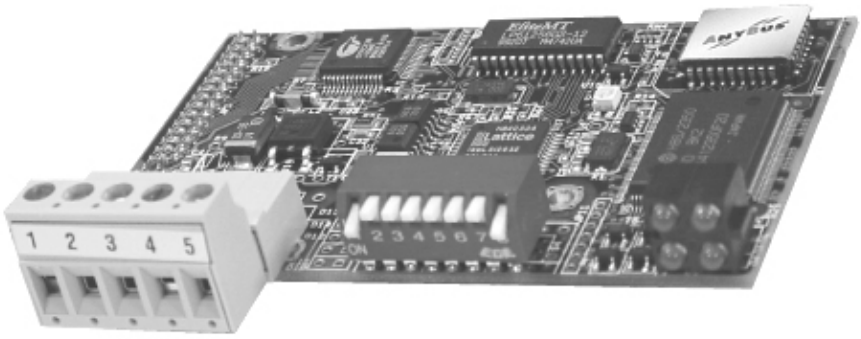
ANYBUS-M DEVICENET MASTER/SCANNER

APPENDIX

Revision 1.02



Revisions

Rev.	Date	Chapter	Author	Description
0.10	1999-11-08	All	HeA	Created
0.20	1999-12-20	All	KaJ	Reviewed
0.30	2000-05-11	All	HeA	Corrections before release
0.40	2000-05-23	All	HeA	Corrections after review
0.50	2000-05-29	3, 6	HeA	Changed pictures
1.00	2000-06-13	All	HeA	Changed Revision to 1.00 for first release
1.02	2003-04-17	4,5	ToT	Error corrections

Table 1 Revisions

Preface

The data and illustrations found in this appendix are not binding. HMS Fieldbus Systems AB reserve the right to modify our products in line with our policy of continuous product development. The information in this appendix is subject to change without notice and should not be considered as a commitment by HMS Fieldbus Systems AB.

HMS Fieldbus Systems AB assumes no responsibility for any errors that may appear in the document.

The product and technology described in this document is patent pending in the following countries:

USA, Canada, Japan, Belgium, Denmark, Finland, Greece, Ireland, Italy, Luxemburg, Monaco, Netherlands, Portugal, Switzerland, Liechtenstein, Spain, United Kingdom, Sweden, Germany, and Austria.

Related Documents

Document	Author
AnyBus-M Design Guide for Parallel Interface	Jörgen Palmhager
AnyBus-M Design Guide for Serial Interface	Jörgen Palmhager
DeviceNet specification Vol I and Vol II Rev 2.0	ODVA

Table of Contents

1. Introduction

This chapter contains an introduction to the AnyBus-M DeviceNet module.

1.1 Company information

HMS Fieldbus Systems AB

Pilefeltsgatan 93-95

S-302 50 Halmstad

Sweden

Phone: +46 (0) 35 172 900

Fax: +46 (0) 35 172 909

Web: http://www.hms.se

http://www.anybus.com

Support Europe: support@hms.se

Support Asia: jp-support@hms.se

Support America: us-support@hms.se

1.2 Overview

ANYBUS® M for DeviceNet is a fieldbus master/scanner module supporting the DeviceNet standard.

ANYBUS® M for DeviceNet follows the ANYBUS® concept, and is compatible with the generic ANYBUS® standard. The application interface is the same as for other ANYBUS®M and ANYBUS®S modules. The module is based on the same hardware platform as the ANYBUS® S for DeviceNet.

ANYBUS® M for DeviceNet can act both as a master/scanner or a slave device in a DeviceNet network. Both the master/scanner and the slave operation modes are configurable with a DeviceNet configuration tool or with mailbox messages sent from the application.

1.3 Reference material and standards

The DeviceNet Specification Revision 2.0 Volume I and Volume II will be used as reference in this document. ODVA (Open DeviceNet Vendor Association) handles the distribution of the DeviceNet specification. For further information, please contact ODVA on e-mail: billmoss@ix.netcom.com or at address:

ODVA - William H. (Bill) Moss, Executive Director

20423 State Road 7 - Suite 499 - Boca Raton, FL 33498 USA

(1) 954 340-5412 or (1) 561 477-7966 Phone

(1) 954 340-5413 or (1) 561 477-6621 Fax

Web: http://www.odva.org

1.4 Abbreviations

Important abbreviations used in this specification:

Abbr.	Description
ACK	Acknowledge
DPRAM	Dual-Port RAM
EMC	Electromagnetic Compatibility
LED	Light Emitting Diode
LSB	Least Significant Bit
MSB	Most Significant Bit
NAK	Negative Acknowledge
NU	Not Used
RO	Read Only
R/W	Read / Write
TBD	To Be Defined
NR	Not relevant
UCMM	Un-Connected Message Manager
MAC ID	Medium Access Identifier
COS	Change Of State
EDS	Electronic Data Sheet

2. Fieldbus introduction

This chapter contains an overview of DeviceNet.

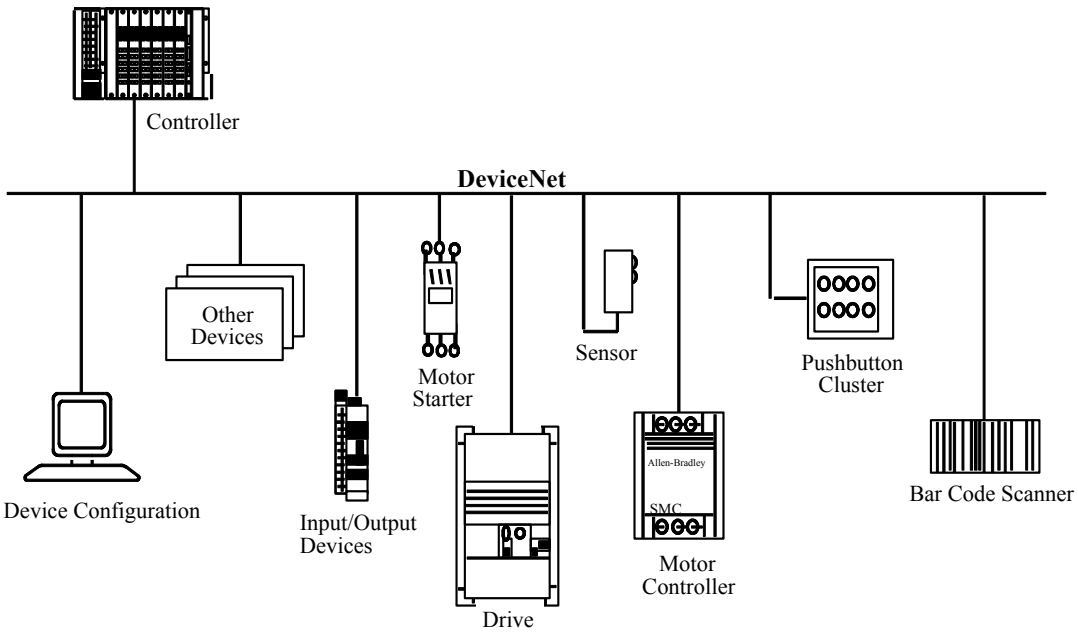
2.1 Introduction to DeviceNet

DeviceNet is a fieldbus system used for industrial automation, normally for the control of valves, sensors and I/O units and other automation equipment. The DeviceNet communication link is based on a broadcast-oriented, communications protocol, the Controller Area Network (CAN). This protocol has I/O response and high reliability even for demanding applications.

DeviceNet has a user organization, the Open DeviceNet Vendor Association (ODVA), which assists members of matters concerning DeviceNet. HMS is a member of ODVA and also represented as a member of the DeviceNet Conformance SIG.

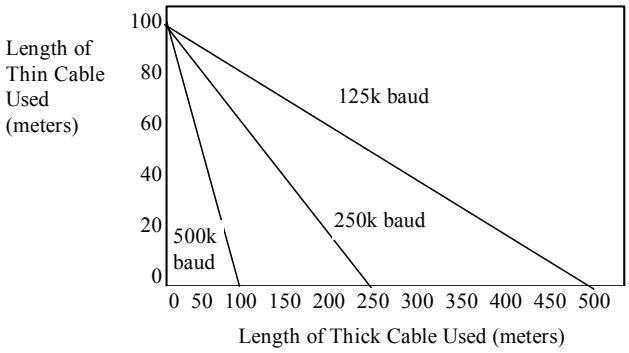
2.2 Network Overview

The physical media for the fieldbus is a shielded copper cable composed of one twisted pair and two cables for the external power supply. The baudrate can be changed between 125k, 250k and 500k bit/sec. Each node in the network is given a MAC ID, which is a number between 0 and 63 and is used to address the node.



2.3 Technical features of DeviceNet

The maximum length of cable is dependent on which baud rate and which DeviceNet cable that is used. Below is a diagram that shows the maximum allowed cable length in the network.



$L_{thick} + 5 \times L_{thin} = 500$ at 125Kbaud
$L_{thick} + 2.5 \times L_{thin} = 250$ at 250Kbaud
$L_{thick} + L_{thin} = 100$ at 500Kbaud

where L_{thick} is the length of thick cable and Lthin is the length of thin cable.

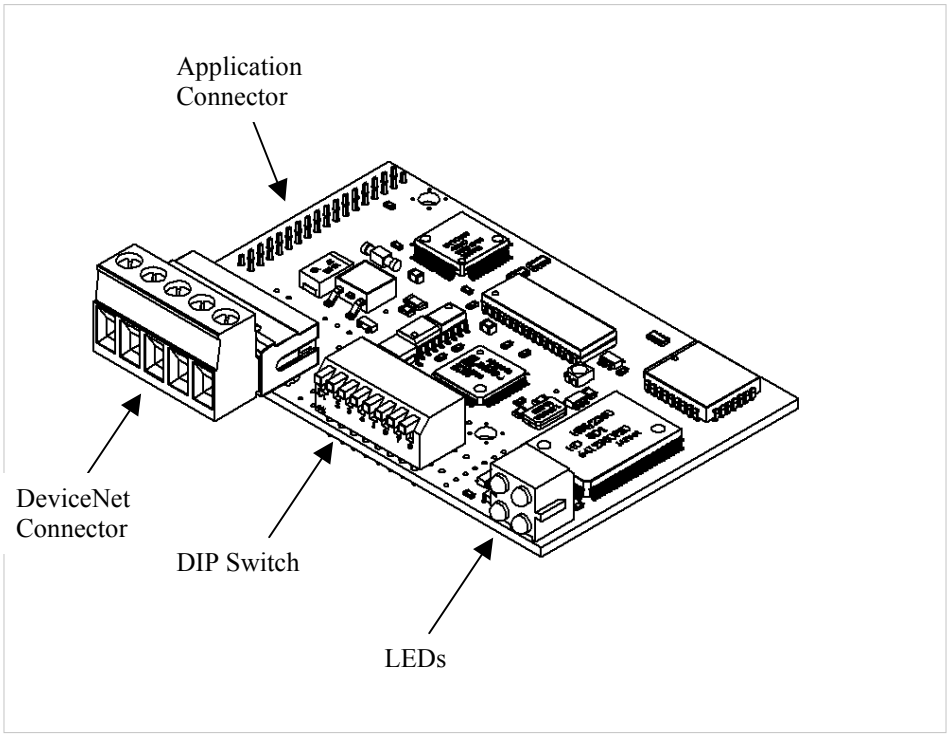
Summary Technical Features DeviceNet

- DeviceNet specific cable (twisted pair)
- Access to intelligence present in low-level devices
- Master/Slave and Peer-to-Peer capabilities
- Trunkline-dropline configuration
- Support for up to 64 nodes
- Node removal without severing the network
- Simultaneous support for both network powered (sensors) and self powered (actuators) devices
- Use of sealed or open style connectors
- Protection from wiring errors
- Selectable data rates of 125k baud, 250k baud, and 500k baud. max. Trunk distance 500 meters and Drop length 156 meters at 125k baud
- Adjustable power configuration to meet individual application needs
- High current capability (up to 16 amps per supply)
- Operation with off-the-shelf power supplies
- Power taps that allow the connection of several power supplies from multiple vendor that comply with DeviceNet standards
- Built-in overload protection
- Power available along the bus: both signal and power lines contained in the trunkline
- Provisions for the typical request/response oriented network communications
- Provisions for the efficient movement of I/O data
- Fragmentation for moving larger bodies of information
- Duplicate MAC ID detection

3. Module Overview

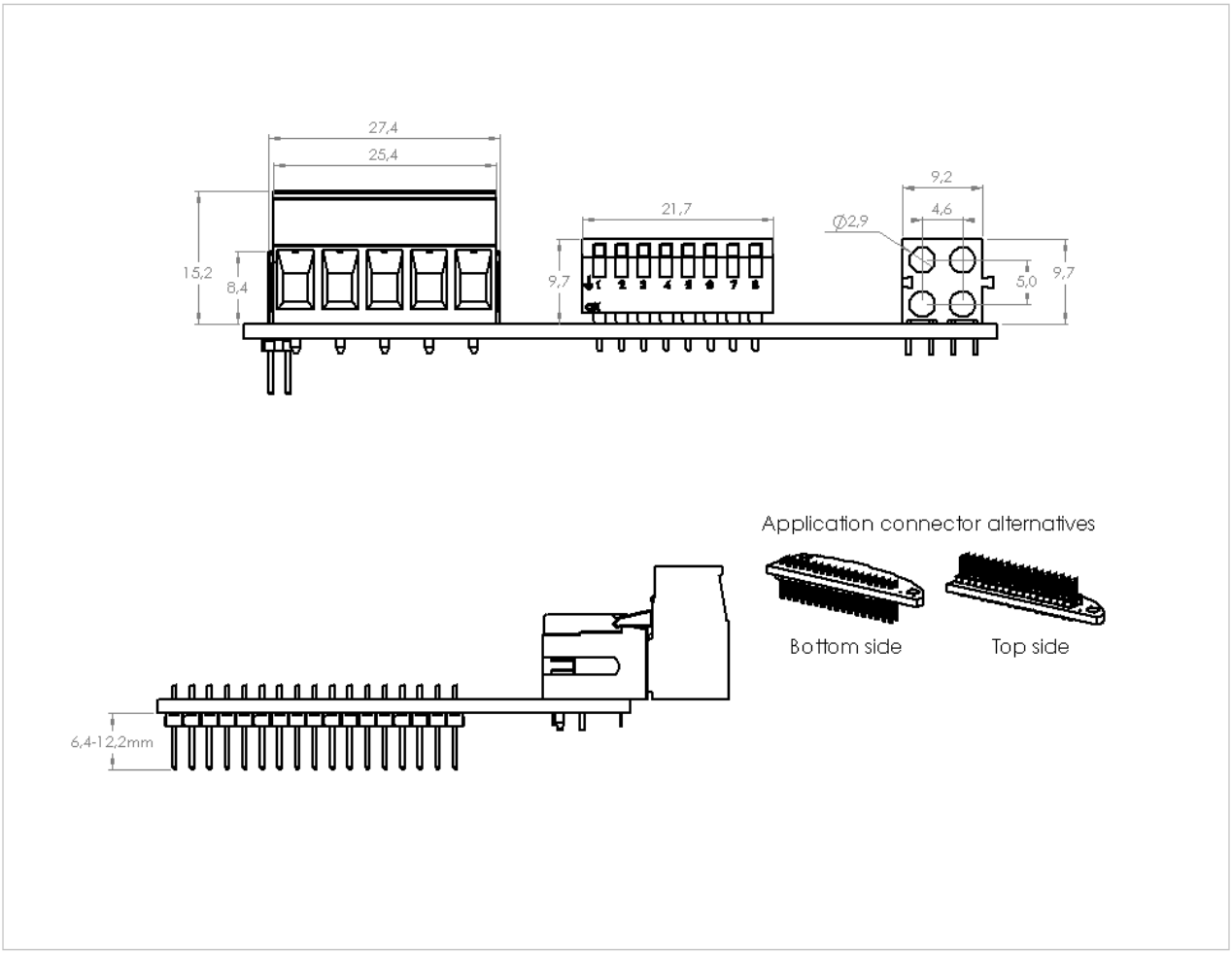
This chapter contains an overview of the AnyBus-M DeviceNet module.

