Fieldbus Appendix

Anybus-S DeviceNet

HMS
Connecting Devices™

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

Table of Contents

Chapter 4 Mailbox Interface

Chapter 5   Fieldbus Specific Area

Appendix A Miscellaneous

Appendix B Connectors

Appendix C Mechanical Specification

Appendix D Technical Specification

Appendix E Quick Connect

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 DeviceNet 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 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
DeviceNet Specification	ODVA
-	-

Document History

Summary of Recent Changes (v2.52... v2.53)

Change	Page(s)
Updated front page info	-
Updated sales and support info	P-3
Corrected data size in SET_MAC_AND_BR to 4 bytes	4-5

Revision List

Revision	Date	Author(s)	Chapter(s)	Description
<2.50	-	-	-	(see previous revisions)
2.50	2007-09-11	PeP	All	Major update to match recent product development
2.51	2008-11-05	HeS	4, E	Minor update
2.52	2009-12-14	KeL	4	Minor update
2.53	2011-01-29	KeL	P, 4	

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.

Sales and Support

Sales		Support	
HMS Sweden (Head Office)			
E-mail:	sales@hms.se	E-mail:	support@hms-networks.com
Phone:	+46 (0) 35 - 17 29 56	Phone:	+46 (0) 35 - 17 29 20
Fax:	+46 (0) 35 - 17 29 09	Fax:	+46 (0) 35 - 17 29 09
Online:	www.anybus.com	Online:	www.anybus.com
HMS North America			
E-mail:	us-sales@hms-networks.com	E-mail:	us-support@hms-networks.com
Phone:	+1-312 - 829 - 0601	Phone:	+1-312-829-0601
Toll Free:	+1-888-8-Anybus	Toll Free:	+1-888-8-Anybus
Fax:	+1-312-629-2869	Fax:	+1-312-629-2869
Online:	www.anybus.com	Online:	www.anybus.com
HMS Germany			
E-mail:	ge-sales@hms-networks.com	E-mail:	ge-support@hms-networks.com
Phone:	+49 (0) 721-96472-0	Phone:	+49 (0) 721-96472-0
Fax:	+49 (0) 721-96472-10	Fax:	+49 (0) 721-96472-10
Online:	www.anybus.de	Online:	www.anybus.de
HMS Japan			
E-mail:	jp-sales@hms-networks.com	E-mail:	jp-support@hms-networks.com
Phone:	+81 (0) 45-478-5340	Phone:	+81 (0) 45-478-5340
Fax:	+81 (0) 45-476-0315	Fax:	+81 (0) 45-476-0315
Online:	www.anybus.jp	Online:	www.anybus.jp
HMS China			
E-mail:	cn-sales@hms-networks.com	E-mail:	cn-support@hms-networks.com
Phone:	+86 (0) 10-8532-3183	Phone:	+86 (0) 10-8532-3023
Fax:	+86 (0) 10-8532-3209	Fax:	+86 (0) 10-8532-3209
Online:	www.anybus.cn	Online:	www.anybus.cn
HMS Italy			
E-mail:	it-sales@hms-networks.com	E-mail:	it-support@hms-networks.com
Phone:	+39 039 59662 27	Phone:	+39 039 59662 27
Fax:	+39 039 59662 31	Fax:	+39 039 59662 31
Online:	www.anybus.it	Online:	www.anybus.it
HMS France			
E-mail:	fr-sales@hms-networks.com	E-mail:	fr-support@hms-networks.com
Phone:	+33 (0) 3 68 368 034	Phone:	+33 (0) 3 68 368 033
Fax:	+33 (0) 3 68 368 031	Fax:	+33 (0) 3 68 368 031
Online:	www.anybus.fr	Online:	www.anybus.fr
HMS UK & Eire			
E-mail:	uk-sales@anybus.co.uk	E-mail:	support@hms-networks.com
Phone:	+44 (0) 1926 405599	Phone:	+46 (0) 35 - 17 29 20
Fax:	+44 (0) 1926 405522	Fax:	+46 (0) 35 - 17 29 09
Online:	www.anybus.co.uk	Online:	www.anybus.com
HMS Denmark			
E-mail:	info@anybus.dk	E-mail:	support@hms-networks.com
Phone:	+45 (0) 22 30 08 01	Phone:	+46 (0) 35 - 17 29 20
Fax:	+46 (0) 35 17 29 09	Fax:	+46 (0) 35 - 17 29 09
Online:	www.anybus.com	Online:	www.anybus.com
HMS India			
E-mail:	in-sales@anybus.com	E-mail:	in-support@hms-networks.com
Phone:	+91 (0) 20 40111201	Phone:	+46 (0) 35 - 17 29 20
Fax:	+91 (0) 20 40111105	Fax:	+46 (0) 35 - 17 29 09
Online:	www.anybus.com	Online:	www.anybus.com

About the Anybus-S DeviceNet

General

The Anybus-S DeviceNet communication module provides instant DeviceNet adapter functionality via the patented Anybus-S application 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 parallel application interface defined in the Anybus-S Parallel Design Guides, making it fully interchangeable with any other device following that specification. Generally, very little additional network related software support is needed, however in order to take advantage of advanced network specific functionality, a certain degree of dedicated software support may be necessary.

Features

- Galvanically isolated bus electronics
- 125kbit, 250kbit and 500kbit operation
- Automatic baudrate detection
- On-board configuration switches
- Up to 512 bytes of I/O in each direction
- Explicit Messaging (up to 2048 bytes in each direction)
- Polled I/O
- Bit-strobed I/O
- Change-of-state / Cyclic I/O
- Unconnected Message Manager (UCMM, 5 servers)
- Supports Quick Connect

Conformance Note

The Anybus-S DeviceNet has been pre-compliance tested stand alone by ODVA's authorized Independent Test Lab, and found to comply with ODVA Conformance Test Software Version A12. However, according to ODVA's conformance test policy, the final product must still be re-certified as a complete product even if the Anybus module itself is certified.

For further information, contact HMS or ODVA.

Overview
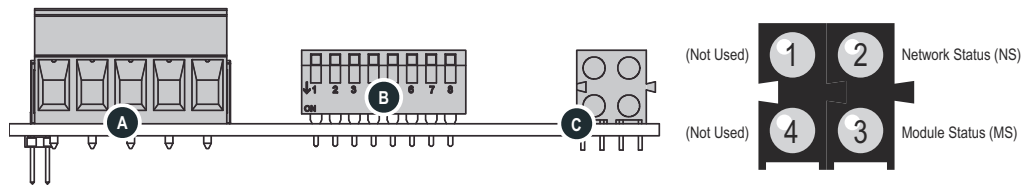
#	Description	Description
A	DeviceNet Connector	See Appendix B-1 "Connectors"
B	Configuration Switches	See 2-3 "MAC ID & Baud Rate Configuration"
C	Network Status LEDs	See 1-2 "Network Status LEDs"

Network Status LEDs

These leds indicate run time status and errors to the user.

LED	State	Meaning
Network Status (NS)	Off	Not powered, not on line
	Green	On line, one or more connections established
	Green, flashing	On line, no connections established
	Red	Critical link failure
	Red, flashing	One or more connections timed out
	Alternating Red/Green	Device self-test in progress
Module Status (MS)	Off	No power
	Green	Normal operation
	Green, flashing	Auto baud in progress
	Red	Major fault
	Red, flashing	Minor fault
	Alternating Red/Green	Device self-test in progress

Basic Operation

General Information

Software Requirements

No additional network support code needs to be written in order to support the Anybus-S DeviceNet. However, due the nature of this networking system, certain restrictions must be taken into account:

- No mailbox commands may be issued for at least 2 seconds after receiving the response to the END_INIT-message.
- If the product is to be mounted vertically or up side down, the value of the on-board switches must be re-interpreted accordingly in order to comply with the DeviceNet specification.

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

Electronic Data Sheet (EDS)

Each device in a DeviceNet network is associated with an EDS file, which describes the implementation of the product. This file is used by network configurations tool during network configuration.

The latest version of EDS file can either be downloaded from HMS website (www.hms-networks.com) or obtained by contacting HMS.

See also...

- 2-2 "CIP Identity Information"
- 3-2 "Identity Object, Class 01h"

IMPORTANT: *The generic EDS-file supplied by HMS can be used with the standard CIP Identity settings. To comply with CIP-specification requirements, any changes in this file also requires a new Vendor ID.*

For further information, contact HMS and/or ODVA.

CIP Identity Information

By default, the module supports the generic CIP-profile with the following identity settings:

- Vendor ID: 005Ah (HMS Industrial Networks)
- Device Type: 000Ch (Communications Adapter)
- Product Code: 000Ch (Anybus-S DeviceNet)
- Product Name: 'Anybus-S DeviceNet'

See also...

