## Fieldbus Appendix

# AnyBus-IC DeviceNet

**Rev. 1.20**


**HMS Industrial Networks AB**

| | ☎ | ✉ |
|---|---|---|
| Germany | +49- 721 - 96472 - 0 | sales-ge@hms-networks.com |
| Japan | +81- 45 - 478 -5340 | sales-jp@hms-networks.com |
| Sweden | +46- 35 - 17 29 20 | sales@hms-networks.com |
| U.S.A | +1- 773 - 404 - 2271 | sales-us@hms-networks.com |

# Table of Contents

## Chapter 6 Fieldbus Specific Parameters



## Chapter 7 Application Parameters



## Appendix A Electrical Characteristics



## Appendix B Mechanical Specification

# About This Manual

## How To Use This Manual

This manual provides an overview of the ABIC-DEV-1. It describes how to configure and operate the module, and provides examples showing how to use its features when communicating over DeviceNet.

This document is intended as a complement to the AnyBus-IC Design Guide. For general information about the AnyBus-IC platform, consult the general AnyBus-IC Design Guide.

The reader of this document is expected to be familiar with hardware and software design as well as to have basic knowledge in the DeviceNet fieldbus, and communication systems in general.

## 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.

## Related Documentation

| Document | Author |
|---|---|
| AnyBus-IC Design Guide | HMS |
| AnyBus-IC Profibus Fieldbus Appendix | HMS |
| AnyBus-IC Ethenet/IT Fieldbus Appendix | HMS |
| AnyBus-IC Ethenet/IP Fieldbus Appendix | HMS |

## Revision List

| Revision | Date | Author | Chapter(s) | Description |
|---|---|---|---|---|
| 1.00 | 2002-08-30 | PeP | All | First version |
| 1.10 | 2002-11-28 | PeP | All | Major update |
| 1.20 | 2003-07-02 | PeP | All<br>5 | Various minor adjustments<br>Rewritten |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

# Conventions Used In This Manual

The following conventions are used throughout this manual:

- Numbered lists provide sequential steps
- Bulleted lists provide information, not procedural steps
- The term 'module' is used when referring to the ABIC-DEV-1.
- The term 'application' is used when referring to the hardware that is connected to the Application connector of the module.
- Hexadecimal values are written in the format NNNNh where NNNN is the hexadecimal value.
- 16/32 bit values are written in big endian Motorola format
- The term 'NV RAM' is used when referring to non-volatile memory.

# Support

### Europe (Sweden)


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


### HMS America


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


### HMS Germany


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


### HMS Japan


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

# About the AnyBus-IC DeviceNet

The AnyBus-IC DeviceNet integrates all analog and digital functionality required to communicate on a DeviceNet network, into a single chip.

Being a member of the AnyBus-IC family, it can be used with intelligent as well as non-intelligent applications, using a serial communication interface, and/or using external shift-registers to form digital inputs and outputs.

## Features

- **DeviceNet Group 2 Slave**
- **Supports up to 5 Explicit Server Connections opened by the UCMM**
- **Identity Object Customization**

  This makes it possible for a configuration tool to identify the module as a vendor specific device instead of a generic AnyBus-IC module.
- **I/O data**

  It is possible to configure the in / out data size from the application. The module supports up to 48 bytes in and 48 bytes out data. Supported communication types:

  - Polled I/O
  - COS/Cyclic I/O
  - Bitstrobe I/O
- **Application Parameters**

  Application specific parameters can be created by the application during startup.
- **Acyclic Data and Parameter Data Mapping**

  Application Parameters can be accessed from the fieldbus by mapping them to Vendor Specific DeviceNet Objects.

## Compatible Products

This product is a member of the AnyBus concept of interchangeable fieldbus modules. Standardization of mechanical, electrical and software interfaces ensures that the different AnyBus-IC models are fully interchangeable with only little or no required software and/or hardware adjustments, depending on the application.

## Electronic Data Sheet (.EDS) File

Each device in a DeviceNet network is associated with an EDS-file, containing the necessary information about the device. This file is used by the network configuration utility during network configuration. The latest version of the generic EDS-file for the AnyBus-IC DeviceNet is available at HMS web site, ‘www.hms-networks.com’.

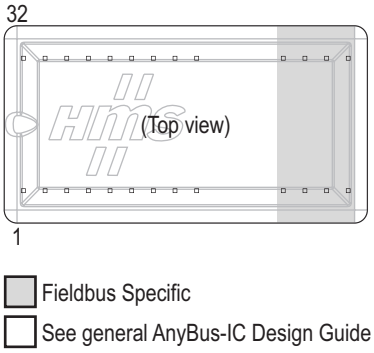
**Note:** If the I/O configuration or the information in the Identity Object has been changed, the generic EDS file must be modified.

# Connectors, Switches & Indicators

## Application Connector

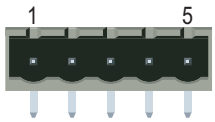
| Pin | Signal | Description |
|---|---|---|
| 1-12 | (See AnyBus-IC Design Guide) | - |
| 13 | FB1 | CAN_H |
| 14 | FB2 | CAN_L |
| 15 | FB3 | NC |
| 16 | FB4 | NC |
| 17 | PE | PE |
| 18 | SHIELD | SHIELD |
| 19 | FB5[a] | V- |
| 20 | FB6[a] | V+ |
| 21-32 | (See AnyBus-IC Design Guide) | - |

a. See Appendix A-1 “Power Requirements” for more information.




## DeviceNet Connector

| Pin | Signal | Connect to |
|---|---|---|
| 1 | V- | FB5 |
| 2 | CAN_L | FB2 |
| 3 | SHIELD | SHIELD |
| 4 | CAN_H | FB1 |
| 5 | V+ | FB6 |



# Switches (Fieldbus Specific Input Register)

The module supports baudrate and MacID configuration via the Fieldbus Specific Input Register on the SSC interface. It supports two kinds of switches; BCD coded and binary/hex coded.

- **BCD Switch**

  This type of switch can only be used to set the MAC ID.

- **Binary Switch**

  This type of switch can be used for both MAC ID and Baud rate configuration.

The type of switch used must be specified using parameter #9 ("Switch Coding"), see general AnyBus-IC Design Guide for more information.

On the AnyBus-IC DeviceNet, the default value of this parameter is 01h (Binary switches). However, the default value of this parameter is fieldbus dependant, i.e. there is no guarantee that the same type of switch is used by default on other versions of the AnyBus-IC.

