

Fieldbus Appendix Anybus-IC DeviceNet

Doc. Id. HMSI-168-68
Rev. 2.01

Important User Information

This document is intended to provide a good understanding of the functionality offered by Anybus-IC DeviceNet. The document only describes the features that are specific to the Anybus-IC DeviceNet. For general information regarding the Anybus-IC, consult the Anybus-IC design guides.

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

Liability

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

There are many application 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

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

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

Intellectual Property Rights

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

Trademark Acknowledgements

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

Warning:	This is a class A product. in a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.
ESD Note:	This product contains ESD (Electrostatic Discharge) sensitive parts that may be damaged if ESD control procedures are not followed. Static control precautions are required when handling the product. Failure to observe this may cause damage to the product.

Table of Contents

Preface	About This Document	
	Related Documents.....	7
	Document History	7
	Conventions & Terminology.....	8
	Sales and Support	9
Chapter 1	About the Anybus-IC DeviceNet	
	General.....	10
	Features	10
	Fieldbus Conformance Notes	10
Chapter 2	Basic Operation	
	General Information.....	11
	<i>Software Requirements</i>	11
	<i>Electronic Data Sheet (EDS)</i>	11
	Device Identity	11
	Communication Settings	12
	<i>Baud Rate</i>	12
	<i>Mac ID</i>	12
	<i>Quick Connect</i>	12
	Data Exchange.....	13
	<i>General</i>	13
	<i>Fieldbus I/O</i>	13
	<i>Application Parameters (Explicit Data)</i>	13
	Status Indicators (Fieldbus Specific Output)	14
	<i>General</i>	14
	<i>Compliance Notes</i>	14
	Switches (Fieldbus Specific Input).....	15
	<i>General</i>	15
	<i>Auto/Net</i>	16
	<i>Compliance Notes</i>	16

Chapter 3 Fieldbus Specific Parameters

<i>FB Status (#100)</i>	18
<i>FB Password (#102)</i>	19
<i>FB Node Address Config (#103)</i>	19
<i>SCC FB Node Address (#104)</i>	20
<i>FB Actual Node Address (#105)</i>	20
<i>DeviceNet Band Rate Config (#106)</i>	21
<i>DeviceNet Band Rate Actual (#107)</i>	21
<i>DeviceNet Vendor ID (#108)</i>	22
<i>DeviceNet Product Type (#109)</i>	22
<i>DeviceNet Product Code (#110)</i>	22
<i>DeviceNet Product Name (#111)</i>	23
<i>DeviceNet Revision (#112)</i>	23
<i>Connection Status Polled (#113)</i>	23
<i>Connection Status COS (#114)</i>	24
<i>Connection Status Bit Strobed (#115)</i>	24
<i>Connection Status Cyclic (#116)</i>	25
<i>FB Auto Baud Cfg (#117)</i>	25
<i>FB Quick Connect Cfg (#118)</i>	26
<i>FB Serial Number (#119)</i>	26

Chapter 4 CIP Object Implementation

General Information.....	27
Identity Object (01h)	28
<i>General Information</i>	28
<i>Class Attributes</i>	28
<i>Instance #1 Attributes</i>	28
<i>Service Details: Reset Service</i>	29
Message Router (02h)	30
<i>General Information</i>	30
<i>Class Attributes</i>	30
<i>Instance Attributes</i>	30
DeviceNet Object (03h).....	31
<i>General Information</i>	31
<i>Class Attributes</i>	31
<i>Instance #1 Attributes</i>	31
Assembly Object (04h)	32
<i>General Information</i>	32
<i>Class Attributes</i>	32
<i>Instance 64h Attributes (Producing Instance)</i>	32
<i>Instance 96h Attributes (Consuming Instance)</i>	32
Connection Object (05h)	33
<i>General Information</i>	33
<i>Class Attributes</i>	33
<i>Instances #1, #10... #14 (Explicit messaging)</i>	33
<i>Instance #2 (Poll or “COS/Cyclic consuming”)</i>	34
<i>Instance #3 (Bit-strobe)</i>	35
<i>Instance #4 (COS/Cyclic producing)</i>	36
Acknowledge Handler Object (2Bh)	37
<i>General Information</i>	37
<i>Class Attributes</i>	37
<i>Instances Attributes (01h)</i>	37
Vendor Specific Objects	38