- 3-2 "Identity Object, Class 01h"
- 4-15 "Set Product Info (SET_PRODUCT_INFO)"
- 4-16 "Set Product Info All (SET_PRODUCT_INFO_ALL)"

IMPORTANT: *The default identity information is valid only when using the standard EDS-file supplied by HMS. To comply with CIP-specification requirements, custom EDS-implementations require a new Vendor ID.*

For further information, contact HMS and/or ODVA.

MAC ID & Baud Rate Configuration

On a DeviceNet network, each node must be assigned it's own unique Mac ID. The Mac ID is a value between 0 an 63 used to identify each node. The Mac ID and Baud rate settings can be set either using a physical DIP-switch or via the application interface using mailbox commands.

Switches 1 and 2 specify the baudrate, while switches 3 though 8 specify the Mac ID:

S1	S2	Baudrate
OFF	OFF	125k
OFF	ON	250K
ON	OFF	500K
ON	ON	(reserved)

S3	S4	S5	S6	S7	S8	Mac ID
OFF	OFF	OFF	OFF	OFF	OFF	0
OFF	OFF	OFF	OFF	OFF	ON	1
OFF	OFF	OFF	OFF	ON	OFF	2
OFF	OFF	OFF	OFF	ON	ON	3
...	...	...	...	...	...	...
ON	ON	ON	ON	OFF	OFF	60
ON	ON	ON	ON	OFF	ON	61
ON	ON	ON	ON	ON	OFF	62
ON	ON	ON	ON	ON	ON	63

Depending on the physical orientation of the module is mounted, it may be necessary to change the bit definition of the switches to comply with the DeviceNet specification. This is achieved by reading the physical switch value (see 4-4 "Get Dip Switch (GET_DIPSWITCH)"), alter it as required, and write it back (see 4-5 "Set MAC ID and Baud Rate (SET_MAC_AND_BR)").

Optionally, the MAC ID and Baud Rate can be specified from the network. This functionality is disabled by default, but can be enabled using the 'SET_MAC_AND_BR'-command.

See also...

- 4-4 "Get Dip Switch (GET_DIPSWITCH)"
- 4-5 "Set MAC ID and Baud Rate (SET_MAC_AND_BR)"

IMPORTANT: *The recommendations regarding the switches has changed somewhat compared to older revisions of the product. Note however that the module is fully backwards compatible regarding this functionality, and still supports the 'SET_DIPSWITCH'- and 'SET_DIPSWITCH2'-commands.*

Data Exchange

General Information

DeviceNet is based on the Control and Information protocol (CIP) which is also the application layer for ControlNet and EtherNet/IP.

See also...

- 3-1 "CIP Object Implementation"

I/O Data

I/O Data is represented on DeviceNet through dedicated instances in the Assembly Object.

See also...

- 3-6 "Assembly Object, Class 04h"
- 3-13 "I/O Data Input Mapping Object, Class A0h"
- 3-14 "I/O Data Output Mapping Object, Class A1h"
- 4-11 "I/O Data Input Area Mapping (IO_INPUT_MAP)"
- 4-13 "I/O Data Output Area Mapping (IO_OUTPUT_MAP)"

Parameter Data (Explicit Data)

Parameter Data can be accessed through explicit messages towards the Parameter Input- and Output Mapping Objects.

The actual mapping between Parameter Data and instance attributes can be specified using the 'PARAMETER_INPUT_MAP' and 'PARAMETER_OUTPUT_MAP' commands.

See also...

- 3-16 "Parameter Data Input Mapping Object, Class B0h"
- 3-18 "Parameter Data Output Mapping Object, Class B1h"
- 4-7 "Parameter Data Input Area Mapping (PARAMETER_INPUT_MAP)"
- 4-9 "Parameter Data Output Area Mapping (PARAMETER_OUTPUT_MAP)"

CIP Object Implementation

General Information

The Anybus-S DeviceNet implements the following standard objects:

- Identity Object, Class 01h
- Message Router, Class 02h
- DeviceNet Object, Class 03h
- Assembly Object, Class 04h
- Connection Object, Class 05h
- Acknowledge Handler Object, Class 2Bh

In addition, the following vendor specific objects are implemented:

- I/O Data Input Mapping Object, Class A0h
- I/O Data Output Mapping Object, Class A1h
- Diagnostic Object, Class AAh
- Parameter Data Input Mapping Object, Class B0h
- Parameter Data Output Mapping Object, Class B1h

Identity Object, Class 01h

General Information

Object Description

-

Implemented Services

Class services:	Get Attribute Single
Instance services:	Get Attribute Single Reset (see 3-3 "Service Details: Reset")

Class Attributes

#	Access	Name	Type	Value
1	Get	Revision	UINT	0001h

Instance Attributes

#	Access	Name	Type	Value
1	Get[a]	Vendor ID	UINT	005Ah (HMS Industrial Networks AB)
2	Get[a]	Device Type	UINT	000Ch (Communications Adapter)
3	Get[a]	Product Code	UINT	000Ch (Anybus-S DeviceNet)
4	Get[a]	Revision	Struct of: USINT, USINT	 (Major fieldbus version) (Minor fieldbus version)
5	Get	Status	WORD	(see 3-3 "Device Status")
6	Get[a]	Serial Number	UDINT	(assigned at manufacturing)
7	Get[a]	Product Name	SHORT_STRING	'Anybus-S DeviceNet'

a. Can be customized.

Device Status

bit(s)	Name
0	Module Owned (A master/scanner has allocated the module)
1	(reserved)
2	Configured (always set to zero)
3	(reserved)
4... 7	Extended Device Status: Value: Meaning: 0000b Unknown 0010b Faulted I/O Connection (not implemented) 0011b No I/O connection established 0100b Non-volatile configuration bad (not implemented) 0110b Connection in Run mode 0111b Connection in Idle mode (other) (reserved)
8	Set for minor recoverable faults
9	Set for minor unrecoverable faults
10	Set for major recoverable faults
11	Set for major unrecoverable faults
12... 15	(reserved)

Service Details: Reset

There are two types of network reset requests on DeviceNet:

- **Type 0: 'Power Cycling Reset'**

  This causes the module to restart the DeviceNet software layer. The communication between the application and the Anybus module remains unaffected.

- **Type 1: 'Out of box reset'**

  This causes the module to revert to an 'out of box' configuration and restart the internal Device-Net software layer. The communication between the application and the Anybus module remains unaffected.

Optionally, the application can be notified of network reset requests through the Event Notification functionality (RST-bit).

The application can then retrieve the type of reset through the 'GET_ID_RESET_PARAM'-mailbox command, and perform the desired actions based on this information. Note however that the module will still perform the actions described above.

See also...

- 4-19 "Get ID Reset Parameter (GET_ID_RESET_PARAM)"
- Anybus-S/M Parallel Design Guide ('Event Notification')

Message Router, Class 02h

General Information

Object Description

-

Supported Services

Class services: -

Instance services: -

Class Attributes

-

Instance Attributes

-

DeviceNet Object, Class 03h

General Information

Object Description

-

Supported Services

Class	Get Attribute Single
Instance:	Get Attribute Single
	Set Attribute Single
	Allocate Master/Slave Connection Set (4Bh)
	Release Group 2 Identifier Set (4Ch)

Class Attributes

#	Name	Access	Type	Value
1	Revision	Get	UINT	0002h

Instance #1 Attributes

#	Name	Access	Type	Value
1	MAC ID[a]	Get (Set)	USINT	(actual MAC ID)
2	Baud Rate[a]	Get (Set)	USINT	Value: Meaning: 0 Operating at 125 kbps 1 Operating at 250 kbps 2 Operating at 500 kbps
3	BOI	Get/Set	BOOL	False
4	Bus off Counter	Get/Set	USINT	00h
5	Allocation Information	Get	Struct of: BYTE, USINT	 Allocation choice byte MAC ID of master
6	MAC ID Switch changed	Get	BOOL	Value: Meaning True MAC ID switches has changed since startup False MAC ID switches has not changed since startup
7	Baud rate Switch changed	Get	BOOL	Value: Meaning True Baudrate switches has changed since startup False Baudrate switches has not changed since startup
8	MAC ID Switch Value	Get	USINT	(actual value of node address switches)
9	Baud Rate Switch Value	Get	USINT	(actual value of baud rate switches)
10	Quick Connect[b]	Get/Set	BOOL	Value: Meaning False Disable Quick Connect (default) True Enable Quick Connect **Note:** This setting is stored in non-volatile memory.

a. Attributes only settable if enabled using 'SET_MAC_AND_BR'. See also 4-5 "Set MAC ID and Baud Rate (SET_MAC_AND_BR)".

b. Optional; attribute only available if this feature has been enabled using 'ENABLE_QUICK_CONNECT'. See also 4-18 "Enable Quick Connect (ENABLE_QUICK_CONNECT)".