Below is a picture of an AnyBus-M DeviceNet module seen from above. The host connector is used for an application to communicate with the AnyBus module. The bus connector is where the DeviceNet fieldbus cable is connected. DIP-switches are used to configure network baudrate and network address (MAC ID). LED indicators are used to display status.



3.1 Mechanical measurements

The ANYBUS® M DeviceNet follows the mechanical specification for the ANYBUS® S and the ANYBUS® M series (see the ANYBUS® Design guides for more information).



3.2 AnyBus-M DeviceNet Technical Information

Technical data for the AnyBus-M DeviceNet module:

Description	Text string	Dec	Hex
Vendor ID	HMS Fieldbus Systems AB	90	0x5A
Product Type	Communications adapter	12	0x0C
Product Code	-	14	0x0E
Product Name	AnyBus-M DeviceNet	-	-

3.3 Compliance with predefined standards

The ANYBUS® M DeviceNet follows the DeviceNet standard that has been developed by ODVA. It is fully compatible with the DeviceNet specification rev. 2.0 Vol I and Vol II. The module operates according to the communication adapter profile (product type 12, see DeviceNet specification for more information). The module supports the I/O connections Bit strobe, Polled I/O, Change of state and Cyclic I/O data.

DeviceNet Features			
Device Type	Communication adapter	Master/Scanner	Yes
Explicit peer-to-peer messaging	Yes	I/O slave messaging	
I/O peer-to-peer messaging	No	• Bit strobe	Yes
Configuration consistency value	Yes	• Polling	Yes
Faulted node recovery	No	• Cyclic	Yes
Baudrates	125K, 250K, 500K	• Change of state (COS)	Yes

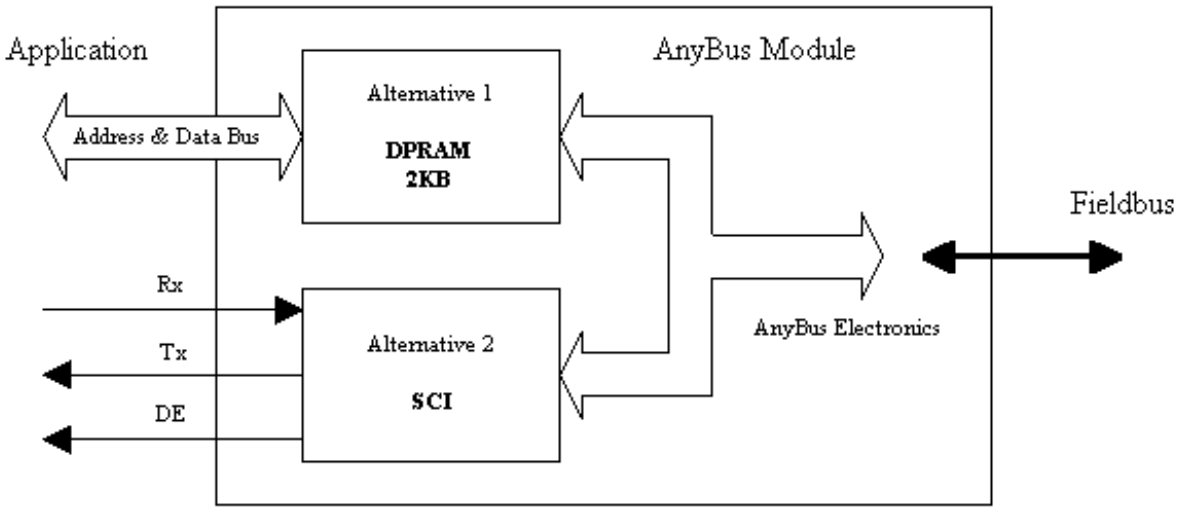
DeviceNet™
CONFORMANCE TESTED

The ANYBUS® M DeviceNet has been conformance tested inside customer's products by ODVA's authorized Independent Test Lab. The products have been found to comply with the latest ODVA Conformance Test Software (1999-10-06 the composite test version is A12). When this has been done, the Identity information of the AnyBus-M DeviceNet module was changed to the customers Vendor IDs, Product Codes, and Product Names.

The module has not been tested standalone with HMS Identity information, since ODVA's conformance test policy has been changed. A product that uses a daughter card (i.e. an AnyBus module) to communicate with a DeviceNet network is not certified even if the daughter card (the AnyBus module) is certified. The whole assembled product needs to be re-certified. For more information regarding conformance testing, please contact HMS or ODVA.

4. Host Application Interface

This chapter describes the application interface of the AnyBus-M DeviceNet module. This interface is fully compatible with the patented standard AnyBus interface, and has been adapted to the features of a DeviceNet Master/Scanner.



The AnyBus master module use a parallel DPRAM interface like the other AnyBus module. The parallel interface contains two data areas, DPRAM data and extended data. The DPRAM data is stored in the DPRAM, and the extended data is stored in the internal memory. The DPRAM data is memory mapped, and the extended data is accessed with mailbox messages.

Parameter data is not used in the same way as on the AnyBus-S module, where it was possible to store parameter data in the DPRAM or the extended memory. Instead all these data areas are dedicated for I/O data, and parameter data is stored in the application. When the module shall exchange parameter data, this is done with mailbox messages towards the application.

4.1 DPRAM Memory Map

Below is a picture of the memory map of the DPRAM on the AnyBus module. Please see the AnyBus Design Guides for more information.

Address	*Memory Area*	*Description*
000h - 1FFh	**Input data** (512 byte)	*Fieldbus input data* *Application ⇒ AnyBus module*
200h - 3FFh	**Output data** (512 byte)	*Fieldbus output data* *Application ⇐ AnyBus module*
400h – 51Fh	**Mailbox in** (288 byte)	*Mailbox input data* *Application ⇒ AnyBus module*
520h – 63Fh	**Mailbox out** (288 byte)	*Mailbox output data* *Application ⇐ AnyBus module*
640h - 7BFh	**Fieldbus specific data** (384 byte)	*Fieldbus specific data* *Application ⇔ AnyBus module*
7C0h - 7FDh	**Control registers** (62 byte)	*Control registers* *Application ⇔ AnyBus module*
7FEh - 7FFh	**Handshake registers** (2 byte)	*Handshake registers* *Application ⇔ AnyBus module*

4.2 DPRAM Input Data Area

The DPRAM input area consists of 512 bytes. Data that shall be sent to the DeviceNet network shall be written here. To write data in this area, the application has to ask the module for access. All data that is written in this area is area consistent, since only data is written when access is granted (see ANYBUS® design guides for more information about the handshaking protocol).

4.3 DPRAM Output Data Area

The DPRAM output area consists of 512 bytes. Data that is received from the DeviceNet network is displayed here. To read the data in this area, the application has to ask the module for access. All data that is read from this area is consistent, since only data is read when access is granted (see ANYBUS® design guides for more information about the handshaking protocol).

4.4 Control Registers

The registers in the control area display status and data for the module.

4.5 Fieldbus Specific Area

The fieldbus specific area contains information that is specific for each AnyBus module. The fieldbus specific area for the AnyBus-M module is a read only area from the application side, and is described on the following pages.

Address	Size (Bytes)	Name	Description
640h-647h	8	Module Status Area	Used for module info. From FB to APP
648h-64Fh	8	Node Active (bit-field)	Nodes that are configured in the master. From FB to APP
650h-657h	8	Node Idle (bit-field)	Nodes that are in configuring state. From FB to APP
658h-65Fh	8	Node Faulted (bit-field)	Nodes that are faulted. From FB to APP
660h-69Fh	64	Node Status (byte-field)	Status of the nodes. From FB to APP
6A0h-7BFh		Reserved	

4.5.1 Module Status Area

The module status area is used to display fieldbus specific information about current settings and current status of the module.

Address	Usage	Description	Data type
640 – 641	Active Connections	The number of established connections	UINT
642	Reserved		USINT
643	Reserved		USINT
644 – 645	Expected packet rate	The expected packet rate for the I/O communication	UINT
646	DIP settings	The settings for Baud rate and MAC ID	USINT
647	Scan flags	Idle mode or run mode is shown here	USINT

The number of connections that have been established between the AnyBus-M DeviceNet module, and other DeviceNet nodes in the Network are shown on address 0x640-0x641.

Expected packet rate for the polled and bit-strobe I/O connections are shown on address 0x644-0x645.

The value that the DIP has is displayed on address 646. The DIP sets the MAC ID and Baud rate. See chapter 5 for more information.

The parameter called Scan flags, contain information about the operation mode of the scanner. The major purpose of this parameter is to display the idle or run mode of the module. See table below for information about the meaning of the bits in the register. The module uses the reserved bits internally, but they do not give the user any useful information.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Strobe active	Reserved	Idle mode	Reserved	ISD active	Poll	Reserved

The bit “Poll” is set when the ISD timer has expired and the master shall produce data on established poll connections.

The bit “ISD active” (Inter Scan Delay) is set while the master is waiting for the Inter Scan Delay timer to expire. When the timer has expired, the master produces and sends data on all I/O connections that are using the ISD timer.

The “Idle mode” bit is set when the master is in idle mode.

The “Strobe active” bit is set when the master has any active bit-strobe connections.

4.5.2 Node Active Area

The Node Active Area is an 8 byte long bit-array, which contains information about which nodes that are configured in the master. If the bit is set (= 1), the node is configured in the scanlist, and the master will try to establish connections to the node. If the bit is cleared (= 0), the node is not configured, and the master will not communicate with the node.

Note: Node means a module in the network corresponding to a certain MAC ID.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
648h	Node 7	Node 6	Node 5	Node 4	Node 3	Node 2	Node 1	Node 0
649h	Node 15	Node 14	Node 13	Node 12	Node 11	Node 10	Node 9	Node 8
⇓	…	…	…	…	…	…	…	…
64Fh	Node 63	Node 62	Node 61	Node 60	Node 59	Node 58	Node 57	Node 56

4.5.3 Node Idle Area

The Node Idle Area is an 8 byte long bit-array, which tells which nodes in the network that are in idle (or configuring) mode. If the bit is set (= 1), the node is idle. If the bit is cleared (= 0), the node is not idle, i.e. it is in run mode.

Note: Node means a module in the network corresponding to a certain MAC ID.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
650h	Node 7	Node 6	Node 5	Node 4	Node 3	Node 2	Node 1	Node 0
651h	Node 15	Node 14	Node 13	Node 12	Node 11	Node 10	Node 9	Node 8
⇓	…	…	…	…	…	…	…	…
657h	Node 63	Node 62	Node 61	Node 60	Node 59	Node 58	Node 57	Node 56

4.5.4 Node Faulted Area

The Node Faulted Area is an 8 byte long bit-array, which tells which nodes in the network that are not running correctly. If the bit is set (= 1), the corresponding node is faulted. If the bit is cleared (= 0), the node is operating correctly. For more information about the fault of the node, see the corresponding information in the Node Status Area.

Note: Node means a module in the network corresponding to a certain MAC ID or Node ID.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
658h	Node 7	Node 6	Node 5	Node 4	Node 3	Node 2	Node 1	Node 0
659h	Node 15	Node 14	Node 13	Node 12	Node 11	Node 10	Node 9	Node 8
⇓	…	…	…	…	…	…	…	…
65Fh	Node 63	Node 62	Node 61	Node 60	Node 59	Node 58	Node 57	Node 56

4.5.5 Node Status Area

The node-status field is a 64 bytes long array, which tells the status of the nodes in the network. If any node is faulted, an error code shall be presented here that describes the fault (if possible). The Status of the ANYBUS® M module is also presented here, in the byte that corresponds to the MAC ID of the module.

Note: Node means a module in the network corresponding to a certain MAC ID or Node ID.

Address	Corresponding Node
660h	Status for Node 0
661h	Status for Node 1
662h	Status for Node 2
⇓	…
69Fh	Status for Node 63

The following table describes the meaning of the values that can appear in the Node status area. Some of these parameters may only appear in the Node Status byte that corresponds to the masters MAC ID.

Value Dec	Value Hex	Meaning
00	0x00	OK or Not in scan list
70	0x46	Duplicate MAC ID failure
71	0x47	Scanner configuration error
72	0x48	Device communication error
73	0x49	Wrong device type
74	0x4A	Port over-run error
75	0x4B	Network failure
76	0x4C	No CAN messages detected
77	0x4D	Wrong data size
78	0x4E	No such device found
79	0x4F	Transmit failure
80	0x50	Node in IDLE mode
81	0x51	Node in fault mode
82	0x52	Fragmentation error
83	0x53	Unable to initialise node

Value Dec	Value Hex	Meaning
84	0x54	Node not yet initialised
85	0x55	Receive buffer overflow
86	0x56	Node changed to IDLE mode
87	0x57	Shared master error (not used)
88	0x58	Shared choice error (not used)
89	0x59	Keeper object failure (not used)
90	0x5A	CAN port disabled (not used)
91	0x5B	Bus off
92	0x5C	No bus power detected
95	0x5F	Updating flash (not used)
96	0x60	In test mode (not used)
97	0x61	Halted by user cmd. (not used)
98	0x62	Firmware failure (not used)
99	0x63	System failure

4.6 Start-up/initialisation sequence

After power on the AnyBus-M module will be waiting for an initialisation sequence. This initialisation sequence consists of a series of mailbox messages. The AnyBus Design guides describe how this initialisation mailbox messages shall be sent to the module.

The AnyBus-M module is only possible to initialise with two different I/O sizes. The first one is with DPRAM memory only, and the second is with full data size, both DPRAM memory and extended memory. Unlike the AnyBus-S modules, it is only possible to initialise the module to these specific pre-defined sizes, since the usage of the I/O areas in the DPRAM are defined when the module is configured.

The I/O sizes that can be used in the AnyBus init command are described below:

	DPRAM only [dec]	DPRAM only [hex]	Full size [dec]	Full size [hex]
I/O size	512	0x200	2048	0x800
DPRAM size	512	0x200	2048	0x800
Total size	512	0x200	2048	0x800

NOTE: The full size configurations are not available in the first release, i.e. only the DPRAM will be available to read and write I/O data. The extended memory area is not used in this version.

4.7 System tests

4.7.1 LED Test

After initialisation a LED test is performed. This test shall be performed according to the DeviceNet specification, see table below.

Test	MS LED	NS LED	Vendor Specific LED	WD LED
1	lit RED 0.25s → lit green 0.25s	Turned off	Turned off	Standard indication
2	Lit green	lit green 0.25s → lit red 0.25s	Turned off	Standard indication
3	Lit green	Turned off	Turned off	Standard indication
END	Standard indication	Standard indication	Standard indication	Standard indication

4.7.2 Hardware self test

A mailbox telegram is implemented in the same way as in the AnyBus DeviceNet module that do a hardware self test. This telegram can be called before the module is initialised, see AnyBus-S Design Guide for further information.

4.8 Normal operation

At normal operation, there are two modes, **run mode** and **idle mode**. In both modes, the application interface works in the same way, i.e. mailbox messages can be exchanged and data can be written and read to/from the I/O areas. In idle mode, no I/O data will be sent out from the master to the network.

4.8.1 Run mode

In run mode, the module will be capable of exchanging data with all configured nodes that have one ore more established I/O connections. Data is written into the input area in the DPRAM from the application. The data in the DPRAM is mapped to the nodes on the network, which means that when the module will exchange data with a node, it will check which bytes in the DPRAM (or the extended memory) that is mapped to the node.

