


# Fieldbus Appendix

# Anybus-IC DeviceNet

Rev 1.53

---

## HMS Industrial Networks AB

  
Germany +49 - 721 - 96472 - 0  
Japan +81 - 45 - 478 - 5340  
Sweden +46 - 35 - 17 29 20  
U.S.A. +1 - 312 - 829 - 0601  
France +33 - 3 89 32 76 76  
Italy +39 - 347 - 00894 - 70  
China +86 - 10 - 8532 - 3183

  
ge-sales@hms-networks.com  
jp-sales@hms-networks.com  
sales@hms-networks.com  
us-sales@hms-networks.com  
fr-sales@hms-networks.com  
it-sales@hms-networks.com  
cn-sales@hms-networks.com





# Table of Contents

<b>Preface</b>	<b>About This Document</b>	
	How To Use This Document .....	P-1
	Important User Information .....	P-1
	Related Documents.....	P-2
	Document History .....	P-2
	Conventions & Terminology.....	P-3
	Support .....	P-3
<b>Chapter 1</b>	<b>About the Anybus-IC DeviceNet</b>	
	General.....	1-1
	Features.....	1-1
	Fieldbus Conformance Notes .....	1-1
<b>Chapter 2</b>	<b>Basic Operation</b>	
	General Information.....	2-1
	<i>Software Requirements</i> .....	2-1
	<i>Electronic Data Sheet (EDS)</i> .....	2-1
	Device Identity .....	2-1
	Communication Settings .....	2-2
	<i>Baudrate</i> .....	2-2
	<i>Mac ID</i> .....	2-2
	<i>Quick Connect</i> .....	2-2
	Data Exchange.....	2-3
	<i>General</i> .....	2-3
	<i>Fieldbus I/O</i> .....	2-3
	<i>Application Parameters (Explicit Data)</i> .....	2-3
	Status Indicators (Fieldbus Specific Output) .....	2-4
	<i>General</i> .....	2-4
	<i>Compliance Notes</i> .....	2-4
	Switches (Fieldbus Specific Input).....	2-5
	<i>General</i> .....	2-5
	<i>Auto/Net</i> .....	2-6
	<i>Compliance Notes</i> .....	2-6

**Chapter 3    Fieldbus Specific Parameters**

<i>FB Status (#100)</i> .....	3-2
<i>FB Password (#102)</i> .....	3-3
<i>FB Node Address Config (#103)</i> .....	3-3
<i>SCC FB Node Address (#104)</i> .....	3-4
<i>FB Actual Node Address (#105)</i> .....	3-4
<i>DeviceNet Baud rate Config (#106)</i> .....	3-5
<i>DeviceNet Baud rate Actual (#107)</i> .....	3-5
<i>DeviceNet Vendor ID (#108)</i> .....	3-6
<i>DeviceNet Product Type (#109)</i> .....	3-6
<i>DeviceNet Product Code (#110)</i> .....	3-6
<i>DeviceNet Product Name (#111)</i> .....	3-7
<i>DeviceNet Revision (#112)</i> .....	3-7
<i>Connection Status Polled (#113)</i> .....	3-7
<i>Connection Status COS (#114)</i> .....	3-8
<i>Connection Status Bit Strobed (#115)</i> .....	3-8
<i>Connection Status Cyclic (#116)</i> .....	3-9
<i>FB Auto Baud Cfg (#117)</i> .....	3-9
<i>FB Quick Connect Cfg (#118)</i> .....	3-10
<i>FB Serial Number (#119)</i> .....	3-10

<b>Chapter 4</b>	<b>CIP Object Implementation</b>	
	General Information .....	4-1
	Identity Object (01h) .....	4-2
	<i>General Information</i> .....	4-2
	<i>Class Attributes</i> .....	4-2
	<i>Instance #1 Attributes</i> .....	4-2
	<i>Service Details: Reset Service</i> .....	4-3
	Message Router (02h) .....	4-4
	<i>General Information</i> .....	4-4
	<i>Class Attributes</i> .....	4-4
	<i>Instance Attributes</i> .....	4-4
	DeviceNet Object (03h) .....	4-5
	<i>General Information</i> .....	4-5
	<i>Class Attributes</i> .....	4-5
	<i>Instance #1 Attributes</i> .....	4-5
	Assembly Object (04h) .....	4-6
	<i>General Information</i> .....	4-6
	<i>Class Attributes</i> .....	4-6
	<i>Instance 64h Attributes (Producing Instance)</i> .....	4-6
	<i>Instance 96h Attributes (Consuming Instance)</i> .....	4-6
	Connection Object (05h) .....	4-7
	<i>General Information</i> .....	4-7
	<i>Class Attributes</i> .....	4-7
	<i>Instances #1, #10... #14 (Explicit messaging)</i> .....	4-7
	<i>Instance #2 (Poll or "COS/Cyclic consuming")</i> .....	4-8
	<i>Instance #3 (Bit-strobe)</i> .....	4-9
	<i>Instance #4 (COS/Cyclic producing)</i> .....	4-10
	Acknowledge Handler Object (2Bh) .....	4-11
	<i>General Information</i> .....	4-11
	<i>Class Attributes</i> .....	4-11
	<i>Instances Attributes (01h)</i> .....	4-11
	Vendor Specific Objects .....	4-12
<b>Chapter 5</b>	<b>Fieldbus Interface</b>	
	General Considerations .....	5-1
	Typical Implementation .....	5-1
	Connector Pinout .....	5-1
<b>Appendix A</b>	<b>Application Parameters</b>	
	<i>General Information</i> .....	A-1
	Creating an Application Parameter .....	A-2
	<i>Query - "Application Parameter Object"</i> .....	A-2
	<i>Response - "Application Parameter Object"</i> .....	A-4
	<i>Example</i> .....	A-5
	Mapping an Application Parameter to CIP .....	A-6
	<i>Query - "CIP Mapping Object"</i> .....	A-6
	<i>Response - "CIP Mapping Object"</i> .....	A-7
	<i>Example</i> .....	A-7

---

## Appendix B HMS Object Implementation

General Information.....	B-1
Application Parameter Object (Class 85h).....	B-2
<i>General Information</i> .....	B-2
<i>Class Attributes</i> .....	B-2
<i>Instance Attributes</i> .....	B-2
<i>Create (Class Service)</i> .....	B-3
CIP Mapping Object (Class A5h).....	B-4
<i>General Information</i> .....	B-4
<i>Class Attributes</i> .....	B-4
<i>Instance Attributes</i> .....	B-4
<i>Create (Class Service)</i> .....	B-5

## Appendix C Technical Specification

Electrical Specification.....	C-1
<i>Protective Earth (PE) Requirements</i> .....	C-1
<i>Power Supply</i> .....	C-1
Environmental Specification.....	C-1
EMC Compliance (CE).....	C-2

## About This Document

## How To Use This Document

This document is intended to be used as a supplement to the Anybus-IC 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-IC Design Guide for further information about the Anybus-IC 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.

