

Fieldbus Appendix **ANYBUS[®]-S LonWorks**

DOC. ABS-APPENDIX-LON Rev 1.51

Table of Contents

1 Fieldbus Introduction	5
1.1 Introduction to LonWorks	5
1.1.1 Nodes	5
1.1.2 Network variables (SNVT's)	6
1.1.3 Configuration Properties (SCPT's)	6
1.1.4 Implementation of Network variables in the ABS-LON	6
1.1.5 Installation	7
1.2 Technical Features for LonWorks	7
2 About the AnyBus-S LonWorks Module	8
2.1 Features	8
2.2 Compatible Products	8
2.3 Overview	8
2.3.1 Connectors	9
2.3.2 Service Pin Push button	9
2.3.3 Indicators	9
3 Functions & Operation	10
3.1 Initialization	10
3.2 Error Handling	10
3.3 DPRAM Structure of the LonWorks Module	10
3.3.1 IN/OUT Data Area	10
3.3.2 Control Area	11
3.3.3 Fieldbus Specific Area	11
3.4 Data Exchange	14
3.4.1 Data coming from the network	14
3.4.2 Data coming from the application	15
4 Fieldbus Specific Mailbox Commands	16
4.1 Fault information	16
4.2 Configuration Commands	17
4.2.1 Configure Data Exchange (CONFIG_EXCHANGE)	17
4.2.2 Configure Force Interrupt (SET_FORCE_INT)	18
4.2.3 Set Mapping of Variable (SET_NV_MAPPING)	19
4.2.4 Get Mapping of Variable (GET_NV_MAPPING)	20
4.2.5 Set Location (SET_LOCATION)	21
4.2.6 Get Location (GET_LOCATION)	22
4.2.7 Get Neuron ID (GET_NEURON_ID)	23
4.2.8 Set Network Variable Data (SET_NETVAR_DATA)	24
4.3 Runtime Commands	25
4.3.1 Set Application LED (SET_APPL_LED)	25
4.3.2 Set Node Mode (SET_NODE_MODE)	26
4.3.3 Get Node Mode (GET_NODE_MODE)	27
4.3.4 Propagate NV to Network (PROPAGATE_NV)	28
4.3.5 Get Errors (GET_ERRORS)	29
4.3.6 Clear Errors (CLEAR_ERRORS)	30
4.3.7 Get Node Status (GET_NODE_STATUS)	31
4.3.8 Clear Node Status (CLEAR_NODE_STATUS)	34
4.4 Network Management Commands	35

4.4.1 Set Domain (SET_DOMAIN)	36
4.4.2 Get Domain (GET_DOMAIN)	38
4.4.3 Set Address Table (SET_ADDRESS)	40
4.4.4 Get Address Table (GET_ADDRESS)	42
4.4.5 Set Network Variable Config Table (SET_NV_CONFIG)	43
4.4.6 Get Network Variable Config Table (GET_NV_CONFIG)	45
4.5 Advanced Configuration	46
4.5.1 Start Network Variable Change (START_NV_CHANGE)	46
4.5.2 Add a network variable (ADD_NV)	47
4.5.3 End Network Variable Change (END_NV_CHANGE)	49
5 Connector Pinouts	50
5.1 Application Connector	50
5.2 Fieldbus Connector	50
5.2.1 5-pin Pluggable Screw Terminal (3.84mm and 5.08mm)	50
5.2.2 9-Pin D-SUB connector (Female)	50
5.2.3 10-pin Board to Board Connector (2mm)	50
6 Mechanical specification	51
6.1 Measurements, PCB	51
6.2 Standard Configuration	52
7 Electrical Characteristics	53
7.1 Supply Voltage	53
7.2 Power Consumption	53
7.3 Protective Earth	53
8 Environmental Specification	54
8.1 Temperature	54
8.2 Relative Humidity	54
8.3 EMC Compliance	54

About This Manual

How To Use This Manual

This manual provides an overview of the AnyBus-S LonWorks communication module and is intended to be used in conjunction with the AnyBus-S Design Guide.

The reader of this document is expected to have basic knowledge in the LonWorks networking system, and communication systems in general.

Important User Information

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

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

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

Related Documentation

Document name	Author
AnyBus-S Parallel Design Guide	HMS Industrial Networks AB
LonWorks Technology Device Data	Motorola
The SNVT Master List and Programmer's Guide, May 1997	LonMark Organization
The SNVT Master List and Programmer's Guide, May 1997	LonMark Organization
The SCPT Master List, May 1997	LonMark Organization
Introduction to the LonWorks System	Echelon
LonMaker User's Guide	Echelon
LonMark Application Layer Interoperability Guidelines	LonMark Organization

Revision notes

Revision	Date	Author	Chapter	Notes
0.10a - 1.40a	-	lcl	All	Original version
1.30b - 1.31b	-	HMS	All	Transferred to Framemaker, minor updates
1.50	2003-03-11	PeP / HMS	All	Reworked
1.51	2003-04-08	PeP / HMS	All	Minor corrections

Conventions used in this manual

The following conventions are used throughout this manual:

- Numbered lists provide sequential steps
- Bulleted lists provide information, not procedural steps
- The term ‘module’ is used when referring to the AnyBus module
- The term ‘application’ is used when referring to the hardware that is connected to the AnyBus Application Connector
- Hexadecimal values are written in the format NNNNh, where NNNN is the hexadecimal value.

Support

HMS Sweden

Email: support@hms-networks.se
Tel: +46 (0)35-17 29 22
Fax: +46 (0)35-17 29 09
Online: www.hms-networks.com

HMS America

Email: us-support@hms-networks.se
Tel: +1.773.404.2271
Fax: +1.773.404.1797
Online: www.hms-networks.com

HMS Germany

Email: ge-support@hms-networks.se
Tel: +49 721 964 72157
Fax: +49 721 964 7210
Online: www.hms-networks.com

HMS Japan

Email: jp-support@hms-networks.se
Tel: +81 45 478 5340
Fax: +81 45 476 0315
Online: www.hms-networks.com

1 Fieldbus Introduction

LonWorks technology offers a powerful means for implementing a wide variety of distributed systems that perform sensing, monitoring, and control. During installation, individual intelligent nodes are assembled into an interoperating network. This installation process, along with the on-going maintenance of the system, is referred to as network management.

LonWorks technology provides a very flexible environment, with many ways to install nodes and to tune network parameters based on application needs. This flexibility makes a LonWorks network suitable as a replacement for a wiring harness or for a master/slave control system; it also allows you to build peer-to-peer control systems. However, it is not necessary to use all the power of LonWorks technology to enjoy many of its benefits. The task of designing LonWorks networks becomes one of picking and choosing the options that are of value in a given application and presenting them in a way that is convenient to the end user or installer. The AnyBus-S LonWorks module communicates according to the LonTalk protocol. This means that it can communicate with all LonWorks nodes that comply with this protocol, but it does not necessarily mean that all services available in the LonTalk protocol is supported.

For more information about LonWorks technology, see www.echelon.com and www.lonmark.org.

1.1 Introduction to LonWorks

The LonWorks technology is a complete platform for implementing control network systems. These networks consists of intelligent control devices or nodes that interact with their environment and communicate with each other using a common, message based protocol (LonTalk®). A LonWorks network can consist of up to 32,385 nodes divided into 255 subnets (127 nodes/subnet).