## BCD Switch

Two switches are used to specify the MAC ID, one for each decimal digit. If a value outside the valid range (0-63) is specified, the the actual value will be 63.

It is not possible to set the Baud Rate using this type of switch. Instead, the baud rate must be set using parameter #106 ("DeviceNet baudrate config"), see 6-1 "Fieldbus Specific Parameters".

## Binary Switch

One 8 pole binary switch is used to specify both MAC ID and Baud rate settings.

- **Baud Rate**

  The 2 most significant switches are used for baudrate configuration.

| Bit 7 (MSB) | Bit 6 | Baud Rate |
|---|---|---|
| 0 | 0 | 125kbps |
| 0 | 1 | 250kbps |
| 1 | 0 | 500kbps |
| 1 | 1 | (reserved) |

- **MAC ID**

  The 6 least significant switches are used for MAC ID configuration.

| Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 (LSB) | MAC ID |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 2 |
| ... | ... | ... | ... | ... | ... | ... |
| 1 | 1 | 1 | 1 | 0 | 1 | 61 |
| 1 | 1 | 1 | 1 | 1 | 0 | 62 |
| 1 | 1 | 1 | 1 | 1 | 1 | 63 |

# Status Indicators (Fieldbus Specific Output Register)

The Fieldbus Specific Output Register on the SSC interface is used as follows:

- **Module Status (MS)**

| Bit | State | Indication |
|---|---|---|
| - | (off) | Device not powered |
| 0 (LSB) | Green | Device operational |
| 1 | Red | Unrecoverable fault |
| | Red, flashing | Minor fault recoverable |

- **Network Status (NS)**

| Bit | State | Indication |
|---|---|---|
| - | (off) | Device not powered / Not online |
| 2 | Green | Link OK, On line, Connected |
| | Green, flashing | On line, Not connected |
| 3 (MSB) | Red | Critical link failure |
| | Red, flashing | Connection timeout |

- **Power up LED Test**

  During power up, these leds are tested with a series of red-green flashes according to the DeviceNet specification (see table below).

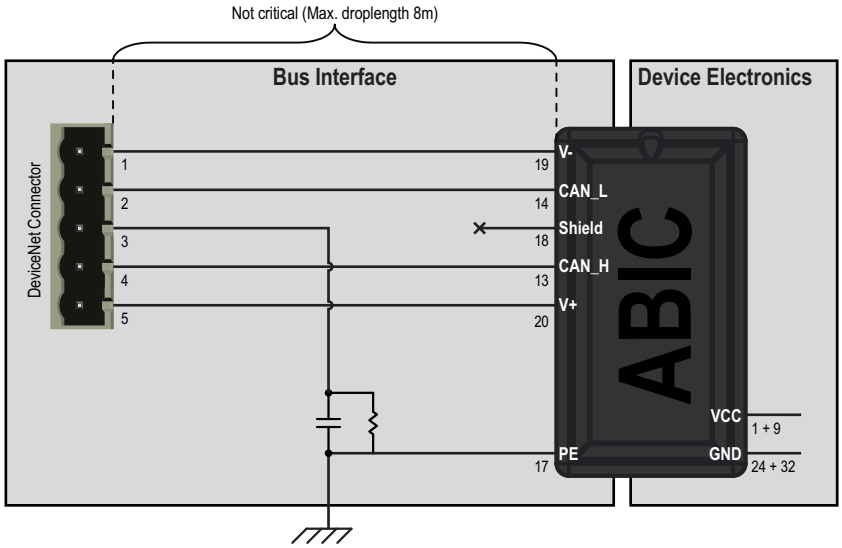
| Duration | Module Status (MS) | Network Status (NS) |
|---|---|---|
| 0.25s | Green | (off) |
| 0.25s | Red | (off) |
| 0.25s | (off) | Green |
| 0.25s | (off) | Red |

**Note:** The application can read the state of these leds using parameter #7 ("LED State").

# Design Considerations

As the PE/Shield requirements vary between different fieldbus systems, special care has to be taken when designing the carrier board if compatibility with other versions of the AnyBus-IC is desired.

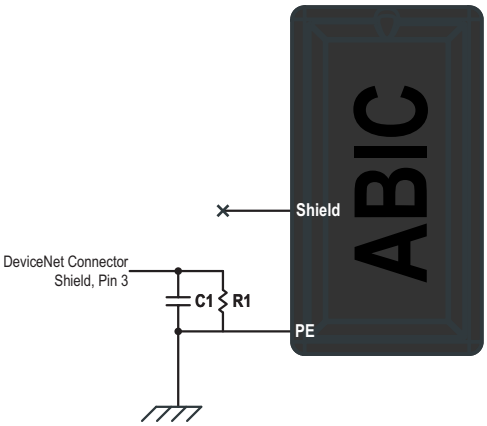
Consult the general AnyBus-IC Design Guide for more information. It is also recommended to study the design examples for the AnyBus-IC Ethernet/IT/IP and the AnyBus-IC Profibus.




# Shield and Protective Earth Connection

The module and the fieldbus interface must be properly connected to protective earth for proper EMC behaviour. Although the module features an on board R/C filter circuit according to the DeviceNet standard, it is recommended to use an external circuit according to the figure below.

| Ref. | Item |
|---|---|
| C1 | 10nF/2kV |
| R1 | 1M |



# Configuration

## Fieldbus Settings

### Mac ID

If bit 4 in parameter #8 (“Configuration Bits”, see general AnyBus-IC Design Guide) is set, the MAC ID is read from parameter #103 (“FB Node Address Config”).

If not, the MAC ID is read from the Fieldbus Specific Input Register on the SSC interface, see 2-2 “Switches (Fieldbus Specific Input Register)”.

### Baud rate

The module supports fieldbus communication at baud rates 125kbit/s, 250kbit/s and 500kbit/s. Automatic baud rate detection is not supported on the fieldbus interface.

If bit 5 in parameter #8 (“Configuration Bits”, see general AnyBus-IC Design Guide) is set, the baud rate is read from parameter #106 (“DeviceNet Baud rate Config”)

If not, the baud rate is read from the Fieldbus Specific Input Register, see 2-2 “Switches (Fieldbus Specific Input Register)”.