<b>Warning:</b>	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.
-----------------	--

<b>ESD Note:</b>	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-IC Design Guide	HMS
Anybus-IC Profibus Fieldbus Appendix	HMS
Anybus-IC Ethernet/IT Fieldbus Appendix	HMS
Anybus-IC Ethernet/IP Fieldbus Appendix	HMS
Planning and Installation Manual, DeviceNet Cable System	ODVA
-	-

## Document History

### Summary of Recent Changes (v1.51... v1.53)

Change	Page(s)
Updated Frontpage Contact info and Support info	1, P-3
Added parameters #112 DeviceNet Revision and #119 FB Serial Number in section 'Fieldbus Specific Parameters'	3-7, 3-10
Updated environmental specifications for temperature range in section 'Environmental Specification'	C-1
Updated info regarding parameter #112 and #119	3-1, 3-7, 3-10

### Revision List

Revision	Date	Author(s)	Chapter(s)	Description
<=1.22	-	-	-	(see previous releases)
1.50	2006-08-30	PeP	-	Major rewrite
1.51	2007-09-09	PeP	-	Minor update
1.52	2008-11-07	HeS	P, 3, C	Minor update
1.53	2008-11-13	HeS	3	Minor update

## 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’ is 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.
- Binary values are written in the format NNNNb, where NNNN is the binary value.
- 16/32 bit values are written in big endian Motorola format
- Floating point values are in the IEEE Standard 754 format

## Support

### HMS Sweden (Head Office)

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

### HMS North America

E-mail: [us-support@hms-networks.com](mailto:us-support@hms-networks.com)  
Phone: +1-312-829-0605  
Toll Free: 888-8-Anybus  
Fax: +1-312-738-5873  
Online: [www.anybus.com](http://www.anybus.com)

### HMS Germany

E-mail: [ge-support@hms-networks.com](mailto:ge-support@hms-networks.com)  
Phone: +49-721-96472-0  
Fax: +49-721-964-7210  
Online: [www.anybus.com](http://www.anybus.com)

### HMS Japan

E-mail: [jp-support@hms-networks.com](mailto:jp-support@hms-networks.com)  
Phone: +81-45-478-5340  
Fax: +81-45-476-0315  
Online: [www.anybus.com](http://www.anybus.com)

### HMS China

E-mail: [cn-support@hms-networks.com](mailto:cn-support@hms-networks.com)  
Phone: +86 10 8532 3023  
Online: [www.anybus.com](http://www.anybus.com)

### **HMS Italy**

E-mail: [it-support@hms-networks.com](mailto:it-support@hms-networks.com)  
Phone: +39 039 59662 27  
Fax: +39 039 59662 31  
Online: [www.anybus.com](http://www.anybus.com)

### **HMS France**

E-mail: [mta@hms-networks.com](mailto:mta@hms-networks.com)  
Phone: +33 (0) 3 89 32 76 41  
Fax: +33 (0) 3 89 32 76 31  
Online: [www.anybus.com](http://www.anybus.com)

# About the Anybus-IC DeviceNet

## General

The Anybus-IC DeviceNet communication module provides instant DeviceNet adapter functionality through the generic Anybus-IC 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.

## Features

- Galvanically isolated bus electronics
- Up to 144 bytes of fieldbus I/O in each direction
- Group 2 slave operation
- UCMM Capable (up to 5 explicit server connections)
- Device identity customization
- Generic EDS-file provided
- Supports Auto Baud
- Quick Connect capable
- Application Parameters (used for Explicit Messaging)

## Fieldbus Conformance Notes

The Anybus-IC DeviceNet has been tested standalone by ODVA's authorized Independent Test Lab and found to comply with ODVA Conformance Test Software Version A12.

However, to comply with the ODVA conformance test policy, the end product must be conformance tested as a complete product.

For more information, please contact HMS Industrial Networks or the ODVA.

# Basic Operation

## General Information

### Software Requirements

Generally, no network-specific support code needs to be written in order to support the Anybus-IC DeviceNet. However, due to natural reasons, advanced fieldbus-specific features may require the use of certain DeviceNet-specific parameters.

For general information about the Anybus-IC software interface, consult the Anybus-IC Design Guide.

See also...

- 3-1 “Fieldbus Specific Parameters”

### Electronic Data Sheet (EDS)

On DeviceNet, the characteristics of a device is stored in an ASCII data file with the suffix ‘EDS’. This file is used by DeviceNet configuration tools etc. when setting up the network.

HMS provides a generic EDS-file, which corresponds to the default settings in the module. However, due to the flexible nature of the Anybus concept, it is possible to alter the behaviour of the product in ways that invalidates the generic EDS-file.

See also...

- 1-1 “Fieldbus Conformance Notes”

## Device Identity

On the DeviceNet network, the module identifies itself as follows:

Information	Default Value	Comments
Vendor ID	005Ah	(HMS Industrial Networks)
Product Type	000Ch	(Communications Adapter)
Product Code	003Dh	(Anybus-IC)
Product Name	'Anybus-IC DEV'	-

The information above can be customized. Note that in such case, the EDS-file needs to be altered as well. Note that the identity information is protected by a password to prevent unintentional changes. This password can be obtained by contacting HMS.

**Note:** 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 supplied using parameter #102 (‘FB Password’).

See also...