Assembly Object, Class 04h

General Information

Object Description

This object provides access to the I/O Data in the Input- and Output Data areas. The Assembly Object uses static assemblies. The instance IDs are in the vendor specific range.

See also...

- 3-13 "I/O Data Input Mapping Object, Class A0h"
- 3-14 "I/O Data Output Mapping Object, Class A1h"
- 4-11 "I/O Data Input Area Mapping (IO_INPUT_MAP)"
- 4-13 "I/O Data Output Area Mapping (IO_OUTPUT_MAP)"

Services

Class services: Get Attribute Single

Instance services: Get Attribute Single
                         Set Attribute Single

Class Attributes

#	Access	Name	Type	Value
1	Get	Revision	UINT	0001h

Instance Attributes - Instance/Connection Point 64h

This instance is used to access the I/O Data area (Input).

#	Access	Name	Type	Value
3	Get	Data	Array of BYTE	Corresponds to the Input I/O data area.

Instance Attributes - Instance/Connection Point 65h... 69h

These instances are not available by default, but can optionally be created by the application using the 'IO_INPUT_MAP'-command. Each instance corresponds to a specified block of Input I/O Data, and is also represented as instance attributes in the I/O Data Input Mapping Object, Class A0h.

#	Access	Name	Type	Value
3	Get	Data	Array of BYTE	Mapped block of Input I/O data

See also...

- 3-13 "I/O Data Input Mapping Object, Class A0h"
- 4-11 "I/O Data Input Area Mapping (IO_INPUT_MAP)"

Instance Attributes - Instance/Connection Point 96h

This instance is used to access the I/O Data area (Output).

#	Access	Name	Type	Value
3	Get	Data	Array of BYTE	Corresponds to the Output I/O data area.

Instance Attributes - Instance/Connection Point 97h... 9Bh

These instances are not available by default, but can optionally be created by the application using the 'IO_OUTPUT_MAP'-command. Each instance corresponds to a specified block of Output I/O Data, and is also represented as instance attributes in the I/O Data Output Mapping Object, Class A1h.

#	Access	Name	Type	Value
3	Set	Data	Array of BYTE	Mapped block of Output I/O data

See also...

- 3-13 "I/O Data Input Mapping Object, Class A0h"
- 4-13 "I/O Data Output Area Mapping (IO_OUTPUT_MAP)"

Connection Object, Class 05h

General Information

Object Description

-

Implemented Services

Class services: Get Attribute Single

Instance services: Get Attribute Single
                      Set Attribute Single

Class Attributes

#	Access	Name	Type	Value				
1	Get	Revision	UINT	0002h				
100[a]	Get	Produced Data for Poll Connection	USINT	Value: 0: 1: 2:	Instance no.: 100 (default) 101 102	Value: 3: 4: 5:	Instance no.: 103 104 105	
101[a]	Get	Consumed Data for Poll Connection	USINT	Value: 0: 1: 2:	Instance no.: 150 (default) 151 152	Value: 3: 4: 5:	Instance no.: 153 154 155	
102[a]	Get	Produced Data for Bit Strobe Connection	USINT	Value: 0: 1: 2:	Instance no.: 100 (default) 101 102	Value: 3: 4: 5:	Instance no.: 103 104 105	
103[a]	Get	Consumed Data for Bit Strobe Connection	USINT	Value: 0: 1: 2:	Instance no.: 150 (default) 151 152	Value: 3: 4: 5:	Instance no.: 153 154 155	
104[a]	Get	Produced Data for COS/Cyclic Connection	USINT	Value: 0: 1: 2:	Instance no.: 100 (default) 101 102	Value: 3: 4: 5:	Instance no.: 103 104 105	

a. Value saved in non-volatile memory.

Instance Attributes

#	Access	Name	Type	Value
1	Get	State	USINT	Value: Meaning: 0 Non existent 1 Configuring 3 Established 4 Timeout 5 Deferred delete
2	Get	Instance type	USINT	Value: Meaning: 0 Explicit messaging connection 1 I/O Connection
3	Get	Transport Class trigger	BYTE	(defines the behaviour of the connection)
4	Get	Produced Connection ID	UINT	(CAN ID for transmission)
5	Get	Consumed Connection ID	UINT	(CAN ID for reception)
6	Get	Initial Comm Characteristics	BYTE	0Fh (No ACK) - Produces over message group - Does not consume 01h (ACK) - Produces over message group 1 - Consumes over message group 2
7	Get	Produced Connection Size	UINT	-
8	Get	Consumed Connection Size	UINT	0
9	Get/Set	Expected Packet Rate	UINT	0
12	Get	Watchdog timeout action	USINT	Value: Meaning: 0 Transition to the timed out state 1 Auto Delete 2 Auto Reset 3 Deferred Delete
13	Get	Produced Connection path length	UINT	0006h
14	Get	Produced Connection Path	EPATH	-
15	Get	Consumed Connection path length	UINT	0004h
16	Get	Consumed Connection Path	EPATH	20 2B 24 01h
17				
18				

Instance 1 (Explicit Messaging Connection) Attributes

#	Access	Name	Type	Value
1	Get	State	USINT	Value: Meaning: 0 Non existent 1 Configuring 3 Established 4 Timeout 5 Deferred delete
2	Get	Instance type	USINT	0 (Explicit messaging connection)

Instance 2 (Polled Connection) Attributes

#	Access	Name	Type	Value
1	Get	State	USINT	Value: Meaning: 0 Non existent 1 Configuring 3 Established 4 Timeout
2	Get	Instance type	USINT	1 (I/O Connection)

Instance 3 (Bit-strobe connection) Attributes

#	Access	Name	Type	Value
1	Get	State	USINT	Value: Meaning: 0 Non existent 1 Configuring 3 Established 4 Timeout
2	Get	Instance type	USINT	1 (I/O Connection)

Instance 4 (COS/Cyclic connection) Attributes

#	Access	Name	Type	Value
1	Get	State	USINT	Value: Meaning: 0 Non existent 1 Configuring 3 Established 4 Timeout
2	Get	Instance type	USINT	1 (I/O Connection)
3	Get	Transport Class trigger	BYTE	(defines the behaviour of the connection)
4	Get	Produced Connection ID	UINT	(CAN ID for transmission)
5	Get	Consumed Connection ID	UINT	(CAN ID for reception)
6	Get	Initial Comm Characteristics	BYTE	0Fh (No ACK) - Produces over message group - Does not consume 01h (ACK) - Produces over message group 1 - Consumes over message group 2
7	Get	Produced Connection Size	UINT	-
8	Get	Consumed Connection Size	UINT	0
9	Get/Set	Expected Packet Rate	UINT	0
12	Get	Watchdog timeout action	USINT	Value: Meaning: 0 Transition to the timed out state 1 Auto Delete 2 Auto Reset 3 Deferred Delete
13	Get	Produced Connection path length	UINT	0006h
14	Get	Produced Connection Path	EPATH	-
15	Get	Consumed Connection path length	UINT	0004h
16	Get	Consumed Connection Path	EPATH	20 2B 24 01h

Instance 10... 14 (UCMM Explicit Server Instances) Attributes

#	Access	Name	Type	Value
1	Get	State	USINT	Value: Meaning: 0 Non existent 1 Configuring 3 Established 4 Timeout 5 Deferred delete
2	Get	Instance type	USINT	0 (Explicit messaging connection)

Acknowledge Handler Object, Class 2Bh

General Information

Object Description

-

Implemented Services

Class services: Get Attribute Single

Instance services: Get Attribute Single
                       Set Attribute Single

Class Attributes

#	Access	Name	Type	Value
1	Get	Revision	UINT	0001h
2	Get	Max Instance	UINT	0001h

Instance Attributes

#	Access	Name	Type	Value
1	Get/Set	Acknowledge Timer	UINT	16
2	Get/Set	Retry Limit	USINT	1
3	Get/Set	Producing Connection Instance	UINT	4
4	Get	Ack List Size	Byte	1
5	Get	Ack List	Array of USINT	N/A
6	Get	Data with Ack Path List Size	Byte	1
7	Get	Data with Ack Path List	Array of USINT	N/A

Note: Instance 1 is created when initiating an acknowledged COS/Cyclic connection.

I/O Data Input Mapping Object, Class A0h

General Information

Object Description

This object can be used to access I/O Data, and is set up dynamically by the application using the 'IO_INPUR_MAP'-command. The mapped data is also available through vendor specific instances (64h to 69h) in the Assembly Object.