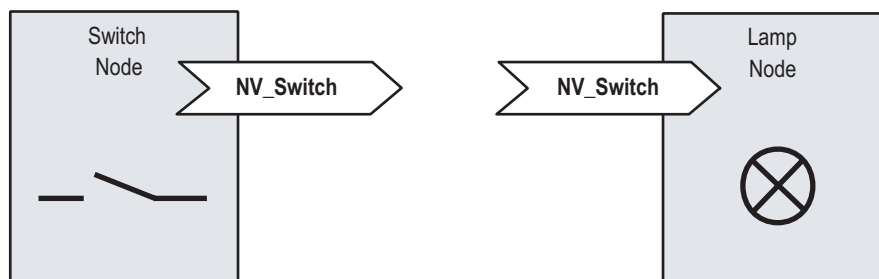
1.1.1 Nodes

Nodes are objects that interact with physically attached I/O devices and communicating over a network using the LonTalk protocol. Each node in the network contains embedded intelligence that implements the protocol and performs control functions (handling of I/O signals). No central control or master-slave architecture is needed and therefor a LonWorks network often is called "distributed I/O network". Each node includes a physical interface, transceiver that interface with the communication media. The ABS-LON module uses the FTT-10 A (Free Topology Twisted Pair) from Echelon Corporation. This is the most commonly used twisted-pair media in building automation and this architecture supports star, bus, and loop wiring. The FTT-10A transceiver connects to a twisted pair cable with a baudrate of 78 kbit/s. The FTT-10A transceiver appears as a high impedance to the network when unpowered, and does not interfere with network communications when powered down.

1.1.2 Network variables (SNVT's)

A network variable is an object on one node that can be connected to one or more network variables on one or more additional nodes. A node's network variables define its inputs and output from a network point of view and allow the sharing of data in a distributed application. Whenever a program writes into one of its output network variables, the new value of the network variable is propagated across the network to all nodes with input network variables connected to that output network variable.

For example to turn on the light over a LonWorks network, a switch node has its output network variable (nvoSwitch) connected or bound to the input variable nviSwitch on the lamp node. When the switch is activated, the network variable is propagated over the network and received in the lamp node, which turn on the light.



The LonMark organization has approved a number of network variables that represent different types of standard data representation, for example temperature, pressure, percent, voltage. They are named Standard Network Variables or SNVT's and are listed in the "SNVT Master List and Programmers Guide" from Echelon Corporation. A SNVT contains information about type, resolution and range. The LonMark association defines objects that can be described as a group of SNVT's used for a specific application. Note that network variables also can be defined by the user as non-standard and they are usually called just network variables or NV's.

1.1.3 Configuration Properties (SCPT's)

Configuration properties are used to store parameters that need to be preserved in non-volatile memory. In LonWorks there are several methods to implement SCPT's, but in the ABS-LON they can only be implemented as network variables. In this case, the network variables used for parameters are defined to store the information in FLASH memory onboard the AnyBus module.

1.1.4 Implementation of Network variables in the ABS-LON

A LonWorks node is application specific, which means that it is impossible to have a generic set of network variables for all types of applications. Therefore the ABS-LON needs to be configured with the set of network variables that is needed for a certain type of application. It is possible for experienced LonWorks users to add the network variables by special commands to the module but we recommend you to contact HMS for more information about this procedure.

1.1.5 Installation

The AnyBus-S LonWorks module supports two basic ways of address assignment / installation, self installing or by a network management tool.

Network Management Tool

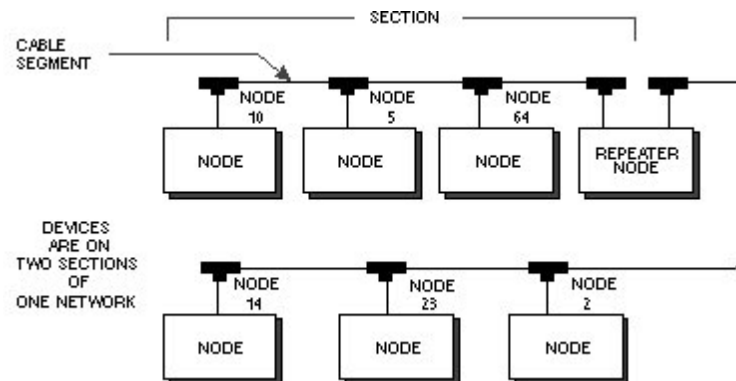
Address assignment and network variables bindings can be done using a network management tool. The bindings connect network variables on different nodes. Once the binding is done, the network management tool can be removed from the network. At installation time each node gets its unique network identity. For more information about how LonWorks nodes are installed, see the LonWorks Technology Device Data from Motorola or LonWorks documentation from Echelon.

Self-installation

Self-installation is used when the network shall be installed without a network management tool. The application must then set all the address information and bindings by sending mailbox commands to the module.

1.2 Technical Features for LonWorks

Summary Technical Features LonWorks	
Transmission Technique: <ul style="list-style-type: none"> - Free topology (FTT-10A) twisted pair cable (recommended cable Belden 7703NH) - Transmission rate 78kbits/s 	Medium Access: <ul style="list-style-type: none"> - Predictive CSMA
Max Bus length: <ul style="list-style-type: none"> - Bus up to 2000 m (with repeaters) 	Data types: <ul style="list-style-type: none"> - Peer-to-peer communication
Max. node-to-node distance: <ul style="list-style-type: none"> - 500m 	Maximum number of I/O: 256 input and 256 output network variables divided over 512 input bytes, 512 output byte
Max amount of nodes: 32.385 nodes divided as 127 nodes/subnet, 255 subnets/ domain	



2 About the AnyBus-S LonWorks Module

This section provides an overview over the module and describes the technical features.

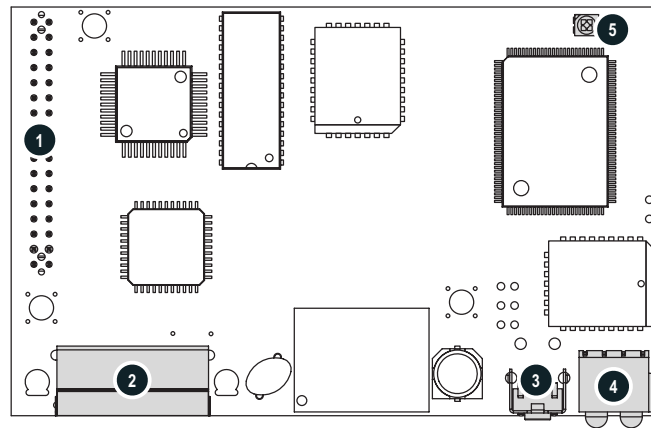
2.1 Features

- Conforms to LonMark Guidelines v. 3.2.
- Transmission media: FTT-10A
- Topology: Free topology, star or bus
- Fieldbus connectors: 9 pin female Dsub, 5-pin 3.81mm or 2mm plug-able or screw connector.
- Cable: Shielded or unshielded copper cable, One Twisted pair. ex: 7703NH from Belden (recommended from HMS)
- ASIC's and circuits: Module is based on chip-set and firmware from Echelon.
- Baudrate: 78kbps
- Supports network variables. Max number of network variables: 256 input and 256 output

2.2 Compatible Products

This product is a member of the AnyBus concept of interchangeable fieldbus modules. This makes it fully compatible with any fieldbus system supported by the AnyBus-S platform.