Chapter 5 Fieldbus Interface

General Considerations	39
Typical Implementation	39
Connector Pinout.....	39

Appendix A Application Parameters

<i>General Information</i>	40
Creating an Application Parameter.....	41
<i>Query - “Application Parameter Object”</i>	41
<i>Response - “Application Parameter Object”</i>	43
<i>Example</i>	44
Mapping an Application Parameter to CIP.....	45
<i>Query - “CIP Mapping Object”</i>	45
<i>Response - “CIP Mapping Object”</i>	46
<i>Example</i>	46

Appendix B HMS Object Implementation

General Information.....	47
Application Parameter Object (Class 85h)	48
<i>General Information</i>	48
<i>Class Attributes</i>	48
<i>Instance Attributes</i>	48
<i>Create (Class Service)</i>	49
CIP Mapping Object (Class A5h)	50
<i>General Information</i>	50
<i>Class Attributes</i>	50
<i>Instance Attributes</i>	50
<i>Create (Class Service)</i>	51

Appendix C Technical Specification

Electrical Specification	52
<i>Protective Earth (PE) Requirements</i>	52
<i>Power Supply</i>	52
Environmental Specification	52
EMC Compliance (CE)	53

About This Document

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 (2.00... 2.01)

Change	Page(s)
Replaced the obsolete varistor V30MLA1812TX1884 with V30MLA1812NTX2538	39

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
2.00	2012-05-22	KaD	1	Major update
2.01	2012-09-07	KaD	5	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

Sales and Support

Sales		Support	
HMS Sweden (Head Office)			
E-mail:	sales@hms-networks.com	E-mail:	support@hms-networks.com
Phone:	+46 (0) 35 - 17 29 56	Phone:	+46 (0) 35 - 17 29 20
Fax:	+46 (0) 35 - 17 29 09	Fax:	+46 (0) 35 - 17 29 09
Online:	www.anybus.com	Online:	www.anybus.com
HMS North America			
E-mail:	us-sales@hms-networks.com	E-mail:	us-support@hms-networks.com
Phone:	+1-312 - 829 - 0601	Phone:	+1-312-829-0601
Toll Free:	+1-888-8-Anybus	Toll Free:	+1-888-8-Anybus
Fax:	+1-312-629-2869	Fax:	+1-312-629-2869
Online:	www.anybus.com	Online:	www.anybus.com
HMS Germany			
E-mail:	ge-sales@hms-networks.com	E-mail:	ge-support@hms-networks.com
Phone:	+49 (0) 721-989777-000	Phone:	+49 (0) 721-989777-000
Fax:	+49 (0) 721-989777-010	Fax:	+49 (0) 721-989777-010
Online:	www.anybus.de	Online:	www.anybus.de
HMS Japan			
E-mail:	jp-sales@hms-networks.com	E-mail:	jp-support@hms-networks.com
Phone:	+81 (0) 45-478-5340	Phone:	+81 (0) 45-478-5340
Fax:	+81 (0) 45-476-0315	Fax:	+81 (0) 45-476-0315
Online:	www.anybus.jp	Online:	www.anybus.jp
HMS China			
E-mail:	cn-sales@hms-networks.com	E-mail:	cn-support@hms-networks.com
Phone:	+86 (0) 10-8532-3183	Phone:	+86 (0) 10-8532-3023
Fax:	+86 (0) 10-8532-3209	Fax:	+86 (0) 10-8532-3209
Online:	www.anybus.cn	Online:	www.anybus.cn
HMS Italy			
E-mail:	it-sales@hms-networks.com	E-mail:	it-support@hms-networks.com
Phone:	+39 039 59662 27	Phone:	+39 039 59662 27
Fax:	+39 039 59662 31	Fax:	+39 039 59662 31
Online:	www.anybus.it	Online:	www.anybus.it
HMS France			
E-mail:	fr-sales@hms-networks.com	E-mail:	fr-support@hms-networks.com
Phone:	+33 (0) 3 68 368 034	Phone:	+33 (0) 3 68 368 033
Fax:	+33 (0) 3 68 368 031	Fax:	+33 (0) 3 68 368 031
Online:	www.anybus.fr	Online:	www.anybus.fr
HMS UK & Eire			
E-mail:	uk-sales@hms-networks.com	E-mail:	support@hms-networks.com
Phone:	+44 (0) 1926 405599	Phone:	+46 (0) 35 - 17 29 20
Fax:	+44 (0) 1926 405522	Fax:	+46 (0) 35 - 17 29 09
Online:	www.anybus.co.uk	Online:	www.anybus.com
HMS Denmark			
E-mail:	dk-sales@hms-networks.com	E-mail:	support@hms-networks.com
Phone:	+45 (0) 35 38 29 00	Phone:	+46 (0) 35 - 17 29 20
Fax:	+46 (0) 35 17 29 09	Fax:	+46 (0) 35 - 17 29 09
Online:	www.anybus.com	Online:	www.anybus.com
HMS India			
E-mail:	in-sales@hms-networks.com	E-mail:	in-support@hms-networks.com
Phone:	+91 (0) 20 40111201	Phone:	+91 (0) 20 40111201
Fax:	+91 (0) 20 40111105	Fax:	+91 (0) 20 40111105
Online:	www.anybus.com	Online:	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 autobaud
- 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 have been found to comply with ODVA conformance test software version A12. However, in accordance with ODVA's conformance test policy, the final product must still be compliance tested to ensure fieldbus conformance. In order to be able to do this, the vendor information in the fieldbus specific parameters must be customized.