- 3-3 “FB Password (#102)”
- 3-6 “DeviceNet Vendor ID (#108)”
- 3-6 “DeviceNet Product Type (#109)”
- 3-6 “DeviceNet Product Code (#110)”
- 3-7 “DeviceNet Product Name (#111)”

---

## Communication Settings

### Baudrate

On DeviceNet, the module supports 125kbps, 250kbps and 500kbps operation. The baudrate is specified either via the on-board switches (in case of binary switches), via the SCI-interface, or by the network. Automatic baudrate detection is supported, but not enabled by default.

See also...

- Anybus-IC Design Guide (parameter #8 ‘Configuration Bits’)
- 2-5 “Switches (Fieldbus Specific Input)”
- 2-6 “Auto/Net”
- 3-5 “DeviceNet Baud rate Config (#106)”
- 3-5 “DeviceNet Baud rate Actual (#107)”
- 3-9 “FB Auto Baud Cfg (#117)”

### Mac ID

The module supports BCD-coded switches (range 0... 99) as well as binary switches (range 0... 126). The node address can also be set via the SCI-interface or via the network.

See also...

- Anybus-IC Design Guide (parameter #8 ‘Configuration Bits’)
- 2-5 “Switches (Fieldbus Specific Input)”
- 3-3 “FB Node Address Config (#103)”
- 3-4 “SCC FB Node Address (#104)”
- 3-4 “FB Actual Node Address (#105)”

### Quick Connect

The module supports Quick Connect, a DeviceNet-specific feature which enables faster connect times by allowing devices to go online before the duplicate address check has completed.

See also...

- 3-10 “FB Quick Connect Cfg (#118)”

## Data Exchange

### General

The module is a Group 2 server with explicit message server/client capabilities. Up to 5 explicit message connections can be open simultaneously (Group 1 or Group 3), and the UCMM supports the Unconnected Explicit Message Request port, Group 3, Message ID = 6.

See also...

- 4-7 “Connection Object (05h)”

### Fieldbus I/O

The module supports up to 144 bytes of fieldbus I/O in each direction. On DeviceNet, this data is represented through dedicated instances in the Assembly Object.

The module supports Polled, Change Of State (COS), Cyclic, and Bit strobe connections. The bit strobe connection outputs 1 bit of data and this bit is located in byte 0, bit 0 in the Fieldbus I/O. The bit strobe bit is only output if no other I/O connection is set up to the module.

See also...

- 4-7 “Connection Object (05h)”
- 4-6 “Assembly Object (04h)”

### Application Parameters (Explicit Data)

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

See also...

- 4-12 “Vendor Specific Objects”
- A-1 “Application Parameters”
- B-1 “HMS Object Implementation”

## Status Indicators (Fieldbus Specific Output)

### General

DeviceNet uses bi-coloured status indications as follows:

Bit	Activates Colour	LED	Comments
0	Green	Module Status	-
1	Red		-
2	Green	Network Status	-
3	Red		-
4... 7	-	-	(not used on DeviceNet)

The standard indications on DeviceNet are as follows:

LED	State	Status
Module Status	Off	No power
	Green	Normal operation
	Green, flashing	Auto Baud in progress
	Red	Unrecoverable Fault(s) detected
	Red, flashing	Recoverable Fault(s) detected
	Alternating Red/Green	Self test in progress
Network Status	Off	Not online - or - no power
	Green	On-line, one or more connections established
	Green, flashing	On-line, no connections established
	Red	Critical link failure
	Red, flashing	Connection timeout
	Alternating Red/Green	Self test in progress

See also...

- Anybus-IC Design Guide (parameter #7 'LED State')

### Compliance Notes

The following issues should be kept in mind when designing for DeviceNet compliance.

- Module Status LED must be labelled 'Module Status' or 'MS'
- Network Status LED must be labelled 'Network Status' or 'NS'

## Switches (Fieldbus Specific Input)

### General

The Fieldbus Specific Input is used for fieldbus-specific configuration settings and supports two types of switches/coding.

- **BCD-coded Switches**

Two switches specifies the Mac ID; one for each decimal digit. Illegal settings (i.e. >63) will be interpreted as 63. The baudrate cannot be specified using switches of this kind.

- **Binary Switches**

This type of switch can be used to specify both Mac ID and baud rate as follows:

b7	b6	b5	b4	b3	b2	b1	b0	Baudrate	Mac ID
-	-	0	0	0	0	0	0	-	0
-	-	0	0	0	0	0	1	-	1
-	-	0	0	0	0	1	0	-	2
-	-	0	0	0	0	1	1	-	3
-	-	...	...	...	...	...	...	-	...
-	-	1	1	1	1	0	1	-	61
-	-	1	1	1	1	1	0	-	62
-	-	1	1	1	1	1	1	-	63
0	0	-	-	-	-	-	-	125kbps	-
0	1	-	-	-	-	-	-	250kbps	-
1	0	-	-	-	-	-	-	500kbps	-
1	1	-	-	-	-	-	-	Auto/Net <sup>a</sup> (see below)	-

a. See 2-6 "Auto/Net"

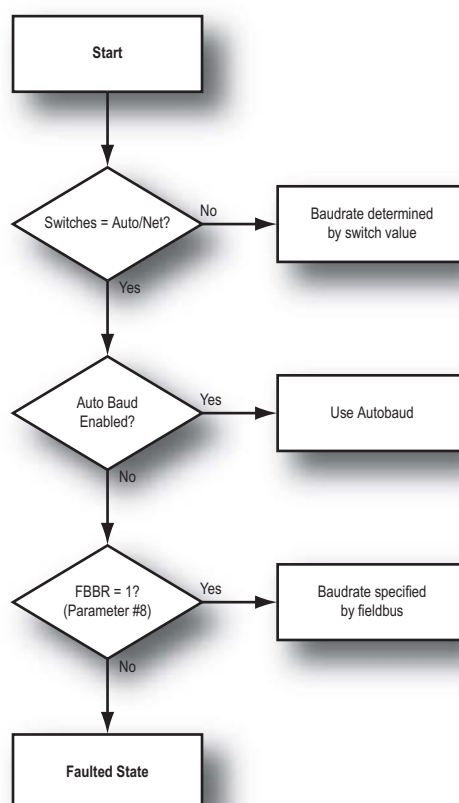
**Note:** The type of switch used is specified in parameter #9 ("Switch Coding"). On DeviceNet, the default value for this parameter is 01h (Binary Switches). However, since this is fieldbus dependant, there is no guarantee that the same type of switch is used by default on other networks.