## Customization

### Identity Object

The information in the Identity Object can be changed, i.e it is possible to customize Vendor ID, Device Type, Product Code, and Product Name. This information is protected by a password to prevent unauthorized changes. This password can be obtained by contacting HMS.

| | Default Value: | Parameter # |
|---|---|---|
| **Vendor ID:** | 005Ah (HMS Networks) | 108 |
| **Device Type:** | 000Ch (Communications Adapter) | 109 |
| **Product Code:** | 003Dh | 110 |
| **Product Name:** | ‘AnyBus-IC DEV’ | 111 |

**Note 1:** These settings are read once during startup, i.e. it is required to restart the module in order for any changes to have effect.

**Note 2:** These settings are stored in non-volatile memory and will not be affected by a reset or a ‘Set Defaults’-operation unless the correct password has been entered using parameter #102 (‘FB Password’).

### Application Parameters

Application Parameters are user specific parameters created by the application during startup. See 7-1 “Application Parameters” for more information.

# DeviceNet Object Implementation

## Mandatory Objects

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

## Vendor Specific Object

It is possible to map objects that are used as a connection between the DeviceNet specific part and the general AnyBus-IC part of the module. For more information, see 7-6 “Acyclic Data & Parameter Data Mapping”.

| Object Name | Class | Description |
|---|---|---|
| Application | - | (mapping dependent) |

# Identity Object, Class 01h

## Services

Class services: Get Attribute Single

Instance services: Get Attribute Single
Reset

## Class Attributes

| # | Access | Name | Type | Value | Description |
|---|---|---|---|---|---|
| 1 | Get | Revision | UINT | 0001h | Revision 1 |

## Instance Attributes

| # | Access | Name | Type | Value | Description |
|---|---|---|---|---|---|
| 1 | Get | Vendor ID | UINT | Default: 005Ah | HMS Networks[a] |
| 2 | Get | Device Type | UINT | Default: 000Ch | Communications Adapter[a] |
| 3 | Get | Product Code | UINT | Default: 003Dh | AnyBus-IC DEV[a] |
| 4 | Get | Revision | Struct of: | | - |
| | | | USINT | | Major fieldbus version |
| | | | USINT | | Minor fieldbus version |
| 5 | Get | Status | WORD | - | Device status, see table below |
| 6 | Get | Serial Number | UDINT | Module serial number | Serial number of the module |
| 7 | Get | Product Name | SHORT_STRING | AnyBus-IC DEV | Name of product[a] |

a. Can be customized, see 4-1 “Customization”

- **Status Attribute (#5)**

| bit(s) | Name |
|---|---|
| 0 | Module Owned. A master has allocated the module. |
| 1 | (reserved) |
| 2 | Configured |
| 3 -7 | (reserved) |
| 8 | Minor recoverable fault |
| 9 | Minor recoverable fault |
| 10 | Major recoverable fault |
| 11 | Major unrecoverable fault |
| 12 - 15 | (reserved) |

### Reset Service

The Identity Object provides a reset service. There are two different types of reset requests:

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

  The module will by default perform a reset of the entire module. However, if the Interrupt Config (RES) bit (Parameter 12, bit 5) is set, the module will instead issue an interrupt to the application. The application is then responsible for resetting itself and the AnyBus IC.

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

  If the Interrupt Config (DEF) bit (Parameter 12, bit 4) is set, the module will issue an interrupt to the application. The interrupt cause register will show 'Set Default' as cause. The application is then responsible for resetting configuration settings before resetting itself and the module.

  If the Interrupt Config (RES) bit (Parameter 12, bit 4) is set, the module will issue an interrupt to the application. The interrupt cause register will show 'Reset' as cause. The application is then responsible for resetting itself and the module.

## Message Router, Class 02h

### Services

Class services: Get Attribute Single

Instance services: -

### Class Attributes

| # | Access | Name | Type | Value | Description |
|---|---|---|---|---|---|
| 1 | Get | Revision | UINT | 0001h | Revision 1 |

# DeviceNet Object, Class 03h

### Services

Class services: Get Attribute Single

Instance services: Get Attribute Single
Allocate Master/Slave Connection Set
Release group 2 Identifier Set

### Class Attributes

| # | Access | Name | Type | Value | Description |
|---|---|---|---|---|---|
| 1 | Get | Revision | UINT | 0002h | Revision 2 |

### Instance Attributes

| # | Access | Name | Type | Value | Description |
|---|---|---|---|---|---|
| 1 | Get | MAC ID | USINT | N/A | Currently used MacID |
| 2 | Get | Baud Rate | USINT | N/A | Currently used baudrate:<br>1 = 125 kbps<br>2 = 250 kbps<br>3 = 500 kbps |
| 5 | Get | Allocation Information | Struct of: | | - |
| | | | BYTE | N/A | Allocation choice byte |
| | | | USINT | N/A | Master's MAC ID |

# Assembly Object, Class 04h

The Assembly object uses static assemblies. The assembly instance ID's used are in the vendor specific range.

### Services

Class services: -

Instance services: Get Attribute Single

### Instance Attributes, Instance/Connection Point 64h

| # | Access | Name | Type | Description |
|---|---|---|---|---|
| 3 | Get | Data | Array of BYTE | Data produced by the AnyBus IC to the master |

### Instance Attributes, Instance/Connection Point 96h

| # | Access | Name | Type | Description |
|---|---|---|---|---|
| 3 | Get | Data | Array of BYTE | Data consumed by the AnyBus-IC from the master |

# Connection Object, Class 05h

### Services

Class services: Get Attribute Single

Instance services: Get Attribute Single
Set Attribute Single

### Class Attributes

| # | Access | Name | Type | Value | Description |
|---|---|---|---|---|---|
| 1 | Get | Revision | UINT | 0001h | Revision 1 |

Instance 1 = Explicit messaging connection (Predefined in DeviceNet object)
Instance 2 = Polled connection / COS / Cyclic consuming connection
Instance 3 = Bit strobe connection
Instance 4 = COS / Cyclic producing connection
Instance 10 - 14 = Explicit connection (UCMM allocated)