It is strongly recommended to customize the information in the identity object (CIP), to enable the product to appear as a vendor specific implementation rather than a generic Anybus module. ODVA requires that all manufacturers use their own Vendor ID. A Vendor ID can be applied for from ODVA.

The EDS file associated with this product must be altered to match the final implementation. See also "Electronic Data Sheet (EDS)" on page 11 .

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...

- “Fieldbus Specific Parameters” on page 17

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 behavior of the product in ways that invalidates the generic EDS file.

See also...

- “Fieldbus Conformance Notes” on page 10

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 nonvolatile 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...

- “FB Password (#102)” on page 19
- “DeviceNet Vendor ID (#108)” on page 22
- “DeviceNet Product Type (#109)” on page 22
- “DeviceNet Product Code (#110)” on page 22
- “DeviceNet Product Name (#111)” on page 23

Communication Settings

Baud Rate

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

See also...

- Anybus-IC Design Guide (parameter #8 ‘Configuration Bits’)
- “Switches (Fieldbus Specific Input)” on page 15
- “Auto/Net” on page 16
- “DeviceNet Baud Rate Config (#106)” on page 21
- “DeviceNet Baud Rate Actual (#107)” on page 21
- “FB Auto Baud Cfg (#117)” on page 25

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’)
- “Switches (Fieldbus Specific Input)” on page 15
- “FB Node Address Config (#103)” on page 19
- “SCC FB Node Address (#104)” on page 20
- “FB Actual Node Address (#105)” on page 20

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...

- “FB Quick Connect Cfg (#118)” on page 26

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...

- “Connection Object (05h)” on page 33

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...

- “Connection Object (05h)” on page 33
- “Assembly Object (04h)” on page 32

Application Parameters (Explicit Data)

Application parameters are user specific parameters created by the application during startup. See “Application Parameters” on page 40 for more information.

See also...

- “Vendor Specific Objects” on page 38
- “Application Parameters” on page 40
- “HMS Object Implementation” on page 47

Status Indicators (Fieldbus Specific Output)