See also...

- Anybus-IC Design Guide (parameter #9 "Switch Coding")
- 2-2 "Communication Settings"
- 2-6 "Auto/Net"
- 3-3 "FB Node Address Config (#103)"
- 3-4 "SCC FB Node Address (#104)"
- 3-4 "FB Actual Node Address (#105)"
- 3-5 "DeviceNet Baud rate Config (#106)"
- 3-5 "DeviceNet Baud rate Actual (#107)"

## Auto/Net

When the baudrate switches (b6... b7) are set to 'Auto/Net' (11), the baudrate is established according to the following flowchart:



## Compliance Notes

The following issues should be kept in mind when designing for DeviceNet compliance.

- Switches should be placed so that the most significant digit is located to the left or to the top of the product.

## Fieldbus Specific Parameters

The following fieldbus specific parameters are implemented in the module:

#	Modbus Address	Name	Size	Default	Access
100	0x7000	FB Status	2 byte	-	R
102	0x7022	FB Password	2 byte	-	W
103	0x7002	FB Node Address Config	1 byte	0x3F	R/W
104	0x7003	SSC FB Node Address	1 byte	-	R
105	0x7004	FB Actual Node Address	1 byte	-	R
106	0x7005	DeviceNet baudrate config	1 byte	0x00	R/W
107	0x7006	DeviceNet baudrate actual	1 byte	-	R
108	0x7007 <sup>a</sup>	DeviceNet Vendor ID	2 byte	0x005A	R(W)
109	0x7008 <sup>a</sup>	DeviceNet Product Type	2 byte	0x000C	R(W)
110	0x7009 <sup>a</sup>	DeviceNet Product Code	2 byte	0x003D	R(W)
111	0x700A... 0x7019 <sup>a</sup>	DeviceNet Product Name	32 byte <sup>b</sup>	'Anybus-IC DEV'	R(W)
112	0x701A <sup>a</sup>	DeviceNet Revision	2 byte	0x00	R(W)
113	0x701C	Connection Status Polled	1 byte	-	R
114	0x701D	Connection Status COS	1 byte	-	R
115	0x701E	Connection Status Bit Strobed	1 byte	-	R
116	0x701F	Connection Status Cyclic	1 byte	-	R
117	0x7020	FB Auto Baud Cfg	1 byte	0x00	R/W
118	0x7021	FB Quick Connect Cfg	1 byte	0x00	R/W
119	0x7023 <sup>a</sup>	FB Serial Number	4 byte	0x00	R/(W)
200	0x8000...	Application Parameter 1	c	-	RW
201	c	Application Parameter 2	c	-	RW
202	c	Application Parameter 3	c	-	RW
...	...	...	...	...	...
d	cd	Application Parameter 50 <sup>d</sup>	c	-	RW

- a. These parameters can be changed after the correct password has been entered to parameter 102. The command "Set to default" only affect these parameters if the correct password has been entered before.
- b. All 32 bytes must be read/written using a single modbus command
- c. Parameter dependent.
- d. It is possible to register up to 50 Application Parameters with a maximum total size of 1kbyte.

## FB Status (#100)

This parameter holds information about the current bus status.

Parameter number	100
Modbus Address	0x7000
Default value	-
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<sup>1</sup>**
  - 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 (#102)

This parameter is used to unlock the following parameters:

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

(The password value can be obtained by contacting HMS)

Parameter number	102
Modbus Address	0x7022
Default value	-
Range	0x0000h... 0xFFFF
Size	2 bytes
Stored in NV RAM	No
Access	W

## FB Node Address Config (#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 number	103
Modbus Address	0x7002
Default value	0x3F
Range	0x00... 0x3F
Size	1 byte
Stored in NV RAM	Yes
Access	R/W

### SCC FB Node Address (#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 number	104
Modbus Address	0x7003
Default value	-
Range	0x00... 0xFF
Size	1 byte
Stored in NV RAM	No
Access	R

### FB Actual Node Address (#105)

This parameter holds the actual fieldbus node address.

Parameter number	105
Modbus Address	0x7004
Default value	-
Range	0x00... 0x3F
Size	1 byte
Stored in NV RAM	No
Access	R

## DeviceNet Baud rate Config (#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 number	106
Modbus Address	0x7005
Default value	-
Range	0x00... 0x02
Size	1 byte
Stored in NV RAM	Yes
Access	R/W

- **Value**

0x00: 125kbps

0x01: 250kbps

0x02: 500kbps

## DeviceNet Baud rate Actual (#107)

This parameter holds the actual DeviceNet baud rate setting.

Parameter number	107
Modbus Address	0x7006
Default value	-
Range	0x00... 0x02
Size	1 byte
Stored in NV RAM	No
Access	R

- **Value**

0x00: 125kbps

0x01: 250kbps

0x02: 500kbps

## DeviceNet Vendor ID (#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 number	108
Modbus Address	0x7007
Default value	0x005A (HMS Industrial Networks AB)
Range	0x0000... 0xFFFF
Size	2 bytes
Stored in NV RAM	Yes
Access	R(W)

## DeviceNet Product Type (#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 number	109
Modbus Address	0x7008
Default value	0x000C (Communication Adapter)
Range	0x0000... 0xFFFF
Size	2 bytes
Stored in NV RAM	Yes
Access	R(W)

## DeviceNet Product Code (#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 number	110
Modbus Address	0x7009
Default value	0x003D
Range	0x0000h... 0xFFFF
Size	2 bytes
Stored in NV RAM	Yes
Access	R(W)

## DeviceNet Product Name (#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 number	111
Modbus Address	0x700A... 0x7019
Default value	'AnyBus-IC Dev'
Range	-
Size	32 bytes <sup>a</sup>
Stored in NV RAM	Yes
Access	R(W)

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

## DeviceNet Revision (#112)

This parameter is used to change the Revision.