### Instance 1 & 10...14 (Explicit Messaging Connection) Attributes

| # | Access | Name | Type | Value | Description |
|---|---|---|---|---|---|
| 1 | Get | State | USINT | N/A | 0 = Non existent<br>1 = Configuring<br>2 = Waiting for connection ID<br>3 = Established<br>4 = Timeout |
| 2 | Get | Instance type | USINT | 0 | Explicit messaging connection |
| 3 | Get | Transport Class trigger | BYTE | 83h | Server/Transport Class 3 |
| 4 | Get | Produced Connection ID | UINT | N/A | CAN ID for transmission |
| 5 | Get | Consumed Connection ID | UINT | N/A | CAN ID for reception |
| 6 | Get | Initial Comm Characteristics | BYTE | N/A | The message group over which the communication occurs |
| 7 | Get | Produced Connection Size | UINT | 0100h | 256 bytes |
| 8 | Get | Consumed Connection Size | UINT | 0100h | 256 bytes |
| 9 | Get/Set | Expected Packet Rate | UINT | 09C4h | Timing associated with this connection (2500ms) |
| 12 | Get/Set | Watchdog timeout action | USINT | 1 | 1 = Auto delete<br>3 = Deferred delete |
| 13 | Get | Produced Connection path length | UINT | 0 | Number of bytes in the produced connection path attribute |
| 14 | Get | Produced Connection Path | EPATH | No value | No connection path |
| 15 | Get | Consumed Connection path length | UINT | 0 | Number of bytes in the consumed connection path instance |
| 16 | Get | Consumed Connection Path | EPATH | No value | No connection path |

### Instance 2 (Polled or COS/Cyclic Consuming Connection) Attributes

| # | Access | Name | Type | Value | Description |
|---|---|---|---|---|---|
| 1 | Get | State | USINT | N/A | 0 = Non existent<br>1 = Configuring<br>2 = Waiting for connection ID<br>3 = Established<br>4 = Timeout |
| 2 | Get | Instance type | USINT | 1 | IO Connection |
| 3 | Get | Transport Class trigger | BYTE | 80h | Server/Transport Class 0<br>(COS//Cyclic w/o ACK) |
| | | | | 83h | Server/Transport Class 3<br>(Polled or COS/Cyclic with ACK) |
| 4 | Get | Produced Connection ID | UINT | N/A | CAN ID for transmission |
| 5 | Get | Consumed Connection ID | UINT | N/A | CAN ID for reception |
| 6 | Get | Initial Comm Characteristics | BYTE | 01h<br>(Polled or COS/Cyclic with ACK) | Produces over message group 1<br>Consumes over message group 2 |
| | | | | F1h<br>(COS/Cyclic w/o ACK) | Does not produce<br>Consumes over message group 2 |
| 7 | Get | Produced Connection Size | UINT | N/A | 0 - 48 bytes depending on AnyBus IC input size configuration (Polled) |
| | | | | 0 | COS/Cyclic |
| 8 | Get | Consumed Connection Size | UINT | N/A | 0 - 48 bytes depending on AnyBus-IC output size configuration |
| 9 | Get/Set | Expected Packet Rate | UINT | 0 | Timing associated with this connection |
| 12 | Get/Set | Watchdog timeout action | USINT | 0 | 0 = Transition to the timed out state |
| 13 | Get | Produced Connection path length | UINT | 0007h (Polled) | Number of bytes in the produced connection path attribute |
| | | | | 0000h (COS/Cyclic) | |
| 14 | Get | Produced Connection Path | EPATH | 20 04 25 64 00 30 03h | Assembly Object, Instance 100, Attribute 3 is producing the data |
| | | | | No value<br>(COS/Cyclic) | No producing object |
| 15 | Get | Consumed Connection path length | UINT | 0007h | Number of bytes in the consumed connection path instance |
| 16 | Get | Consumed Connection Path | EPATH | 20 04 25 96 00 30 03h | Assembly Object, Instance 150, Attribute 3 is consuming the data. |
| 17 | Get | Production Inhibit Time | UINT | 0 | Minimum time between new data production |

### Instance 3 (Bit-strobe connection) Attributes

| # | Access | Name | Type | Value | Description |
|---|---|---|---|---|---|
| 1 | Get | State | USINT | N/A | 0 = Non existent<br>1 = Configuring<br>2 = Waiting for connection ID<br>3 = Established<br>4 = Timeout |
| 2 | Get | Instance type | USINT | 1 | IO Connection |
| 3 | Get | Transport Class trigger | BYTE | 82h | Server/Transport Class 2 |
| 4 | Get | Produced Connection ID | UINT | N/A | CAN ID for transmission |
| 5 | Get | Consumed Connection ID | UINT | N/A | CAN ID for reception |
| 6 | Get | Initial Comm Characteristics | BYTE | 02h | Produces over message group 1<br>Consumes over message group 2 |
| 7 | Get | Produced Connection Size | UINT | N/A | 0 - 8 depending on AnyBus IC input size configuration |
| 8 | Get | Consumed Connection Size | UINT | 8 | Consumes 8 bytes. 1 bit used for the AnyBus IC slave. |
| 9 | Get/Set | Expected Packet Rate | UINT | 0 | Timing associated with this connection |
| 12 | Get | Watchdog timeout action | USINT | 0 | 0 = Transition to the timed out state |
| 13 | Get | Produced Connection path length | UINT | 0007h | Number of bytes in the produced connection path attribute |
| 14 | Get | Produced Connection Path | EPATH | 20 04 25 64 00 30 03h | Assembly Object, Instance 100, Attribute 3 is producing the data |
| 15 | Get | Consumed Connection path length | UINT | 0007h | Number of bytes in the consumed connection path instance |
| 16 | Get | Consumed Connection Path | EPATH | 20 04 25 96 00 30 03h | Assembly Object, Instance 150, Attribute 3 is consuming the data |
| 17 | Get | Production Inhibit Time | UINT | 0 | Minimum time between new data production |