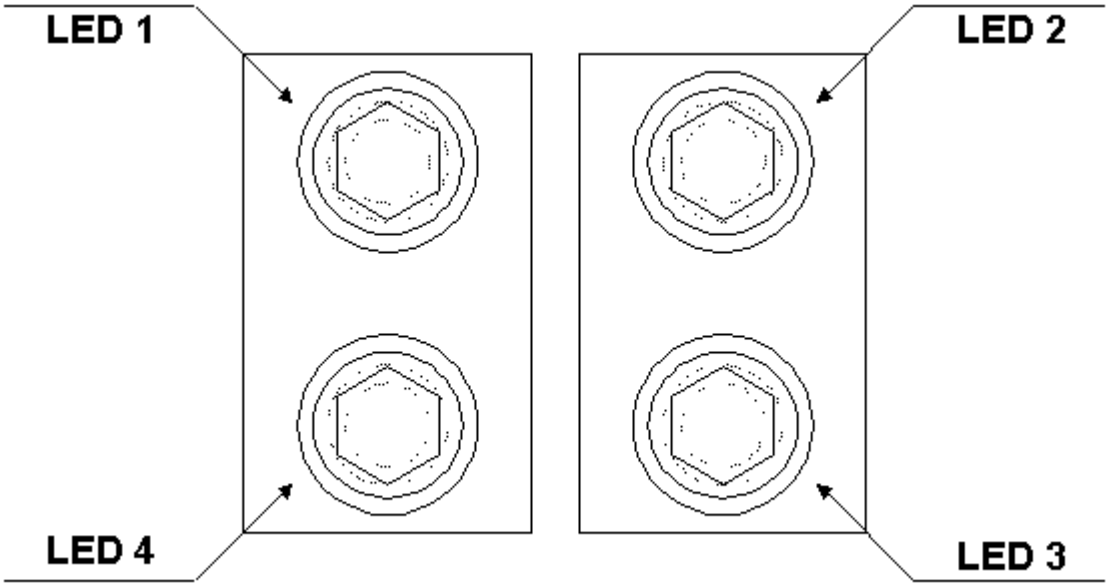
4.8.2 Idle mode

In idle mode, it is possible to change the configuration of the master. Nodes can be added and removed from the scan list and corresponding I/O data can be mapped into the DPRAM and the extended I/O areas. No I/O data is sent to the network when the module is in this mode, since it is possible to change the configuration.

4.9 Indications

The ANYBUS® M DeviceNet module is equipped with four LED indicators that are mounted on the front of the module. There is also an additional on-board watchdog LED. These are used for displaying the status of the module.

4.9.1 Front LEDs



1. Reserved for future usage
2. Network Status
3. Module Status
4. Vendor specific LED, Operation mode

LED	LED Status	Description
Module Status	Off	No power or not initialised
	Green	Module status is OK
	Flashing Red	Minor Fault
	Red	Major Fault

LED	LED Status	Description
Network Status	Off	No power, not initialised or no connections established
	Flashing Green	On-line, no connections established
	Green	On-line, one or more connections are established
	Flashing Red	Minor fault, one or more connections have a minor fault
	Red	Critical link failure

LED	LED Status	Description
Operation mode (Vendor specific LED)	Off	No power or not initialised
	Flashing Green	Idle mode
	Green	Run mode

4.9.2 On-board Watchdog LED

The on-board watchdog LED is placed next to the CPU. See AnyBus Design guides for more information.

5. Configuration

This chapter describes the different ways to configure the AnyBus-M DeviceNet module.

5.1 Configuration of Baud rate and MAC ID

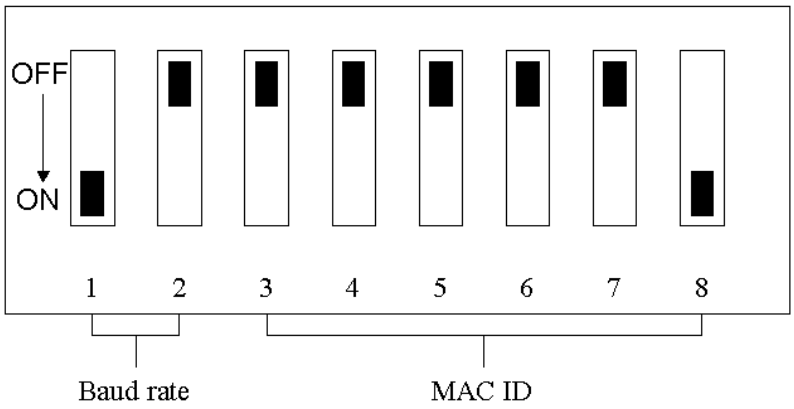
In a DeviceNet network, each node in the network has a MAC ID (the address in the network). The MAC ID is a number between 0 and 63. Each node's MAC ID has to be unique, since it is used to address the node.

In a DeviceNet network it is also possible to configure the Baud rate. The following Baud rates are possible to use in the network: 125, 250, and 500 kBit/sec. All nodes in the network have to communicate with the same Baud rate.

On the AnyBus-M DeviceNet module it is possible to set the MAC ID and the Baud rate with a physical DIP-switch mounted on the module or with a mailbox message. If the AnyBus-M module is mounted inside a product it can be hard to reach the physical DIP-switch. In this case it can be useful to set the MAC ID and Baud rate with a mailbox message. The DIP-switch is mounted at the front of the module.

The Baud rate and the MAC ID can be configured either via dipswitches mounted on the module or with a fieldbus specific mailbox message. In the AnyBus-S DeviceNet module it is necessary to set the DIP-switch in 0xFF to be able to set the DIP with a mailbox message. This has been removed in the AnyBus-M module, since a customer may want to change the bit definitions of the DIP. For example, according to the DeviceNet specification, if the module is mounted vertically in a product, the switch that is at top shall always be the high-bit.

Dip 1 and 2 are used to configure the Baud rate and dips 3 to 8 are used to configure the node address (MAC ID). The DIP-switch configuration is set according to the description on the next page.



(ON = "1", OFF = "0")

Baudrate [kBit/sec]	DIP 1-2
125	0 0
250	0 1
500	1 0
Reserved	1 1

Address	DIP 3 - 8
0	0 0 0 0 0 0
1	0 0 0 0 0 1
2	0 0 0 0 1 0
3	0 0 0 0 1 1
…	
62	1 1 1 1 1 0
63	1 1 1 1 1 1

5.2 Configuration of nodes and data mapping

According to the AnyBus-M concept, configuration of an AnyBus-M module is fieldbus specific. It is possible to configure the ANYBUS® M DeviceNet both from the application and from the fieldbus. If configuration is done from the application, the user has to implement this with mailbox messages. The configuration from the fieldbus side can be done with RSNetWorx from Rockwell. In the next release of the AnyBus-M DeviceNet, it will also be able to configure it with NetView from Cutler Hammer.

At the moment, it does not exist any generic ANYBUS® configuration tool capable of configuring the AnyBus-M DeviceNet. Please contact HMS for more information about when this will be released.

5.2.1 Scan list object

The ANYBUS® M DeviceNet module contains an object called the "Scan list" (please refer to the DeviceNet specification for more information about DeviceNet objects). The Scan list is an object that contains 64 instances (1 to 64), which each are corresponding to a DeviceNet node in the network (node 0 to instance 1, node 1 to instance 2… node 63 to instance 64). Each of these instances contains information about the configuration that the master has about each node. Configuration data corresponding to the nodes that is stored in the scan list is for example: I/O sizes, Vendor ID, Product Code, Product type etc. (see table below).

The parameters of an instance of the scan list are described in the table on the next pages. Each instance of the scan list can be uploaded and downloaded with mailbox messages or with a DeviceNet configuration tool. The parameters in the scan list can only be changed if the scanner is in idle mode (configuration mode).

Table describing the parameters in the Scan list object:

Byte no.	Parameter	Data type	Description
0	Configuration Valid	USINT	1 = Valid Record, 0 = Not valid
1	DeviceNet port	USINT	Set To Zero
2	MAC ID	USINT	0-63
3, 4	Vendor	UINT	Identity Object
5, 6	Product Type	UINT	Identity Object
7, 8	Product Code	UINT	Identity Object
9	Scan Type	USINT	I/O communication type
10	Poll Hz	USINT	0 = Every scan, 1 = Background poll rate
11	Strobe In Length	USINT	# of bytes received from device after strobe
12	Reserved	USINT	Set to Zero
13	Poll In Length	USINT	# of bytes received from device after poll
14	COS/Cyclic In Length	USINT	# of bytes received from device after poll
15	Poll Out Length	USINT	# of bytes sent out to device at poll
16	COS/Cyclic Out Length	USINT	# of bytes sent out to device at poll
17	In_DNet_Msg_Type_1	USINT	Corresponding I/O connection
18	In_DNet_Map_Type_1	USINT	1 = map to data table, 0 and 2-225 are reserved
19, 20	In_DNet_Byte_Off_1	UINT	# of bytes offset in the DeviceNet message
21	In_DNet_Bit_Off_1	USINT	# of bits offset in the DeviceNet message
22, 23	In_MapTable_Length_1	UINT	# of bits that will be mapped to the data table
24, 25	In_MapTable_Off_1	UINT	# of bits offset in the data table message
26	In_DNet_Msg_Type_2	USINT	Corresponding I/O connection
27	In_DNet_Map_Type_2	USINT	1 = map to data table, 0 and 2-225 are reserved
28, 29	In_DNet_Byte_Off_2	UINT	# of bytes offset in the DeviceNet message
30	In_DNet_Bit_Off_2	USINT	# of bits offset in the DeviceNet message
31, 32	In_MapTable_Length_2	UINT	# of bits that will be mapped to the data table
33, 34	In_MapTable_Off_2	UINT	# of bits offset in the data table message
35	In_DNet_Msg_Type_3	USINT	Corresponding I/O connection
36	In_DNet_Map_Type_3	USINT	1 = map to data table, 0 and 2-225 are reserved
37, 38	In_DNet_Byte_Off_3	UINT	# of bytes offset in the DeviceNet message
39	In_DNet_Bit_Off_3	USINT	# of bits offset in the DeviceNet message
40, 41	In_MapTable_Length_3	UINT	# of bits that will be mapped to the data table
42, 43	In_MapTable_Off_3	UINT	# of bits offset in the data table message
44	In_DNet_Msg_Type_4	USINT	Corresponding I/O connection
45	In_DNet_Map_Type_4	USINT	1 = map to data table, 0 and 2-225 are reserved
46, 47	In_DNet_Byte_Off_4	UINT	# of bytes offset in the DeviceNet message
48	In_DNet_Bit_Off_4	USINT	# of bits offset in the DeviceNet message
49, 50	In_MapTable_Length_4	UINT	# of bits that will be mapped to the data table

51, 52	In_MapTable_Off_4	UINT	# of bits offset in the data table message
53	Out_DNet_Msg_Type_1	USINT	Corresponding I/O connection
54	Out_DNet_Map_Type_1	USINT	1 = map to data table, 0 and 2-225 are reserved
55, 56	Out_DNet_Byte_Off_1	UINT	# of bytes offset in the DeviceNet message
57	Out_DNet_Bit_Off_1	USINT	# of bits offset in the DeviceNet message
58, 59	Out_MapTable_Length_1	UINT	# of bits that will be mapped to the data table
60, 61	Out_MapTable_Off_1	UINT	# of bits offset in the data table message
62	Out_DNet_Msg_Type_2	USINT	Corresponding I/O connection
63	Out_DNet_Map_Type_2	USINT	1 = map to data table, 0 and 2-225 are reserved
64, 65	Out_DNet_Byte_Off_2	UINT	# of bytes offset in the DeviceNet message
66	Out_DNet_Bit_Off_2	USINT	# of bits offset in the DeviceNet message
67, 68	Out_MapTable_Length_2	UINT	# of bits that will be mapped to the data table
69, 70	Out_MapTable_Off_2	UINT	# of bits offset in the data table message
71	Out_DNet_Msg_Type_3	USINT	Corresponding I/O connection
72	Out_DNet_Map_Type_3	USINT	1 = map to data table, 0 and 2-225 are reserved
73, 74	Out_DNet_Byte_Off_3	UINT	# of bytes offset in the DeviceNet message
75	Out_DNet_Bit_Off_3	USINT	# of bits offset in the DeviceNet message
76, 77	Out_MapTable_Length_3	UINT	# of bits that will be mapped to the data table
78, 79	Out_MapTable_Off_3	UINT	# of bits offset in the data table message
80	Out_DNet_Msg_Type_4	USINT	Corresponding I/O connection
81	Out_DNet_Map_Type_4	USINT	1 = map to data table, 0 and 2-225 are reserved
82, 83	Out_DNet_Byte_Off_4	UINT	# of bytes offset in the DeviceNet message
84	Out_DNet_Bit_Off_4	USINT	# of bits offset in the DeviceNet message
85, 86	Out_MapTable_Length_4	UINT	# of bits that will be mapped to the data table
87, 87	Out_MapTable_Off_4	UINT	# of bits offset in the data table message
88	Major Revision	USINT	Device keying info
90	Minor Revision	USINT	Device keying info
91, 92	EPR/Heartbeat	UINT	COS/Cyclic heartbeat expected packet rate
93, 94	ACK timer	UINT	ACK timer value used for COS/Cyclic
95, 96	Inhibit Timer	UINT	Inhibit timer
97	Options	USINT	Option bits
98	Unused	USINT	Not used
99	Unused	USINT	Not used

The lowest two bits, bit 0 and bit 1, of the scan-type (byte 9) contain information about the strobe and poll connection. See the table below how to select the poll and strobe connections:

Binary	Description
X X X X X X 0 0	Strobe
X X X X X X 0 1	Poll
X X X X X X 1 0	Poll and strobe
X X X X X X 1 1	Neither strobe nor poll

The next two bits, bit 2 and bit 3, configure the COS/Cyclic connection. See the table below how to configure the COS cyclic bits:

Binary	Description
X X X X 0 0 X X	Neither COS nor Cyclic
X X X X 0 1 X X	COS
X X X X 1 0 X X	Cyclic
X X X X 1 1 X X	Illegal

The next bit, bit 4, configures the acknowledge message for the COS/Cyclic connection. If the ACK message shall be used, the bit shall be set to zero, if the ACK message shall not be used, set the bit to 1.

The three highest bits are reserved and shall always be set to 0.

5.2.2 Scanner configuration object

The ANYBUS® M DeviceNet module contains an object called the "Scanner Configuration Object". The Scanner Configuration Object contains information about the configuration of the module.

The parameters (except from the read only parameters) can be changed with a DeviceNet configuration tool or with mailbox messages. The parameters can only be changed when the module is in idle mode (configuration mode).

The table below describes the parameters of the Scanner configuration object.