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

Parameter number	112
Modbus Address	0x701A
Default value	0x00
Range	0x0000... 0xFFFF
Size	2 byte
Stored in NV RAM	Yes
Access	R(W)

## Connection Status Polled (#113)

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

Parameter number	113
Modbus Address	0x701C
Default value	-
Range	0x00... 0xFF
Size	1 byte
Stored in NV RAM	No
Access	R

- **Value**

0x00: Connection non existant

0x01: Connection configuring

0x03: Connection established

0x04: Connection timed out

## Connection Status COS (#114)

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

Parameter number	114
Modbus Address	0x701D
Default value	-
Range	0x00... 0xFF
Size	1 byte
Stored in NV RAM	No
Access	R

- **Value**

- 0x00: Connection non existant
- 0x01: Connection configuring
- 0x03: Connection established
- 0x04: Connection timed out

## Connection Status Bit Strobed (#115)

This parameter returns the connection status for bitstrobed connections.

Parameter number	115
Modbus Address	0x701E
Default value	-
Range	0x00... 0xFF
Size	1 byte
Stored in NV RAM	No
Access	R

- **Value**

- 0x00: Connection non existant
- 0x01: Connection configuring
- 0x03: Connection established
- 0x04: Connection timed out

## Connection Status Cyclic (#116)

This parameter returns the connection status for cyclic connections.

Parameter number	116
Modbus Address	0x701F
Default value	-
Range	0x00... 0xFF
Size	1 byte
Stored in NV RAM	No
Access	R

- **Value**

- 0x00: Connection non existant
- 0x01: Connection configuring
- 0x03: Connection established
- 0x04: Connection timed out

## FB Auto Baud Cfg (#117)

This parameter enables/disables the auto baud feature.

Parameter number	117
Modbus Address	0x7020
Default value	0x00
Range	0x00... 0x01
Size	1 byte
Stored in NV RAM	Yes
Access	R/W

- **Value**

- 0x00: Auto Baud disabled (default)
- 0x01: Auto Baud enabled

## FB Quick Connect Cfg (#118)

This parameter enables/disables the Quick Connect feature.

Parameter number	118
Modbus Address	0x7021
Default value	0x00
Range	0x00... 0x01
Size	1 byte
Stored in NV RAM	Yes
Access	R/W

- **Value**
  - 0x00: Quick Connect disabled (default)
  - 0x01: Quick Connect enabled

## FB Serial Number (#119)

This parameter is used to change the module serial number. To be able to alter the parameter value, the correct FB password has to be entered in parameter #102.

Parameter number	119
Modbus Address	0x7023
Default value	0x00
Range	0x00000000... 0xFFFFFFFF
Size	4 bytes
Stored in NV RAM	Yes
Access	R(W)

# **CIP Object Implementation**

## **General Information**

This chapter specifies the CIP-objects implementation in the module.

Mandatory Objects:

- 4-2 “Identity Object (01h)”
- 4-4 “Message Router (02h)”
- 4-5 “DeviceNet Object (03h)”
- 4-6 “Assembly Object (04h)”
- 4-7 “Connection Object (05h)”
- 4-11 “Acknowledge Handler Object (2Bh)”

Vendor Specific Objects:

- 4-12 “Vendor Specific Objects”

# Identity Object (01h)

## General Information

### Object Description

-

### Supported Services

Class                   Get Attribute Single

Instance:               Get Attribute Single

Reset

## Class Attributes

#	Access	Name	Type	Comments
1	Get	Revision	UINT	0001h

## Instance #1 Attributes

#	Access	Name	Type	Comments
1	Get	Vendor ID	UINT	005Ah (HMS Industrial Networks AB <sup>a</sup> )
2	Get	Device Type	UINT	000Ch (Communications Adapter <sup>a</sup> )
3	Get	Product Code	UINT	003Dh (Anybus-IC DeviceNet <sup>a</sup> )
4	Get	Revision	Struct of: {USINT, USINT}	Major and minor firmware revision
5	Get	Status	WORD	See 4-3 "Device Status"
6	Get	Serial Number	UDINT	Assigned by HMS at production
7	Get	Product Name	SHORT_STRING	"Anybus-IC Dev" <sup>a</sup>

a. Can be customized, see 2-1 "Device Identity"

## Device Status

bit(s)	Name
0	Module Owned
1	(reserved)
2	Configured <sup>a</sup>
3	(reserved)
4... 7	Extended Device Status: <u>Value:</u> <u>Meaning:</u> 0000b    Unknown 0010b    Faulted I/O Connection 0011b    No I/O connection established 0100b    Non-volatile configuration bad 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)

a. This bit shows if the product has other settings than "out-of-box". The value is set to true if the configured attribute in the Application Object is set and/or the module's NV storage is changed from default.

## Service Details: Reset Service

The module can forward reset requests from the network to the application as interrupts. For more information about network reset handling, consult the general Anybus-IC Design Guide.

There are two types of network reset requests on DeviceNet:

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

By default, the module performs a reset of the module. Optionally, the module can issue an interrupt to the application, which in turn is responsible for resetting itself and the Anybus module.  
See also...

- Anybus-IC Design Guide (Parameter #12 'Interrupt Config', 'RES'-bit)

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

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

If the 'RES'-bit (Parameter #12, bit 5) is set, the module will issue an interrupt to the application. The interrupt cause register will indicate 'Reset' as cause. The application is then responsible for resetting itself and the module.

See also...

- Anybus-IC Design Guide (Parameter #12 'Interrupt Config', 'RES' and 'DES'-bits)

## Message Router (02h)

### General Information

#### Object Description

-

#### Supported Services

Class -

Instance: -

### Class Attributes

-

### Instance Attributes

-

# DeviceNet Object (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	Comments
1	Revision	Get	UINT	0002h

## Instance #1 Attributes

#	Name	Access	Type	Comments
1	MAC ID	Get/Set <sup>a</sup>	USINT	Currently used MacID
2	Baud Rate	Get/Set <sup>a</sup>	USINT	Value: <u>Currently Used Baud rate:</u> 0 125 kbps 1 250 kbps 2 500 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 <sup>b</sup>	BOOL	Value: <u>Meaning</u> True MacID switches has changed since startup False No change
7	Baud rate Switch changed	Get <sup>b</sup>	BOOL	Value: <u>Meaning</u> True Baudrate switches has changed since startup False No change
8	MAC ID Switch value	Get <sup>b</sup>	USINT	Actual value of node address switches
9	Baud rate Switch value	Get <sup>b</sup>	USINT	Actual value of baud rate switches
10	Quick Connect <sup>c</sup>	Get/Set	BOOL	Value: <u>Meaning</u> True Disable Quick Connect (default) False Enable Quick Connect
100	Disable auto baud <sup>c</sup>	Set	BOOL	Value: <u>Meaning</u> True Disable auto baud (default) False Enable auto baud

a. Attribute is settable only if parameter #8 ('Configuration Bits') is set up accordingly

b. Attribute is only present when using the appropriate switches on the Fieldbus Specific Input register.

c. This setting is stored in NV memory.