### Instance 4 (COS/Cyclic connection) Attributes

| # | Access | Name | Type | Value | Description |
|---|---|---|---|---|---|
| 1 | Get | State | USINT | N/A | 0 = Non existent<br>1 = Configuring<br>2 = Waiting for connection ID<br>3 = Established<br>4 = Timeout |
| 2 | Get | Instance type | USINT | 1 | IO Connection |
| 3 | Get | Transport Class trigger | BYTE | N/A | 00h = Cyclic<br>03h = Cyclic + ACK<br>10h = COS<br>13h = COS + ACK |
| 4 | Get | Produced Connection ID | UINT | N/A | CAN ID for transmission |
| 5 | Get | Consumed Connection ID | UINT | N/A | CAN ID for reception |
| 6 | Get | Initial Comm Characteristics | BYTE | 0Fh (No ACK) | Produces over message group 1<br>Does not consume. |
| | | | | 01h (ACK) | Produces over message group 1<br>Consumes over message group 2 |
| 7 | Get | Produced Connection Size | UINT | N/A | 0 - 48 bytes depending on AnyBus IC input size configuration |
| 8 | Get | Consumed Connection Size | UINT | 0 | Consumes 0 bytes on this connection |
| 9 | Get/Set | Expected Packet Rate | UINT | 0 | Timing associated with this connection |
| 12 | Get | Watchdog timeout action | USINT | 0 | 0 = Transition to the timed out state |
| 13 | Get | Produced Connection path length | UINT | 0007h | Number of bytes in the produced connection path attribute |
| 14 | Get | Produced Connection Path | EPATH | 20 04 25 64 00 30 03h | Assembly Object, Instance 100, Attribute 3 is producing the data |
| 15 | Get | Consumed Connection path length | UINT | 0007h (No ACK) | Number of bytes in the consumed connection path instance |
| | | | | 0005h (ACK) | |
| 16 | Get | Consumed Connection Path | EPATH | No value (No ACK) | Empty |
| | | | EPATH | 20 2B 25 01 00h | Acknowledge Handler Object, Instance 1 |
| 17 | Get | Production Inhibit Time | UINT | 0 | Minimum time between new data production |

# Acknowledge Handler Object, Class 2Bh

## Services

Class services: Get Attribute Single

Instance services: Get Attribute Single
Set Attribute Single

## Class Attributes

| # | Access | Name | Type | Value | Description |
|---|---|---|---|---|---|
| 1 | Get | Revision | UINT | 0001h | Revision 1 |

## Instance[1] Attributes

| # | Access | Name | Type | Value | Description |
|---|---|---|---|---|---|
| 1 | Get/Set | Acknowledge Timer | UINT | 16 | Time (in ms) to wait for acknowledge before re-sending |
| 2 | Get/Set | Retry Limit | USINT | 1 | Number of ACK timeouts before retry limit reached event |
| 3 | Get | Producing connection Instance | UINT | 4 | Connection instance, which contains the path of the producing I/O application object, which will be notified of Ack Handler events. |

1. Instance 1 is created when using an ACK:ed COS/Cyclic connection.

# Fieldbus Specific Parameters

To be able to use the full functionality of every fieldbus, the Fieldbus Specific Parameters are used. These parameters are specific to the actual fieldbus used and must be configured accordingly. The table below shows an overview of the parameters, followed by a detailed description of each parameter.

| # | R/W | Name | Size | Default value | Modbus Address |
|---|---|---|---|---|---|
| 100 | R | FB Status | 2 byte | - | 7000h |
| 102 | W | FB Password | 2 byte | - | 7022h |
| 103 | R/W | FB Node address Config | 1 byte | 3Fh | 7002h |
| 104 | R | SCC FB Node address | 1 byte | - | 7003h |
| 105 | R | FB Actual Node Address | 1 byte | - | 7004h |
| 106 | R/W | DeviceNet baudrate config | 1 byte | 00h | 7005h |
| 107 | R | DeviceNet baudrate actual | 1 byte | - | 7006h |
| 108 | R(W) | DeviceNet Vendor ID | 2 byte | 005Ah | 7007h |
| 109 | R(W) | DeviceNet Product Type | 2 byte | 000Ch | 7008h |
| 110 | R(W) | DeviceNet Product Code | 2 byte | 003Dh | 7009h |
| 111 | R(W) | DeviceNet Product Name | 32 byte[a] | 'AnyBus-IC DEV' | 700Ah-7019h |
| 113 | R | Connection Status Polled | 1 byte | - | 701Ch |
| 114 | R | Connection Status COS | 1 byte | - | 701Dh |
| 115 | R | Connection Status Bit Strobed | 1 byte | - | 701Eh |
| 116 | R | Connection Status Cyclic | 1 byte | - | 701Fh |
| 200 | RW | Application Parameter 1 | b | - | 8000h - |
| 201 | RW | Application Parameter 2 | b | - | b |
| 202 | RW | Application Parameter 3 | b | - | b |
| ... | ... | ... | ... | ... | ... |
| c | RW | Application Parameter 50[c] | b | - | bc |

a. All 32 bytes must be read/written using a single modbus command
b. Parameter dependent.
c. It is possible to register up to 50 Application Parameters with a maximum total size of 1kbyte.

## FB Status (Parameter #100)

This parameter holds information about the current bus status.

| | |
|---|---|
| Parameter Name | 'FB Status' |
| Parameter number | 100 |
| Modbus Address | 7000h |
| Default value | N/A |
| Range | Bit field, see below. |
| Size | 2 bytes |
| Stored in NV RAM | No |
| Access | R |

### Bit layout

| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | PWR | IDLE | BUS | | | | | | | | |

- **BUS**
  1: Bus is on line
  0: Bus is off line
- **IDLE**[1]
  1: Bus is in Idle mode
  0: Bus is not in Idle mode
- **PWR**
  1: Power is supplied to the bus electronics (24V)
  0: Power is not supplied to the bus electronics

1. This bit is valid only if the connection consumes data. If not, there is no way to determine if the master is in idle mode or not.

## FB Password (Parameter #102)

This parameter is used to unlock the following parameters:

- Parameter #108 - DeviceNet Vendor ID
- Parameter #109 - DeviceNet Product Type
- Parameter #110 - DeviceNet Product Code
- Parameter #111 - DeviceNet Product Name

(The password can be obtained by contacting HMS)

| | |
|---|---|
| Parameter Name | 'FB Password' |
| Parameter number | 102 |
| Modbus Address | 7022h |
| Default value | N/A |
| Range | 0000h - FFFFh |
| Size | 2 bytes |
| Stored in NV RAM | No |
| Access | W |

## FB Node Address Config (Parameter #103)

This parameter holds the manually configured DeviceNet node address. Note that in order for this value to be valid, bit 4 in parameter #8 ("Configuration Bits") must be set.

**Note:** This information is stored in non-volatile memory.

| | |
|---|---|
| Parameter Name | 'Cfg FB Address' |
| Parameter number | 103 |
| Modbus Address | 7002h |
| Default value | 3Fh |
| Range | 00h - 3Fh |
| Size | 1 byte |
| Stored in NV RAM | Yes |
| Access | R/W |

## SCC FB Node Address (Parameter #104)

This parameter holds the auto configured fieldbus node address from the SSC interface. Note that in order for this value to be valid, bit 5 in parameter #8 ("Configuration Bits") must be set.