General

DeviceNet uses bicolored 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	Autobaud 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
	Green	Online, one or more connections established
	Green, flashing	Online, 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 baud rate 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	Baud rate	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	-	-	-	-	-	-	125 kbps	-
0	1	-	-	-	-	-	-	250 kbps	-
1	0	-	-	-	-	-	-	500 kbps	-
1	1	-	-	-	-	-	-	Auto/Net ^a (see below)	-

a. See "Auto/Net" on page 16

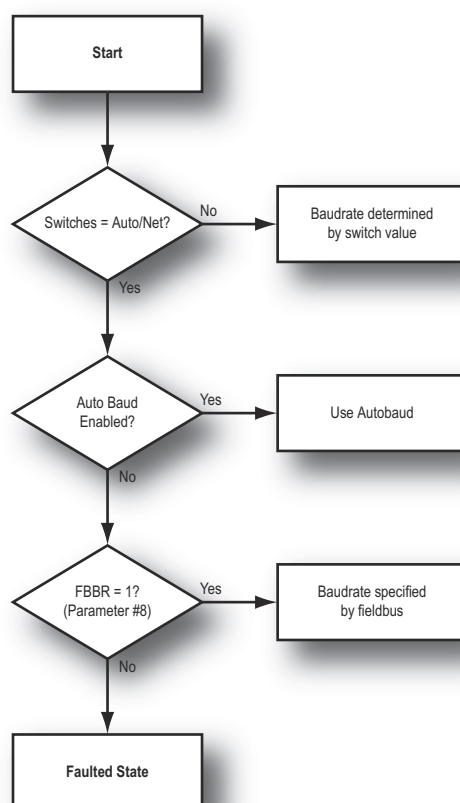
Note: The type of switch used is specified in parameter #9 ('Switch Coding'). On DeviceNet, the default value for this parameter is 01h (Binary Switches). However, since this is fieldbus dependent, there is no guarantee that the same type of switch is used by default on other networks.

See also...

- Anybus-IC Design Guide (parameter #9 'Switch Coding')
- "Communication Settings" on page 12
- "Auto/Net" on page 16
- "FB Node Address Config (#103)" on page 19
- "SCC FB Node Address (#104)" on page 20
- "FB Actual Node Address (#105)" on page 20
- "DeviceNet Baud Rate Config (#106)" on page 21
- "DeviceNet Baud Rate Actual (#107)" on page 21

Auto/Net

When the baud rate switches (b6... b7) are set to 'Auto/Net' (11), the baud rate 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 baud rate config	1 byte	0x00	R/W
107	0x7006	DeviceNet baud rate actual	1 byte	-	R
108	0x7007 ^a	DeviceNet Vendor ID	2 byte	0x005A	R(W)
109	0x7008 ^a	DeviceNet Product Type	2 byte	0x000C	R(W)
110	0x7009 ^a	DeviceNet Product Code	2 byte	0x003D	R(W)
111	0x700A... 0x7019 ^a	DeviceNet Product Name	32 byte ^b	'Anybus-IC DEV'	R(W)
112	0x701A ^a	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 ^a	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 ^d	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 1 kbyte.

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 online
 - 0: Bus is offline
- IDLE¹**
 - 1: Bus is in idle mode
 - 0: Bus is not in idle mode
- PWR**
 - 1: Power is supplied to the bus electronics (24 V)
 - 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 nonvolatile 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 autoconfigured 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 nonvolatile 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: 125 kbps

0x01: 250 kbps

0x02: 500 kbps

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: 125 kbps

0x01: 250 kbps

0x02: 500 kbps

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 ^a
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 nonexistent
 - 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-existent
- 0x01: Connection configuring
- 0x03: Connection established
- 0x04: Connection timed out

Connection Status Bit Strobed (#115)

This parameter returns the connection status for bit strobed 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 nonexistent
- 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 nonexistent
- 0x01: Connection configuring
- 0x03: Connection established
- 0x04: Connection timed out

FB Auto Baud Cfg (#117)

This parameter enables/disables the autobaud 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: Autobaud disabled (default)
- 0x01: Autobaud 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 object implementation in the module.