2.3 Overview



#	Description
1	Application Connector
2	Fieldbus Connector
3	Service Pin Push button
4	Status Indicators
5	AnyBus Watchdog

2.3.1 Connectors

Application Connector

The application connector features a standard AnyBus-S 2kbyte parallel DPRAM interface. For further information, please consult the general AnyBus-S Parallel Design Guide.

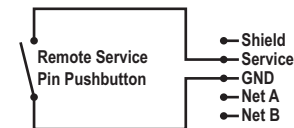
LonWorks Connector

The module supports 9 pin dsub, board to board, and pluggable screw connectors. For more information about connectors and pin assignments, see “Connector Pinouts” on page 50.

Signal	Description
Net A, Net B	This is the 2-wire LonWorks bus, which is designed according to the LonMark design guidelines. The bus is of non polarization character and supports Line Powered networks.
Service Pin	This pin is used for remote functionality of the SERVICE PIN PUSHBUTTON. See figure below.
GND	This pin is used for the Service Pin function, see connections in figure below.
Shield:	This pin is supposed to be connected to the shield of the LonWorks cable, in cases where such cable is used. The pin is, via a RC-filter, connected to the GND on the AnyBus.

2.3.2 Service Pin Push button

The module features an onboard service pin button. However, the service pin functionality can also be wired through the fieldbus connector for remote operation, see figure on the right.



2.3.3 Indicators

AnyBus Watchdog

Consult the general AnyBus-S Design Guide for further information.

Status Indicators

These leds indicate run time status and errors to the user



LED no.	State	Description
1 - Application Led	-	This led is controlled by the application using the SET_APPL_LED mailbox command, see “Set Application LED (SET_APPL_LED)” on page 25.
2 - Service Led	Green	Node is unconfigured and applicationless / errors detected by the Neuron self test.
	Green, flashing	This node has an application but is not yet installed in a network (unconfigured)
	Off	This node is configured and installed in a network.
3 - Wink Led	Red, flashing	A wink command is received on the LonWorks network
	Off	Normal state.
4 - Module Status	Green	The module has been initialized correctly and is working ok.
	Red	A hardware error has occurred
	Red, flashing	Software error. Restart the module.

3 Functions & Operation

3.1 Initialization

Initialization of the AnyBus-S module in standard format is performed with the ANYBUS_INIT mailbox telegram, where the sizes of the input/output areas are set. The sizes of these areas must be set to 512 bytes each on the AnyBus-S LonWorks module. The sequence for this type of initialization is START_INIT, ANYBUS_INIT and END_INIT.

3.2 Error Handling

There are a number of errors that can occur during runtime. The latest error will be available in the fieldbus specific OUT area, starting at offset 0x700. The error is represented as an error number together with optional data. The possible errors that can occur is listed in the table below:

Error Number	Meaning of optional data	Explanation
0x00	-	No Error
0x01	Index of network variable update that failed	Network Variable Update Failure. No ACK from receiver to network variable update.
0x02	-	Network Communication Error. An internal command to the network failed.
0x03	-	Network driver error. The module has a problem communicating with the local neuronchip.
0x10->	-	Internal Error in the module. Contact HMS.

3.3 DPRAM Structure of the LonWorks Module

The application interface is thoroughly described in the AnyBus-S Design Guide. Many items are the same for all AnyBus-S modules. In the sections below, only the fieldbus specific items are described, i.e., items which are individual for the AnyBus-S LonWorks module.

3.3.1 IN/OUT Data Area

The I/O areas are directly mapped against network variables with no scaling. This means that the application needs to take care of reading and writing data that corresponds to the data type of the network variable. Variables are mapped with an offset to the most significant byte of the data type. **The I/O area must always be initiated to 512 bytes input and 512 bytes output.**

Area (HEX)	Contents	Access
000h - 1FFh	IN area Input data area ; 512 bytes	R/W
200h - 3FFh	OUT area Output data area; 512 Bytes	RO

Example:

Lets say that the first output network variable in the node (index 0) is of the type SNVT_switch and mapped with offset 0 to the IN area. The second output network variable is if the type SNVT_lev_percent and mapped to offset 2 of the IN area. The type declaration of the variables looks like this (Check the SNVT Master List and Programmer's Guide, for more information about these types):

```
Typedef struct {
    unsigned char value; // 1byte, valid range 0 - 100%, resolution 0.5%
    signed state;        // 1 byte, 0 = Off, 1 = ON, 0xFF = Undefined
} SNVT_switch;
```

Typedef signed long SNVT_lev_percent; // 2 bytes, -163.84% ... 163.83% , resolution 0.005%

NV Index	Meaning	IN Area (HEX)
0	Value (of SNVT_switch)	000h
	State (of SNVT_switch)	001h
1	High byte of SNVT_lev_percent	002h
	Low byte of SNVT_lev_percent	003h

3.3.2 Control Area

These registers contain information about the status and control environment, both for the fieldbus and the module itself, e.g. Fieldbus type, Module Software version, Module Type, configuration, etc. The following registers are specific for the AnyBus-S LonWorks module. For further information about the Control Area contents, please see the AnyBus-S Design Guide.

Area (HEX)	Contents	Access
7CCh - 7CDh	Fieldbus Type (0x0015 for LonWorks)	RO
7CEh - 7CFh	Module Software Version	RO

3.3.3 Fieldbus Specific Area

This area has two purposes, information about network variable updates and error reporting.

Area (HEX)	Contents	Access
640h - 6BFh	Fieldbus Specific IN area Input data area ; 128 bytes	R/W
6C0h - 73Fh	Fieldbus Specific OUT area Output data area ; 128 bytes	RO

For specific information about the content of each area address, see next page.

Fieldbus Specific IN area

Area (HEX)	Contents	Valid data	Access
640h	Network Variable update notification ACK (0-7)	Bitfield	R/W
641h	Network Variable update notification ACK (8-15)	Bitfield	R/W
642h	Network Variable update notification ACK (16-23)	Bitfield	R/W
643h	Network Variable update notification ACK (24-31)	Bitfield	R/W
644h	Network Variable update notification ACK (32-39)	Bitfield	R/W
645h	Network Variable update notification ACK (40-47)	Bitfield	R/W
646h	Network Variable update notification ACK (48-55)	Bitfield	R/W
647h	Network Variable update notification ACK (56-63)	Bitfield	R/W
648h	Network Variable update notification ACK (64-71)	Bitfield	R/W
649h	Network Variable update notification ACK (72-79)	Bitfield	R/W
64Ah	Network Variable update notification ACK (80-87)	Bitfield	R/W
64Bh	Network Variable update notification ACK (88-95)	Bitfield	R/W
64Ch	Network Variable update notification ACK (96-103)	Bitfield	R/W
64Dh	Network Variable update notification ACK (104-111)	Bitfield	R/W
64Eh	Network Variable update notification ACK (112-119)	Bitfield	R/W
64Fh	Network Variable update notification ACK (120-127)	Bitfield	R/W
650h	Network Variable update notification ACK (128-135)	Bitfield	R/W
651h	Network Variable update notification ACK (136-143)	Bitfield	R/W
652h	Network Variable update notification ACK (144-151)	Bitfield	R/W
653h	Network Variable update notification ACK (152-159)	Bitfield	R/W
654h	Network Variable update notification ACK (160-167)	Bitfield	R/W
655h	Network Variable update notification ACK (168-175)	Bitfield	R/W
656h	Network Variable update notification ACK (176-183)	Bitfield	R/W
657h	Network Variable update notification ACK (184-191)	Bitfield	R/W
658h	Network Variable update notification ACK (192-199)	Bitfield	R/W
659h	Network Variable update notification ACK (200-207)	Bitfield	R/W
65Ah	Network Variable update notification ACK (208-215)	Bitfield	R/W
65Bh	Network Variable update notification ACK (216-223)	Bitfield	R/W
65Ch	Network Variable update notification ACK (224-231)	Bitfield	R/W
65Dh	Network Variable update notification ACK (232-239)	Bitfield	R/W
65Eh	Network Variable update notification ACK (240-247)	Bitfield	R/W
65Fh	Network Variable update notification ACK (248-255)	Bitfield	R/W
660h	Not Used		
...	Not Used		-
6BFh	Not Used		

