.

Emergency Message Indication (EMCY_MESSAGE)

This command issues an emergency telegram (EMCY) on the bus. The structure of the emergency telegram is specified in the DS301 specification from CiA (CAN in Automation).

Note: This command may only be issued during runtime.

Command initiator	Application
Command number	0030h
Extended Header data	Fault information
Message data	EMCY error information
Response message	-
Firmware Revision	All

Command and response layout:

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	
Command	0030h	0030h	EMCY_MESSAGE
Data size	0007h	0007h	
Frame count	0001h	0001h	
Frame number	0001h	0001h	
Offset high	0000h	0000h	
Offset low	0000h	0000h	
Extended word 1	-	-	
Extended word 2	-	-	
Extended word 3	-	-	
Extended word 4	-	-	
Extended word 5	-	-	
Extended word 6	-	-	
Extended word 7	-	-	
Extended word 8	-	Fault Information	
Message data byte 1	Emergency Error Code 1	Emergency Error Code 1	Emergency Error Code
Message data byte 2	Emergency Error Code 2	Emergency Error Code 2	Emergency Error Code
Message data byte 3	Emergency Error Code 3	Emergency Error Code 3	Manufacturer Specific Error
Message data byte 4	Emergency Error Code 4	Emergency Error Code 4	Manufacturer Specific Error
Message data byte 5	Emergency Error Code 5	Emergency Error Code 5	Manufacturer Specific Error
Message data byte 6	Emergency Error Code 6	Emergency Error Code 6	Manufacturer Specific Error
Message data byte 7	Emergency Error Code 7	Emergency Error Code 7	Manufacturer Specific Error

- Fault Information**

(see 4-1 “Fault Information”)

- Emergency Error Code [1... 7]**

For more information regarding the structure of the EMCY telegram, consult the DS301 specification.

Fieldbus Specific Area

Address	Contents
640h	EtherCAT State
641h... 7BFh	(reserved)

- **EtherCAT State**

This register reflects the current EtherCAT application layer state:

bit(s)	Description
0	INIT
1	PRE-OPERATIONAL
2	SAFE-OPERATIONAL
3	OPERATIONAL
4	State change flag
5... 7	(reserved)

Miscellaneous

Control Register Area

Fieldbus Type

The fieldbus type value for this product is 0087h (EtherCAT).

Module Type

The module type value for this product is 0101h (Anybus-S).

Watchdog Counter Input (7D2h... 7D3h)

If the application has enabled the Watchdog Counter Input and doesn't update it properly, the Anybus module will go back to the INIT state and indicate that a PDI watchdog timeout has occurred on the 'ERR' LED.

This situation is considered fatal; there is no way to recover from it besides power cycling the module.

Event Notification Cause/Source Registers

- **ON/OFF Line Indication (FBON/FBOF)**

The On/Off line indication is set when the module is in the OPERATIONAL state.

- **Network Reset Functionality (RST)**

(not implemented)

Slave Information Interface Area Defaults

General

Adr.	Description	Data (LE)	Comment
0000h	PDI Control	0005h	SPI slave
0001h	PDI Configuration	0003h	PushPull low INT, low SEL
0002h	SyncImpulseLen	0000h	-
0003h	(reserved)	0000h	-
0004h	Configured station alias	0000h	-
0005h... 0006h	(reserved)	0000h	-
0007h	Checksum	00AFh	-
0008h... 0009h	Vendor ID	-	See 2-1 "Identity Customization"
000Ah... 000Bh	Product Code	-	
000Ch... 000Dh	Revision Number	-	
000Eh... 000Fh	Serial number	-	
0010h	Execution delay	0000h	-
0011h	Port0 delay	0000h	-
0012h	Port1 delay	0000h	-
0013h	(reserved)	0000h	-
0014h	Bootstrap mbx rx offset	0000h	-
0015h	Bootstrap mbx rx size	0000h	-
0016h	Bootstrap mbx tx offset	0000h	-
0017h	Bootstrap mbx tx size	0000h	-
0018h	Standard mbx rx offset	1C00h	-
0019h	Standard mbx rx size	00C0h	-
001Ah	Standard mbx tx offset	1E00h	-
001Bh	Standard mbx tx size	00C0h	-
001Ch	Mailbox protocol	0004h	CoE
001Dh... 001Fh	(reserved)	0000h	-
0020h	Port0 Tx delay	YYYY	-
0021h	Port1 Tx delay	YYYY	-
0022h	Port2 Tx delay	0000h	-
0023h	Port3 Tx delay	0000h	-
0024h	Port0 Rx delay	YYYY	-
0025h	Port1 Rx delay	YYYY	-
0026h	Port2 Rx delay	0000h	-
0027h	Port3 Rx delay	0000h	-
0028h	Forward port 0 to next port	0000h	-
0029h	Forward non-port 0 to next port	0000h	-
002Ah	Additive forward time closed port	0000h	-
002Bh... 003Dh	(reserved)	0000h	-
003Eh	Size	001Fh	Size of EEPROM in kBit-1
003Fh	Version	0001h	This is version 1