Mandatory Objects:

- “Identity Object (01h)” on page 28
- “Message Router (02h)” on page 30
- “DeviceNet Object (03h)” on page 31
- “Assembly Object (04h)” on page 32
- “Connection Object (05h)” on page 33
- “Acknowledge Handler Object (2Bh)” on page 37

Vendor Specific Objects:

- “Vendor Specific Objects” on page 38

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 ^a)
2	Get	Device Type	UINT	000Ch (Communications Adapter ^a)
3	Get	Product Code	UINT	003Dh (Anybus-IC DeviceNet ^a)
4	Get	Revision	Struct of: {USINT, USINT}	Major and minor firmware revision
5	Get	Status	WORD	See "Device Status" on page 29
6	Get	Serial Number	UDINT	Assigned by HMS at production
7	Get	Product Name	SHORT_STRING	"Anybus-IC Dev" ^a

a. Can be customized, see "Device Identity" on page 11

Device Status

Bit(s)	Name
0	Module Owned
1	(reserved)
2	Configured ^a
3	(reserved)
4... 7	Extended Device Status: <div> <div>Value:</div> <div>Meaning:</div> <div>0000b Unknown</div> <div>0010b Faulted I/O Connection</div> <div>0011b No I/O connection established</div> <div>0100b Nonvolatile configuration bad</div> <div>0110b Connection in Run mode</div> <div>0111b Connection in Idle mode</div> <div>(other) (reserved)</div> </div>
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 'DEF'-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 ^a	USINT	Currently used MAC ID
2	Baud Rate	Get/Set ^a	USINT	<u>Value:</u> <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 ^b	BOOL	<u>Value:</u> <u>Meaning</u> True MacID switches has changed since startup False No change
7	Baud rate Switch changed	Get ^b	BOOL	<u>Value:</u> <u>Meaning</u> True Baud rate switches has changed since startup False No change
8	MAC ID Switch value	Get ^b	USINT	Actual value of node address switches
9	Baud rate Switch value	Get ^b	USINT	Actual value of baud rate switches
10	Quick Connect ^c	Get/Set	BOOL	<u>Value:</u> <u>Meaning</u> True Disable Quick Connect (default) False Enable Quick Connect
100	Disable auto baud ^c	Set	BOOL	<u>Value:</u> <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...

- “Data Exchange” on page 13

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...

- “Fieldbus I/O” on page 13

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...

- “Fieldbus I/O” on page 13

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 Nonexistent 1 Configuring 2 Waiting for connection ID 3 Established 4 Timeout 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 ^a	UINT	2500 ms
12	Watchdog timeout action	Get/Set ^a	USINT	<u>Value:</u> <u>Action:</u> 0001h Autodelete (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 Nonexistent 1 Configuring 2 Waiting for connection ID 3 Established 4 Timeout
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	<div>Value: State:</div> <div>0 Nonexistent</div> <div>1 Configuring</div> <div>2 Waiting for connection ID</div> <div>3 Established</div> <div>4 Timeout</div>
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 Nonexistent 1 Configuring 2 Waiting for connection ID 3 Established 4 Timeout
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...

- “Application Parameters” on page 40
- “HMS Object Implementation” on page 47

Fieldbus Interface

General Considerations

The recommended connector for DeviceNet is a 5-pole pluggable screw connector (5.08 mm). 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 25 mm apart.

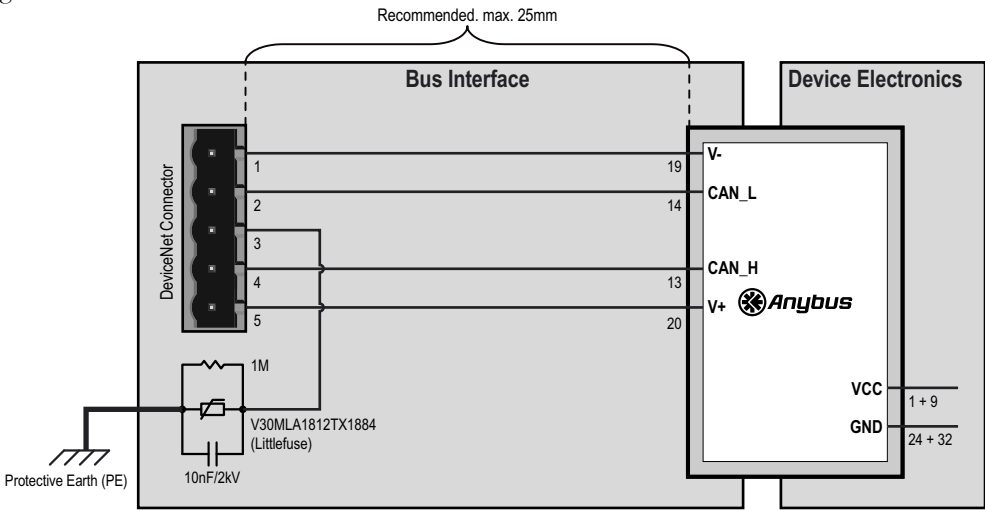
Note: Due to the individual requirements (i.e. differences in cable shield filters, max. stub line length etc.) for each networking system, special care has to be taken if compatibility with several networking systems is required. It is therefore generally recommended to also study the design examples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