# Assembly Object (04h)

## General Information

### Object Description

The Assembly object uses static assemblies and holds the Fieldbus I/O. The assembly instance IDs used are in the vendor specific range.

See also...

- 2-3 “Data Exchange”

### Supported Services

Class -

Instance: Get Attribute Single  
Set Attribute Single

## Class Attributes

-

## Instance 64h Attributes (Producing Instance)

#	Name	Access	Type	Comments
3	Produced Data	Get	Array of BYTE	Data corresponds to Fieldbus Input (FB IN)

See also...

- 2-3 “Fieldbus I/O”

## Instance 96h Attributes (Consuming Instance)

#	Name	Access	Type	Comments
3	Consumed Data	Set	Array of BYTE	Data corresponds to Fieldbus Output (FB OUT)

See also...

- 2-3 “Fieldbus I/O”

# Connection Object (05h)

## General Information

### Object Description

-

### Supported Services

Class                   Get Attribute Single

Instance:               Get Attribute Single  
Set Attribute Single

## Class Attributes

#	Name	Access	Type	Comments
1	Revision	Get	UINT	0001h

## Instances #1, #10... #14 (Explicit messaging)

#	Name	Access	Type	Comments
1	State	Get	USINT	<u>Value:</u> <u>State:</u> 0     Non existent 1     Configuring 2     Waiting for connection ID 3     Established 4     Time out 5     Deferred Delete
2	Instance type	Get	USINT	0000h (Explicit messaging connection)
3	Transport Class trigger	Get	BYTE	83h (Server, Transport class 3)
4	Produced connection ID	Get	UINT	-
5	Consumed connection ID	Get	UINT	-
6	Initial Comm Characteristics	Get	BYTE	The message group over which the communication occurs: <u>Value:</u> <u>Message Group</u> 21     Instance #1 33     Instances #10... #14
7	Produced Connection Size	Get	UINT	262 bytes
8	Consumed Connection Size	Get	UINT	262 bytes
9	Expected Packet Rate	Get/Set <sup>a</sup>	UINT	2500ms
12	Watchdog timeout action	Get/Set <sup>a</sup>	USINT	<u>Value:</u> <u>Action:</u> 0001h   Auto delete (default) 0003h   Deferred delete
13	Produced Connection path length	Get	UINT	0000h (No connection path)
14	Produced Connection path	Get	EPATH	-
15	Consumed Connection path length	Get	UINT	0000h (No connection path)
16	Consumed Connection path	Get	EPATH	-
17	Production Inhibit Time	Get	UINT	0000h

a. Not settable on UCMM Explicit Message connections (Instances 10... 14)

**Instance #2 (Poll or “COS/Cyclic consuming”)**

#	Name	Access	Type	Comments
1	State	Get	USINT	<u>Value:</u> <u>State:</u> 0 Non existent 1 Configuring 2 Waiting for connection ID 3 Established 4 Time out
2	Instance type	Get	USINT	0001h (I/O Connection)
3	Transport Class trigger	Get	BYTE	<u>Value:</u> <u>Meaning:</u> 82h Server, Polled, Class 2 80h Server, COS/Cyclic, Class 0, No Ack. 82h Server, COS/Cyclic, Class 2, Ack.
4	Produced connection ID	Get	UINT	<u>Value:</u> <u>Meaning:</u> FFFFh Not consuming (COS/Cyclic) Other CAN ID for transmission
5	Consumed connection ID	Get	UINT	-
6	Initial Comm Characteristics	Get	BYTE	<u>Value:</u> <u>Meaning:</u> 01h Polled - Produces over message group 1 - Consumes over message group 2 F1h COS/Cyclic, No Ack - Consumes only over message group 2 01h COS/Cyclic, Ack - Produces over message group 1 (Ack) - Consumes over message group 2
7	Produced Connection Size	Get	UINT	<u>Value:</u> <u>Meaning:</u> 0000h COS/Cyclic Other Size of Fieldbus Input data (Polled)
8	Consumed Connection Size	Get	UINT	Size of Fieldbus Output data
9	Expected Packet Rate	Get/Set	UINT	-
12	Watchdog timeout action	Get	USINT	0000h (Transition to the timed out state)
13	Produced Connection path length	Get	UINT	0000h (COS/Cyclic) 0007h (Polled)
14	Produced Connection path	Get	EPATH	No value (COS/Cyclic) 20 04 25 64 00 30 03h (Polled)
15	Consumed Connection path length	Get	UINT	0007h
16	Consumed Connection path	Get	EPATH	20 04 25 96 00 30 03h
17	Production Inhibit Time	Get	UINT	0000h

**Instance #3 (Bit-strobe)**

#	Name	Access	Type	Comments
1	State	Get	USINT	<u>Value:</u> <u>State:</u> 0        Non existent 1        Configuring 2        Waiting for connection ID 3        Established 4        Time out
2	Instance type	Get	USINT	0001h (I/O Connection)
3	Transport Class trigger	Get	BYTE	82h (Transport class & Trigger Server, Cyclic, Class 2)
4	Produced connection ID	Get	UINT	-
5	Consumed connection ID	Get	UINT	-
6	Initial Comm Characteristics	Get	BYTE	Produces over message group 1 Consumes over message group 2
7	Produced Connection Size	Get	UINT	0... 8 bytes depending on Fieldbus Input size
8	Consumed Connection Size	Get	UINT	0008h
9	Expected Packet Rate	Get/Set	UINT	-
12	Watchdog timeout action	Get	USINT	0000h (Transition to the timed out state)
13	Produced Connection path length	Get	UINT	0007h
14	Produced Connection path	Get	EPATH	20 04 25 64 00 30 03h
15	Consumed Connection path length	Get	UINT	0007h
16	Consumed Connection path	Get	EPATH	20 04 25 96 00 30 03h
17	Production Inhibit Time	Get	UINT	0000h