See also...

- 3-6 "Assembly Object, Class 04h"
- 3-14 "I/O Data Output Mapping Object, Class A1h"
- 4-11 "I/O Data Input Area Mapping (IO_INPUT_MAP)"

Supported Services

Class services: Get Attribute All

Instance services: Get Attribute Single

Class Attributes

#	Access	Name	Type	Value
1	Get	Data	UINT	0001h

Instance Attributes, Instance 01h

Each attribute corresponds to a block of Input I/O Data.

#	Access	Name	Type	Description
1[a]	Get	Data	Array of USINT	Mapped block of Input I/O data (also available as Assembly Instance 64h)
2[b]	Get	Data	Array of USINT	Mapped block of Input I/O data (also available as Assembly Instance 65h)
3[b]	Get	Data	Array of USINT	Mapped block of Input I/O data (also available as Assembly Instance 66h)
4[b]	Get	Data	Array of USINT	Mapped block of Input I/O data (also available as Assembly Instance 67h)
5[b]	Get	Data	Array of USINT	Mapped block of Input I/O data (also available as Assembly Instance 68h)
6[b]	Get	Data	Array of USINT	Mapped block of Input I/O data (also available as Assembly Instance 69h)

a. This attribute is created automatically if the Input I/O Data is larger than 0 (zero).
b. These attributes are not available by default, but can optionally be created using the 'IO_INPUT_MAP'-command.

See also...

- 3-7 "Instance Attributes - Instance/Connection Point 65h... 69h"
- 4-11 "I/O Data Input Area Mapping (IO_INPUT_MAP)"

I/O Data Output Mapping Object, Class A1h

General Information

Object Description

This object can be used to access I/O Data, and is set up dynamically by the application using the 'IO_OUTPUT_MAP'-command. The mapped data is also available through vendor specific instances (96h to 6Bh) in the Assembly Object.

See also...

- 3-6 "Assembly Object, Class 04h"
- 3-13 "I/O Data Input Mapping Object, Class A0h"
- 4-13 "I/O Data Output Area Mapping (IO_OUTPUT_MAP)"

Supported Services

Class services: Get Attribute All

Instance services: Get Attribute Single
                        Set Attribute Single

Class Attributes

#	Access	Name	Type	Value
1	Get	Data	UINT	0001h

Instance Attributes, Instance 01h

Each attribute corresponds to a block of Output I/O Data.

#	Access	Name	Type	Description
1[a]	Get/Set	Data	Array of USINT	Mapped block of Output I/O data (also available in Assembly Instance 96h)
2[b]	Get/Set	Data	Array of USINT	Mapped block of Output I/O data (also available in Assembly Instance 97h)
3[b]	Get/Set	Data	Array of USINT	Mapped block of Output I/O data (also available in Assembly Instance 98h)
4[b]	Get/Set	Data	Array of USINT	Mapped block of Output I/O data (also available in Assembly Instance 99h)
5[b]	Get/Set	Data	Array of USINT	Mapped block of Output I/O data (also available in Assembly Instance 9Ah)
6[b]	Get/Set	Data	Array of USINT	Mapped block of Output I/O data (also available in Assembly Instance 9Bh)

a. This attribute is created automatically if the Output I/O Data is larger than 0 (zero).
b. These attributes are not available by default, but can optionally be created using the 'IO_OUTPUT_MAP'-command.

See also...

- 3-7 "Instance Attributes - Instance/Connection Point 97h... 9Bh"
- 4-13 "I/O Data Output Area Mapping (IO_OUTPUT_MAP)"

Diagnostic Object, Class AAh

General Information

Object Description

This vendor specific object provides diagnostic information from the module.

Implemented Services

Class services: Get Attribute All

Instance services: Get Attribute Single

Class Attributes

#	Access	Name	Type	Value
1	Get	Revision	UINT	0001h

Instance Attributes, Instance 01h

#	Access	Name	Type	Description
01h	Get	Module serial number	UDINT	(unique serial number, assigned during manufacturing)
04h	Get	Module Software version	UINT	(fieldbus software revision)
0Fh	Get	Input I/O Size	UINT	(total size of Input I/O Data in bytes)
11h	Get	Input Total Size	UINT	(total size of Input I/O+ Parameter Data in bytes)
12h	Get	Output I/O Size	UINT	(total size of Output I/O Data in bytes)
14h	Get	Output Total Size	UINT	(total size of Input I/O+ Parameter Data in bytes)

Parameter Data Input Mapping Object, Class B0h

General Information

Object Description

This object can be used to access Parameter Data acyclically, and is set up dynamically by the application using the 'PARAMETER_INPUT_MAP'-command.

See also...

- 3-18 "Parameter Data Output Mapping Object, Class B1h"
- 4-7 "Parameter Data Input Area Mapping (PARAMETER_INPUT_MAP)"

Supported Services

Class services: Get Attribute All

Instance services: Get Attribute Single

Class Attributes

#	Access	Name	Type	Value
1	Get	Revision	UINT	0001h

Instance Attributes, Instance 01h

Each attribute corresponds to a block of Input Parameter Data. The actual size and location of each block can be specified by the application using the 'PARAMETER_INPUT_MAP'-command during initialization.

If no mapping information has been provided (i.e. if the 'PARAMETER_INPUT_MAP' has not been issued), the module falls back to a simple default mapping scheme where all Output Parameter Data is mapped to consecutive attribute (starting with 01h), each one carrying up to 512 bytes of data.

Note: Some masters may have trouble accessing blocks larger than 256 bytes. It is therefore generally recommended to customize the Parameter Data map, since the default mapping scheme exceeds this limitation.

Example:

Input Parameter Data size is 1280 bytes, no mapping information provided.

Resulting attributes (default mapping scheme):

#	Access	Name	Type	Description
01h	Get	Data	Array of USINT	512 bytes of Input Parameter Data
02h	Get	Data	Array of USINT	512 bytes of Input Parameter Data
03h	Get	Data	Array of USINT	256 bytes of Input Parameter Data

Example:

Input Parameter Data size is 100 bytes, four blocks comprising of 25 bytes each are mapped using the 'PARAMETER_INPUT_MAP'-command.

Resulting attributes (custom mapping scheme):

#	Access	Name	Type	Description
01h	Get	Data	Array of USINT	25 bytes of Input Parameter Data
02h	Get	Data	Array of USINT	25 bytes of Input Parameter Data
03h	Get	Data	Array of USINT	25 bytes of Input Parameter Data
03h	Get	Data	Array of USINT	25 bytes of Input Parameter Data

See also...

- 3-18 "Parameter Data Output Mapping Object, Class B1h"
- 4-7 "Parameter Data Input Area Mapping (PARAMETER_INPUT_MAP)"

Parameter Data Output Mapping Object, Class B1h

General Information

Object Description

This object can be used to access Parameter Data acyclically, and is set up dynamically by the application using the 'PARAMETER_OUTPUT_MAP'-command.

See also...

- 3-16 "Parameter Data Input Mapping Object, Class B0h"
- 4-9 "Parameter Data Output Area Mapping (PARAMETER_OUTPUT_MAP)"

Supported Services

Class services: Get Attribute All

Instance services: Get Attribute Single
                       Set Attribute Single

Class Attributes

#	Access	Name	Type	Value
1	Get	Revision	UINT	0001h

Instance Attributes, Instance 01h

Each attribute corresponds to a block of Output Parameter Data. The actual size and location of each block can be specified by the application using the 'PARAMETER_OUTPUT_MAP'-command during initialization.

If no mapping information has been provided (i.e. if the 'PARAMETER_OUTPUT_MAP' has not been issued), the module falls back to a simple default mapping scheme where all Output Parameter Data is mapped to consecutive attribute (starting with 01h), each one carrying up to 512 bytes of data.

Note: Some masters may have trouble accessing blocks larger than 256 bytes. It is therefore generally recommended to customize the Parameter Data map, since the default mapping scheme exceeds this limitation.

Example:

Output Parameter Data size is 1280 bytes, no mapping information provided.

Resulting attributes (default mapping scheme):

#	Access	Name	Type	Description
01h	Get/Set	Data	Array of USINT	512 bytes of Output Parameter Data
02h	Get/Set	Data	Array of USINT	512 bytes of Output Parameter Data
03h	Get/Set	Data	Array of USINT	256 bytes of Output Parameter Data

Example:

Output Parameter Data size is 100 bytes, four blocks comprising of 25 bytes each are mapped using the 'PARAMETER_OUTPUT_MAP'-command.

