Anybus® X-gateway Network Interface Addendum

DeviceNet™ Adapter

Doc: HMSI-27-243
Rev: 2.10

Connecting Devices™

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

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

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

Important User Information

This document is intended to provide a good understanding of the functionality offered by the network interface described here.

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

Liability

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

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

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

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

Intellectual Property Rights

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

Trademark Acknowledgements

Anybus® is a registered trademark of HMS Industrial Networks AB. Microsoft® and Windows® are registered trademarks of Microsoft, Inc. ODVA™ and DeviceNet™ are trademarks of ODVA, Inc. All other trademarks are the property of their respective holders.

!

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

ESD Note: This product contains ESD (Electrostatic Discharge) sensitive parts that may be damaged if ESD control procedures are not followed. Static control precautions are required when handling the product. Failure to observe this may cause damage to the product.

Anybus X-gateway Network Interface Addendum
DeviceNet Adapter

HMSI-27-243

Table of Contents

P. About This Document

P.1 How To Use This Document

This document describes network specific features and procedures needed when operating the DeviceNet Adapter Interface for the Anybus X-gateway. For general information and operating instructions for the Anybus X-gateway, consult the Anybus X-gateway User Manual.

The reader of this document is expected to be familiar with DeviceNet networking technology, and communication systems in general.

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

P.2 Related Documents

Document	Author
Anybus X-gateway User Manual	HMS
Anybus-S DeviceNet Fieldbus Appendix	HMS
Anybus X-gateway DeviceNet Adapter Interface Network Installation Sheet	HMS
Common Industrial Protocol (CIP) specification	ODVA
DeviceNet Adaptation of CIP	ODVA

P.3 Document History

Summary of Recent Changes (2.00... 2.10)

Change	Page(s)
Added trademark acknowledgments	2
Modified list of related documents	5
Added connector assignments, replaced graphics	8
Added description of fieldbus-specific ACM settings	10
Added compliance info	31

Revision List

Revision	Date	Author	Chapter	Description
1.00	2004-04-02	PeP	All	First release
1.10	2007-11-19	PeP	All	Major rewrite
1.11	2007-11-21	PeP	4	Minor update
2.00	May 2014	SDa	Multiple	New hardware and Anybus Configuration Manager
2.01		SDa		Minor corrections
2.10	March 2015	ThN	All	Misc. corrections and additions

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 'X-gateway' refers to the Anybus X-gateway.
- The term 'Interface' refers to the DeviceNet Adapter interface for the Anybus X-gateway.
- The term 'user manual' refers to the Anybus X-gateway User Manual.
- Hexadecimal values are written in the format NNNNh, where NNNN is the hexadecimal value.
- 16/32 bit values are generally stored in Motorola (big endian) format unless otherwise stated.

P.5 Support

For general contact information and global support, please visit www.anybus.com.

1. About the DeviceNet Adapter Interface

1.1 General Description

The DeviceNet Adapter Interface for the Anybus X-gateway implements a DeviceNet communications adapter (profile no. 12). The interface acts as an adapter, which means it can be accessed by a DeviceNet scanner, but it will not initiate communication by itself.

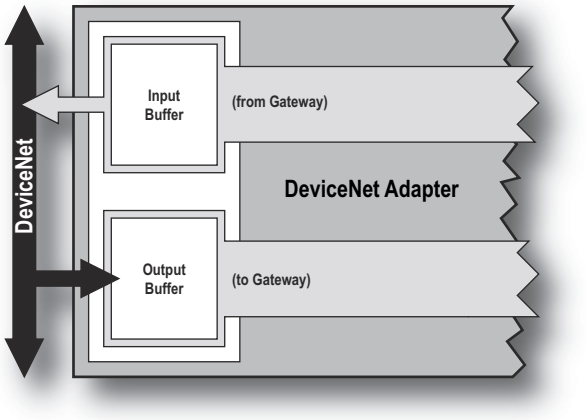
The interface exchanges data through two buffers as follows:

- **Input Buffer**

  This buffer holds data forwarded *from* the other network, i.e. data which can be read by the DeviceNet scanner.