Fieldbus Specific OUT area

Area (HEX)	Contents	Valid data	Access
6C0h	Network Variable update notification ACK (0-7)	Bitfield	RO
6C1h	Network Variable update notification ACK (8-15)	Bitfield	RO
6C2h	Network Variable update notification ACK (16-23)	Bitfield	RO
6C3h	Network Variable update notification ACK (24-31)	Bitfield	RO
6C4h	Network Variable update notification ACK (32-39)	Bitfield	RO
6C5h	Network Variable update notification ACK (40-47)	Bitfield	RO
6C6h	Network Variable update notification ACK (48-55)	Bitfield	RO
6C7h	Network Variable update notification ACK (56-63)	Bitfield	RO
6C8h	Network Variable update notification ACK (64-71)	Bitfield	RO
6C9h	Network Variable update notification ACK (72-79)	Bitfield	RO
6CAh	Network Variable update notification ACK (80-87)	Bitfield	RO
6CBh	Network Variable update notification ACK (88-95)	Bitfield	RO
6CCh	Network Variable update notification ACK (96-103)	Bitfield	RO
6CDh	Network Variable update notification ACK (104-111)	Bitfield	RO
6CEh	Network Variable update notification ACK (112-119)	Bitfield	RO
6CFh	Network Variable update notification ACK (120-127)	Bitfield	RO
6D0h	Network Variable update notification ACK (128-135)	Bitfield	RO
6D1h	Network Variable update notification ACK (136-143)	Bitfield	RO
6D2h	Network Variable update notification ACK (144-151)	Bitfield	RO
6D3h	Network Variable update notification ACK (152-159)	Bitfield	RO
6D4h	Network Variable update notification ACK (160-167)	Bitfield	RO
6D5h	Network Variable update notification ACK (168-175)	Bitfield	RO
6D6h	Network Variable update notification ACK (176-183)	Bitfield	RO
6D7h	Network Variable update notification ACK (184-191)	Bitfield	RO
6D8h	Network Variable update notification ACK (192-199)	Bitfield	RO
6D9h	Network Variable update notification ACK (200-207)	Bitfield	RO
6DAh	Network Variable update notification ACK (208-215)	Bitfield	RO
6DBh	Network Variable update notification ACK (216-223)	Bitfield	RO
6DCh	Network Variable update notification ACK (224-231)	Bitfield	RO
6DDh	Network Variable update notification ACK (232-239)	Bitfield	RO
6DEh	Network Variable update notification ACK (240-247)	Bitfield	RO
6DFh	Network Variable update notification ACK (248-255)	Bitfield	RO
6E0h	Not Used		RO
...	Not Used		RO
700h	Last Error	Byte	RO
701h	Optional data (MSB)	Unsigned 32bit	RO
702h	Optional data		RO
703h	Optional data		RO
704h	Optional data (LSB)		RO
...	Not used		
73Fh	Not Used		-

3.4 Data Exchange

3.4.1 Data coming from the network

Data/updates from the LonWorks network will come as an update on an input network variable on the Any-Bus-S module and then be mapped to the corresponding offset in the OUT data area of the DPRAM. As a network variable can be updated without changed data, updates to a variable will be indicated through the fieldbus specific OUT area. Each network variable index is represented as a bit in the "Network Variable update notification" bytes in this area. When an update on a variable occurs, the corresponding bit will be toggled in this area. The data will then be available in the mapped offset in the I/O area for this variable.

NOTE! All network variable indexes has a corresponding bit in this area even if they are outputs, but updates will only be notified if they are coming from input network variables.

When the application has read the update, it must clear the update by setting the corresponding bit in the "Network Variable update notification ACK" bytes to an equal level in the fieldbus specific IN area, to prepare for a new update. The sequence can be described like this:

1. An update occurs on the network variable index 1, which is notified with the corresponding "Network Variable update notification".

6C0h

-	1	-	-	-	-	-	-
---	---	---	---	---	---	---	---

2. The application reads the data in the I/O Area and then acknowledges the update by setting the corresponding "Network Variable update notification ACK" bit to the same level.

640h

-	1	-	-	-	-	-	-
---	---	---	---	---	---	---	---

3. If a new update occurs on the network variable index 1, this is notified with the corresponding "Network Variable update notification". As the bit was set to '1' before, the indication is done by clearing this bit this time.

6C0h

-	0	-	-	-	-	-	-
---	---	---	---	---	---	---	---

4. The application reads the data in the I/O Area and then acknowledges the update by clearing the corresponding "Network Variable update notification ACK".

640h

-	0	-	-	-	-	-	-
---	---	---	---	---	---	---	---

3.4.2 Data coming from the application

Data/updates from the application will be mapped from the IN data area in the DPRAM and sent over the LonWorks network as an update to an output network variable.

There are two methods to update output network variables to the network, automatic or manual mode. Modes are selected at initialization with the "CONFIG_EXCHANGE" mailbox call. When auto mode is selected, network variable updates will be propagated to the network when data changes in the mapped I/O area. When manual mode is selected, variables will be propagated to the network when the "PROPAGATE_NV" mailbox call is sent.

To update an output network variable with index 3 mapped to IN-area 640h in manual mode:

1. Application writes data to adress 640h.
2. Application sends the "PROPAGATE_NV" mailbox call for network variable index 3.
3. The AnyBus sends out a network variable update message to the network if the variable is bound to another node.

4 Fieldbus Specific Mailbox Commands

This chapter describes the fieldbus specific mailbox commands in the module. Consult the AnyBus-S Design Guide for more information regarding mailbox functionality.

4.1 Fault information

When a mailbox command cannot be processed the Message Information register in the header of the response will indicate that an error occurred. Consult the general AnyBus-S Design Guide for more information.

If the error code is 'Invalid Other' (Fh), extended error information is available in the Fault Information register (Extended word 8).

The fault codes in the Fault Information register are:

Register Value	Description
0x0001	Index not valid
0x0002	Invalid Data
0x0003	Update Failure
0x0004	Internal Error

4.2 Configuration Commands

4.2.1 Configure Data Exchange (CONFIG_EXCHANGE)

This mailbox command configures if output network variables should be automatically propagated to the network when changed. Auto mode becomes default if this command isn't sent. If auto mode is chosen, the variables will only propagate to the network if the value have changed, which makes this method unsuitable when heartbeat updates are needed. If heartbeats are used in the network, manual propagates should be used instead (see "Propagate NV to Network (PROPAGATE_NV)" on page 28).

Note: This command can only be sent during module initialization.