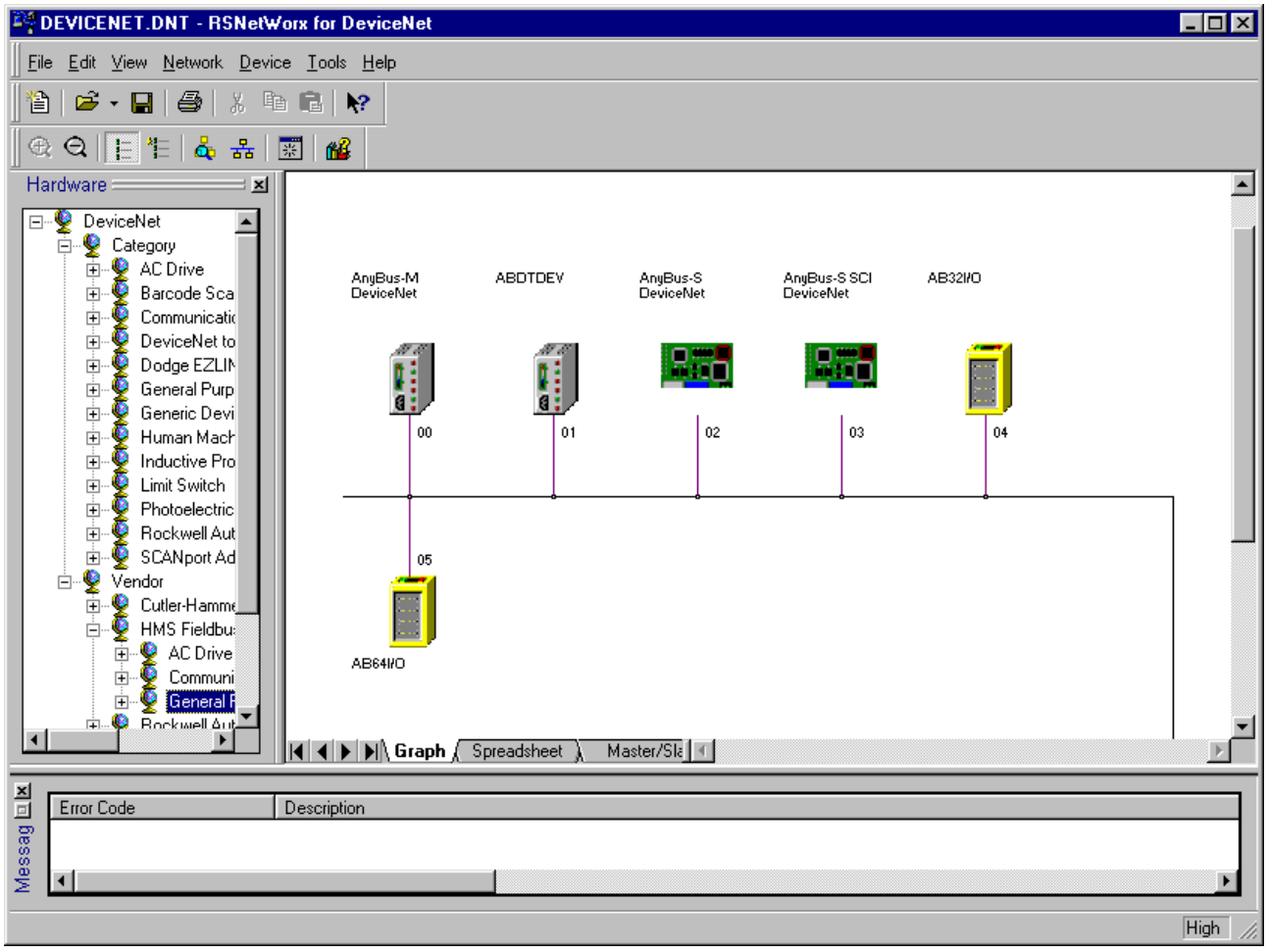
Byte no.	Attribute	Data type	Description
0, 1	Net EPR	UINT	Network expected packet rate
2, 3	Net ISD	UINT	Interscan delay
4	NET STAT	USINT	Network status (Read only)
5, 6	Net background poll rate	UINT	Background poll rate
7	Options	USINT	Reserved (Read only)
8	Unused	USINT	Reserved (Read only)
9	Unused	USINT	Reserved (Read only)
10	Unused	USINT	Reserved (Read only)
11	Net tx retry	USINT	Internal usage (Read only)
12, 13	Scan Counter	UINT	Count of scan cycles (Read only)

5.3 Configuration with RSNetWorx

One way to configure the AnyBus-M DeviceNet module is to use RSNetWorx for DeviceNet (DeviceNet configuration tool from Rockwell). See the picture below. This program is run on a PC in the Windows environment. The program needs a physical link towards the DeviceNet network where the AnyBus-M module is connected and also the modules that the AnyBus-M shall communicate with. This physical link can be a serial adapter (for example 1770-KFD), a PCI or ISA card (for example 1784 scanner), or a PCMCIA interface.

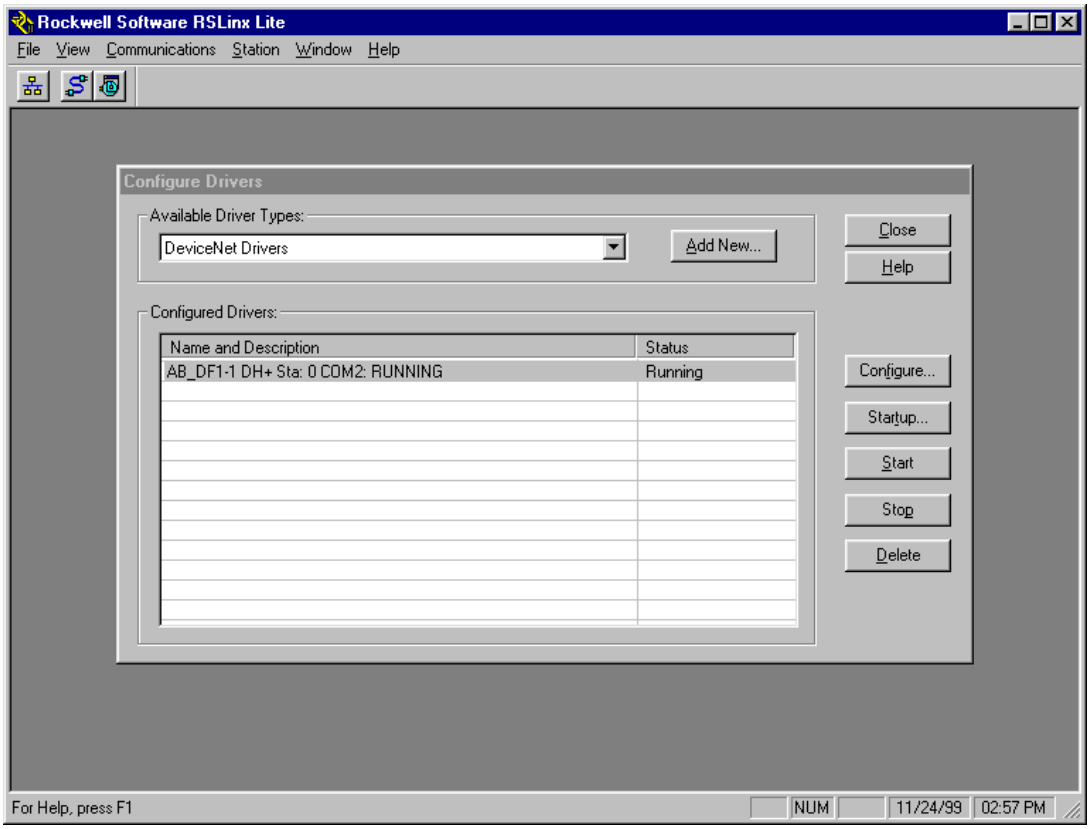
When a node in a DeviceNet network shall be configured with a DeviceNet configuration tool, it is necessary to have an EDS file that describes the node for the configuration tool. Please contact each vendor for all products that shall be configured for correct EDS files.

Some of the buttons on the pictures in this chapter have Swedish labels, since they has been captured on a computer with a Swedish operating system. We apologise for that.

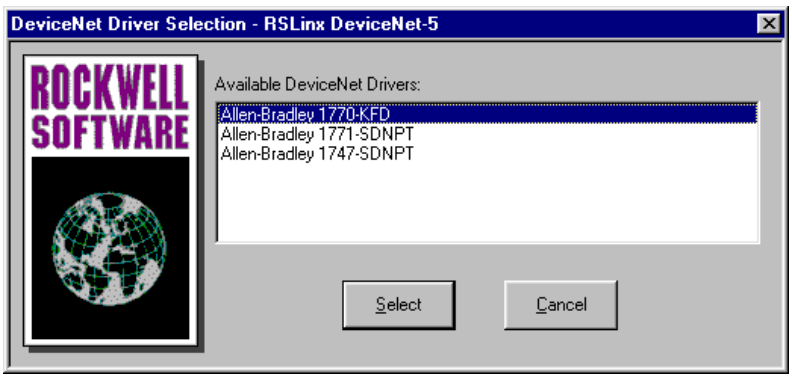


5.3.1 Getting started

The first thing to do when the RSNetWorx shall be used for the first time is to select which driver that shall be used. RSNetWorx uses a program called RSLinx to access the DeviceNet network. To install the driver, start the RSLinx program, and then select configure drivers in the communications menu. A new window will now appear called "Configure drivers" (see below). Select DeviceNet drivers in the "Available driver types" scroll list, and press add new.

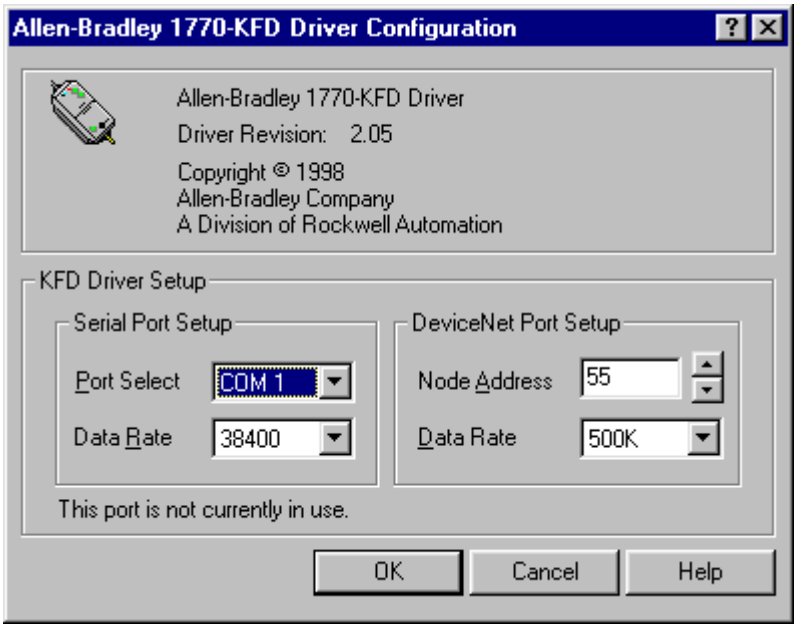


Now another window will appear, which will display available DeviceNet drivers. Select the driver that corresponds to the physical interface that is used (in our case 1770-KFD as seen below). Choose the corresponding driver and press the select button.



Now the program wants the user to select the correct settings, that the RSNetWorx shall use on the DeviceNet network. In our case, the 1770-KFD is connected to a serial COM port on the PC, so we select the COM port that it is connected to and which data rate we will use on the serial channel.

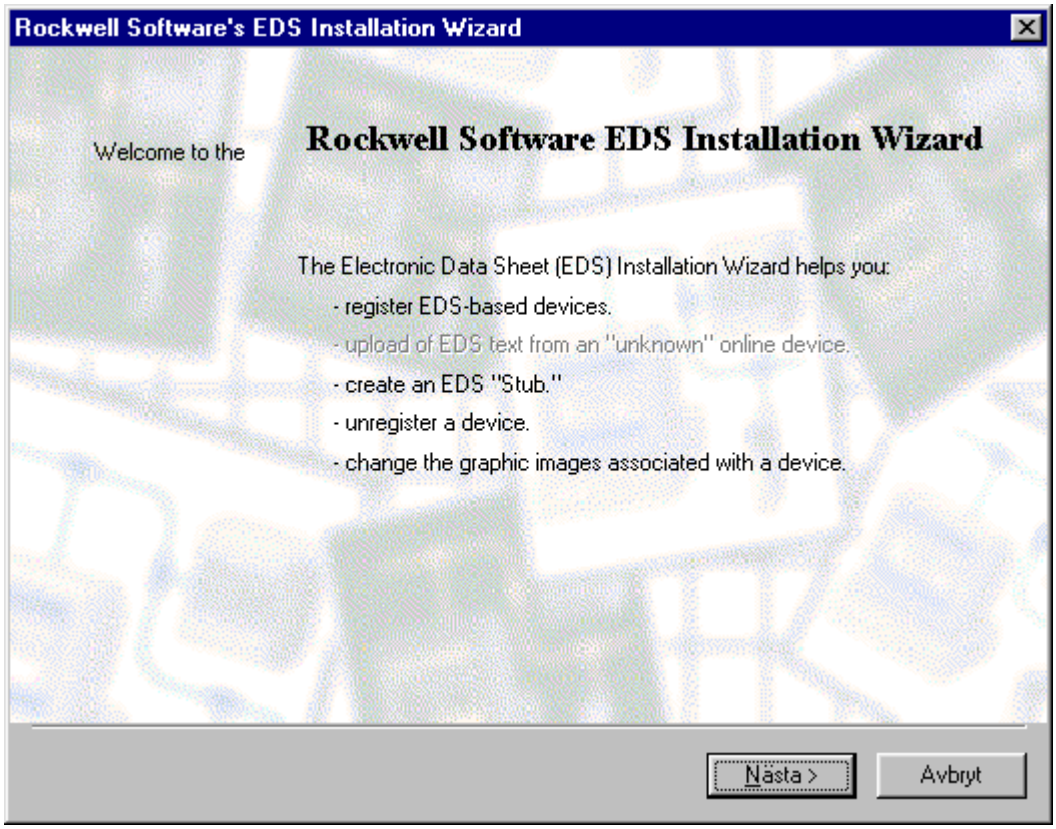
The node address (MAC ID) and data rate (Baud rate), which the configuration tool will use on the DeviceNet network, has to be selected. In the example below, we have chosen 55 as MAC ID and 500 kbit/sec, since the rest of the nodes are configured for that Baud rate.



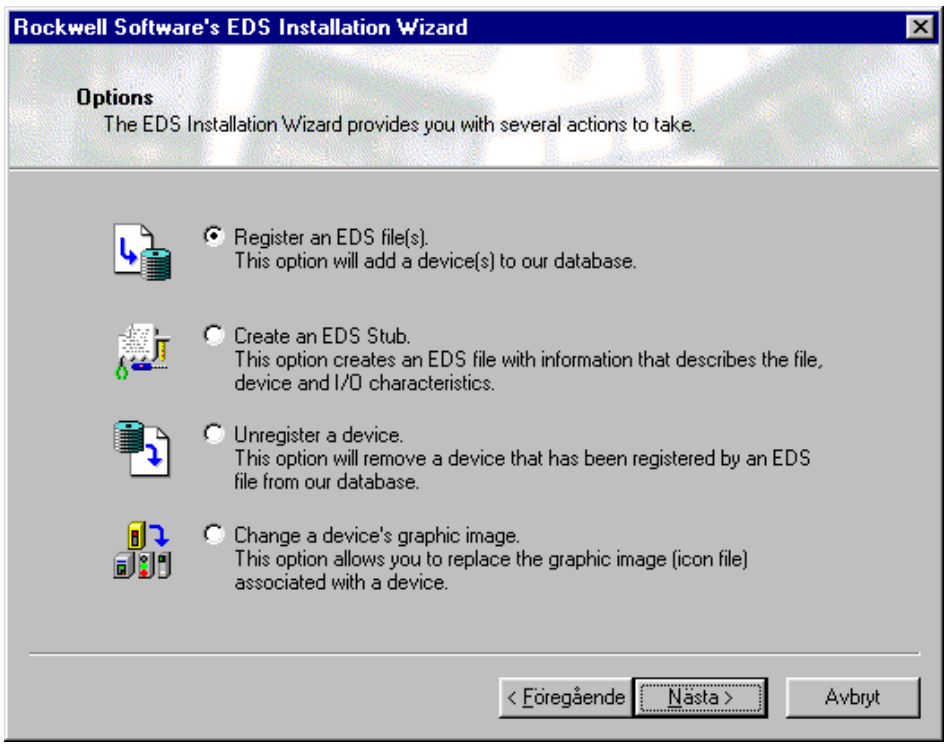
Then press the OK button, and the RSLinx shall start the driver.