**Instance #4 (COS/Cyclic producing)**

#	Name	Access	Type	Value
1	State	Get	USINT	<u>Value:</u> <u>State:</u> 0 Non existent 1 Configuring 2 Waiting for connection ID 3 Established 4 Time out
2	Instance type	Get	USINT	0001h (I/O Connection)
3	Transport Class trigger	Get	BYTE	<u>Value:</u> <u>Meaning:</u> 00h Client, Cyclic, Class 0 (No Ack.) 10h Client, COS, Class 0 (No Ack.) 02h Client, Cyclic, Class 2 (Ack.) 12h Client, COS, Class 2 (Ack.)
4	Produced connection ID	Get	UINT	CAN ID for transmission
5	Consumed connection ID	Get	UINT	<u>Value:</u> <u>Meaning:</u> FFFFh Not acknowledged Other CAN ID for reception (Ack.)
6	Initial Comm Characteristics	Get	BYTE	<u>Value:</u> <u>Meaning:</u> 0Fh Producing only over message group 1 (No Ack.) 01h Produces over message group 1 Consumes over message group 2 (Ack.)
7	Produced Connection Size	Get	UINT	Corresponds to Fieldbus Input size
8	Consumed Connection Size	Get	UINT	0000h (Consumes 0 bytes on this connection)
9	Expected Packet Rate	Get/Set	UINT	Timing associated with this connection.
12	Watchdog timeout action	Get	USINT	0000h (Transition to the timed out state)
13	Produced Connection path length	Get	UINT	0007h
14	Produced Connection path	Get	EPATH	20 04 25 64 00 30 03h
15	Consumed Connection path length	Get	UINT	0000h (No ack.) 0005h (Acknowledged)
16	Consumed Connection path	Get	EPATH	No value (No ack.) 20 2B 25 01 00h (Acknowledged)
17	Production Inhibit Time	Get/Set	UINT	0000h

## Acknowledge Handler Object (2Bh)

### General Information

#### Object Description

-

#### Supported Services

Class                   Get Attribute Single

Instance:               Get Attribute Single  
                          Set Attribute Single

### Class Attributes

#	Name	Access	Type	Value
1	Revision	Get	UINT	0001h

### Instances Attributes (01h)

#	Name	Access	Type	Value
1	Acknowledge Timer	Get/Set	UINT	16ms
2	Retry Limit	Get/Set	USINT	01h
3	Producing Connection Instance	Get	UINT	04h

## Vendor Specific Objects

It is possible to create application-specific parameters and map them to a Vendor Specific CIP Class, Instance and Attribute.

See also...

- A-1 “Application Parameters”
- B-1 “HMS Object Implementation”

# Fieldbus Interface

## General Considerations

The recommended connector for DeviceNet is a 5-pole pluggable screw connector (5.08mm). In order to minimize the drop line length, it is recommended to place the connector and the module as close as possible, preferable less than 25mm apart.

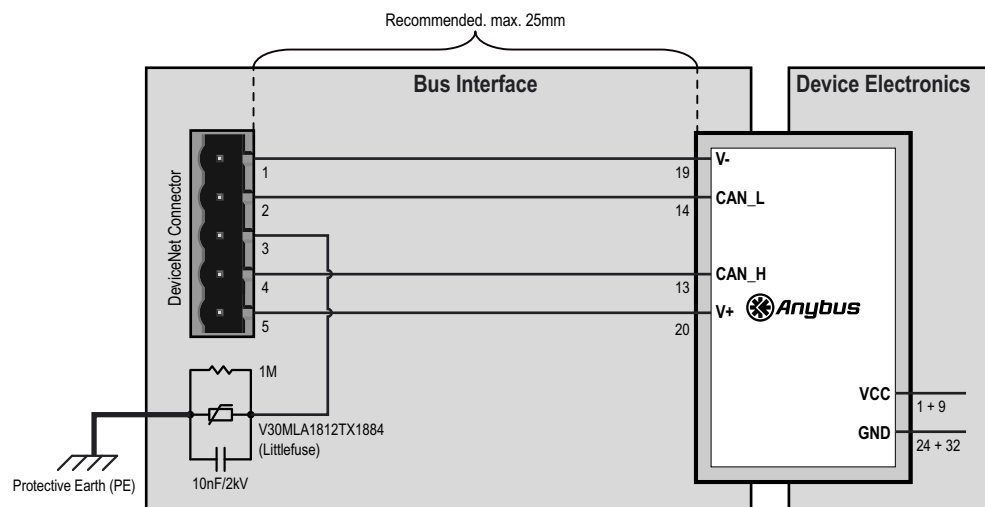
**Note:** Due to the individual requirements (i.e. differences in cable shield filters, max. stub-line length etc.) for each networking system, special care has to be taken if compatibility with several networking systems is required. It is therefore generally recommended to also study the design examples in the fieldbus appendices for Anybus-IC PROFIBUS and Anybus-IC Ethernet/IT/IP.

See also...

- Planning and Installation Manual, DeviceNet Cable System

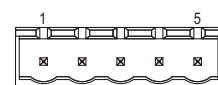
## Typical Implementation

To achieve proper EMC behaviour, DeviceNet requires a filter between Shield and PE as shown below. Also note the varistor (V30MLA1812TX1884), which is required in order to comply with modern CE surge demands.



## Connector Pinout

DeviceNet Connector (5.08mm Pluggable Screw)		Anybus	
Pin	Signal	Pin	Signal
1	V-	19	FB5
2	CAN_L	14	FB2
3	SHIELD	-	-
4	CAN_H	13	FB1
5	V+	20	FB6



# Application Parameters

## General Information

An Application Parameter is an application-specific parameter created by the application during startup. Application Parameters works just like normal parameters and can be accessed via the MIF and SCI interfaces. In addition, Application Parameters can be accessed from the network by mapping them to CIP objects.

See also...