- **Output Buffer**

  This buffer is forwarded *to* the other network, i.e. data which can be written by the DeviceNet scanner.



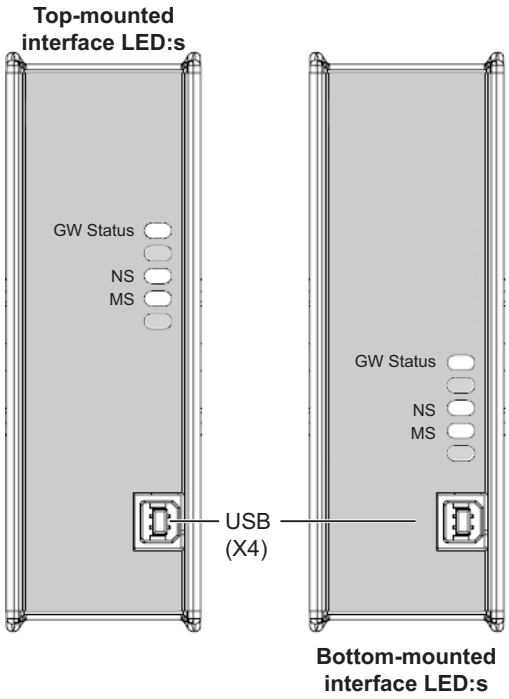
1.2 Features

- Galvanically isolated bus electronics
- 125 kbit, 250 kbit and 500 kbit operation
- On-board configuration switches
- Up to 512 byte of I/O in each direction
- Explicit Messaging (up to 512 bytes in each direction)
- Polled I/O
- Bit-strobed I/O
- Change-of-state / Cyclic I/O

1.3 External View

1.3.1 DeviceNet Status LED:s

LED	Colour	Indication
GW Status	See the X-gateway User Manual	
NS	Off	No power or not online
	Green	Link OK, online, connected
	Green, flashing	Online, not connected
	Red	Critical link failure
	Red, flashing	Connection timeout
MS	Off	No power
	Green	Operational
	Green, flashing	Data size error
	Red	Unrecoverable fault
	Red, flashing	Minor fault



1.3.2 Connectors and Switches

DeviceNet Connector (X1)

See "DeviceNet Connector Pinout" on page 31.

Gateway Power Connector (X3)

See the X-gateway User Manual.

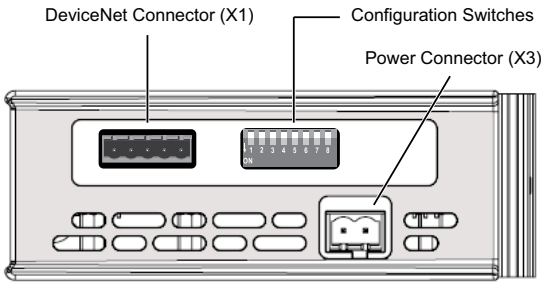
USB Gateway Config Connector (X4)

See the X-gateway User Manual.

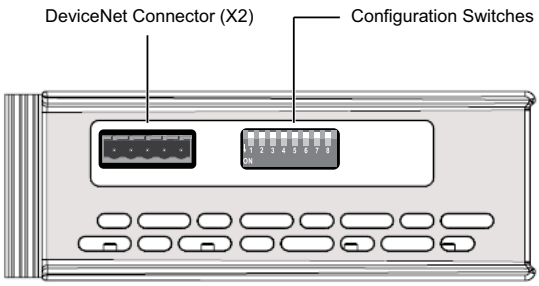
Configuration Switches

See "Configuration Switches" on page 9.

Top-mounted Interface



Bottom-mounted Interface



2. Installation and Configuration

2.1 Configuration Switches

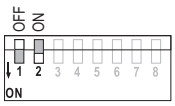
2.1.1 Baudrate Configuration

Switches 1 and 2 specifies the operating baudrate for the Adapter Interface as follows:

Switch 1	Switch 2	Switch 3	Switch 4	Switch 5	Switch 6	Switch 7	Switch 8	Baudrate
OFF	OFF	-	-	-	-	-	-	125 kbps
OFF	ON	-	-	-	-	-	-	250 kbps
ON	OFF	-	-	-	-	-	-	500 kbps
ON	ON	-	-	-	-	-	-	(reserved)

Example:

Baudrate set to 250 kbps.