5.3.2 Installing EDS files

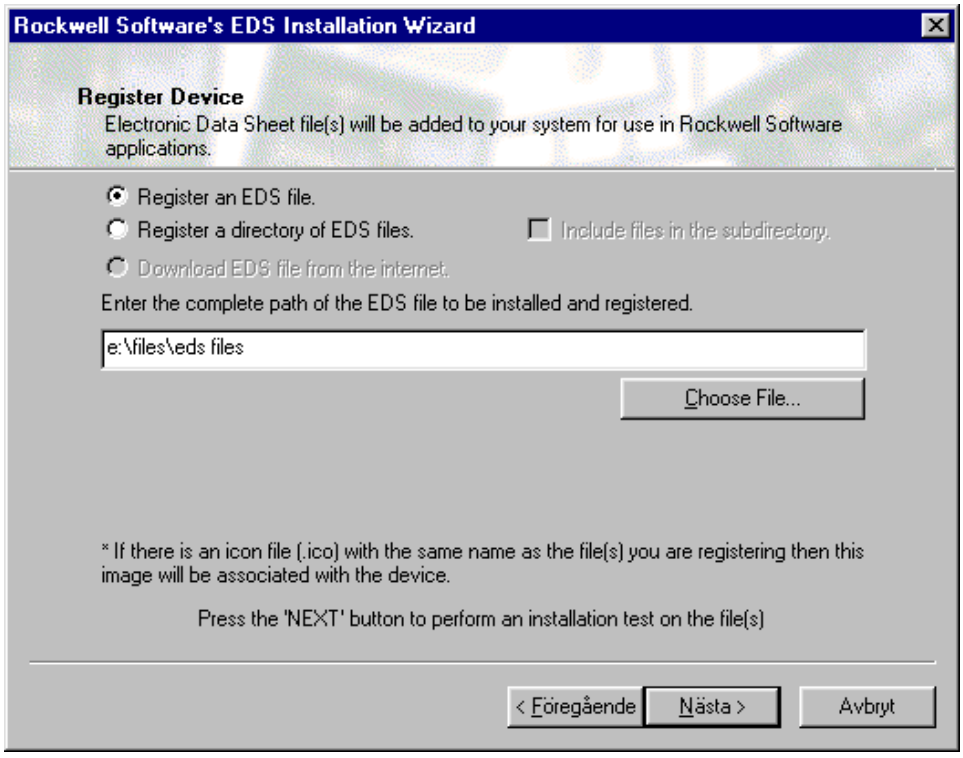
Before the RSNetWorx can go on-line, the correct EDS files for all nodes in the network need to be installed. To install an EDS file, select EDS wizard in the tools menu. A new window shall now appear (see below) and guide the user on how to install the files. Start with pressing the next button ("nästa" in Swedish).



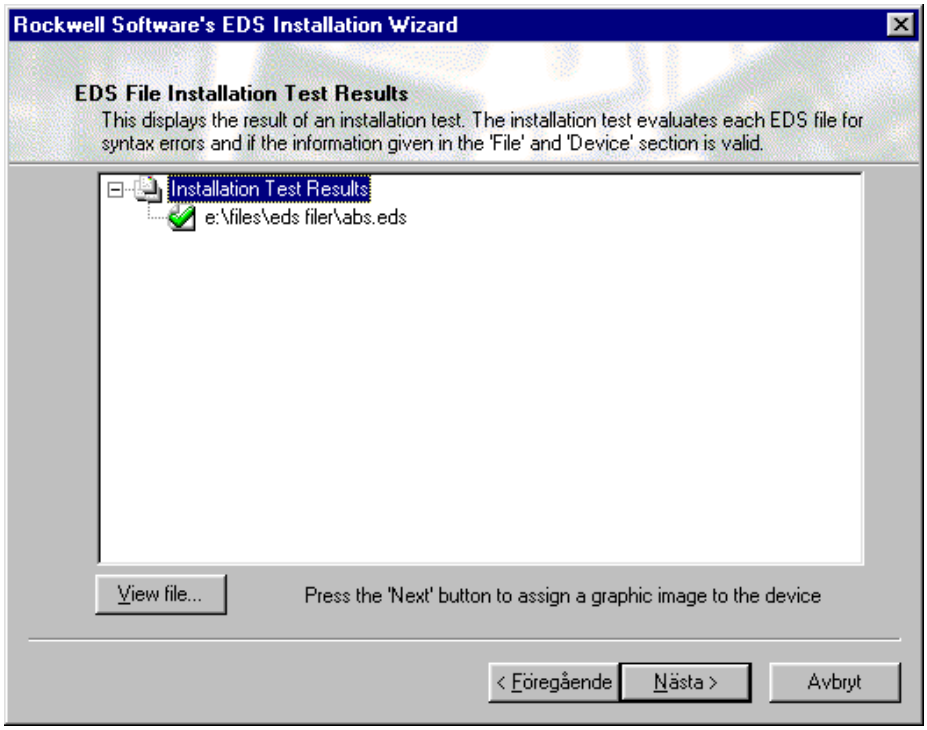
Select the alternative "Register an EDS file(s)" and press next.



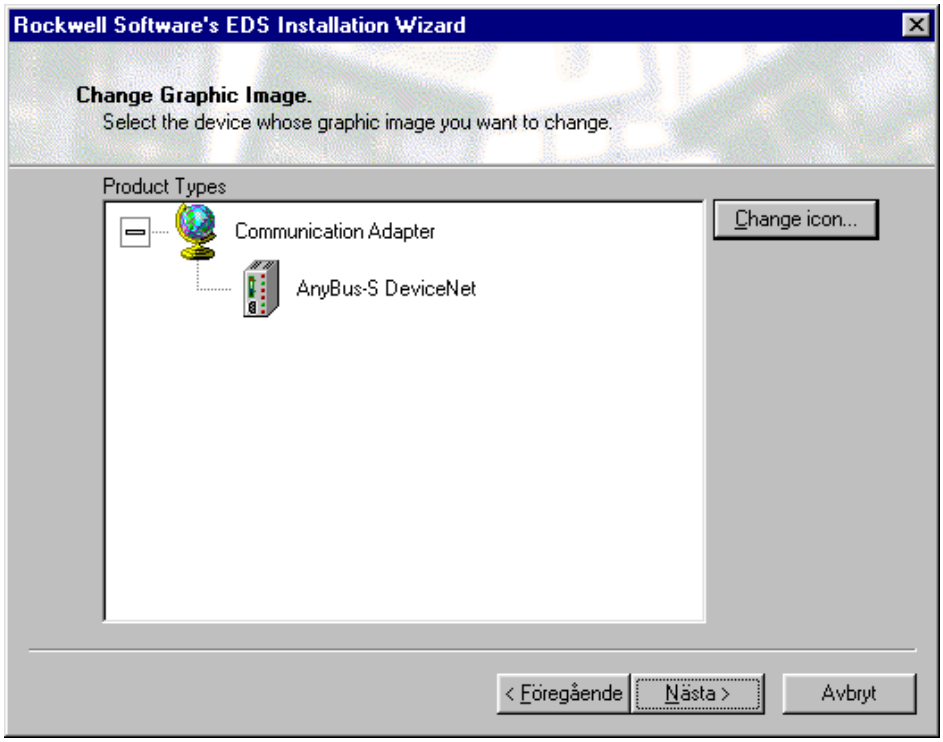
Select "Register an EDS file" and choose file by pressing the "Choose file" button. Then press the next button.



The EDS wizard will now test the EDS file, to make sure that it does not have any error. If no errors or warnings are found, the EDS file will appear with a green “checked” sign on it as seen below. If no errors are found just press next, otherwise contact the vendor for the product for a correct EDS file.



The user now can select an icon for the module, which will be used in the configuration tool when the module is displayed. In the picture below, the default icon is displayed. Press the next button to continue.



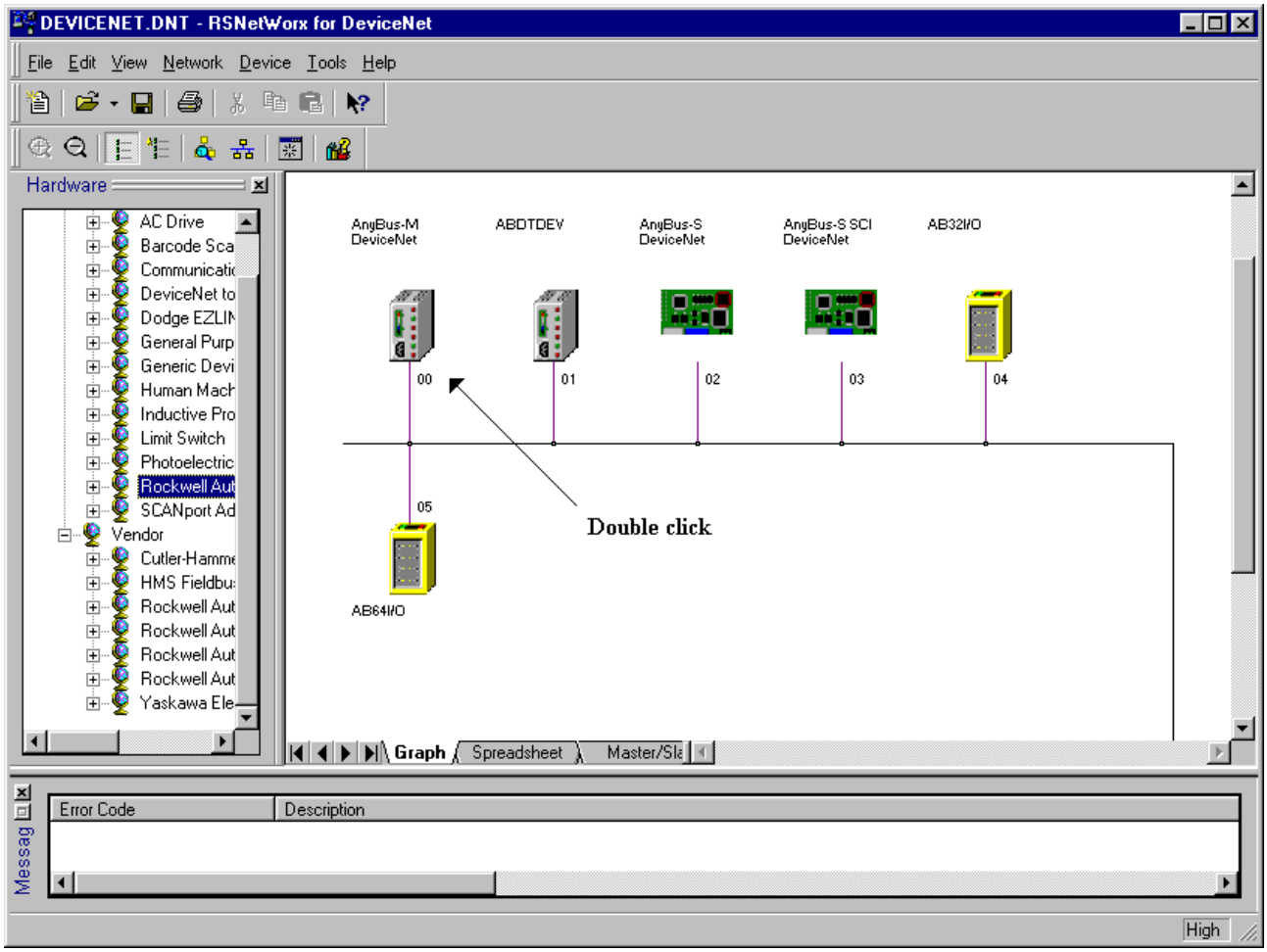
The EDS wizard will now install the EDS file into the RSNetWorx configuration tool. Press the finish button (see below, "Slutför" in Swedish) to finish the installation.

5.3.3 Get On-line

When RSNetWorx has been correctly installed and connected to the DeviceNet network, the user can go on-line by selecting ONLINE in the Network menu, or by pressing F10. The user now has to select which physical interface that is used. Just double click the alternative in the pop-up window that appears (if the correct alternative do not appear in the window, the driver has not been correctly installed in RSLinx, see last chapter on how to install the RSLinx driver).

The configuration tool will now browse the network, and display all nodes that are found in the network. All nodes that shall be configured need corresponding EDS files to be able to be configured.

To configure the AnyBus-M DeviceNet module, double click the icon for the module in the main window (see below).

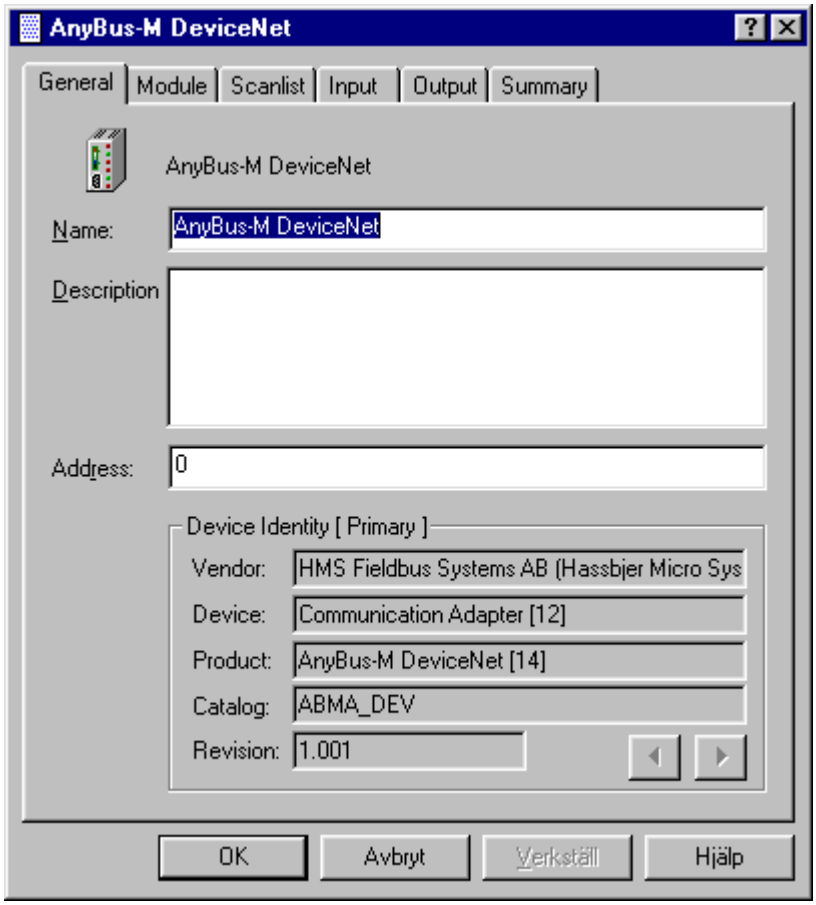


5.3.4 General information

In this window, which appears after double clicking the modules icon, all configuration of the AnyBus-M module can be done. The window contains six pages:

- General
- Module
- Scanlist
- Input
- Output
- Summary

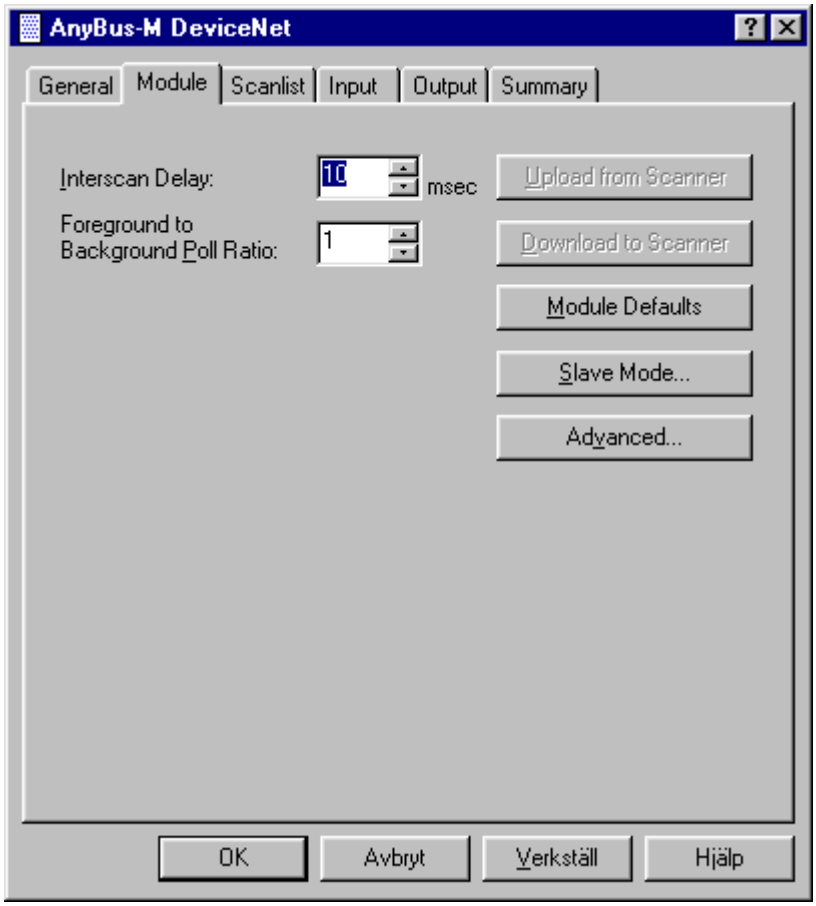
The general page (shown in the picture) contains general information, which is obtained from the EDS file, and verified with the module in the network.