See also...

- Planning and Installation Manual, DeviceNet Cable System

Typical Implementation

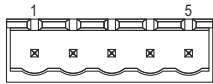
To achieve proper EMC behavior, DeviceNet requires a filter between Shield and PE as shown below. Also note the varistor (V30MLA1812NTX2538), which is required in order to comply with modern CE surge demands.



Connector Pinout

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

a. FB3, FB4, FB7 and FB8 should be left unconnected.



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...

- “CIP Object Implementation” on page 27 (“Vendor Specific Objects” on page 38)
- “Creating an Application Parameter” on page 41
- “Mapping an Application Parameter to CIP” on page 45
- “HMS Object Implementation” on page 47

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 Subfield)

Object Message Subfield

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

Descriptor¹

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 ^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 bytes, null terminated)	Name of parameter, e.g "Speed" ^a
Unit	String (16 bytes, 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 bytes, null terminated)	Name of parameter ^a
Unit	String (16 bytes, 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 bytes, null terminated)	Name of parameter ^a
Unit	String (16 bytes, 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
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 Subfield)

Object Message Subfield

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	
Instance	0000h	
Service Code	0005h	
Attribute	0000h	
Parameter Size	0002h	
Descriptor MSB	0000h	
Descriptor LSB	0003h	
Min value	0000h	
Max value	FFFFh	
Init value	8000h	
Name	53h ('S')	70h ('p')
	65h ('e')	65h ('e')
	64h ('d')	00h
	-	-
	-	-
	-	-
	-	-
	-	-
Unit	72h ('r')	70h ('p')
	6Dh ('m')	00h
	-	-
	-	-
	-	-
	-	-
	-	-
	-	-
Extension Word	0001h	
	CRC	

Application Parameter Class
Create
Parameter Size = 2 bytes
UINT, DEC, R/W
Minimum allowed value: 0
Maximum allowed value: 65535
Initial value: 8000h
“Speed”
“rpm”
Request Modbus Address

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 “Creating an Application Parameter” on page 41)
- **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 Subfield)

Object Message Subfield

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, 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 Subfield)

Object Message Subfield

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:

- “Application Parameter Object (Class 85h)” on page 48
- “CIP Mapping Object (Class A5h)” on page 50

Application Parameter Object (Class 85h)

General Information

Object Description

This object manages application parameters.

See also...

- “Application Parameters” on page 40
- “Creating an Application Parameter” on page 41
- “CIP Mapping Object (Class A5h)” on page 50

Supported Services

Class: Get_Attribute
 Create(see “Create (Class Service)” on page 49)

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...

- “Data Exchange” on page 13 (“Application Parameters (Explicit Data)” on page 13)
- “CIP Object Implementation” on page 27 (“Vendor Specific Objects” on page 38)
- “Application Parameters” on page 40 (“Mapping an Application Parameter to CIP” on page 45)
- “Application Parameter Object (Class 85h)” on page 48

Supported Services

Class: Get_Attribute
 Create(see “Create (Class Service)” on page 51)

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 “Typical Implementation” on page 39 .

Power Supply

Supply Voltage

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

Power Consumption

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

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% noncondensing.

Test performed according to IEC 68-2-30.

EMC Compliance (CE)

EMC precompliance testing has been conducted according to the following standards:

- **Emission:** EN 61000-6-4
Tested per EN 55016-2-3
- **Immunity:** EN 61000-6-2
Tested per EN 61000-4-2
EN 61000-4-3
EN 61000-4-4
EN 61000-4-5
EN 61000-4-6