Note: The orientation of the switches follows that of the Adapter Interface (top or bottom).

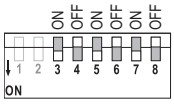
2.1.2 MAC ID Configuration

Switches 3 to 8 specifies the MAC ID for the Adapter Interface in binary format as follows:

Switch 1	Switch 2	Switch 3	Switch 4	Switch 5	Switch 6	Switch 7	Switch 8	Mac ID
-	-	OFF	OFF	OFF	OFF	OFF	OFF	0
-	-	OFF	OFF	OFF	OFF	OFF	ON	1
-	-	OFF	OFF	OFF	OFF	ON	OFF	2
...	...	...	...	...	...	...	...	...
-	-	ON	ON	ON	ON	OFF	ON	61
-	-	ON	ON	ON	ON	ON	OFF	62
-	-	ON	ON	ON	ON	ON	ON	63

Example:

MAC ID set to 42.

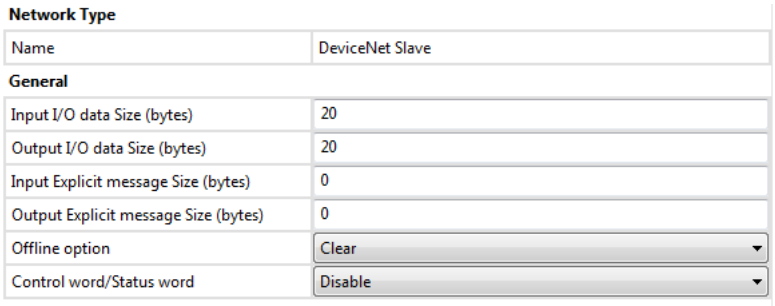


Note: The orientation of the switches follows that of the Adapter Interface (top or bottom).

2.2 Gateway Config Interface

The X-gateway and the DeviceNet slave interface may be configured by using the software tool **Anybus Configuration Manager** (ACM), which is available from www.anybus.com/support

When ACM is connected to the gateway via the USB configuration connector, the following settings are available:



Setting	Description
Input I/O data Size (bytes)	Specifies the amount of Input I/O Data to exchange on DeviceNet. This data is represented through the Assembly Object and the I/O Data Input Mapping Object.
Output I/O data Size (bytes)	Specifies the amount of Output I/O Data to exchange on DeviceNet. This data is represented through the Assembly Object and the I/O Data Output Mapping Object.
Input Explicit message Size (bytes)	Specifies the amount of Explicit Input Data. This data is represented through the Parameter Data Input Mapping Object.
Output Explicit message Size (bytes)	Specifies the amount of Explicit Output Data. This data is represented through the Parameter Data Output Mapping Object.
Offline option	Determines what action to perform if the network goes offline. The gateway can either freeze (keep the current value) or clear (set to zero) the data from the offline network.
Control word/Status word	Enables/disables representation of the Control/Status word on DeviceNet.

See also:

- The Anybus X-gateway User Manual, for full details on using ACM.
- The online help in ACM, for further help on the available settings.
- "Assembly Object, Class 04h" on page 20
- "I/O Data Input Mapping Object, Class A0h" on page 26
- "I/O Data Output Mapping Object, Class A1h" on page 27
- "Parameter Data Input Mapping Object, Class B0h" on page 29
- "Parameter Data Output Mapping Object, Class B1h" on page 30

2.3 Network Configuration (RSNetWorx)

The following example describes how to include the Adapter Interface in a DeviceNet network using RSNetWorx.(The procedure used for other network configuration tools is similar, although individual steps may be slightly different.)

2.3.1 Step 1: Install the .EDS-file

Each device in a DeviceNet network is associated with an .EDS-file[1]. This file contains information about the device and is used by the network configuration tool, in this case RSNetWorx.