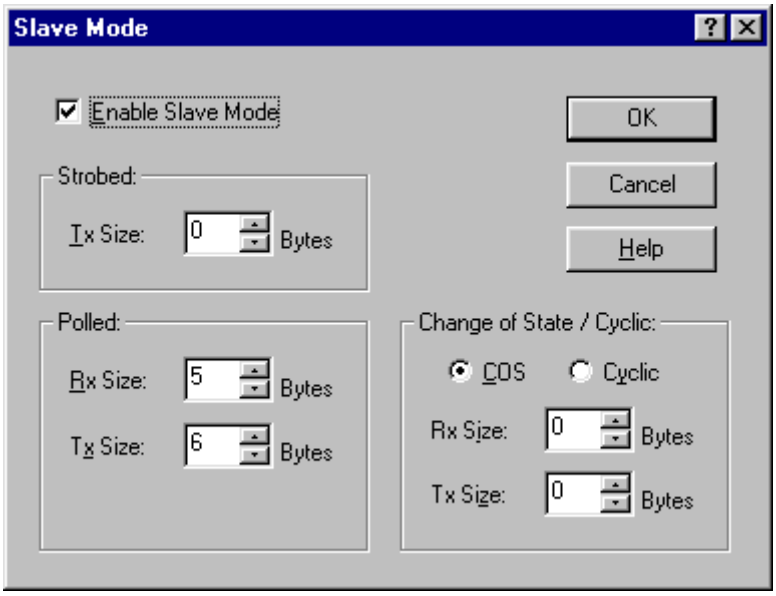
5.3.5 Scanner Configuration and Slave mode configuration

In the configuration window, on the module page it is possible to change the "Inter scan delay" and "Foreground to background poll ratio". "Inter scan delay" is the delay between each I/O poll the master sends to the configured nodes. A node can be configured to be polled with a "background poll rate". This means that the module will poll the foreground-nodes *X* number of times as often as the background nodes. It is also possible to configure the slave connection in this window. There is also a button called "Advanced" which gives the user the possibility to change the maximum expected packet rate (the time the master shall wait for a I/O response) and the number of times the master shall retry to transmit if a module is not responding.

To upload the settings that are stored in the module by pressing the "Upload from scanner" button. If the settings are changed, they need to be downloaded to the module. Pressing the "Download to scanner" button does this.



In the slave mode window, the user can enable the slave connections that shall be used, and select the amount of data for each connection. This data will have to be mapped in the Input and Output areas of the DPRAM (see the next pages). In the Example below, we have chosen to enable a polled slave connection, with 5 bytes as input data and 6 bytes as output data.

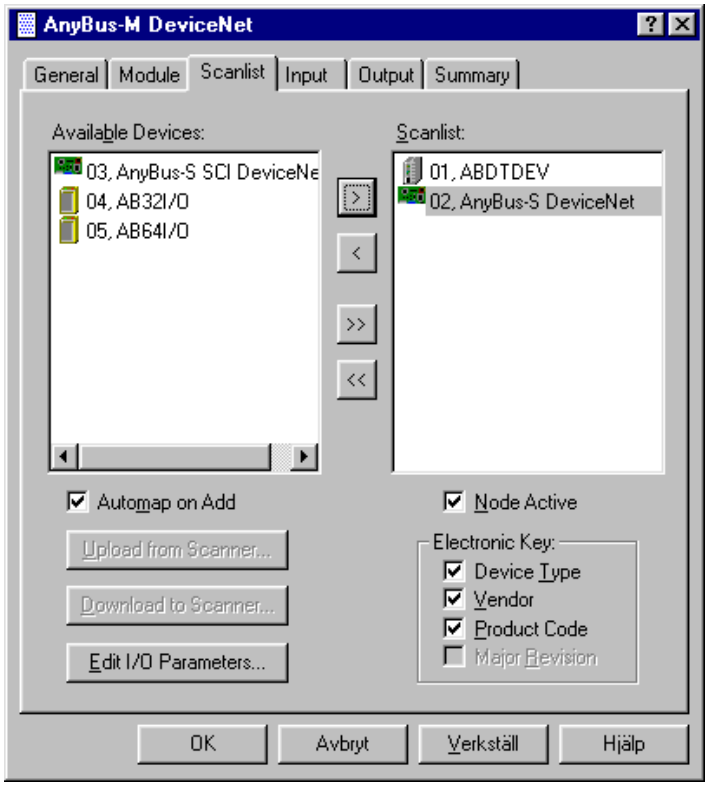


5.3.6 Scan list configuration with RSNetworx

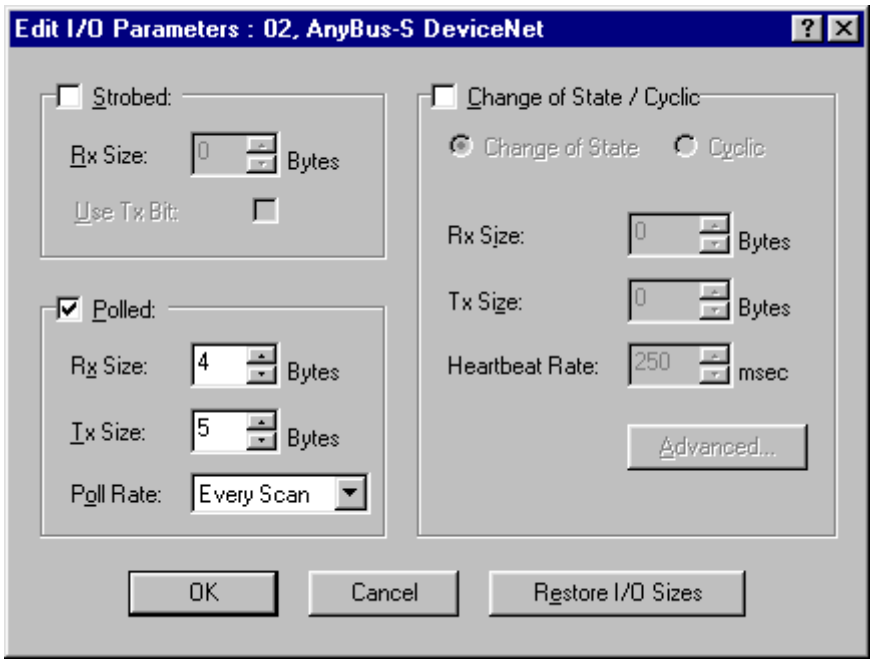
On the Scanlist page, nodes are mapped into the masters scan list. In the picture below, two modules have been mapped into the masters scan list. To add a node to the scanlist, mark the node and press the ">" button. The node shall now be moved into the scan list. To add all nodes into the scan list, press the ">>" button.

When a node is added into the scan list, the node will automatically be mapped into the input and output areas, if the check box "Automap on Add" is marked. If a node does not use the default I/O sizes that are presented in the EDS file, it might be necessary to correct the I/O configuration for a node. To do this, mark the node and press the "Edit I/O parameters" button.

The scanlist data can be uploaded and downloaded from / to the module by pressing the upload and download buttons.



Select the I/O connections that shall be used, by marking the checkboxes corresponding to the I/O connections. Then fill in the number of bytes that shall be sent and received. Also set Heartbeat rate for the COS/Cyclic connection, and select if the Tx bit shall be used for the strobed connection, if those connections shall be configured. In our picture below we have selected 4 bytes input data and 5 bytes output data for the poll connection, and we have chosen not to use the other connections. Press the OK button when the editing is finished.



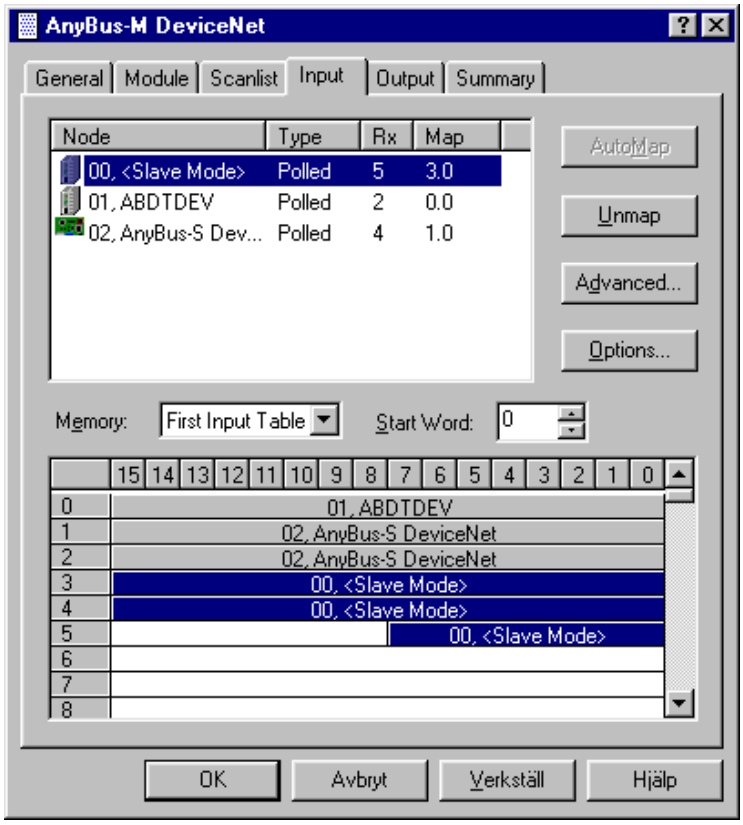
5.3.7 Mapping data into the DPRAM with RSNetworx

On the input and output pages there are information about how the I/O connections are mapped into the I/O areas. If the "Automap on add" checkbox was marked, and the default I/O connections and data sizes are used, the nodes have automatically been mapped into the I/O areas. If a node has not been mapped into the I/O areas, it will say "No" in the column "Map" (the column where it says 3.0 for node 0).

Below we can see that there are three connections that have been mapped into the input area, each corresponding to different nodes. The first node that has been mapped is node 1 (the node with MAC ID 1). That node has 2 bytes of input corresponding to its polled connection, which will be sent from the node to the AnyBus-M module. The data is mapped into the input area from byte 0 to byte 1 (word 0). The next node that has been mapped is node 2. This node also uses the polled I/O connection, and it will send 4 bytes of polled I/O data to the AnyBus-M module. The data has been mapped into the bytes 2 to 5 (word 1 and 2) in the input area.

The master's slave connection also has been configured to 5 bytes of input data for the polled connection. This data has been mapped into the bytes 6 to 10 (word 3, 4 and the low byte of word 5).

The RSNetWorx uses word addressing. This is the reason why it says 3.0 for node 0 in the map column. This means that the data is mapped from word 3, bit 0 (= byte 6, bit 0).

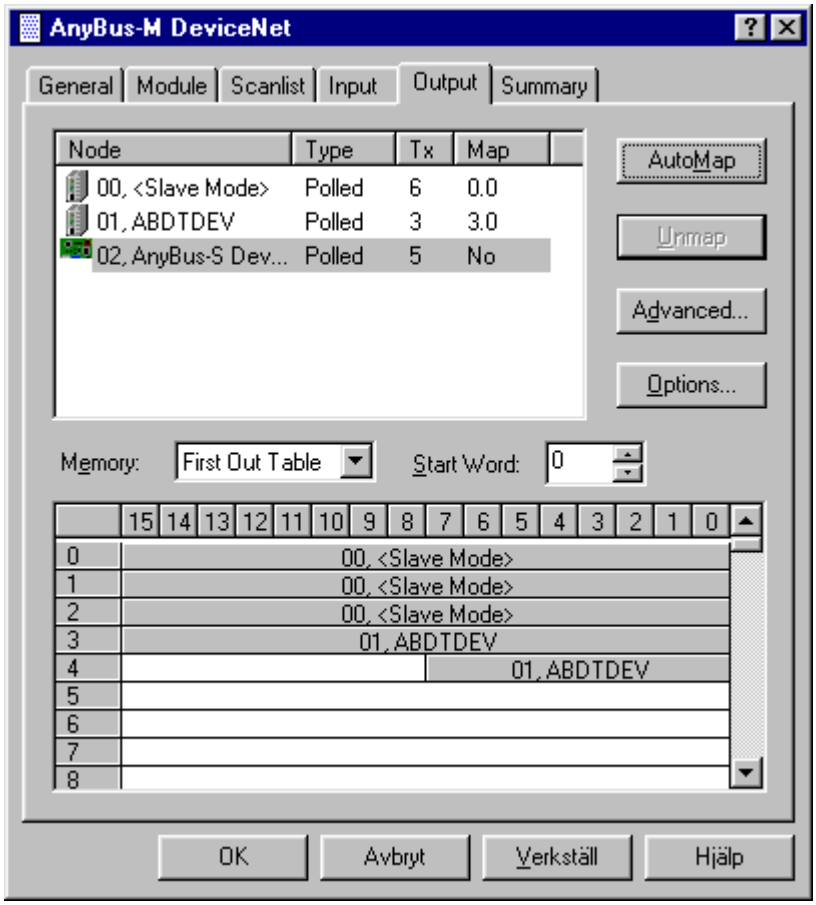


Below we can see information for the output data mapping. Here are also three I/O connections that are configured, 6 bytes polled output for the masters slave connection, 3 bytes of polled output data for node 1, and 5 bytes of polled output data for node 2.

In this example, the AnyBus-M module's slave connection has been mapped into the first four bytes in the output data area, bytes 0 to 5 (word 0, 1, and 2). The output data corresponding to node 1 has been mapped into the bytes 6 to 8 (word 3 and the low byte of word 4).

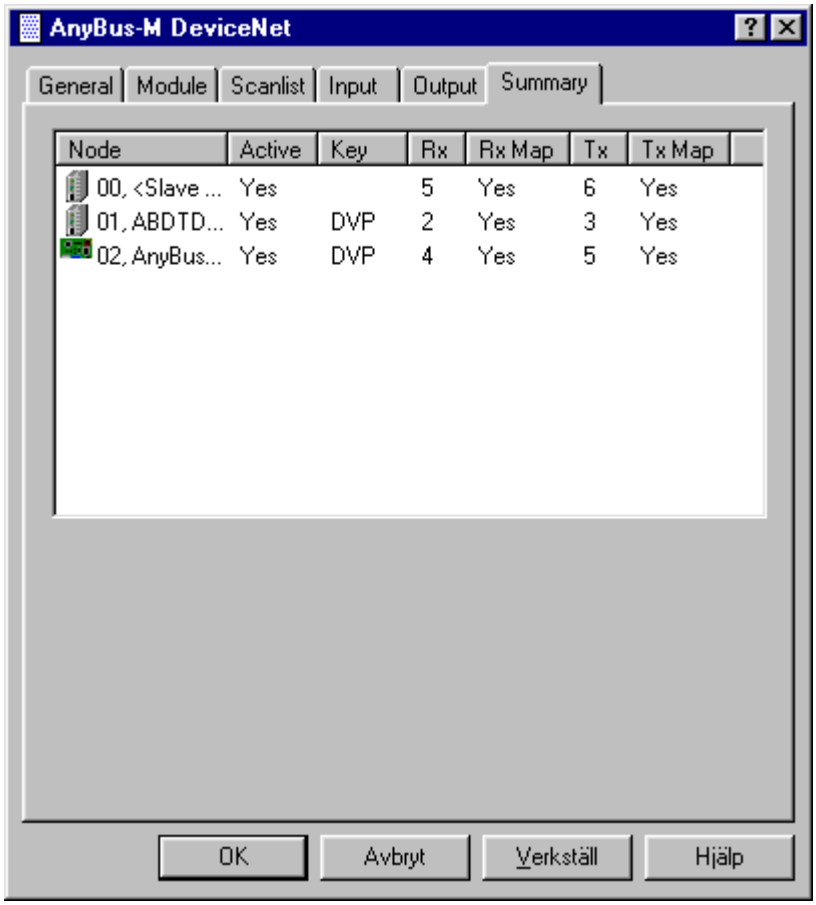
No I/O data has yet been mapped for node 2. To map the data for that node, just mark the node by clicking it once and the press the button "Automap". Now the node will be mapped directly after the last mapped node in the Output data area.