If the SCC node address is larger than allowed by the fieldbus, the value will be forced to 63.

| | |
|---|---|
| Parameter Name | 'SCC FB Address' |
| Parameter number | 104 |
| Modbus Address | 7003h |
| Default value | N/A |
| Range | 00h - FFh |
| Size | 1 byte |
| Stored in NV RAM | No |
| Access | R |

## FB Actual Node Address (Parameter #105)

This parameter holds the actual fieldbus node address.

| | |
|---|---|
| Parameter Name | 'Act FB Address' |
| Parameter number | 105 |
| Modbus Address | 7004h |
| Default value | N/A |
| Range | 00h - 3Fh |
| Size | 1 byte |
| Stored in NV RAM | No |
| Access | R |

## DeviceNet Baud rate Config (Parameter #106)

This parameter is used to change the DeviceNet baud rate. Note that in order for this value to be valid, bit 5 in parameter #8 ("Configuration Bits") must be set.

**Note:** This information is stored in non-volatile memory.

| | |
|---|---|
| Parameter Name | 'Cfg DEV Baudrate' |
| Parameter number | 106 |
| Modbus Address | 7005h |
| Default value | N/A |
| Range | 00h - 02h |
| Size | 1 byte |
| Stored in NV RAM | Yes |
| Access | R/W |

- **Value**

  00h:125kbps

  01h:250kbps

  02h:500kbps

## DeviceNet Baud rate Actual (Parameter #107)

This parameter holds the actual DeviceNet baud rate setting.

| | |
|---|---|
| Parameter Name | 'Act DEV Baudrate' |
| Parameter number | 107 |
| Modbus Address | 7006h |
| Default value | N/A |
| Range | 00h - 02h |
| Size | 1 byte |
| Stored in NV RAM | No |
| Access | R |

- **Value**

  00h:125kbps

  01h:250kbps

  02h:500kbps

## DeviceNet Vendor ID (Parameter #108)

This parameter is used to change the Vendor ID. Before the application can gain access (including a 'set default') to this parameter, the module has to be unlocked using parameter #102 ("FB Password").

**Note:** If this parameter is altered, the EDS file must be modified accordingly.

| | |
|---|---|
| Parameter Name | 'Act DEV Vendor ID' |
| Parameter number | 108 |
| Modbus Address | 7007h |
| Default value | 005Ah (HMS Industrial Networks AB) |
| Range | 0000h - FFFFh |
| Size | 2 bytes |
| Stored in NV RAM | Yes |
| Access | R(W) |

## DeviceNet Product Type (Parameter #109)

This parameter is used to change the Product Type. Before the application can gain access (including a 'set default') to this parameter, the module has to be unlocked using parameter #102 ("FB Password").

**Note:** If this parameter is altered, the EDS file must be modified accordingly.

| | |
|---|---|
| Parameter Name | 'DeviceNet Product Type' |
| Parameter number | 109 |
| Modbus Address | 7008h |
| Default value | 000Ch (Communication Adapter) |
| Range | 0000h - FFFFh |
| Size | 2 bytes |
| Stored in NV RAM | Yes |
| Access | R(W) |

## DeviceNet Product Code (Parameter #110)

This parameter is used to change the Product Code. Before the application can gain access (including a 'set default') to this parameter, the module has to be unlocked using parameter #102 ("FB Password").

**Note:** If this parameter is altered, the EDS file must be modified accordingly.

| | |
|---|---|
| Parameter Name | 'DeviceNet Product Code' |
| Parameter number | 110 |
| Modbus Address | 7009h |
| Default value | 003Dh |
| Range | 0000h - FFFFh |
| Size | 2 bytes |
| Stored in NV RAM | Yes |
| Access | R(W) |

## DeviceNet Product Name (Parameter #111)

This parameter is used to change the Product Name. Before the application can gain access (including a 'set default') to this parameter, the module has to be unlocked using parameter #102 ("FB Password").

**Note:** If this parameter is altered, the EDS file must be modified accordingly.

| | |
|---|---|
| Parameter Name | 'DeviceNet Product Name' |
| Parameter number | 111 |
| Modbus Address | 700Ah - 7019h |
| Default value | 'AnyBus-IC Dev' |
| Range | - |
| Size | 32 bytes[a] |
| Stored in NV RAM | Yes |
| Access | R(W) |

a. All 32 bytes must be read/written with one modbus command.

## Connection Status Polled (Parameter #113)

This parameter returns the connection status for polled I/O connections.

| | |
|---|---|
| Parameter Name | 'Conn Status Polled' |
| Parameter number | 113 |
| Modbus Address | 701Ch |
| Default value | N/A |
| Range | 00h - FFh |
| Size | 1 byte |
| Stored in NV RAM | No |
| Access | R |

- **Value**

  00h:Connection non existant

  01h:Connection configuring

  03h:Connection established

  04h:Connection timed out

## Connection Status COS (Parameter #114)

This parameter returns the connection status for change-of-state connections.

| | |
|---|---|
| Parameter Name | 'Conn Status COS' |
| Parameter number | 114 |
| Modbus Address | 701Dh |
| Default value | N/A |
| Range | 00h - FFh |
| Size | 1 byte |
| Stored in NV RAM | No |
| Access | R |

- **Value**

  00h:Connection non existant

  01h:Connection configuring

  03h:Connection established

  04h:Connection timed out

## Connection Status Bit Strobed (Parameter #115)

This parameter returns the connection status for bitstrobed connections.

| | |
|---|---|
| Parameter Name | 'Conn Status Bit Strobed' |
| Parameter number | 115 |
| Modbus Address | 701E |
| Default value | N/A |
| Range | 00h - FFh |
| Size | 1 byte |
| Stored in NV RAM | No |
| Access | R |

- **Value**

  00h:Connection non existant

  01h:Connection configuring

  03h:Connection established

  04h:Connection timed out

## Connection Status Cyclic (Parameter #116)

This parameter returns the connection status for cyclic connections.

| | |
|---|---|
| Parameter Name | 'Conn Status Cyclic' |
| Parameter number | 116 |
| Modbus Address | 701Fh |
| Default value | N/A |
| Range | 00h - FFh |
| Size | 1 byte |
| Stored in NV RAM | No |
| Access | R |

- **Value**

  00h:Connection non existant

  01h:Connection configuring

  03h:Connection established

  04h:Connection timed out

# Application Parameters

## Introduction

An Application Parameter is a user specific AnyBus-IC parameter created by the application during start-up. Just like other parameters, Application Parameters can be accessed by the application via the SCI interface using Modbus messages, or by the user via the MIF interface.