Resulting attributes (custom mapping scheme):

#	Access	Name	Type	Description
01h	Get/Set	Data	Array of USINT	25 bytes of Output Parameter Data
02h	Get/Set	Data	Array of USINT	25 bytes of Output Parameter Data
03h	Get/Set	Data	Array of USINT	25 bytes of Output Parameter Data
03h	Get/Set	Data	Array of USINT	25 bytes of Output Parameter Data

See also...

- 3-16 "Parameter Data Input Mapping Object, Class B0h"
- 4-9 "Parameter Data Output Area Mapping (PARAMETER_OUTPUT_MAP)"

Mailbox Interface

General Information

The module implements the following fieldbus specific mailbox commands:

Configuration Switch

Set Dip Switch (SET_DIPSWITCH)

IMPORTANT: *This command is included for backwards compatibility only. For new implementations, see 4-5 "Set MAC ID and Baud Rate (SET_MAC_AND_BR)".*

This command can be used to specify the MAC ID and operating baud rate, and overrides the on-board switches provided that all switches are set to 'ON'-position.

Note: This command may only be issued during initialization.

Command Initiator	Application
Command Number	0001h
Extended Header data	-
Message Data	Logical Switch Value
Response Message	(the response holds a copy of the command data)

Command and response layout:

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	
Command	0001h	0001h	*SET_DIPSWITCH*
Data size	0001h	0001h	
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	-	-	
Message data byte 1	Logical Switch Value	Logical Switch Value	

- **Logical Switch Value**

  Each bit corresponds to a switch value as follows:

Switch:	S8	S7	S6	S5	S4	S3	S2	S1
Bit:	b7	b6	b5	b4	b3	b2	b1	b0

  0: Switch in 'OFF'-position
  1: Switch in 'ON'-position

Set Dip Switch 2 (SET_DIPSWITCH_2)

IMPORTANT: *This command is included for backwards compatibility only. For new implementations, see 4-5 "Set MAC ID and Baud Rate (SET_MAC_AND_BR)".*

This command can be used to specify the MAC ID and operating baud rate, and overrides the on-board switches (there is no need to set the switches to 'ON'-position in order for this command to have effect).

This command can be used in conjunction with GET_DIPSWITCH to alter the actual interpretation of the on-board switches. This is useful if the module is mounted e.g. upside down or vertical, as this changes the orientation of the switches which does not comply with the DeviceNet specification.

Note: This command will only have effect if issued during module initialization; if issued during runtime, the module will update instance attributes #6 and #8 in the DeviceNet Object, and indicate a minor fault in the Module Status LED.

Command Initiator	Application
Command Number	000Ah
Extended Header data	-
Message Data	Logical Switch Value
Response Message	(the response holds a copy of the command data)

Command and response layout:

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	
Command	000Ah	000Ah	*SET_DIPSWITCH_2*
Data size	0001h	0001h	
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	-	-	
Message data byte 1	Logical Switch Value	Logical Switch Value	

- **Logical Switch Value**

  Each bit corresponds to a switch value as follows:

Switch:	S8	S7	S6	S5	S4	S3	S2	S1
Bit:	b7	b6	b5	b4	b3	b2	b1	b0

  0: Switch in 'OFF'-position
  1: Switch in 'ON'-position

Get Dip Switch (GET_DIPSWITCH)

This command can be used to retrieve the binary value of the on-board switches.

This command can be used in conjunction with SET_DIPSWITCH_2 to alter the actual interpretation of the on-board switches. This is useful if the module is mounted e.g. upside down or vertical, as this changes the orientation of the switches which does not comply with the DeviceNet specification.

Command Initiator	Application
Command Number	0008h
Extended Header data	-
Message Data	-
Response Message	Physical switch value.

Command and response layout:

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	
Command	0008h	0008h	*GET_DIPSWITCH*
Data size	0000h	0001h	
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	-	-	
		Physical Switch Value	**Message data byte 1**

- **Physical Switch Value**

  Each bit corresponds to a switch value as follows:

Switch:	S8	S7	S6	S5	S4	S3	S2	S1
Bit:	b7	b6	b5	b4	b3	b2	b1	b0

Set MAC ID and Baud Rate (SET_MAC_AND_BR)

This command can be used to specify the value and source for the MAC ID and baud rate and overrides the on-board switches.

Note: This command will only have effect if issued during module initialization; if issued during runtime, the module will update instance attributes #6 and #8 in the DeviceNet Object, and indicate a minor fault in the Module Status LED.

Command Initiator	Application
Command Number	000Bh
Extended Header data	-
Message Data	MAC ID Source, MAC ID, Baud Rate Source, Baud Rate
Response Message	(the response holds a copy of the command data)

Command and response layout:

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	
Command	000Bh	000Bh	*SET_MAC_AND_BR*
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	-	-	
Message data byte 1	MAC ID Source	MAC ID Source	
Message data byte 2	MAC ID Switch Value	MAC ID Switch Value	
Message data byte 3	Baud Rate Source	Baud Rate Source	
Message data byte 4	Baud Rate Switch Value	Baud Rate Switch Value	

- **MAC ID Source**

   0: Specify the MAC ID using this mailbox command
   1: Specify the MAC ID via DeviceNet

  (other settings are invalid)

- **MAC ID Switch Value**

  Valid Range: 0... 63

  (other settings are invalid, specified value will be used if the 'MAC ID Source' is set to 0.

- **Baud Rate Source**

   0: Specify the baud rate using this mailbox command
   1: Specify the baud rate via DeviceNet
   2: Enable automatic baudrate detection

  (other settings are invalid)

- **Baud Rate Switch Value**

   0: 125kbps
   1: 250kbps
   2: 500kbps
   3: Enable automatic baudrate detection

  (other settings are invalid, specified value will be used if 'Baud Rate Source' is set to 0)

Mapping

Parameter Data Input Area Mapping (PARAMETER_INPUT_MAP)

This command maps blocks of Parameter Input Data to instance attributes in the Parameter Data Input Mapping Object. The blocks are mapped to consecutive attribute numbers starting with attribute #1. Specifying zero length for a block prevents the corresponding attribute number from being mapped. This makes it possible to for example map only attributes 1 and 10 by setting the length-value for attributes 2... 9 to zero.

It is possible to specify up to 50 attribute mappings using this command. It is only necessary to provide information up to the last mapped attribute (i.e. if data should only be mapped to attributes 1... 3, it is not necessary to include information associated with attribute mappings 4... 50).

Should the offset and/or length for a particular attribute mapping be invalid, these parameters will be set to zero in the response, and the corresponding attributes will not be mapped.

See also...

- 3-16 "Parameter Data Input Mapping Object, Class B0h"
- 4-9 "Parameter Data Output Area Mapping (PARAMETER_OUTPUT_MAP)"

Note: This command may only be issued during initialization.

Command Initiator	Application
Command Number	0004h
Extended Header data	-
Message Data	Attribute mapping information (Offset and Length).
Response Message	(the response holds a copy of the command data)

Command and response layout:

	Command	Expected response	
	Command	**Expected response**	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	
Command	0004h	0004h	*PARAMETER_INPUT_MAP*
Data size	0014h	0014h	
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	-	-	
Message data word 1	Offset for Attribute 1	Offset for Attribute 1	
Message data word 2	Length for Attribute 1	Length for Attribute 1	
Message data word 3	Offset for Attribute 2	Offset for Attribute 2	
Message data word 4	Length for Attribute 2	Length for Attribute 2	
Message data word 5	Offset for Attribute 3	Offset for Attribute 3	
Message data word 6	Length for Attribute 3	Length for Attribute 3	
Message data word 7	Offset for Attribute 4	Offset for Attribute 4	
Message data word 8	Length for Attribute 4	Length for Attribute 4	
Message data word 9	Offset for Attribute 5	Offset for Attribute 5	
Message data word 10	Length for Attribute 5	Length for Attribute 5	

Note: The example above specifies mapping information for attributes 1... 5.

- **Offset for Attribute N**

  Specifies the offset from the start of the Parameter Data area.

- **Length for Attribute N**

  Specifies the number of bytes to map to the corresponding attribute.

Parameter Data Output Area Mapping (PARAMETER_OUTPUT_MAP)

This command maps blocks of Parameter Output Data to instance attributes in the Parameter Data Output Mapping Object. The blocks are mapped to consecutive attribute numbers starting with attribute #1. Specifying zero length for a block prevents the corresponding attribute number from being mapped. This makes it possible to for example map only attributes 1 and 10 by setting the length-value for attributes 2... 9 to zero.

It is possible to specify up to 50 attribute mappings using this command. It is only necessary to provide information up to the last mapped attribute (i.e. if data should only be mapped to attributes 1... 3, it is not necessary to include information associated with attribute mappings 4... 50).