Parameter	Description
Command initiator	Application
Message name	CONFIG_EXCHANGE
Message type	0x02
Command number	0x0006
Fragmented	No
Extended header data	No Extended Header data is associated with this command.
Message data	Use automatic NV-propagate
Response message	The response data is a copy of the command data.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	Any integer value
Message information	0x4002	0x0002	
Command	0x0006	0x0006	
Data size	0x0001	0x0001	
Frame number	0x0001	0x0001	One frame
Frame count	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	-	Fault information	
Message data byte 1	Exchange Config	Exchange Config	See table below

Message Data:

Exchange Config value	Explanation
TRUE (0x01)	Sets the data exchange in manual mode
FALSE (0x00)	Sets the data exchange in auto mode (default)

4.2.2 Configure Force Interrupt (SET_FORCE_INT)

This mailbox command configures if updates to input network variables should trigger a new data interrupt to the application even if the data haven't changed. Normally, an AnyBus module only triggers a new data interrupt if data in the OUT area are changed. This command will use the last byte of the OUT area as an update counter, which will be incremented on every network variable update. Default configuration is no forcing.

Note: This command can only be sent during module initialization.

Parameter	Description
Command initiator	Application
Message name	SET_FORCE_INT
Message type	0x02
Command number	0x0007
Fragmented	No
Extended header data	No Extended Header data is associated with this command.
Message data	Force update interrupt value, see below.
Response message	The response data is a copy of the command data.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	Any integer value
Message information	0x4002	0x0002	
Command	0x0007	0x0007	
Data size	0x0001	0x0001	
Frame number	0x0001	0x0001	One frame
Frame count	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	-	Fault information	
Message data byte 1	Force Interrupt	Force Interrupt	See table below

Force Interrupt value	Explanation
TRUE (0x01)	Forces new data interrupt even if data is not changed.
FALSE (0x00)	Normal operation (default).

4.2.3 Set Mapping of Variable (SET_NV_MAPPING)

This mailbox command is used to set mapping of a network variable towards the I/O Areas. It maps a given network variable index to an offset in the I/O-Areas. The module will automatically decide if it should be mapped to the input or output area depending on the direction of the network variable.

Note: This command can only be sent during module initialization.

Parameter	Description
Command initiator	Application
Message name	SET_NV_MAPPING
Message type	0x02
Command number	0x0030
Fragmented	No
Extended header data	No Extended Header data is associated with this command.
Message data	Network Variable index and Offset to I/O area, see below.
Response message	The response data is a copy of the command data.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	Any integer value
Message information	0x4002	0x0002	
Command	0x0030	0x0030	
Data size	0x0004	0x0004	
Frame number	0x0001	0x0001	One frame
Frame count	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	-	Fault information	
Message data byte 1	Network Variable Index (Hi)	Network Variable Index (Hi)	
Message data byte 2	Network Variable Index (Lo)	Network Variable Index (Lo)	
Message data byte 3	Offset (Hi)	Offset (Hi)	
Message data byte 4	Offset (Lo)	Offset (Lo)	

Message Data	Range	Explanation
Network Variable Index	0 - Number of NV's declared	Index of the network variable to be mapped, starting from 0.
Offset	0 - 511	Offset in input or output I/O-areas depending on the direction of the network variable.

4.2.4 Get Mapping of Variable (GET_NV_MAPPING)

This mailbox command can be sent to get the mapping towards the I/O area of a network variable from a given index.

Parameter	Description
Command initiator	Application
Message name	GET_NV_MAPPING
Message type	0x02
Command number	0x0031
Fragmented	No
Extended header data	No Extended Header data is associated with this command.
Message data	Network Variable Index, see below.
Response message	Network Variable index, DPRAM Offset and Direction (See below)

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	Any integer value
Message information	0x4002	0x0002	
Command	0x0031	0x0031	
Data size	0x0002	0x0005	
Frame number	0x0001	0x0001	One frame
Frame count	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	Network Variable Index (Hi)	Network Variable Index (Hi)	
Message data byte 2	Network Variable Index (Lo)	Network Variable Index (Lo)	
Message data byte 3		Offset (Hi)	
Message data byte 4		Offset (Lo)	
Message data byte 5		Direction	

Message Data	Range	Explanation
Network Variable Index	0 - Number of NV's declared (Max 512)	Index of the network variable to be mapped, starting from 0.
Offset	0 - 511	Offset in input or output I/O-areas depending on the direction of the network variable.
Direction	0: Variable is an input 1: Variable is an output	Direction of the network variable on this index. Example: An input network variable (direction 0) is mapped to the DPM OUT Area.

4.2.5 Set Location (SET_LOCATION)

This mailbox command sets the location string in the node. The location string is a 6-byte ASCII-string, used to describe a physical location of a LonWorks node.

Parameter	Description
Command initiator	Application
Message name	SET_LOCATION
Message type	0x02
Command number	0x0016
Fragmented	No
Extended header data	No Extended Header data is associated with this command.
Message data	Location String, 6 byte ASCII string without termination
Response message	The response data is a copy of the command data.

Command and response layout:

	Command	Expected response	
Message ID	(ID)	(ID)	Any integer value
Message information	0x4002	0x0002	
Command	0x0016	0x0016	
Data size	0x06	0x06	
Frame number	0x0001	0x0001	One frame
Frame count	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	-	Fault information	
Message data byte 1	Location String [0]	Location String [0]	
Message data byte 2	Location String [1]	Location String [1]	
Message data byte 3	Location String [2]	Location String [2]	
Message data byte 4	Location String [3]	Location String [3]	
Message data byte 5	Location String [4]	Location String [4]	
Message data byte 6	Location String [5]	Location String [5]	

4.2.6 Get Location (GET_LOCATION)

This mailbox command returns the location string in the node. The location string is a 6-byte ASCII-string used to describe a physical location of a LonWorks node.

Parameter	Description
Command initiator	Application
Message name	GET_LOCATION
Message type	0x02
Command number	0x0017
Fragmented	No
Extended header data	No Extended Header data is associated with this command.
Message data	-
Response message	Location String, 6 byte ASCII string without termination

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	Any integer value
Message information	0x4002	0x0002	
Command	0x0017	0x0017	
Data size	0x0000	0x0006	
Frame number	0x0001	0x0001	One frame
Frame count	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	-	Fault information	
Message data byte 1		Location String [0]	
Message data byte 2		Location String [1]	
Message data byte 3		Location String [2]	
Message data byte 4		Location String [3]	
Message data byte 5		Location String [4]	
Message data byte 6		Location String [5]	

4.2.7 Get Neuron ID (GET_NEURON_ID)

This mailbox command returns Neuron ID and program ID in the node. The Neuron ID is a 6-byte number that is unique for each manufactured LonWorks node. The program ID is an 8 byte binary value if the node conforms to the LonMark application layer interoperability guidelines or an ASCII string if it doesn't. The program ID should be unique for each network variable configuration.