For each Application Parameter it is possible to specify a number of properties, such as datatype, range, name etc. This enables the application to utilize the MIF user interface for internal functions.

Application Parameters can also be accessed from the fieldbus, by mapping them to a DeviceNet Object using the CIP Mapping Object Class, see 7-6 "Acyclic Data & Parameter Data Mapping".

## HMS Object Software

This functions described in this chapter uses the HMS Object Software (HOS) to create and map application specific parameters. The HOS provides access to parameters and AnyBus-IC functions in an object oriented way using the Modbus Object Messaging protocol (Consult the AnyBus-IC Design Guide for more information).

Two HOS Objects are described in this chapter:

- **Application Parameter Object, Class 85h**

  This class is used to create Application Parameters.

- **CIP Mapping Object, Class A5h**

  This class is used to map HOS Objects (In this case an Application Parameter) to DeviceNet Objects, enabling remote access to Application Parameters from the fieldbus.

# Creating an Application Parameter

## Query - “Application Parameter Object”

To create a new Application Parameter, send the following message using Modbus Object Messaging. (Consult the AnyBus-IC Design Guide for more information about the Object Message Sub Field)

### Object Message Sub Field.

| Fragment byte count | Fragment protocol | Class ID | Instance ID | Service Code | Attribute | Data Field |
|---|---|---|---|---|---|---|
| (size) | 02h | 0085h | 0000h | 0005h | 0000h | (See below) |

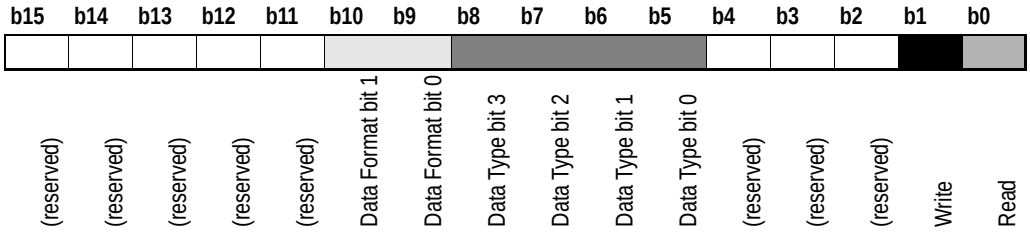
| Parameter Size (WORD) | Descriptor (DWORD) | Parameter Info (Size varies) | Extension Word (Optional) (WORD) |
|---|---|---|---|

### Parameter Size

This value depends on the type of data specified in the Descriptor (see below).

| Data Type | Valid Parameter Size values |
|---|---|
| UINT, INT, BITSTRING | 1, 2, 4 |
| FLOAT | 4 |
| STRING | 1 - 32 (String length including NULL termination) |
| BYTE_ARRAY | N/A |

### Descriptor[1]




| Write Value | Meaning |
|---|---|
| 0 | Write access not allowed |
| 1 | Write access allowed |

| Read Value | Meaning |
|---|---|
| 0 | Read access not allowed |
| 1 | Read access allowed |

| Data Type | Meaning |
|---|---|
| 0 (0000) | UINT |
| 1 (0001) | INT |
| 2 (0010) | BITSTRING |
| 3 (0011) | STRING |
| 4 (0100) | FLOAT |
| 5 (0101) | BYTE ARRAY |

| Data Format | Meaning |
|---|---|
| 0 (00) | Dec. |
| 1 (01) | Hex |
| 2 (10) | Bin |
| 3 (11) | Dotted decimal |

1. Note that the upper 16 bits of the Descriptor is reserved and should always be set to 0000h.

### Parameter Info

The size and contents of this field depends on the Data Type specified in the Descriptor block.

- **Data types UINT, INT, BITSTRING & FLOAT**

| Min Value (size varies) | Max Value (size varies) | Init Value (size varies) | Name[a] (String, 16 bytes) | Unit[a] (String, 16 bytes) |
|---|---|---|---|---|

| Field | Type / Size | Description |
|---|---|---|
| Min Value | Specified in 'Parameter Size' | Minimum allowed parameter value |
| Max Value | Specified in 'Parameter Size' | Maximal allowed parameter value |
| Init Value | Specified in 'Parameter Size' | Initial parameter value |
| Name | String (16 byte, null terminated) | Name of parameter, e.g "Speed"[a] |
| Unit | String (16 byte, null terminated) | Unit, e.g "RPM"[a] |

a. These fields are optional. (However, if used, both fields must be present)

- **Data type STRING**

| Init Value (STRING, 16 bytes) | Name[a] (STRING, 16 bytes) | Unit[a] (STRING, 16 bytes) |
|---|---|---|

| Field | Type / Size | Description |
|---|---|---|
| Init Value | Specified in 'Parameter Size' | Initial value |
| Name | String (16 byte, null terminated) | Name of parameter[a] |
| Unit | String (16 byte, null terminated) | Unit[a] |

a. These fields are optional. (However, if used, both fields must be present)

- **Data type BYTE_ARRAY**

| Min Value (BYTE) | Max Value (BYTE) | Init Value (BYTE) | Name[a] (String, 16 bytes) | Unit[a] (String, 16 bytes) |
|---|---|---|---|---|

| Field | Type / Size | Description |
|---|---|---|
| Min Value | Byte | Min. allowed value of each element in the array |
| Max Value | Byte | Max. allowed value of each element in the array |
| Init Value | Byte | Initial value of all elements in the array |
| Name | String (16 byte, null terminated) | Name of parameter[a] |
| Unit | String (16 byte, null terminated) | Unit[a] |

a. These fields are optional. (However, if used, both fields must be present)

### Extension Word (Optional)

This word is optional and specifies whether the response message should contain the Modbus address of the created Application Parameter or not.

| Value | Description |
|---|---|
| 0000h | - |
| 0001h | Request Modbus Address |
| Other values | (Reserved for future use) |

## Response - “Application Parameter Object”

The AnyBus-IC module will respond with the following message. (Consult the AnyBus-IC Design Guide for more information about the Object Message Sub Field)

### Object Message Sub Field

| Fragment byte count | Fragment protocol | Class ID | Instance ID | Service Code | Error Code | Data Field |
|---|---|---|---|---|---|---|
| (size) | 02h | 0085h | 0000h | 0006h | 0000h | (See below) |

| HOS Instance (WORD) | Parameter Number (WORD) | Modbus Address[a] (WORD) |
|---|---|---|

a. This field is only present if the Extension Word of the query is set to 0001h

### HOS Instance

If the Error Code is 0 (Success), this field contains the HOS Instance of the created Application Parameter.