Should the offset and/or length for a particular attribute mapping be invalid, these parameters will be set to zero in the response, and the corresponding attributes will not be mapped.

See also...

- 3-18 "Parameter Data Output Mapping Object, Class B1h"
- 4-7 "Parameter Data Input Area Mapping (PARAMETER_INPUT_MAP)"

Note: This command may only be issued during initialization.

Command Initiator	Application
Command Number	0005h
Extended Header data	-
Message Data	Attribute mapping information (Offset and Length).
Response Message	(the response holds a copy of the command data)

Command and response layout:

	Command	Expected response	
	Command	**Expected response**	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	
Command	0005h	0005h	*PARAMETER_OUTPUT_MAP*
Data size	0014h	0014h	
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	-	-	
Message data word 1	Offset for Attribute 1	Offset for Attribute 1	
Message data word 2	Length for Attribute 1	Length for Attribute 1	
Message data word 3	Offset for Attribute 2	Offset for Attribute 2	
Message data word 4	Length for Attribute 2	Length for Attribute 2	
Message data word 5	Offset for Attribute 3	Offset for Attribute 3	
Message data word 6	Length for Attribute 3	Length for Attribute 3	
Message data word 7	Offset for Attribute 4	Offset for Attribute 4	
Message data word 8	Length for Attribute 4	Length for Attribute 4	
Message data word 9	Offset for Attribute 5	Offset for Attribute 5	
Message data word 10	Length for Attribute 5	Length for Attribute 5	

Note: The example above specifies mapping information for attributes 1... 5.

- **Offset for Attribute N**

  Specifies the offset from the start of the Parameter Data area.

- **Length for Attribute N**

  Specifies the number of bytes to map to the corresponding attribute.

I/O Data Input Area Mapping (IO_INPUT_MAP)

This command maps blocks of Input I/O Data to instances in the Assembly Object. The blocks are mapped to consecutive instance numbers starting with instance #64h. Specifying zero length for a block prevents the corresponding instance number from being mapped. This makes it possible to for example map only instances 64h and 66h by setting the length-value for instance #65h.

It is possible to specify up to 6 instance mappings using this command. It is only necessary to provide information up to the last mapped instance (i.e. if data should only be mapped to instances 64h... 66h, it is not necessary to include information associated with instance mappings 67h... 69).

Should the offset and/or length for a particular instance mapping be invalid, these parameters will be set to zero in the response, and the corresponding instances will not be mapped.

Note: The mapped instances are also available as attributes in the I/O Input Mapping Object.

See also...

- 3-6 "Assembly Object, Class 04h"
- 3-13 "I/O Data Input Mapping Object, Class A0h"
- 4-13 "I/O Data Output Area Mapping (IO_OUTPUT_MAP)"

Note: This command may only be issued during initialization.

Command Initiator	Application
Command Number	0006h
Extended Header data	-
Message Data	Offset and length of the instance to map.
Response Message	(the response holds a copy of the command data)

Command and response layout:

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	
Command	0006h	0006h	*IO_INPUT_MAP*
Data size	0014h	0014h	
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	-	-	
Message data word 1	Offset for Instance 64h	Offset for Instance 64h	
Message data word 2	Length for Instance 64h	Length for Instance 64h	
Message data word 3	Offset for Instance 65h	Offset for Instance 65h	
Message data word 4	Length for Instance 65h	Length for Instance 65h	
Message data word 5	Offset for Instance 66h	Offset for Instance 66h	
Message data word 6	Length for Instance 66h	Length for Instance 66h	
Message data word 7	Offset for Instance 67h	Offset for Instance 67h	
Message data word 8	Length for Instance 67h	Length for Instance 67h	
Message data word 9	Offset for Instance 68h	Offset for Instance 68h	
Message data word 10	Length for Instance 68h	Length for Instance 68h	
Message data word 11	Offset for Instance 69h	Offset for Instance 69h	
Message data word 12	Length for Instance 69h	Length for Instance 69h	

- **Offset for Instance N**

  Specifies the offset from the start of the Input I/O Data area.

- **Length for Instance N**

  Specifies the number of bytes to map to the corresponding instance.

I/O Data Output Area Mapping (IO_OUTPUT_MAP)

This command maps blocks of Output I/O Data to instances in the Assembly Object. The blocks are mapped to consecutive instance numbers starting with instance #96h. Specifying zero length for a block prevents the corresponding instance number from being mapped. This makes it possible to for example map only instances 96h and 98h by setting the length-value for instance #97h.

It is possible to specify up to 6 instance mappings using this command. It is only necessary to provide information up to the last mapped instance (i.e. if data should only be mapped to instances 96... 97h, it is not necessary to include information associated with instance mappings 98h... 9Bh).

Should the offset and/or length for a particular instance mapping be invalid, these parameters will be set to zero in the response, and the corresponding instances will not be mapped.

Note: The mapped instances are also available as attributes in the I/O Output Mapping Object.

See also...

- 3-6 "Assembly Object, Class 04h"
- 3-14 "I/O Data Output Mapping Object, Class A1h"
- 4-11 "I/O Data Input Area Mapping (IO_INPUT_MAP)"

Note: This command may only be issued during initialization.

Command Initiator	Application
Command Number	0007h
Extended Header data	-
Message Data	Offset and length of the instance to map.
Response Message	(the response holds a copy of the command data)

Command and response layout:

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	
Command	0007h	0007h	*IO_OUTPUT_MAP*
Data size	0014h	0014h	
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	-	-	
Message data word 1	Offset for Instance 96h	Offset for Instance 96h	
Message data word 2	Length for Instance 96h	Length for Instance 96h	
Message data word 3	Offset for Instance 97h	Offset for Instance 97h	
Message data word 4	Length for Instance 97h	Length for Instance 97h	
Message data word 5	Offset for Instance 98h	Offset for Instance 98h	
Message data word 6	Length for Instance 98h	Length for Instance 98h	
Message data word 7	Offset for Instance 99h	Offset for Instance 99h	
Message data word 8	Length for Instance 99h	Length for Instance 99h	
Message data word 9	Offset for Instance 9Ah	Offset for Instance 9Ah	
Message data word 10	Length for Instance 9Ah	Length for Instance 9Ah	
Message data word 9	Offset for Instance 9Bh	Offset for Instance 9Bh	
Message data word 10	Length for Instance 9Bh	Length for Instance 9Bh	

- **Offset for Instance N**

  Specifies the offset from the start of the Output I/O Data area.

- **Length for Instance N**

  Specifies the number of bytes to map to the corresponding instance.

Other

Set Product Info (SET_PRODUCT_INFO)

This command makes it possible to specify custom DeviceNet identity information (i.e. the information in the Identity Object), causing the module to appear as a vendor specific implementation rather than a generic Anybus product. Note that the EDS-file needs to be adjusted accordingly.

Note 1: This command may only be issued during initialization.

Note 2: Vendor IDs must not be chosen arbitrarily and shall be obtained from ODVA international.

Note 3: This mailbox must be sent between ANYBUS_INIT and END_INIT.

Command Initiator	Application
Command Number	0002h
Extended Header data	-
Message Data	Vendor ID, Product Code, Product Name length and Product Name.
Response Message	(the response holds a copy of the command data)

Command and response layout:

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	
Command	0002h	0002h	*PRODUCT_INFO*
Data size	N	N	*Data size N*
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	-	-	
Message data byte 1	Vendor ID (msb)	Vendor ID (msb)	
Message data byte 2	Vendor ID (lsb)	Vendor ID (lsb)	
Message data byte 3	Product Code (msb)	Product Code (msb)	
Message data byte 4	Product Code (lsb)	Product Code (lsb)	
Message data byte 5	Product Name Length	Product Name Length	*Maximum 32 characters*
Message data byte 6	Product Name (1st char.)	Product Name (1st char.)	
Message data byte 7	Product Name (2nd char.)	Product Name (2nd char.)	
...	...	...	
Message data byte N	Product Name (last char.)	Product Name (last char.)	

Set Product Info All (SET_PRODUCT_INFO_ALL)

This command makes it possible to specify custom DeviceNet identity information (i.e. the information in the Identity Object), causing the module to appear as a vendor specific implementation rather than a generic Anybus product. Note that the EDS-file needs to be adjusted accordingly.

Note 1: This command may only be issued during initialization.

Note 2: Vendor IDs must not be chosen arbitrarily and shall be obtained from ODVA international.

Note 3: This mailbox must be sent between ANYBUS_INIT and END_INIT.