- 4-1 “CIP Object Implementation” (4-12 “Vendor Specific Objects”)
- A-2 “Creating an Application Parameter”
- A-6 “Mapping an Application Parameter to CIP”
- B-1 “HMS Object Implementation”

## 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)				
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">Parameter Size (WORD)</td> <td style="width: 25%;">Descriptor (DWORD)</td> <td style="width: 25%;">Parameter Info (Size varies)</td> <td style="width: 25%;">Extension Word (Optional, WORD)</td> </tr> </table>							Parameter Size (WORD)	Descriptor (DWORD)	Parameter Info (Size varies)	Extension Word (Optional, WORD)
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<sup>1</sup>

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
(reserved)						Data Format		Data Type			(reserved)			Write	Read

Write	Meaning
0	Write access not allowed
1	Write access allowed

Read	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 are reserved for future use and must 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 <sup>a</sup> (String, 16 bytes)	Unit <sup>a</sup> (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" <sup>a</sup>
Unit	String (16 byte, null terminated)	Unit, e.g "RPM" <sup>a</sup>

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

- **Data type STRING**

Init Value (STRING, 16 bytes)	Name <sup>a</sup> (STRING, 16 bytes)	Unit <sup>a</sup> (STRING, 16 bytes)
----------------------------------	---	---

Field	Type / Size	Description
Init Value	Specified in 'Parameter Size'	Initial value
Name	String (16 byte, null terminated)	Name of parameter <sup>a</sup>
Unit	String (16 byte, null terminated)	Unit <sup>a</sup>

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 <sup>a</sup> (String, 16 bytes)	Unit <sup>a</sup> (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 <sup>a</sup>
Unit	String (16 byte, null terminated)	Unit <sup>a</sup>

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
0x0000	-
0x0001	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 <sup>a</sup> (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		

## Mapping an Application Parameter to CIP

Acyclic data on the Anybus-IC module is exchanged by means of Application Parameters mapped to Vendor Specific CIP Objects.

The mapping procedure consists of two steps:

- **Creating the Application Parameter**  
(See A-2 “Creating an Application Parameter”)
- **Mapping the created Application Parameter to a CIP 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 CIP 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 CIP 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		

# **HMS Object Implementation**

## **General Information**

The objects described in this chapter can be accessed using the Modbus Object Messaging protocol, and provides access to advanced fieldbus-specific functionality.

The module features the following HOS objects:

- B-2 “Application Parameter Object (Class 85h)”
- B-4 “CIP Mapping Object (Class A5h)”

# Application Parameter Object (Class 85h)

## General Information

### Object Description

This object manages Application Parameters.

See also...

- A-1 “Application Parameters”
- A-2 “Creating an Application Parameter”
- B-4 “CIP Mapping Object (Class A5h)”

### Supported Services

Class:                   Get\_Attribute  
                               Create                    (see B-3 “Create (Class Service)”)

Instance:               -

## Class Attributes

#	Name	Access	Type	Value
1	Revision	Get	Byte	01h
2	No. of Instances	Get	Word	Current no. of instances in class.

## Instance Attributes

#	Name	Access	Type	Value
1	(varies)	(varies)	(varies)	(specified on 'create')
2	Parameter Size	Get	Word	
3	Descriptor	Get	DWord	

## Create (Class Service)

### Service Description

This service creates a new Application Parameter instance.

### Service Request (05h)

#	Contents	Type	Comments
1	Parameter Size	Word	-
2	Descriptor	DWord	Determines access rights, types etc.
3	Min. Value	(varies)	Type depends on Descriptor
4	Max. Value		
5	Initial Value		
6	Name	String[16]	(optional)
7	Unit	String[16]	(optional)

### Service Response (06h)

#	Contents	Type	Comments
1	Instance Number	Word	HOS instance number
2	Parameter Number	Word	Parameter no. of the created parameter

# CIP Mapping Object (Class A5h)

## General Information

### Object Description

Maps Application Parameter instances to CIP-attributes.

See also...

- 2-3 “Data Exchange” (2-3 “Application Parameters (Explicit Data)”)
- 4-1 “CIP Object Implementation” (4-12 “Vendor Specific Objects”)
- A-1 “Application Parameters” (A-6 “Mapping an Application Parameter to CIP”)
- B-2 “Application Parameter Object (Class 85h)”

### Supported Services

Class:                   Get\_Attribute  
                           Create                    (see B-5 “Create (Class Service)”)

Instance:               -

## Class Attributes

#	Name	Access	Type	Value
1	Revision	Get	Byte	01h
2	No. of Instances	Get	Word	Current no. of instances in class.
3	Max. Instance	Get	Word	Max. no. of allowed instances.

## Instance Attributes

#	Name	Access	Type	Value
1	CIP Class	Get	Word	Mapping information
2	CIP Instance	Get	Word	
3	CIP Attribute	Get	Word	
4	HOS Class	Get	Word	
5	HOS Instance	Get	Word	
6	HOS Attribute	Get	Word	
7	Size	Get	Word	Size of attribute data

## Create (Class Service)

### Service Description

This service creates a new mapping instance.

### Service Request (05h)

#	Contents	Type	Comments
1	CIP Class	Word	CIP instance target attribute; Class must be in vendor-specific range
2	CIP Instance	Word	
3	CIP Attribute	Word	
4	HOS Class	Word	Source Application Parameter instance attribute to map
5	HOS Instance	Word	
6	HOS Attribute	Word	
7	Size	Word	Size of attribute

### Service Response (06h)

#	Contents	Type	Comments
-	-	-	-

---

# Technical Specification

## Electrical Specification

### Protective Earth (PE) Requirements

DeviceNet requires a filter circuit between the cable shield and protective earth.

For more information, see 5-1 “Typical Implementation”.

### Power Supply

#### Supply Voltage

The module requires a regulated  $5V \pm 5\%$  DC power supply as specified in the Anybus-IC Design Guide.

#### Power Consumption

Interface	Max.	Typ.
Module Electronics, 5V	85mA	-
Bus Interface, 24V	55mA	-

## Environmental Specification

- **Temperature**

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

Operating: -40 to +85 °C (-40 to 185°F)

Storage: -40 to +85 °C (-40 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)

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

- **Emission:** EN 50 081-2:1993  
Tested per EN 55011:1998, 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