### Parameter Number

If the Error Code is 0 (Success). this field contains the parameter number of the created Application Parameter.

### Modbus Address

If the Error Code is 0 (Success), this field contains the Modbus Address of the created Application Parameter.

**Note:** This field is only present if the Extension Word of the query is set to 0001h.

## Example

The example below creates an Application Parameter with the following properties:

- Parameter Name “Speed”, unit “rpm”
- Type 16 bit unsigned INT, range 0 - 65535, initial parameter value 32768.
- R/W access

**Query**

| | | | |
|---|---|---|---|
| | 01h | 5Bh | |
| | 35h | 02h | |
| Class | 0085h | | Application Parameter Class |
| Instance | 0000h | | |
| Service Code | 0005h | | Create |
| Attribute | 0000h | | |
| Parameter Size | 0002h | | Parameter Size = 2 bytes |
| Descriptor MSB | 0000h | | |
| Descriptor LSB | 0003h | | UINT, DEC, R/W |
| Min value | 0000h | | Minimum allowed value: 0 |
| Max value | FFFFh | | Maximum allowed value: 65535 |
| Init value | 8000h | | Initial value: 8000h |
| Name | 53h (‘S’) | 70h (‘p’) | “Speed” |
| | 65h (‘e’) | 65h (‘e’) | |
| | 64h (‘d’) | 00h | |
| | - | - | |
| | - | - | |
| | - | - | |
| | - | - | |
| | - | - | |
| Unit | 72h (‘r’) | 70h (‘p’) | “rpm” |
| | 6Dh (‘m’) | 00h | |
| | - | - | |
| | - | - | |
| | - | - | |
| | - | - | |
| | - | - | |
| | - | - | |
| Extension Word | 0001h | | Request Modbus Address |
| | CRC | | |

**Response**

| | | | |
|---|---|---|---|
| | 01h | 5Bh | |
| | 0Fh | 02h | |
| Class | 0085h | | Application Parameter Object Class |
| Instance | 0000h | | |
| Service Code | 0006h | | Create Response |
| Error Code | 0000h | | Success |
| HOS Instance | 0001h | | HOS Instance 1 |
| Parameter no. | 00C8h | | Parameter no. = 200 |
| Modbus Address | 8000h | | Modbus Address = 8000h |
| | CRC | | |

# Acyclic Data & Parameter Data Mapping

Acyclic data on the AnyBus-IC DeviceNet module is sent using Application Parameters mapped to Vendor Specific DeviceNet Objects.

The mapping procedure consists of two steps:

- **Creating the Application Parameter**
  (See 7-2 "Creating an Application Parameter")
- **Mapping the created Application Parameter to a DeviceNet Object**
  This is done by creating a new instance in the AnyBus-IC CIP Mapping Object Class (A5h). This class is used to map a vendor specific DeviceNet Object Attribute onto an AnyBus-IC Object Attribute.

## Query - "CIP Mapping Object"

(Consult the AnyBus-IC Design Guide for more information about the Object Message Sub Field)

### Object Message Sub Field

| Fragment byte count | Fragment protocol | Class ID | Instance ID | Service Code | Data Field |
|---|---|---|---|---|---|
| (size) | 02h | 00A5h | 0000h | 0005h | (See below) |

| CIP Class (WORD) | CIP Instance (WORD) | CIP Attribute (WORD) | HOS Class (WORD) | HOS Instance (WORD) | HOS Attribute (WORD) | Attribute Size (WORD) |
|---|---|---|---|---|---|---|

### CIP Class

CIP Class to map

### CIP Instance

CIP Instance to map

### CIP Attribute

CIP Attribute to map

### Attribute Size

Size of attribute. This value should match the Parameter Size value in the Application Parameter request.

### HOS Class

HOS Class to map.
(In this case 85h "Application Parameter Object Class")

### HOS Instance

HOS Instance to map
(In this case, use the HOS Instance value returned from the Application Parameter Object request when the Application Parameter was created.)

### HOS Attribute

HOS Attribute to map
In this case, the 0001h (=Parameter Value)

## Response - “CIP Mapping Object”

The response contains no additional data. (Consult the AnyBus-IC Design Guide for more information about the Object Message Sub Field)

### Object Message Sub Field

| Fragment byte count | Fragment protocol | Class ID | Instance ID | Service Code | Error Code |
|---|---|---|---|---|---|
| (8 bits) | 02h | 00A5h | 0000h | 0006h | (16 bits) |

## Example

This example will map the Application Parameter created earlier in this chapter to DeviceNet Class 144, Instance 1, Attribute 1.

**Query**

| | | | |
|---|---|---|---|
| | 01h | 5Bh | |
| | 17h | 02h | |
| Class | 00A5h | | CIP Mapping Object |
| Instance | 0000h | | |
| Service Code | 0005h | | Create |
| Attribute | 0000h | | |
| CIP Class | 0090h | | CIP Class 144 |
| CIP Instance | 0001h | | CIP Instance 1 |
| CIP Attribute | 0001h | | CIP attribute 1 |
| HOS Class | 0085h | | Application Parameter Object Class |
| HOS Instance | 0001h | | HOS Instance 1 |
| HOS Attribute | 0001h | | 0001h = Parameter value |
| Attribute Size | 0002h | | Size = Word |
| | CRC | | |

**Response**

| | | | |
|---|---|---|---|
| | 01h | 5Bh | |
| | 09h | 02h | |
| Class | 00A5h | | CIP Mapping Object |
| Instance | 0000h | | |
| Service Code | 0006h | | Create Response |
| Error Code | 0000h | | Success |
| | CRC | | |

# Electrical Characteristics

## Power Requirements

### Module Electronics

The module electronics requires 5V DC. For more information regarding power requirements, consult the AnyBus-IC Design Guide.

### Bus Interface

The fieldbus interface requires 24V DC bus power according to the DeviceNet specification.

### Power Consumption

| Interface | Max. | Typ. |
|---|---|---|
| Module Electronics, 5V | 250mA | - |
| Bus Interface, 24V | 70mA | 30mA |

## PE Grounding

Pin 17 in the Application connector must be connected to PE for proper EMC behaviour. For more information, see 3-1 “Shield and Protective Earth Connection”.

# Mechanical Specification

(Consult the general AnyBus-IC Design Guide for more information)