Parameter	Description
Command initiator	Application
Message name	GET_NEURON_ID
Message type	0x02
Command number	0x0019
Fragmented	No
Extended header data	No Extended Header data is associated with this command.
Message data	-
Response message	6 byte Neuron ID and 8 byte Program ID (without termination)

Command and response layout

	Command	Expected response	
<i>Message ID</i>	(ID)	(ID)	<i>Any integer value</i>
<i>Message information</i>	0x4002	0x0002	
<i>Command</i>	0x0019	0x0019	
<i>Data size</i>	0x0000	0x000E	
<i>Frame number</i>	0x0001	0x0001	<i>One frame</i>
<i>Frame count</i>	0x0001	0x0001	<i>This is frame 1</i>
<i>Offset high</i>	0x0000	0x0000	
<i>Offset low</i>	0x0000	0x0000	
<i>Extended word 1</i>	-	-	<i>No message header data used</i>
<i>Extended word 2</i>	-	-	"
<i>Extended word 3</i>	-	-	"
<i>Extended word 4</i>	-	-	"
<i>Extended word 5</i>	-	-	"
<i>Extended word 6</i>	-	-	"
<i>Extended word 7</i>	-	-	"
<i>Extended word 8</i>	-	-	"
<i>Message data byte 1</i>		Fault information	
<i>Message data byte 2</i>		Neuron ID [0]	
<i>Message data byte 3</i>		Neuron ID [1]	
<i>Message data byte 4</i>		Neuron ID [2]	
<i>Message data byte 5</i>		Neuron ID [3]	
<i>Message data byte 6</i>		Neuron ID [4]	
<i>Message data byte 7</i>		Neuron ID [5]	
<i>Message data byte 8</i>		Program ID [0]	
<i>Message data byte 9</i>		Program ID [1]	
<i>Message data byte 10</i>		Program ID [2]	
<i>Message data byte 11</i>		Program ID [3]	
<i>Message data byte 12</i>		Program ID [4]	
<i>Message data byte 13</i>		Program ID [5]	
<i>Message data byte 14</i>		Program ID [6]	
		Program ID [7]	

4.2.8 Set Network Variable Data (SET_NETVAR_DATA)

This mailbox command sets startup data for network variables.

Parameter	Description
Command initiator	Application
Message name	SET_NETVAR_DATA
Message type	0x02
Command number	0x0040
Fragmented	No
Extended header data	No Extended Header data is associated with this command.
Message data	Network Variable Index and data, see below.
Response message	The response data is a copy of the command data.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	Any integer value
Message information	0x4002	0x0002	
Command	0x0040	0x0040	
Data size	0x0002 + n	0x0002 + n	
Frame number	0x0001	0x0001	One frame
Frame count	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	-	Fault information	
Message data byte 1	NetVar Index (Hi)	NetVar Index (Hi)	
Message data byte 2	NetVar Index (Lo)	NetVar Index (Lo)	
Msg. data	Data [0] ... [n]	Data [0] ... [n]	

Message Data	Range	Explanation
Network Variable Index	0 - Number of NV's declared	Index of the network variable to be propagated, starting from 0.
Data	Netvar Size	Data with the same size as the network variable

4.3 Runtime Commands

4.3.1 Set Application LED (SET_APPL_LED)

This command is used to set the status of the LED that can be controlled by the application.

Parameter	Description
Command initiator	Application
Message name	SET_APPL_LED
Message type	0x02
Command number	0x0050
Fragmented	No
Extended header data	No Extended Header data is associated with this command.
Message data	Led function, see below.
Response message	The response data is a copy of the command data.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	Any integer value
Message information	0x4002	0x0002	
Command	0x0050	0x0050	
Data size	0x0002	0x0002	
Frame number	0x0001	0x0001	One frame
Frame count	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	-	Fault information	
Message data byte 1	Green LED Function	Green LED Function	
Message data byte 2	Red LED Function	Red LED Function	

Red / Green LED Function value	Explanation
0x00: LED OFF	The application LED is a bicolored LED (Green/Red), and this controls the state of the green and red color. NOTE! You will get unpredictable results if both colors are used simultaneously!
0x01: FLASH 1 Hz	
0x02: FLASH 2 Hz	
0x04: FLASH 4 Hz	
0x08: FLASH 8 Hz	
0xFF: LED ON	

4.3.2 Set Node Mode (SET_NODE_MODE)

This mailbox command used to set the mode of the module on the LonWorks network. **NOTE!** Normally it's only necessary to use this command if the node is using self installation. If the node is installed by a network management tool, this tool should be used to set the node into the appropriate mode.

Parameter	Description
Command initiator	Application
Message name	SET_NODE_MODE
Message type	0x02
Command number	0x0054
Fragmented	No
Extended header data	No Extended Header data is associated with this command.
Message data	Node Mode, see below.
Response message	The response data is a copy of the command data.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	Any integer value
Message information	0x4002	0x0002	
Command	0x0054	0x0054	
Data size	0x0001	0x0001	
Frame number	0x0001	0x0001	One frame
Frame count	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	Node Mode	Fault information	
		Node Mode	

Node Mode	Explanation
0x00: Set to Online	Controls the network status mode of the module. The module will only send network variable updates if the module is Configured and Online.
0x01: Set to Offline	When the module is in Offline state, it will not send out any network variable updates. It will however answer to network management messages and network variable fetch messages coming from the network.
0x02: Set to Configured	-
0x03: Set to Unconfigured	-
0x04: Set to Hard Offline	The mode Hard Offline is the same as Offline but with the difference that this state is preserved when the module is reset.
0x05: Reset Network Interface	The command 0x05 only resets the network communication chip and not the whole module.
0x06: Send out a Service Pin Message	The command 0x06 sends out a Service Pin Message equal to pressing the Service Pin button on the module.

4.3.3 Get Node Mode (GET_NODE_MODE)

This mailbox command is used to get the mode that the module have on the LonWorks network.

Parameter	Description
Command initiator	Application
Message name	GET_NODE_MODE
Message type	0x02
Command number	0x0055
Fragmented	No
Extended header data	No Extended Header data is associated with this command.
Message data	-
Response message	0x00 = Online, 0x01 = Offline (Other values are reserved for future software revisions)

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	Any integer value
Message information	0x4002	0x0002	
Command	0x0055	0x0055	
Data size	0x0000	0x0002	
Frame number	0x0001	0x0001	One frame
Frame count	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		Fault information	
Message data byte 2		Node Mode	
		Node State	

Data Word	Range	Explanation
Node Mode	0x00: Online 0x01: Offline	The network mode.
Node State	0x00: Configured 0x01: Unconfigured	Configured means that the node is commissioned and bound to the network.

4.3.4 Propagate NV to Network (PROPAGATE_NV)

This command is used to force an update of a given network variable to the network. The update will be sent with the latest data in the mapped I/O area.

NOTE! This command will work in both Auto and Manual exchange mode. In Auto mode this command will cause the module to send two network variable updates to the network if the data has changed.

Parameter	Description
Command initiator	Application
Message name	PROPAGATE_NV
Message type	0x02
Command number	0x0005
Fragmented	No
Extended header data	No Extended Header data is associated with this command.
Message data	Index of the network variable to be propagated, starting from 0.
Response message	The response data is a copy of the command data.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	Any integer value
Message information	0x4002	0x0002	
Command	0x0005	0x0005	
Data size	0x0002	0x0002	
Frame number	0x0001	0x0001	One frame
Frame count	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	-	Fault information	
Message data byte 1	NetVar Index (Hi)	NetVar Index (Hi)	
Message data byte 2	NetVar Index (Lo)	NetVar Index (Lo)	

4.3.5 Get Errors (GET_ERRORS)

The AnyBus module has an error table containing the last ten errors that have occurred in the node. The latest error is always present in the fieldbus specific out area. This command can be used to read all errors present in the error table.

The data will be ordered from the latest occurred error to the oldest. See section 3.2 for information about possible errors.