Structure Category String

Offset	Description	Data (LE)	Comment
0000h	Category type	000Ah	10 (STRINGS)
0001h	Category word size	0005h	Size may be different if device name is changed by mailbox.
0002h	Number of strings	03h	-
0003h...	Strings	"Anybus-S"	2-1 "Identity Customization"

Structure Category General

Offset	Description	Data (LE)	Comment
0000h	Category type	001Eh	30 (General)
0001h	Category word size	0010h	16 words
0002h	GroupIdx	00h	Group Information (Not used)
	ImgIdx	00h	Image name (Not used)
0003h	OrderIdx	00h	Device order number (Not used)
	NamIdx	01h	Device Name Information (idx to string "Anybus-S" in string structure)
0004h	Port PHY layer	05h	2x100BASE-TX
	CoE details	00010011b	SDO, info, enable upload at startup
0005h	FoE details	00h	FoE disabled
	EoE details	00h	EoE disabled
0006h	SoE Channels	00h	-
	DS402 Channels	00h	-
0007h	SysmanClass	00h	-
	Flags	00h	Bit 0: Enable safeop Bit 1: Enable notLRW
0008h	CurrentOnEBus	0000h	-
0009h... 0012h	PAD_bytes	0000h	-

Structure Category FMMU

Offset	Description	Data (LE)	Comment
0000h	Category type	0028h	40 (FMMU)
0001h	Category word size	0002h	2 words
0002h	FMMU0	02h	Output I/O
	FMMU1	03h	Input I/O
0003h	FMMU2	00h	(not used)
	FMMU3	00h	(not used)

Structure Category SyncM

Offset	Description	Data (LE)	Comment
0000h	Category type	0029h	41 (SyncM)
0001h	Category word size	0012h	18 words
0002h	Physical Start Address	1C00h	-
0003h	Length	00C0h	-
0004h	Control Register	26h	MBoxOut
	Status Register	00h	Don't care
0005h	Activate	01h	Enable SyncM
	PDI CTRL	00h	Don't care
0006h	Physical Start Address	1E00h	-
0007h	Length	00C0h	-
0008h	Control Register	22h	MBoxIn
	Status Register	00h	Don't care
0009h	Activate	01h	Enable SyncM
	PDI CTRL	00h	Don't care
000Ah	Physical Start Address	1000h	-
000Bh	Length	(size)	Equals size of Output I/O Data
000Ch	Control Register	24h	-
	Status Register	00h	Don't care
000Dh	Activate	01h	Enable SyncM
	PDI CTRL	00h	Don't care
000Eh	Physical Start Address	1600h	-
000Fh	Length	(size)	Equals size of Input I/O Data
0010h	Control Register	20h	-
	Status Register	00h	Don't care
0011h	Activate	01h	Enable SyncM
	PDI CTRL	00h	Don't care

Structure Category TxPDO and RxPDO for Each PDO

Due to technical reasons, this structure is not present in the Anybus-S implementation.

Mechanical Specification

tbd

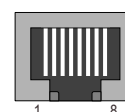
Connectors

Application Connector

(Consult the general Anybus-S Parallel Design Guide for more information)

Ethernet Connector (RJ45)

Pin	Signal	Notes
1	TD+	-
2	TD-	-
3	RD+	-
4	-	Normally left unused; to ensure signal integrity, these pins are tied together and terminated to PE via a filter circuit in the module.
5	-	
6	RD-	-
7	-	Normally left unused; to ensure signal integrity, these pins are tied together and terminated to PE via a filter circuit in the module.
8	-	



Technical Specification (Preliminary)

Electrical Specification

Protective Earth (PE) Requirements

All Anybus-S/M modules features a cable shield filter designed according to each network standard. To be able to support this, the application *must* provide a connection to PE (Protective Earth) as described in the general Anybus-S Parallel Design Guide. HMS cannot guarantee proper EMC behaviour unless this requirement is fulfilled.

Power Supply

Supply Voltage

The module requires a regulated 5V power supply as specified in the Anybus-S Parallel Design Guide.

Power Consumption

The maximum current consumption for this product is TBDmA.

Environmental Specification

- **Temperature**

Test performed according to IEC-68-2-1 and IEC 68-2-2.

Operating: 0 to 70°C (32 to 158°F)

Storage: -25 to +85°C (-13 to 185°F)

- **Humidity**

The product is designed for a relative humidity of 5 to 95% non-condensing.

Test performed according to IEC 68-2-30.

EMC Compliance (CE)

The EMC pre-testing has been conducted according to the following standards:

- **Emission:** EN 50 081-2:1993

Tested per EN 55011:1990, class A, radiated

- **Immunity:** EN 61000-6-2: 1999

Tested per EN 61000-4-2:1995

EN 61000-4-3:1996

EN 61000-4-4:1995

EN 61000-4-5:1995

EN 61000-4-6:1996