Command Initiator	Application
Command Number	0009h
Extended Header data	-
Message Data	Vendor ID, Device Type, Product Code, Major revision, Minor revision, Name length and Name
Response Message	(the response holds a copy of the command data)

Command and response layout:

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	
Command	0009h	0009h	*PRODUCT_INFO_ALL*
Data size	n	n	
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	-	-	
Message data byte 1	Vendor ID (msb)	Vendor ID (msb)	
Message data byte 2	Vendor ID (lsb)	Vendor ID (lsb)	
Message data byte 3	Device Type (msb)	Device Type (msb)	
Message data byte 4	Device Type (lsb)	Device Type (lsb)	
Message data byte 5	Product Code (msb)	Product Code (msb)	
Message data byte 6	Product Code (lsb)	Product Code (lsb)	
Message data byte 7	Major revision	Major revision	
Message data byte 8	Minor revision	Minor revision	
Message data byte 9	Product Name Length	Product Name Length	*Maximum 32 characters*
Message data byte 10	Product Name (1st char.)	Product Name (1st char.)	*Product name, 1st character*
Message data byte 11	Product Name (2nd char.)	Product Name (2nd char.)	
...	...	...	
Message data byte N	Product Name (last char.)	Product Name (last char.)	*Product name, last character*

Set Product Code (SET_PRODUCT_CODE)

IMPORTANT: *This command is included for backwards compatibility only. For new implementations, see 4-15 "Set Product Info (SET_PRODUCT_INFO)" or 4-16 "Set Product Info All (SET_PRODUCT_INFO_ALL)".*

This command makes it possible to specify custom Product Code (represented through the Identity Object), causing the module to appear as a unique product rather than a generic Anybus product. Note that the EDS-file needs to be adjusted accordingly.

Note 1: This command may only be issued during initialization.

Note 2: Product Codes must not be chosen arbitrarily. Contact HMS for further information.

Command Initiator	Application
Command Number	0003h
Extended Header data	-
Message Data	Product Code
Response Message	(the response holds a copy of the command data)

Command and response layout:

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	
Command	0003h	0003h	SET_PRODUCT_CODE
Data size	0001h	0001h	
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	-	-	
Message data byte 1	Product Code	Product Code	

Enable Quick Connect (ENABLE_QUICK_CONNECT)

This command enables the Quick Connect functionality in the module.

Note: This command may only be issued during initialization.

Command Initiator	Application
Command Number	000Ch
Extended Header data	-
Message Data	-
Response Message	-

Command and response layout:

	Command	Expected response
Message ID	(ID)	(ID)
Message information	4002h	0002h
Command	000Ch	000Ch
Data size	0001h	0001h
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	-	-

Get ID Reset Parameter (GET_ID_RESET_PARAM)

This command can be used to retrieve the type of reset that has been received from the network, and should only be issued as a response to an Event Notification (RST-bit).

See also...

- 3-3 "Service Details: Reset"
- Anybus-S/M Parallel Design Guide ('Event Notification')

Command Initiator	Application
Command Number	0095h
Extended Header data	-
Message Data	-
Response Message	Reset Type

Command and response layout:

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	
Command	0095h	0095h	GET_ID_RESET_PARAM
Data size	0000h	0001h	
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	-	-	
		Reset Type	Message data byte 1

- **Reset Type**

    00h: Power on reset
    01h: Out of box configuration
    FFh: (no reset request received from the network).

Disable UCMM functionality (DISABLE_UCMM)

This command disables UCMM functionality so the module becomes a Group 2 Only slave. The mailbox must be sent before END_INIT. The command has been added to enable initiation of the module to behave as the previous version of the ABS-DEV.

Command Initiator	Application
Command Number	000Dh
Extended Header data	-
Message Data	-
Response Message	-

Command and response layout:

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	
Command	000Dh	000Dh	*DISABLE_UCMM*
Data size	0000h	0000h	
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	-	-	

Enable data consistency (ENABLE_DATA_CONSISTENCY)

This command will make the module ignore the Bit Strobe bit if Polled and Bit Strobed connections are run simultaneously to the same data area. If this mailbox is not sent, they will both write to the same area, creating data inconsistence. The mailbox message has to be sent before END_INIT

This is not enabled by default sinec some old Omron PLC:s implements an incorrect way to auto-configure the connections. This version of the ABS-DEV will behave as version 1.X and have data inconsistency between the Bit Strobe bit and the first byte in a Polled connection to retain backwards compatibility.

Command Initiator	Application
Command Number	000Eh
Extended Header data	-
Message Data	-
Response Message	-

Command and response layout:

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	
Command	000Eh	000Eh	ENABLE_DATA_CONSISTENCY
Data size	0000h	0001h	
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	-	-	

Set Serial Number (SET_SERIAL_NUMBER)

This command is used to change the serial number of the module.

Command Initiator	Application
Command Number	0081h
Extended Header data	-
Message Data	Serial number
Response Message	(the response holds a copy of the command data)

Command and response layout:

	Command	Expected response	
Message ID	(ID)	(ID)	
Message information	4002h	0002h	
Command	0081h	0081h	*SET_SERIAL_NUMBER*
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	-	-	
Message data byte 1	Serial number (msb)	Serial number (msb)	
Message data byte 2	Serial number	Serial number	
Message data byte 3	Serial number	Serial number	
Message data byte 4	Serial number (lsb)	Serial number (lsb)	

Fieldbus Specific Area

Memory Map

Address	Register
640h... 641h	Identity Status Attribute
642h	Explicit Connection Status
643h	Polled I/O Connection Status
644h	Strobed Connection Status
645h	COS / Cyclic Connection Status
646h	(reserved)
647h	Master Status
648h - 7BFh	(reserved, do not use)

- **Identity Status**

  This register reflects attribute #5 of the Identity Object.

  See also 3-3 "Device Status".

- **Explicit Connection Status**

  This register reflects attribute #1 in instance #1 of the Connection Object.

  See also 3-10 "Instance 1 (Explicit Messaging Connection) Attributes".

- **Polled I/O Connection Status**

  This register reflects attribute #1 in instance #2 of the Connection Object.

  See also 3-10 "Instance 2 (Polled Connection) Attributes".

- **Bit Strobe Connection Status**

  This register reflects attribute #1 in instance #3 of the Connection Object.

  See also 3-10 "Instance 3 (Bit-strobe connection) Attributes".

- **COS/Cyclic Connection Status**

  This register reflects attribute #1 in instance #4 of the Connection Object.

  See also 3-11 "Instance 4 (COS/Cyclic connection) Attributes".

- **Master Status**

  This register indicates the current master/scanner state.

  00h: (no information available)
  01h: Master in RUN state
  02h: Master in IDLE state

  Note: To receive information about the master state, the module must receive I/O data from the master.

Miscellaneous

Control Register Area

Fieldbus Type

The fieldbus type value for this product is 0025h (DeviceNet).

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 module will reset the CAN-controller and go off line. The Module Status LED will indicate a major fault.

Module Status Register

- **Fieldbus Run/Stop (FBRS)**

  This bit is set as soon as the DUP MAC check has succeeded and the module has gone online, and cleared in case one or more connections have timed out.

  See also...
  - Anybus-S/M Parallel Design ('Module Status')

Event Notification Cause/Source Registers

- **Network Reset Functionality (RST)**

  The module supports network reset requests.

  See also...
  - 3-3 "Service Details: Reset"
  - 4-19 "Get ID Reset Parameter (GET_ID_RESET_PARAM)"

Connectors

Application Interface

(Consult the Anybus-S/M Parallel Interface Design Guide for further information).

DeviceNet Connector

The module supports both 5.08mm and 3.84mm pluggable screw connectors, as well as a 10 pin 2mm board to board connector.