1. Start RSNetWorx for DeviceNet.
2. To add the .EDS-file for the Adapter interface, select 'EDS Wizard' from the 'Tools' menu.
3. Click 'Next'
4. Make sure 'Register an EDS file(s)' is selected
5. Click 'Next'
6. Make sure 'Register single file' is selected
7. Type the path and filename of the EDS file, or click 'Browse' to select it using the file selector.
8. Click 'Next'. (RSNetWorx will now parse and test the EDS file and display the result.)
9. Click 'Next'. (Optionally, an icon to be used with the adapter can be selected now.)
10. Click 'Next'. (A summary of the EDS process is displayed.)
11. Click 'Next'. (The Wizard will indicate that the installation process is finished)
12. Click 'Finish'. The .EDS-file is now included in the DeviceNet configuration tool (RSNetWorx).

2.3.2 Step 2: Scan the network for new devices

In order to be able to add the Adapter Interface to a Scanner's scanlist, it must first be properly detected by the network configuration tool (in this case RSNetWorx).

1. From the 'Network' menu, select 'Online'.
   This will scan the entire network and find any attached nodes.
2. When done, all devices found on the network will be presented graphically.

The Adapter Interface is now included in the DeviceNet configuration tool (RSNetWorx).

1. The latest version of the .EDS-file for this product can be obtained from www.anybus.com.

2.3.3 Step 3: Adding the Adapter Interface to the Scanners scanlist

To exchange data with the Adapter Interface, it must first be included in the Scanlist of a Scanner.

1. Double-click on the desired Scanner.

   This will bring up a window with the configuration of the Scanner.
2. Select the 'Scanlist-'tab.

   You will be prompted whether to download the offline configuration to the Scanner, or upload the Scanner's current configuration.
3. Click 'Upload'.

   When done, a window will appear, showing which devices that are included in the Scanner's scanlist and which ones that are not.
4. Highlight 'Anybus-S DeviceNet' in the list of available devices
5. Click on the right-hand arrow ('>').

The Adapter Interface is now included in the Scanner's scanlist.

2.3.4 Step 4: Configuring I/O Sizes

The I/O sizes for the interface must be configured in the Scanner.

1. From the 'Scanlist'-window, select 'Edit I/O Parameters'. A window will appear, containing the I/O configuration for the Adapter interface.
2. Select 'Polled' by checking it's checkbox.
3. The 'Input' and 'Output' data sizes must be set to match the values specified via the Gateway Config interface. Click 'OK' when done.

   Note: A warning message will appear, indicating that the configured size does not match the settings in the .EDS-file. This is perfectly normal, as the data sizes of the AnyBus-X DeviceNet Adapter interface can be configured freely. Just click 'Yes' to continue.
4. You will now be prompted on whether or not the I/O data should be mapped automatically. Click 'Yes'.
5. Click 'OK'

The configuration will now be downloaded to the Scanner. Once the Scanner is in Run mode, data will be exchanged.

Note: In case of problems, verify that the I/O configuration specified in RSNetWorx matches that of the gateway. Also make sure that the baudrate matches that of the DeviceNet network, and that the MAC ID of the Adapter Interface doesn't conflict with another device.

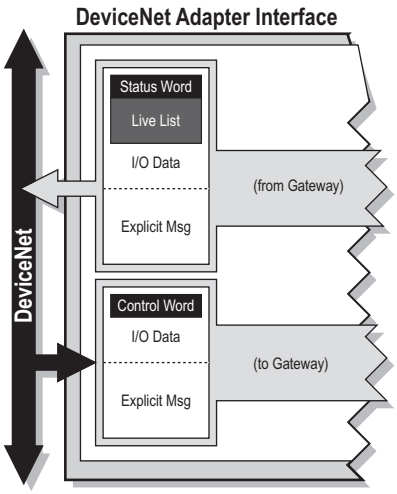
3. Data exchange

3.1 General Information

The Adapter Interface exchanges up to 512 bytes of data in each direction.

This can be I/O Data and/or Explicit Message data:

- **I/O Data (Up to 512 bytes)**
  This data can be accessed by means of I/O connections towards the Assembly Object. It is also represented through the I/O Data Input/Output Mapping Objects.