Parameter	Description
Command initiator	Application
Message name	GET_ERRORS
Message type	0x02
Command number	0x0065
Fragmented	No
Extended header data	No Extended Header data is associated with this command.
Message data	-
Response message	The latest errors in the module.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	Any integer value
Message information	0x4002	0x0002	
Command	0x0065	0x0065	
Data size	0x0000	0x0032	
Frame number	0x0001	0x0001	One frame
Frame count	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		Fault information	
Message data byte 2		Type Of Error 1	
Message data byte 3		Optional Data [0]	
Message data byte 4		Optional Data [1]	
Message data byte 5		Optional Data [2]	
Message data byte 6		Optional Data [3]	
Message data byte 7		Type Of Error 2	
Message data byte 8		Optional Data [0]	
Message data byte 9		Optional Data [1]	
		Optional Data [n]	

Message Data	Range	Explanation
Type Of Error	Enum.	Error type. See "Error Handling" on page 10
Optional Data	Depending on type of error	Optional integer data depending on type of error. See section 'Errorhandling'

4.3.6 Clear Errors (CLEAR_ERRORS)

This command is used to clear the error information table in the module.

Parameter	Description
Command initiator	Application
Message name	CLEAR_ERRORS
Message type	0x02
Command number	0x0060
Fragmented	No
Extended header data	No Extended Header data is associated with this command.
Message data	-
Response message	-

Command and response layout

	Command	Expected response	
<i>Message ID</i>	(ID)	(ID)	<i>Any integer value</i>
<i>Message information</i>	0x4002	0x0002	
<i>Command</i>	0x0060	0x0060	
<i>Data size</i>	0x0000	0x0000	
<i>Frame number</i>	0x0001	0x0001	<i>One frame</i>
<i>Frame count</i>	0x0001	0x0001	<i>This is frame 1</i>
<i>Offset high</i>	0x0000	0x0000	
<i>Offset low</i>	0x0000	0x0000	
<i>Extended word 1</i>	-	-	<i>No message header data used</i>
<i>Extended word 2</i>	-	-	"
<i>Extended word 3</i>	-	-	"
<i>Extended word 4</i>	-	-	"
<i>Extended word 5</i>	-	-	"
<i>Extended word 6</i>	-	-	"
<i>Extended word 7</i>	-	-	"
<i>Extended word 8</i>	-	Fault information	

4.3.7 Get Node Status (GET_NODE_STATUS)

This command queries the module for extended status of the LonWorks network interface. The data that is returned holds information about communication errors/timeouts and last reset cause.

Parameter	Description
Command initiator	Application
Message name	GET_NODE_STATUS
Message type	0x02
Command number	0x0057
Fragmented	No
Extended header data	No Extended Header data is associated with this command.
Message data	-
Response message	The response data contains information about communication errors/timeouts and last reset cause. (See next page)

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	Any integer value
Message information	0x4002	0x0002	
Command	0x0057	0x0057	
Data size	0x0000	0x000F	
Frame number	0x0001	0x0001	One frame
Frame count	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	-	Fault information	
Message data byte 1		Xmit Errors (Hi)	
Message data byte 2		Xmit Errors (Lo)	
Message data byte 3		Trans. Timeouts (Hi)	
Message data byte 4		Trans. Timeouts (Lo)	
Message data byte 5		Rcv Trans. Full (Hi)	
Message data byte 6		Rcv Trans. Full (Lo)	
Message data byte 7		Lost Msgs (Hi)	
Message data byte 8		Lost Msgs (Lo)	
Message data byte 9		Missed Msgs (Hi)	
Message data byte 10		Missed Msgs (Lo)	
Message data byte 11		Last Reset Cause	
Message data byte 12		Error Log	
Message data byte 13		Node State	
Message data byte 14		Reserved	
Message data byte 15		Reserved	

Message Data	Value	Explanation
Xmit Errors	-	The number of CRC errors detected during packet reception. These may be due to collisions or noise on the transceiver input.
Trans. Timeouts	-	The number of times that the node failed to receive expected acknowledgements or responses after retrying the configured number of times. These may be due to destination nodes being inaccessible on the network, transmission failures because of noise on the channel, or if any destination node has insufficient buffers or receive transaction records.
Rcv. Trans. Full	-	The number of times that an incoming packet was discarded because there was no room in the transaction database. This may be due to excessively long receive timers, or inadequate size of the transaction database.
Lost Msgs.	-	The number of times that an incoming packet was discarded because there was no application buffer available. This may be due to an application program being too slow to process incoming packets, insufficient application buffers, or excess traffic on the channel. If the incoming message is too large for the application buffer, an error is logged, but the lost message count is not incremented.
Missed Msgs.	-	The number of times that an incoming packet was discarded because there was no network buffer available. This may be due to excess traffic on the channel, insufficient network buffers, or the network buffers not being large enough to accept all packets on the channel, whether or not addressed to this node.
Last Reset Cause	Power-up: 0bXXXXXX1 External: 0bXXXXXX10 Watchdog: 0bXXXX1100 Software: 0bXXX10100 Cleared: 0b00000000 (X = don't care)	The last reset cause of the network interface (not a reset of the anybus module as whole).

Message Data	Value	Explanation
Error Log	No Error = 0, Bad_event = 129 Nv_length_mismatch = 130 Nv_msg_too_short = 131 Eeprom_write_fail = 132 Bad_address_type = 133 Preemption_mode_timeout = 134 Already_preempted = 135 Sync_nv_update_lost = 136 Invalid_resp_alloc = 137 Invalid_domain = 138 Read_past_end_of_msg = 139 Write_past_end_of_msg = 140 Invalid_addr_table_index = 141 Incomplete_msg = 142 Nv_update_on_output_nv = 143 No_msg_avail = 144 Illegal_send = 145 Unknown_pdu = 146 Invalid_nv_index = 147 Divide_by_zero = 148 Invalid_appl_error = 149 Memory_alloc_failure = 150 Write_past_end_of_net_buffer = 151 Appl_cs_error = 152 Cnfg_cs_error = 153 Invalid_xcvr_reg_addr = 154 Xcvr_reg_timeour = 155 Write_past_end_of_appl_buffer = 156 Io_ready = 157 Self_test_failed = 158 Subnet_router = 159 Authentication_mismatch = 160 Self_inst_semaphore_set = 161 Read_write_semaphore_set = 162 Appl_signature_bad = 163 Router_firmware_version_mismatch = 164	This byte indicates the reason for the last error detected by the firmware in the network interface. The error number is stored in EEPROM. A 0 means that no error has been detected since the last reset. Errors 134, 135, 150 and 151 cause the network interface to reset.
Node State	Unconfigured = 0x02 Configured, Hard Offline = 0x06 Configured, Soft Offline = 0x0C Configured, Bypass Offline = 0x8C Configured, Online = 0x04	This byte returns the current state of the network interface. Configured Online is the normal run time mode, and means that the node is commissioned and online on the network. Soft/Bypass/Hard Offline means that the node is not participating actively on the network.
Reserved	Reserved	Reserved for future implementations

4.3.8 Clear Node Status (CLEAR_NODE_STATUS)

This command is used to clear the status information on the network interface.