It is also possible to do a more advanced mapping of the data by pressing the "Advanced" button. To find more information about this please see the manual for RSNetWorx.



5.3.8 Summary of the configuration

On the summary page, information about all I/O connections, which are configured into the scan list, is displayed. The Active column tells if each connection is active or not. The Key column tells if electronic keying will be used to verify the identity of each node, before any of the I/O connections are established and any data exchange will be done. The Rx and Tx fields display the data size that is configured for each connection, and the Rx Map and Tx Map fields tell if the connections have been mapped into the I/O areas.



When all configurations have been done, press the "Apply" button ("Verkställ" in Swedish). Now all changes will be downloaded to the module. Note that the module needs to be in idle mode to be able to download any changes to the configuration.

5.4 Fieldbus specific mailbox interface

The mailbox interface is used to exchange data with the application. The mailbox messages described below are the so-called fieldbus specific mailbox messages. They are specific for the AnyBus-M DeviceNet module. These mailbox messages can be used to set up parameters in the module and special fieldbus specific services. See the AnyBus-S Design guides for more information about mailbox messages.

The following fieldbus specific mailbox messages are implemented in the AnyBus-M DeviceNet module:

Mailbox message	Mailbox command no.	Direction
SET DIPSWITCH	0x0001	Application -> AnyBus
SET PRODUCT INFO	0x0002	Application -> AnyBus
SET PRODUCT CODE	0x0003	Application -> AnyBus
GET SCANLIST INSTANCE	0x0004	Application -> AnyBus
SET SCANLIST INSTANCE	0x0005	Application -> AnyBus
GET SCANNER CONFIG DATA	0x0006	Application -> AnyBus
SET SCANNER CONFIG DATA	0x0007	Application -> AnyBus
RESET SCANNER CONFIG	0x0008	Application -> AnyBus
SEND EXPLICIT MESSAGE	0x0009	Application -> AnyBus
SET SCANNER OPERATION MODE	0x000A	Application -> AnyBus
GET DIPSWITCH	0x000F	Application -> AnyBus
EXPLICIT MESSAGE RESULT	0x8001	AnyBus -> Application

5.4.1 Mailbox message status/error codes

The following table lists the error codes that the mailbox messages can return.

Value	Description
0x00	No error
0x01-0xFA	Reserved
0xFB	Incorrect string length
0xFC	Error in message data, value outside of allowable range, etc.
0xFD	Reserved
0xFE	Scanner is not in idle mode
0xFF	Error in mailbox message

5.4.2 SET DIPSWITCH

This mailbox message makes it possible for the application to set the MAC ID and the Baud rate with a mailbox message. It shall be seen as a complement to the physical DIP-switch that is mounted on the module. If the AnyBus module is embedded inside a product, it can be difficult to reach the physical DIP-switch. This mailbox message has to be sent before the End Init mailbox message (before the initialisation of the AnyBus module is finished).

See chapter 5.1 for more information on how to configure the Baud rate and the MAC ID.

NOTE: Chapter 5.4.13 describes a mailbox message to read the value of the DIP-switch. Before initialisation of the module, these mailbox messages can be combined to change the meaning of the bits.

Parameter	Description
Command initiator	Application
Command Name	Set DIPSWITCH mailbox telegram
Message type	0x02
Command number	0x0001
Fragmented	No
Extended Header data	No extended header data is associated with this command
Message data	Contains the data settings that correspond to the dipswitch.
Response message	The response indicates if the command was accepted. The response data is a copy of the command data.

Command and response layout

Register Name	*Command*	*Expected response*	
Message ID	0x0001	0x0001	*Any integer value*
Message information	0x4002	0x0002	
Command	0x0001	0x0001	
Data size	0x0001	0x0001	*One byte*
Frame count	0x0001	0x0001	*One frame*
Frame number	0x0001	0x0001	*This is frame 1*
Offset high	0x0000	0x0000	
Offset low	0x0000	0x0000	
Extended word 1	-	-	*No message header data used*
Extended word 2	-	-	*"*
Extended word 3	-	-	*"*
Extended word 4	-	-	*"*
Extended word 5	-	-	*"*
Extended word 6	-	-	*"*
Extended word 7	-	-	*"*
Extended word 8	-	Error code	*See error code table*
Message data byte 1	Dipswitch setting	Dipswitch setting	*The value of the dipswitch*

5.4.3 SET PRODUCT INFO

This mailbox message makes it possible to give the module a new identity. The mailbox message changes the parameters in the identity object. When this mailbox message is used, it is possible to identify the product on the network with other identity information than the default HMS identity information. When the identity information is changed, it is not possible to use the original EDS file. A new EDS file has to be written, with the new identity information.

Parameter	Description
Command initiator	Application
Command Name	Set product info
Message type	0x02
Command number	0x0002
Fragmented	No
Extended Header data	No extended header data is associated with this command
Message data	Vendor ID, Device Type, Product code, Revision number and Product name
Response message	The response indicates if the command was accepted. The response data is a copy of the command data.

Command and response layout

Register Name	*Command*	*Expected response*	
Message ID	0x0001	0x0001	*Any integer value*
Message information	0x4002	0x0002	
Command	0x0002	0x0002	
Data size	n	n	
Frame count	0x0001	0x0001	*One frame*
Frame number	0x0001	0x0001	*This is frame 1*
Offset high	0x0000	0x0000	
Offset low	0x0000	0x0000	
Extended word 1	-	-	*No message header data used*
Extended word 2	-	-	*"*
Extended word 3	-	-	*"*
Extended word 4	-	-	*"*
Extended word 5	-	-	*"*
Extended word 6	-	-	*"*
Extended word 7	-	-	*"*
Extended word 8	-	Error code	*See error code table*
Message data byte 1	Vendor ID (high byte)	Vendor ID (high byte)	*Vendor ID high byte*
Message data byte 2	Vendor ID (low byte)	Vendor ID (low byte)	*Vendor ID low byte*
Message data byte 3	Device Type (high byte)	Device Type (high byte)	*Device type low byte*
Message data byte 4	Device Type (low byte)	Device Type (low byte)	*Device type high byte*
Message data byte 5	Product code (high byte)	Product code (high byte)	*Product code high byte*
Message data byte 6	Product code (low byte)	Product code (low byte)	*Product code low byte*
Message data byte 7	Major revision	Major revision	*Major revision*
Message data byte 8	Minor revision	Minor revision	*Minor revision*

(continued on next page)

(continued from previous page)

Message data byte 9	Product name length	Product name length	*Length of product name (max 32)*
Message data byte 10	Product name 1st char	Product name 1st char	*First character (ascii value)*
Message data byte 11	Product name 2nd char	Product name 2nd char	*Second character (ascii value)*
…	…	…	
Message data byte n	Product name last char	Product name last char	*Last character (ascii value)*

5.4.4 SET PRODUCT CODE

This mailbox message is used to change the product code of the module. The mailbox message will only change the high byte of the product code. The customer will have to apply for a product code from the DeviceNet department at HMS, since the Vendor ID still is HMS. The user needs to apply for a product code to make sure that no other product has the same product code value. This will make it possible for a configuration tool to identify the product on the network, instead of identifying it as a general AnyBus DeviceNet module. The low byte of the product code will remain the same.

Parameter	Description
Command initiator	Application
Command Name	Set product code
Message type	0x02
Command number	0x0003
Fragmented	No
Extended Header data	No extended header data is associated with this command
Message data	Contains the high byte of the new product code.
Response message	The response indicates if the command was accepted. The response data is a copy of the command data.

Command and response layout

Register Name	*Command*	*Expected response*	
Message ID	0x0001	0x0001	*Any integer value*
Message information	0x4002	0x0002	
Command	0x0003	0x0003	
Data size	0x0001	0x0001	
Frame count	0x0001	0x0001	*One frame*
Frame number	0x0001	0x0001	*This is frame 1*
Offset high	0x0000	0x0000	
Offset low	0x0000	0x0000	
Extended word 1	-	-	*No message header data used*
Extended word 2	-	-	*"*
Extended word 3	-	-	*"*
Extended word 4	-	-	*"*
Extended word 5	-	-	*"*
Extended word 6	-	-	*"*
Extended word 7	-	-	*"*
Extended word 8	-	-	
Message data byte 1	Product code	Product code	*High byte of the product code*

5.4.5 GET SCANLIST INSTANCE

This mailbox message is used to read the configuration in the scan list for a node. See chapter 5.2.1 for detailed information about the response data of this mailbox message.

Parameter	Description
Command initiator	Application
Command Name	Get scan list instance
Message type	0x02
Command number	0x0004
Fragmented	No
Extended Header data	No extended header data is associated with this command
Message data	Contains the MAC ID of the targeted node
Response message	The response indicates if the command was accepted. The response data is the scan list data corresponding to the MAC ID.

Command and response layout

Register Name	*Command*	*Expected response*	
Message ID	0x0001	0x0001	*Any integer value*
Message information	0x4002	0x0002	
Command	0x0004	0x0004	
Data size	0x0001	0x0041	
Frame count	0x0001	0x0001	*One frame*
Frame number	0x0001	0x0001	*This is frame 1*
Offset high	0x0000	0x0000	
Offset low	0x0000	0x0000	
Extended word 1	-	-	*No message header data used*
Extended word 2	-	-	*"*
Extended word 3	-	-	*"*
Extended word 4	-	-	*"*
Extended word 5	-	-	*"*
Extended word 6	-	-	*"*
Extended word 7	-	-	*"*
Extended word 8	-	Error code	*See error code table*
Message data byte 1	MAC ID (0x00 to 0x3F)	Config Valid	Scan List Data
byte 2	-	DNet Port	Scan List Data
byte 3	-	MAC ID	Scan List Data
byte 4	-	Vendor ID (High byte)	Scan List Data
	…	…	*"*
byte 100	-	-	Scan List Data

Please see chapter 5.2.1 for more information about the contents of the scanlist table.

5.4.6 SET SCANLIST INSTANCE

This mailbox message can be used to add, remove, or change configuration of nodes in the scan list. The message will set the data in the scan list instance that corresponds to the MAC ID in the data. See chapter 5.2.1 for a detailed description of the scan list data. Only byte 0 - byte 95 of the scan list data can be set (this mailbox message). The size of this message is 100 bytes. The last bytes of the configuration data in the scan list are the status for the node, which cannot be set.

Note: This mailbox message can only be used when the module is in idle mode. To set the module in idle mode, the mailbox message "set scanner operation mode" has to be used.

Parameter	Description
Command initiator	Application
Command Name	Set scan list instance
Message type	0x02
Command number	0x0005
Fragmented	No
Extended Header data	No extended header data is associated with this command
Message data	Scan list data.
Response message	The response indicates if the command was accepted. The response data is a copy of the command data.

Command and response layout

Register Name	*Command*	*Expected response*	
Message ID	0x0001	0x0001	*Any integer value*
Message information	0x4002	0x0002	
Command	0x0005	0x0005	
Data size	0x0064	0x0064	*100 bytes*
Frame count	0x0001	0x0001	*One frame*
Frame number	0x0001	0x0001	*This is frame 1*
Offset high	0x0000	0x0000	
Offset low	0x0000	0x0000	
Extended word 1	-	-	*No message header data used*
Extended word 2	-	-	*"*
Extended word 3	-	-	*"*
Extended word 4	-	-	*"*
Extended word 5	-	-	*"*
Extended word 6	-	-	*"*
Extended word 7	-	-	*"*
Extended word 8	-	Error code	*See error code table*
Message data byte 1	Config Valid	Config Valid	*Scanlist data*
Message data byte 2	Dnet Port	DNet Port	*Scanlist data*
Message data byte 3	MAC ID	MAC ID	*Scanlist data*
Message data byte 4	Vendor ID (High byte)	Vendor ID (High byte)	*Scanlist data*
	…	…	
Message data byte 100	-	-	*Scanlist data*

Please see chapter 5.2.1 for more information about the contents of the scanlist table.

5.4.7 GET SCANNER CONFIG DATA

This mailbox message is used to read the Configuration Object in the module. This

Parameter	Description
Command initiator	Application
Command Name	Get scanner configuration data
Message type	0x02
Command number	0x0006
Fragmented	No
Extended Header data	No extended header data is associated with this command
Message data	Scanner configuration data
Response message	The response indicates if the command was accepted. The response is the scanner configuration data.

Command and response layout

Register Name	*Command*	*Expected response*	
Message ID	0x0001	0x0001	*Any integer value*
Message information	0x4002	0x0002	
Command	0x0006	0x0006	
Data size	0x0000	0x000E	
Frame count	0x0001	0x0001	*One frame*
Frame number	0x0001	0x0001	*This is frame 1*
Offset high	0x0000	0x0000	
Offset low	0x0000	0x0000	
Extended word 1	-	-	*No message header data used*
Extended word 2	-	-	*"*
Extended word 3	-	-	*"*
Extended word 4	-	-	*"*
Extended word 5	-	-	*"*
Extended word 6	-	-	*"*
Extended word 7	-	-	*"*
Extended word 8	-	-	*''*
Message data byte 1	-	Net EPR (high byte)	*Scanner Config Data*
Message data byte 2	-	Net EPR (low byte)	*Scanner Config Data*
Message data byte 3	-	Net ISD (high byte)	*Scanner Config Data*
Message data byte 4	-	Net ISD (low byte)	*Scanner Config Data*
	-	Back plane poll rate (high byte)	*Scanner Config Data*
Message data byte 14	-	Back plane poll rate (high byte)	*Scanner Config Data*

5.4.8 SET SCANNER CONFIG DATA

This mailbox message is used to change the scanner configuration data (see chapter 5.2.2 for more information).

Parameter	Description
Command initiator	Application
Command Name	Set scanner configuration data
Message type	0x02
Command number	0x0007
Fragmented	No
Extended Header data	No extended header data is associated with this command
Message data	Scanner configuration data
Response message	The response indicates if the command was accepted. The response data is a copy of the command data.

Command and response layout

Register Name	*Command*	*Expected response*	
Message ID	0x0001	0x0001	*Any integer value*
Message information	0x4002	0x0002	
Command	0x0007	0x0007	
Data size	0x0006	0x0006	
Frame count	0x0001	0x0001	*One frame*
Frame number	0x0001	0x0001	*This is frame 1*
Offset high	0x0000	0x0000	
Offset low	0x0000	0x0000	
Extended word 1	-	-	*No message header data used*
Extended word 2	-	-	*"*
Extended word 3	-	-	*"*
Extended word 4	-	-	*"*
Extended word 5	-	-	*"*
Extended word 6	-	-	*"*
Extended word 7	-	-	*"*
Extended word 8	-	Error code	*See error code table*
Message data byte 1	Net EPR (high byte)	Net EPR (high byte)	*Scanner Config Data*
Message data byte 2	Net EPR (low byte)	Net EPR (low byte)	*Scanner Config Data*
Message data byte 3	Net ISD (high byte)	Net ISD (high byte)	*Scanner Config Data*
Message data byte 4	Net ISD (low byte)	Net ISD (low byte)	*Scanner Config Data*
Message data byte 5	Back plane poll rate (high byte)	Back plane poll rate (high byte)	*Scanner Config Data*
Message data byte 6	Back plane poll rate (high byte)	Back plane poll rate (high byte)	*Scanner Config Data*

The maximum allowable values for the 9999 for the EPR, 9999 for the ISD and 32000 for the BPR.

5.4.9 RESET SCANNER CONFIG

This mailbox message removes all configuration from the from the scan list and sets the scanner configuration parameters to their default values.