  See also...
  - "Assembly Object, Class 04h" on page 20
  - "I/O Data Input Mapping Object, Class A0h" on page 26
  - "I/O Data Output Mapping Object, Class A1h" on page 27

- **Explicit Message data (Up to 512 bytes)**
  This data can be accessed acyclically by means of Explicit Messaging.

  See also...
  - "Parameter Data Input Mapping Object, Class B0h" on page 29
  - "Parameter Data Output Mapping Object, Class B1h" on page 30

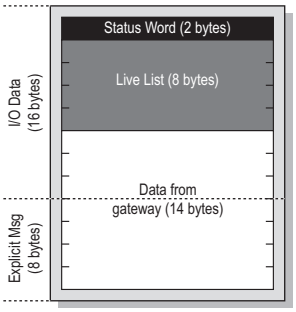
The amount of data that is exchanged as I/O Data and Explicit Message data respectively is specified via the Gateway Config Interface.

See also...

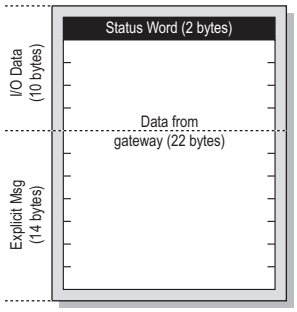
- "Gateway Config Interface" on page 10

3.1.1 Input Data (Gateway to DeviceNet)

Depending on the actual gateway configuration and how it has been set up to operate, parts of the data produced by the DeviceNet Adapter Interface may be used to represent status information (i.e. Status Word, Live List etc.).



Example A:
I/O Data Size = 16
Explicit Message Size = 8
Live List = Enabled
Control & Status Word = Enabled

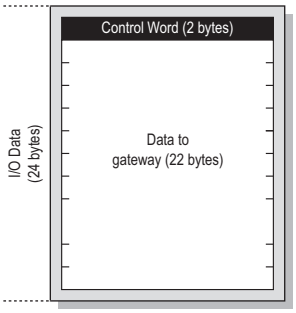


Example B:
I/O Data Size = 10
Explicit Message Size = 14
Live List = Disabled
Control & Status Word = Enabled

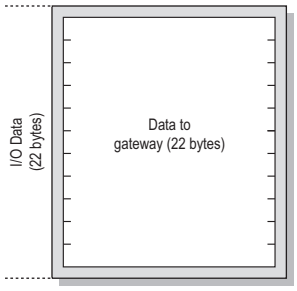
Note: The Live List is only available on master-slave gateway versions.

3.1.2 Output Data (DeviceNet to Gateway)

Depending on the actual gateway configuration and how it has been set up to operate, the first two bytes consumed by the DeviceNet Adapter Interface may be interpreted as control information (i.e. the Control Word).



Example A:
I/O Data Size = 18
Explicit Message Size = 6
Control Word = Enabled



Example B:
I/O Data Size = 6
Explicit Message Size = 16
Control Word = Disabled

4. CIP Object Implementation

4.1 General Information

The DeviceNet Adapter Interface 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

4.2 Identity Object, Class 01h

4.2.1 General Information

Object Description

-

Implemented Services

Class services: Get Attribute Single

Instance services: Get Attribute Single
Reset (see "Service Details: Reset" on page 17)

4.2.2 Class Attributes

#	Access	Name	Type	Value
1	Get	Revision	UINT	0001h

4.2.3 Instance Attributes

#	Access	Name	Type	Value
1	Get	Vendor ID	UINT	005Ah (HMS Industrial Networks AB)
2	Get	Device Type	UINT	000Ch (Communications Adapter)
3	Get	Product Code	UINT	000Ch (Anybus-S DeviceNet)
4	Get	Revision	Struct of: USINT, USINT	 (Major fieldbus version) (Minor fieldbus version)
5	Get	Status	WORD	(see 4-17 "Device Status")
6	Get	Serial Number	UDINT	(assigned at manufacturing)
7	Get	Product Name	SHORT_STRING	'Anybus-S DeviceNet'

4.2.4 Device Status

bit(s)	Name
0	Module Owned (A master/scanner has allocated the Adapter Interface)
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)

4.2.5 Service Details: Reset

There are two types of network reset requests on DeviceNet:

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

  This causes the DeviceNet interface to restart its internal DeviceNet software layer. The overall operation of the gateway remains unaffected, i.e. the gateway will neither reset itself nor the other network interface.