Pluggable Screw (5.08 and 3.84 mm)

Pin	Signal	Description
1	V-	Negative supply voltage[a]
2	CAN_L	CAN_L bus line
3	SHIELD	Cable shield
4	CAN_H	CAN_H bus line
5	V+	Positive supply voltage[a]
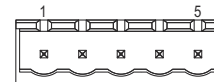
a. The module requires 24VDC bus power.
   See also Appendix D-1 "Supply Voltage"

2mm Board to Board

| Pin	Signal	Description
1	SHIELD	Cable shield
2	-	-
3	CAN_L	CAN_L bus line
4	V-	Negative supply voltage[a]
5	CAN_H	CAN_H bus line
6	V-	Negative supply voltage[a]
7	-	-
8	V+	Positive supply voltage[a]
9	-	-
10	SHIELD	Cable shield

a. The module requires 24VDC bus power.
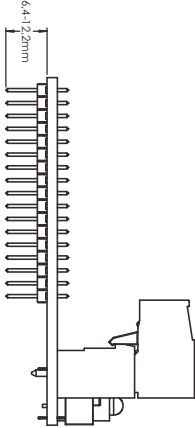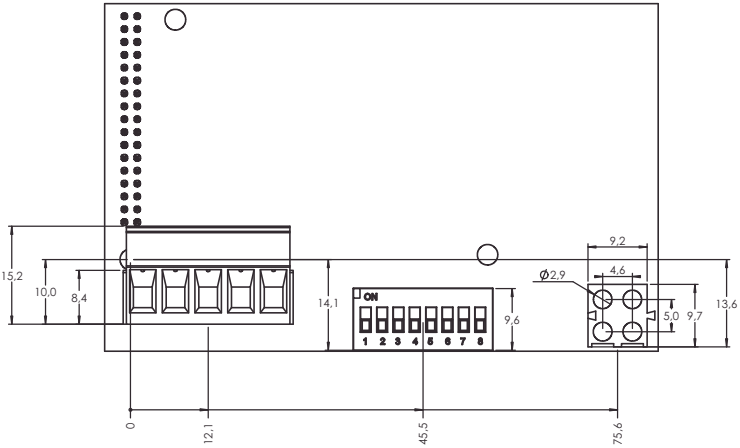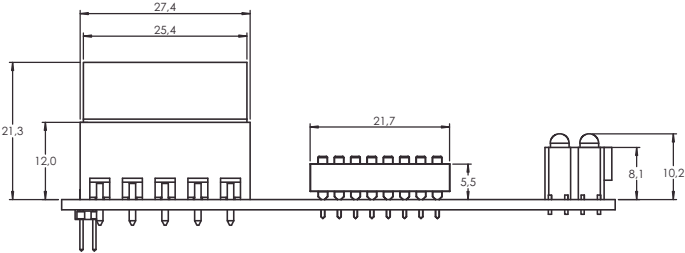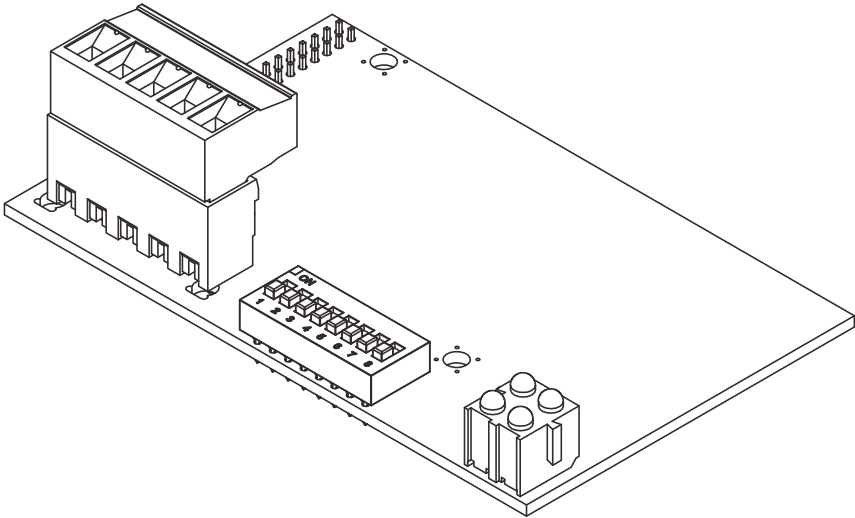   See also Appendix D-1 "Supply Voltage"

Mechanical Specification

All measurements expressed in this chapter have a tolerance of ±0.25mm unless otherwise stated.

Measurements, Connectors & Switches

Angled Switches & Connectors
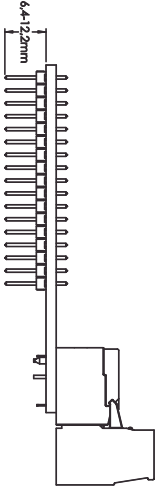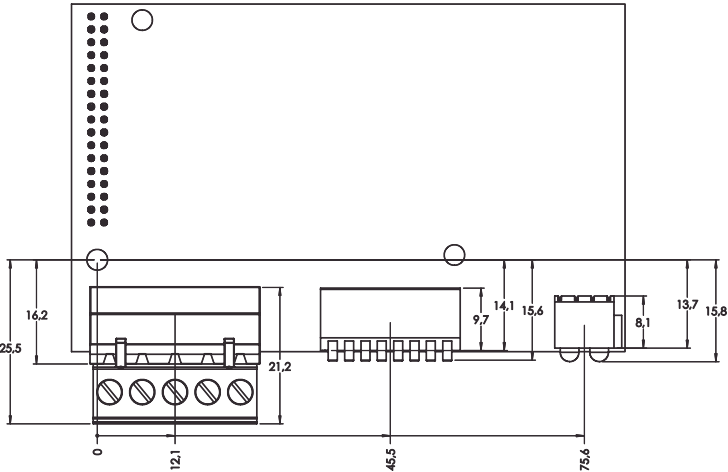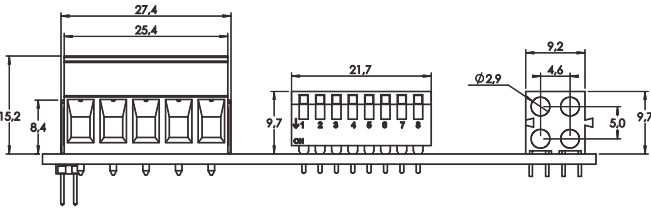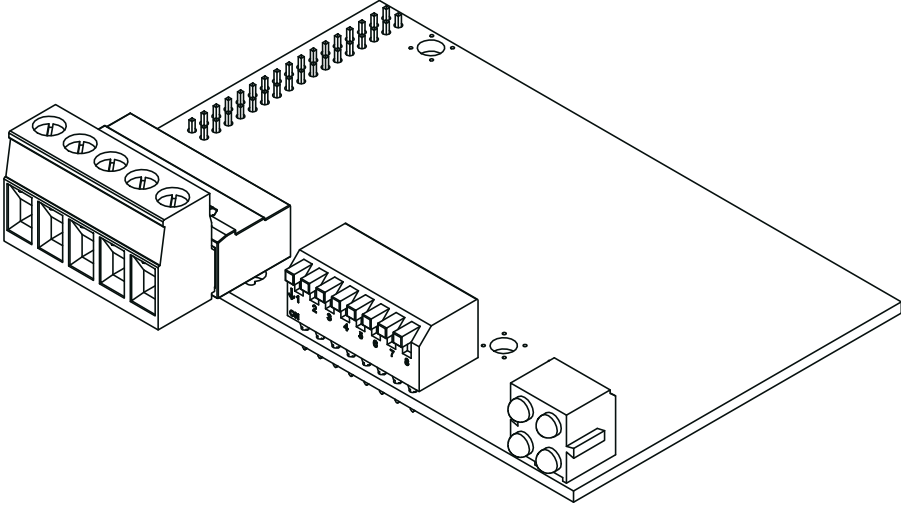
Straight Switches & Connectors

Technical Specification

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.

Supply Voltage

The module electronics requires a regulated 5VDC power supply as specified by the Anybus-S Parallel Design Guide. In addition, the bus interface requires 11... 25VDC power source as stated in the Device-Net specification.

Power Consumption

The maximum current consumption for this product is 130mA on the application interface, and 30mA on the bus interface.

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

Quick Connect

The following steps needs to be taken to enable quick connect:

1. In the Electronic Data Sheet (EDS) file add the following line under the [Device] section.

```
[File]
        DescText   = "ABS Standard Configuration";
        CreateDate = 08-12-1998;
        CreateTime = 10:31:30;
        ModDate    = 10-17-2007;
        ModTime    = 08:40:00;
        Revision   = 2.2;                        $ Revision of EDS file

[Device]
        VendCode   = 90;                         $ Vendor Code
        VendName   = "HMS Networks";
        ProdType   = 12;                         $ Product Type
        ProdTypeStr = "Communications Adapter";
        ProdCode   = 12;                         $ Product Code
        MajRev     = 2;                          $ Major Rev
        MinRev     = 1;                          $ Minor Rev
        ProdName   = "Anybus-S DeviceNet";
        Catalog    = "Anybus-S DeviceNet";

        DNetQC =
              0x0001,           $ Quick Connect supported at Powerup
              250;              $ 250 ms Powerup time


$ Parameter Class Section

[ParamClass]
        MaxInst = 29;                   $ Max Instances - total # configuration parameters
        Descriptor = 0x0000;            $ Parameter Class Descriptor - No parameters
        CfgAssembly = 0;                $ The config assembly is not supported.
```

2. Send the ENABLE_QUICK_CONNECT mailbox command to the module during initialization.