Parameter	Description
Command initiator	Application
Command Name	Reset Scanner Configuration
Message type	0x02
Command number	0x0008
Fragmented	No
Extended Header data	No extended header data is associated with this command
Message data	No data
Response message	The response indicates if the command was accepted. No data.

Command and response layout

Register Name	*Command*	*Expected response*	
Message ID	0x0001	0x0001	*Any integer value*
Message information	0x4002	0x0002	
Command	0x0008	0x0008	
Data size	0x0000	0x0000	
Frame count	0x0001	0x0001	*One frame*
Frame number	0x0001	0x0001	*This is frame 1*
Offset high	0x0000	0x0000	
Offset low	0x0000	0x0000	
Extended word 1	-	-	*No message header data used*
Extended word 2	-	-	*"*
Extended word 3	-	-	*"*
Extended word 4	-	-	*"*
Extended word 5	-	-	*"*
Extended word 6	-	-	*"*
Extended word 7	-	-	*"*
Extended word 8	-	Error code	*See error code table*

5.4.10 SET SCANNER OPERATION MODE

This mailbox message is used to set the operation mode for the scanner. The scanner can be set to run mode or idle mode (see chapter 3.5).

Parameter	Description
Command initiator	Application
Command Name	Set scanner operation mode
Message type	0x02
Command number	0x000A
Fragmented	No
Extended Header data	No extended header data is associated with this command
Message data	Operation mode byte.
Response message	The response indicates if the command was accepted. The response data is a copy of the command data.

Command and response layout

Register Name	*Command*	*Expected response*	
Message ID	0x0001	0x0001	*Any integer value*
Message information	0x4002	0x0002	
Command	0x000A	0x000A	
Data size	0x0001	0x0001	
Frame count	0x0001	0x0001	*One frame*
Frame number	0x0001	0x0001	*This is frame 1*
Offset high	0x0000	0x0000	
Offset low	0x0000	0x0000	
Extended word 1	-	-	*No message header data used*
Extended word 2	-	-	*"*
Extended word 3	-	-	*"*
Extended word 4	-	-	*"*
Extended word 5	-	-	*"*
Extended word 6	-	-	*"*
Extended word 7	-	-	*"*
Extended word 8	-	Error code	*See error code table*
Message data byte 1	Operation mode	Operation mode	*0 = run mode, 1 = idle mode*

5.4.11 GET DIPSWITCH

This mailbox message gives an opportunity to read the settings on the DIP-switch. This is implemented, since the AnyBus module might be mounted in different ways in different products, and the DeviceNet specification says that the DIP most left or most to the top of the product shall be the DIP with highest significance. With this mailbox it is possible to read the DIP, convert the bit order, and use the set DIP mailbox message to write the actual value to the module.

Parameter	Description
Command initiator	Application
Command Name	Get DIP
Message type	0x02
Command number	0x000F
Fragmented	No
Extended Header data	No extended header data is associated with this command
Message data	No data
Response message	The response contains the settings of the DIP

Command and response layout

Register Name	*Command*	*Expected response*	
Message ID	0x0001	0x0001	*Any integer value*
Message information	0x4002	0x0002	
Command	0x000F	0x000F	
Data size	0x0000	0x0000	
Frame count	0x0001	0x0001	*One frame*
Frame number	0x0001	0x0001	*This is frame 1*
Offset high	0x0000	0x0000	
Offset low	0x0000	0x0000	
Extended word 1	-	-	*No message header data used*
Extended word 2	-	-	”
Extended word 3	-	-	”
Extended word 4	-	-	”
Extended word 5	-	-	”
Extended word 6	-	-	”
Extended word 7	-	-	”
Extended word 8	-	-	‘’
Message data byte 1	-	DIP data	*Switch setting*

5.4.12 SEND EXPLICIT MESSAGE

This mailbox message is used to make it possible to send an explicit DeviceNet message (parameter data) from the application, directly to a node in the network. The module will respond directly to this message as a notification that the explicit message will be sent. When the module has been able to get a result of an explicit message, it will automatically send an explicit message result mailbox telegram to the application (see 3.7.11). For more information about the explicit messaging protocol, see the DeviceNet specification Vol. II and the Master tool kit documentation.

Parameter	Description
Command initiator	Application
Command Name	Send explicit message
Message type	0x02
Command number	0x0009
Fragmented	No
Extended Header data	No extended header data is associated with this command
Message data	Explicit DeviceNet message data
Response message	The response indicates if the command was accepted. The response data is a copy of the command data.

Command and response layout

Register Name	*Command*	*Expected response*	
Message ID	0x0001	0x0001	*Any integer value*
Message information	0x4002	0x0002	
Command	0x0009	0x0009	
Data size	N	n	
Frame count	0x0001	0x0001	*One frame*
Frame number	0x0001	0x0001	*This is frame 1*
Offset high	0x0000	0x0000	
Offset low	0x0000	0x0000	
Extended word 1	-	-	*No message header data used*
Extended word 2	-	-	*"*
Extended word 3	-	-	*"*
Extended word 4	-	-	*"*
Extended word 5	-	-	*"*
Extended word 6	-	-	*"*
Extended word 7	-	-	*"*
Extended word 8	-	Error code	*See error code table*
Message data byte 1	Status	Status	*See status code table*
Message data byte 2	TXID	TXID	*Transmission ID*
Message data byte 3	Size	Size	*Data size in bytes*
Message data byte 4	Reserved	Reserved	
Message data byte 5	MAC ID	MAC ID	*Target node ID*
Message data byte 6	Service	Service	*Service code*
Message data byte 7	Explicit message data	Explicit message data	*1st byte of message data*
	…	…	
Message data byte n	Explicit message data	Explicit message data	*Last byte of message data*

5.4.13 EXPLICIT MESSAGE RESULT

Generated by the AnyBus M module when an explicit message task has been completed. Returns the result of the explicit message task to the application. Note that the application shall not send any response mailbox message to the AnyBus module when this mailbox command is received.

Parameter	Description
Command initiator	AnyBus M module
Command Name	Explicit message result
Message type	0x02
Command number	0x8001
Fragmented	No
Extended Header data	No extended header data is associated with this command
Message data	Contains the result of an earlier sent "SEND EXPLICIT MESSAGE" mailbox
Response message	The application shall not respond to this message

Command and response layout

Register Name	*Command*	*Expected response*	
Message ID	0x0001	-	*Any integer value*
Message information	0x4002	-	
Command	0x8001	-	
Data size	N	-	
Frame count	0x0001	-	*One frame*
Frame number	0x0001	-	*This is frame 1*
Offset high	0x0000	-	
Offset low	0x0000	-	
Extended word 1	-	-	*No message header data used*
Extended word 2	-	-	"
Extended word 3	-	-	"
Extended word 4	-	-	"
Extended word 5	-	-	"
Extended word 6	-	-	"
Extended word 7	-	-	"
Extended word 8	-	-	
Message data byte 1	Status	-	
Message data byte 2	TXID	-	
Message data byte 3	Size	-	
Message data byte 4	Reserved	-	
Message data byte 5	MAC ID	-	
Message data byte 6	Service	-	
Message data byte 7	Explicit response data	-	
	…	-	
Message data byte N	Explicit response data	-	

The following table contains a description of the status code that the SEND_EXPLICIT_MESSAGE and the EXPLICIT_MESSAGE_RESULT can return.

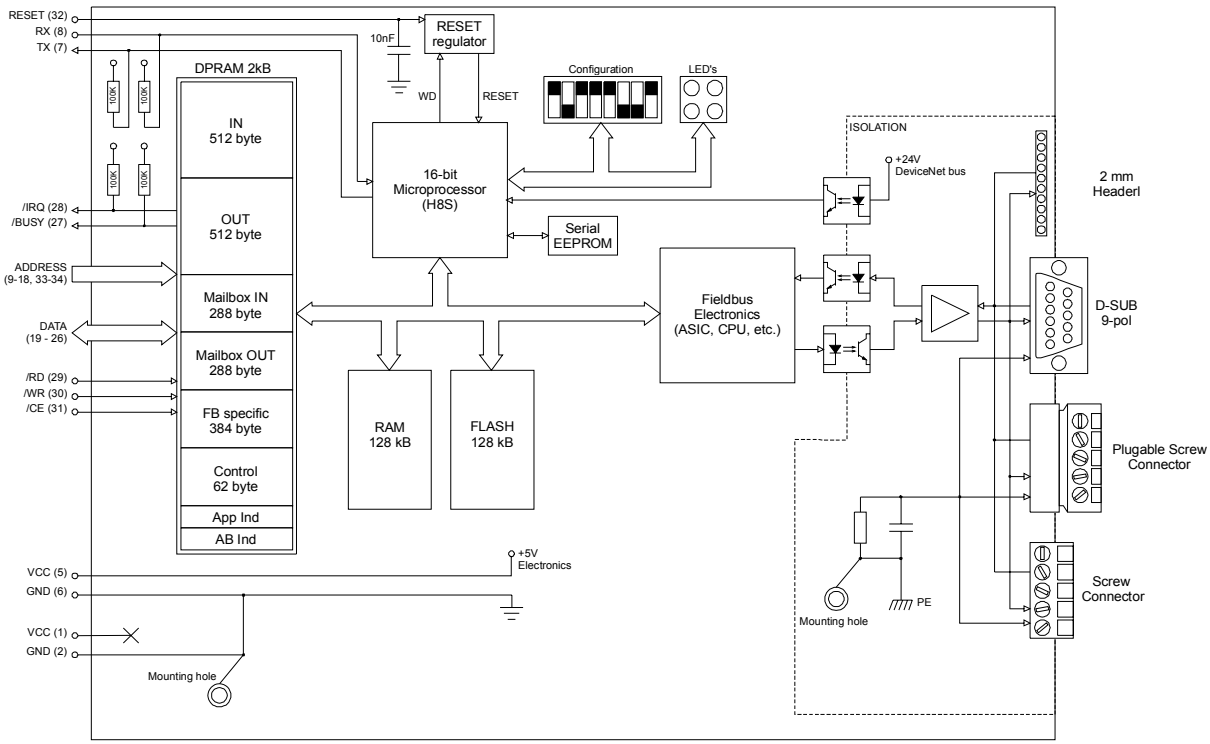
Value	Description
0	No error
1	Message completed successfully
2	Transaction in progress
4	Error – Node offline
5	Error – DeviceNet port disabled/offline
6	Error – TXID unknown
7	Error – Duplicate TXID
9	Error – Scanner out of buffers
12	Error – Response data too large
14	Error – Invalid size specified
15	Error – Device timeout
16	Block queued
17	Block allocated
18	Connection in progress
3, 8, 10, 11, 13, 19-255	Reserved for future use

6. Hardware Overview

This chapter gives an overview of the hardware design of the AnyBus-M module and also describes possible configurations of the hardware.

6.1 Hardware Design

The AnyBus-M module is based on the same hardware as the AnyBus-S DeviceNet module. An overview of the hardware design is shown in the block schematic below.



6.2 Hardware Configuration

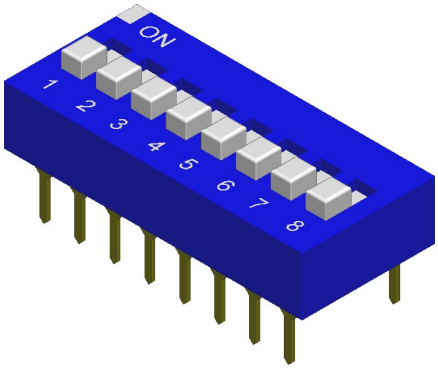
It is possible to order the AnyBus-M module with different types of hardware configurations. It is possible to order the module with different fieldbus connectors, different length of the application connector, different types of DIP-switches.

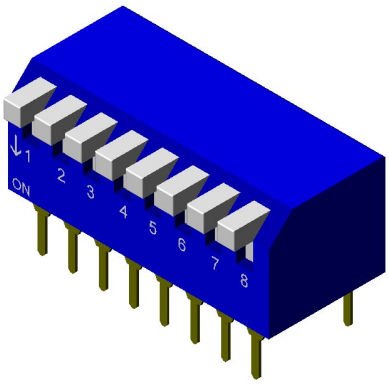
The configurable components can be mounted on the topside or the bottom side of the module.

Also see the Design guides for information about connectors and LEDs.

6.2.1 DIP Switch

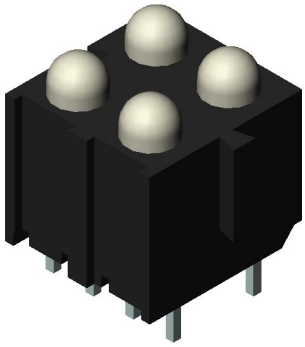
There are two types of DIP-switches that can be mounted on the module. The DIPs can be on top (straight, see left picture below) or angled (see right picture below) towards the front of the module.

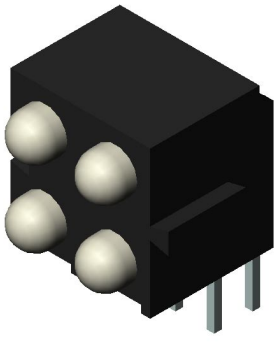




6.2.2 LEDs

There are two types of LEDs that can be mounted on the module. The LEDs can be on top (straight, see left picture below) or angled (see right picture below) towards the front of the module.





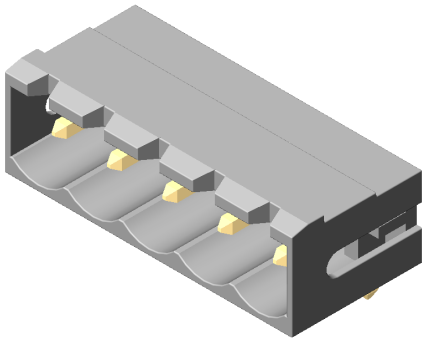
6.2.3 Fieldbus Connectors

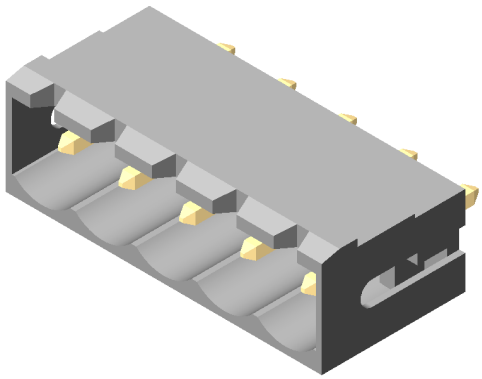
It is possible to order the AnyBus-M DeviceNet with three different fieldbus connector types.

1. 5,08mm plugable screw connector
2. 3,81mm plugable screw connector
3. 2,00mm male strip connector

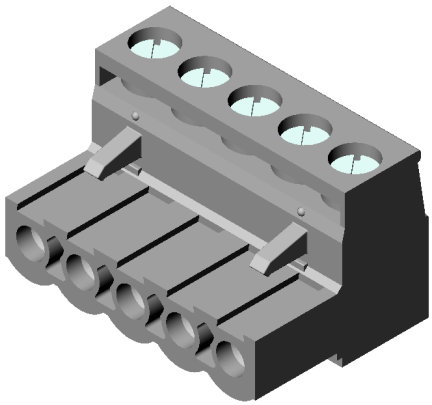
It is possible to get the connectors mounted on the top side or the bottom side of the module.

1. The 5,08mm plugable screw connector (shown below) is the only applicable connector that is approved by the ODVA conformance test. The connector can either be angled (below left) or straight (below right)





When the 5,08mm plugable screw connector is chosen, the module is delivered with both the male part (above) and the female part (below).



2. It is also possible to order the AnyBus-M module with a 3,81mm screw connector. This connector is not approved by the ODVA conformance test. This connector is possible to order, for backwards compatibility reasons, and is not recommended to use in any new in-design.

When the 3,81mm plugable screw connector is chosen, the module is delivered with both the male part (picture does not exist) and the female part (picture does not exist).

3. The 2mm 10 pin connector is a standard connector for all AnyBus modules and can be used if the module shall be embedded into a product. This connector also makes it possible to extend the fieldbus connection to a carrier board or to some other part of the end product.