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

  This causes the DeviceNet interface to revert to an 'out of box' configuration and restart its internal DeviceNet software layer. The overall operation of the gateway remains unaffected, i.e. the gateway will neither reset itself nor the other network interface.

4.3 Message Router, Class 02h

4.3.1 General Information

Object Description

-

Supported Services

Class services: -

Instance services: -

4.3.2 Class Attributes

-

4.3.3 Instance Attributes

-

4.4 DeviceNet Object, Class 03h

4.4.1 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)

4.4.2 Class Attributes

#	Name	Access	Type	Value
1	Revision	Get	UINT	0002h

4.4.3 Instance #1 Attributes

#	Name	Access	Type	Value
1	MAC ID	Get	USINT	(actual MAC ID)
2	Baud Rate	Get	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)

4.5 Assembly Object, Class 04h

4.5.1 General Information

Object Description

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

See also...

- "I/O Data Input Mapping Object, Class A0h" on page 26
- "I/O Data Output Mapping Object, Class A1h" on page 27

Services

Class services:	Get Attribute Single
Instance services:	Get Attribute Single Set Attribute Single

4.5.2 Class Attributes

#	Access	Name	Type	Value
1	Get	Revision	UINT	0001h

4.5.3 Instance Attributes - Instance/Connection Point 64h

This instance is used to access Input I/O Data.

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

4.5.4 Instance Attributes - Instance/Connection Point 96h

This instance is used to access Output I/O Data.

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

4.6 Connection Object, Class 05h

4.6.1 General Information

Object Description

-

Implemented Services

Class services: Get Attribute Single

Instance services: Get Attribute Single
Set Attribute Single

4.6.2 Class Attributes

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

a. Value saved in non-volatile memory.

4.6.3 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)[a]	Expected Packet Rate	UINT	0
12	Get (Set)	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 (Set)[b]	Produced Connection Path	EPATH	-
15	Get	Consumed Connection path length	UINT	0004h
16	Get (Set)[b]	Consumed Connection Path	EPATH	20 2B 24 01h
17	Get	Production Inhibit Time	UINT	0
18	Get (Set)	Connection timeout multiplier	USINT	Default: 0 (x4)

a. Only settable for instance #1
b. Saved in non-volatile memory

4.6.4 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)

4.6.5 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)

4.6.6 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)

4.6.7 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

4.6.8 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)

4.7 Acknowledge Handler Object, Class 2Bh

4.7.1 General Information

Object Description

-

Implemented Services

Class services: Get Attribute Single

Instance services: Get Attribute Single
Set Attribute Single

4.7.2 Class Attributes

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

4.7.3 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.

4.8 I/O Data Input Mapping Object, Class A0h

4.8.1 General Information

Object Description

This object can be used to access Input I/O Data.

The data is also available through instance 64h in the Assembly Object.

See also...

- "Assembly Object, Class 04h" on page 20
- "I/O Data Output Mapping Object, Class A1h" on page 27
- "Parameter Data Output Mapping Object, Class B1h" on page 30

Supported Services

Class services: Get Attribute All

Instance services: Get Attribute Single

4.8.2 Class Attributes

#	Access	Name	Type	Value
1	Get	Data	UINT	0001h

4.8.3 Instance Attributes, Instance 01h

#	Access	Name	Type	Description
1[a]	Get	Data	Array of USINT	Input I/O data (also available as Assembly Instance 64h)

a. This attribute is only available if the Input I/O Data size is larger than 0 (zero)

See also...

- "Instance Attributes - Instance/Connection Point 64h" on page 20

4.9 I/O Data Output Mapping Object, Class A1h

4.9.1 General Information

Object Description

This object can be used to access Output I/O Data.

The data is also available through instance 96h in the Assembly Object.

See also...

- "Assembly Object, Class 04h" on page 20
- "I/O Data Input Mapping Object, Class A0h" on page 26
- "Parameter Data Input Mapping Object, Class B0h" on page 29

Supported Services

Class services:	Get Attribute All
Instance services:	Get Attribute Single Set Attribute Single

4.9.2 Class Attributes

#	Access	Name	Type	Value
1	Get	Data	UINT	0001h

4.9.3 Instance Attributes, Instance 01h

#	Access	Name	Type	Description
1[a]	Get/Set	Data	Array of USINT	Output I/O data (also available in Assembly Instance 96h)

a. This attribute is only available if the Output I/O Data size is larger than 0 (zero)

See also...

- "Instance Attributes - Instance/Connection Point 96h" on page 20

4.10 Diagnostic Object, Class AAh

4.10.1 General Information

Object Description

This vendor specific object provides access to misc. diagnostic information.

Implemented Services

Class services: Get Attribute All

Instance services: Get Attribute Single

4.10.2 Class Attributes

#	Access	Name	Type	Value
1	Get	Revision	UINT	0001h

4.10.3 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 Buffer 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 Output Buffer in bytes)

4.11 Parameter Data Input Mapping Object, Class B0h

4.11.1 General Information

Object Description

This object is used to access Input Explicit Message data.

See also...

- "Assembly Object, Class 04h" on page 20
- "I/O Data Input Mapping Object, Class A0h" on page 26
- "Parameter Data Output Mapping Object, Class B1h" on page 30

Supported Services

Class services: Get Attribute All

Instance services: Get Attribute Single

4.11.2 Class Attributes

#	Access	Name	Type	Value
1	Get	Revision	UINT	0001h

4.11.3 Instance Attributes, Instance 01h

Each attribute carries up to 255 bytes of data.

#	Access	Name	Type	Description
01h	Get	Data	Array of USINT	Input Explicit Message data, bytes 0... 254
02h	Get	Data	Array of USINT	Input Explicit Message data, bytes 255... 509
03h	Get	Data	Array of USINT	Input Explicit Message data, bytes 510... 511

Note: The interface only creates enough attributes to hold the specified amount of Input Explicit Message data (i.e. if using 400 bytes, the interface creates attributes #1 (255 bytes) and #2 (145 bytes).

4.12 Parameter Data Output Mapping Object, Class B1h

4.12.1 General Information

Object Description

This object is used to access Output Explicit Message data.

See also...

- "Assembly Object, Class 04h" on page 20
- "I/O Data Output Mapping Object, Class A1h" on page 27
- "Parameter Data Input Mapping Object, Class B0h" on page 29

Supported Services

Class services:	Get Attribute All
Instance services:	Get Attribute Single Set Attribute Single

4.12.2 Class Attributes

#	Access	Name	Type	Value
1	Get	Revision	UINT	0001h

4.12.3 Instance Attributes, Instance 01h

Each attribute carries up to 255 bytes of data.

#	Access	Name	Type	Description
01h	Get/Set	Data	Array of USINT	Output Explicit Message data, bytes 0... 254
02h	Get/Set	Data	Array of USINT	Output Explicit Message data, bytes 255... 509
03h	Get/Set	Data	Array of USINT	Output Explicit Message data, bytes 510... 511

Note: The interface only creates enough attributes to hold the specified amount of Output Explicit Message data (i.e. if using 400 bytes, the interface creates attributes #1 (255 bytes) and #2 (145 bytes).4.9

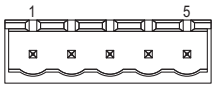
A. Technical Specification

A.1 DeviceNet Interface Details

- Galvanically isolated bus electronics
- 125 kbit, 250 kbit and 500 kbit operation
- On-board configuration switches
- Up to 512 bytes of I/O in each direction
- Explicit Messaging (up to 512 bytes in each direction)
- Polled I/O
- Bit-strobed I/O
- Change-of-state / Cyclic I/O

A.2 DeviceNet Connector Pinout

#	Signal	Description
1	V-	Negative bus power supply
2	CAN_L	CAN low
3	SHIELD	Cable shielding
4	CAN_H	CAN high
5	V+	Positive bus power supply



A.3 Compliance Information

CIP Product Compliance