Parameter	Description
Command initiator	Application
Message name	CLEAR_NODE_STATUS
Message type	0x02
Command number	0x0060
Fragmented	No
Extended header data	No Extended Header data is associated with this command.
Message data	-
Response message	-

Command and response layout

	Command	Expected response	
<i>Message ID</i>	(ID)	(ID)	<i>Any integer value</i>
<i>Message information</i>	0x4002	0x0002	
<i>Command</i>	0x0056	0x0056	
<i>Data size</i>	0x0000	0x0000	
<i>Frame number</i>	0x0001	0x0001	<i>One frame</i>
<i>Frame count</i>	0x0001	0x0001	<i>This is frame 1</i>
<i>Offset high</i>	0x0000	0x0000	
<i>Offset low</i>	0x0000	0x0000	
<i>Extended word 1</i>	-	-	<i>No message header data used</i>
<i>Extended word 2</i>	-	-	"
<i>Extended word 3</i>	-	-	"
<i>Extended word 4</i>	-	-	"
<i>Extended word 5</i>	-	-	"
<i>Extended word 6</i>	-	-	"
<i>Extended word 7</i>	-	-	"
<i>Extended word 8</i>	-	Fault information	

4.4 Network Management Commands

NOTE! The commands in this section should only be used if you have knowledge in how LonWorks address assignment and binding of network variables works. These commands should NOT be used if the node is installed by an external network management tool, as these will destroy the address information set by this tool.

These commands are used when the node is acting in a self-installation network. This gives the application the ability to set all the address information needed to bind the node together with other nodes in the LonWorks network.

4.4.1 Set Domain (SET_DOMAIN)

This mailbox command sets the domain to which this node belongs. A LonWorks node has two domain entries, and this command must be called twice to set both entries.

Note: This command can only be sent during module initialisation.

Parameter	Description
Command initiator	Application
Message name	SET_DOMAIN
Message type	0x02
Command number	0x0010
Fragmented	No
Extended header data	No Extended Header data is associated with this command.
Message data	Index in the domain table, ID length, Domain ID, Subnet ID, Node ID, Authentication Key
Response message	The response data is a copy of the command data.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	Any integer value
Message information	0x4002	0x0002	
Command	0x0010	0x0001	
Data size	0x0010	0x0010	
Frame number	0x0001	0x0001	One frame
Frame count	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	-	Fault information	
Message data byte 1	Domain Index	Domain Index	
Message data byte 2	ID Length	ID Length	
Message data byte 3	Domain ID [0]	Domain ID [0]	
Message data byte 4	Domain ID [1]	Domain ID [1]	
Message data byte 5	Domain ID [2]	Domain ID [2]	
Message data byte 6	Domain ID [3]	Domain ID [3]	
Message data byte 7	Domain ID [4]	Domain ID [4]	
Message data byte 8	Domain ID [5]	Domain ID [5]	
Message data byte 9	Subnet ID	Subnet ID	
Message data byte 10	Node ID	Node ID	
Message data byte 11	Authentication Key [0]	Authentication Key [0]	
Message data byte 12	Authentication Key [1]	Authentication Key [1]	
Message data byte 13	Authentication Key [2]	Authentication Key [2]	
Message data byte 14	Authentication Key [3]	Authentication Key [3]	
Message data byte 15	Authentication Key [4]	Authentication Key [4]	
Message data byte 16	Authentication Key [5]	Authentication Key [5]	

Message Data	Value	Explanation
Domain Index	0 or 1	Index to domain table in the module.
ID Length	0, 1, 3 or 6 0xFF means not in use	Specifies the length of the domain ID.
Domain ID	0x00 - 0x3F	Each domain in a LonWorks network has a unique ID of 0, 1, 3 or 6 bytes in length. If the ID is shorter than 6 bytes, it is left justified in this field.
Subnet ID	1 - 255 0 means invalid	This field specifies the ID of the subnet within this node belongs.
Node ID	1 - 127 0 means not in use	This field specifies the ID of the node within this subnet.
Authentication Key	-	This field specifies the 6-byte authentication key to be used in this domain for authenticated transactions. This key must match the key of all the other nodes on this domain that participate in authenticated transactions with this node.

4.4.2 Get Domain (GET_DOMAIN)

This mailbox command gets the domain to which this node belongs.

Parameter	Description
Command initiator	Application
Message name	GET_DOMAIN
Message type	0x02
Command number	0x0011
Fragmented	No
Extended header data	No Extended Header data is associated with this command.
Message data	Index in the domain table (0 or 1)
Response message	Index in the domain table, ID length, Domain ID, Subnet ID, Node ID, Authentication Key

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	Any integer value
Message information	0x4002	0x0002	
Command	0x0011	0x0011	
Data size	0x0001	0x0010	
Frame number	0x0001	0x0001	One frame
Frame count	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	Domain Index	Fault information	
Message data byte 2		Domain Index	
Message data byte 3		ID Length	
Message data byte 4		Domain ID [0]	
Message data byte 5		Domain ID [1]	
Message data byte 6		Domain ID [2]	
Message data byte 7		Domain ID [3]	
Message data byte 8		Domain ID [4]	
Message data byte 9		Domain ID [5]	
Message data byte 10		Subnet ID	
Message data byte 11		Node ID	
Message data byte 12		Authentication Key [0]	
Message data byte 13		Authentication Key [1]	
Message data byte 14		Authentication Key [2]	
Message data byte 15		Authentication Key [3]	
Message data byte 16		Authentication Key [4]	
		Authentication Key [5]	

Message Data	Value	Explanation
Domain Index	0 or 1	Index to domain table in the module
ID Length	0, 1, 3 or 6 0xFF means not in use	Specifies the length of the domain ID
Domain ID	0x00 - 0x3F	Each domain in a LonWorks network has an unique ID of 0, 1, 3 or 6 bytes in length. If the ID is shorter than 6 bytes, it is left justified in this field.
Subnet ID	1 - 255 0 means invalid	This field specifies the ID of the subnet within this node belongs.
Node ID	1 - 127 0 means not in use	This field specifies the ID of the node within this subnet.
Authentication Key	0xFF means that authentication is off.	This field specifies the 6-byte authentication key to be used in this domain for authenticated transactions. This key must match the key of all the other nodes on this domain that participate in authenticated transactions with this node.

4.4.3 Set Address Table (SET_ADDRESS)

This mailbox command sets an address in the specified address table index. The address table defines the network addresses to which this node may send implicitly addressed messages and network variables. It also defines the groups to which this node belongs.

Parameter	Description
Command initiator	Application
Message name	SET_ADDRESS
Message type	0x02
Command number	0x0012
Fragmented	No
Extended header data	No Extended Header data is associated with this command.
Message data	Index in the address table, Address type, Domain, Node, Repeat_timer, Retry count, Recieve timer, Tx_timer, Group / Subnet.
Response message	The response data is a copy of the command data.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	Any integer value
Message information	0x4002	0x0002	
Command	0x0012	0x0012	
Data size	0x0006	0x0006	
Frame number	0x0001	0x0001	One frame
Frame count	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	-	Fault information	
Message data byte 1	Address Index	Address Index	
Message data byte 2	Address Type / Size	Address Type / Size	
Message data byte 3	Domain / Node	Domain / Node	
Message data byte 4	Repeat Timer / Retries	Repeat Timer / Retries	
Message data byte 5	Receive / Tx Timer	Receive / Tx Timer	
Message data byte 6	Group / Subnet	Group / Subnet	

Message Data Word Information:

Message Data	Value	Explanation
Address Index	0 - 14	Index to a address table entry in the module.
Address Type / Size	0:Not in use / Turnaround 1:Subnet / Node Format 3:Broadcast Format 128 - 255:Group Format	Each entry in the address table can be of one of five formats: Group address, subnet/node address, broadcast address, turnaround address or not in use. If group format is specified, the bits 0 - 6 specifies the size of the group (2 - 64).

Message Data	Value	Explanation
Domain / Node	Bitfield	The contents of this field is depends on which address method that is used.
		Bit 7 Index to domain table. Represents the index domain to the domain table. (0 or 1)
		Bits 0 - 6 ... when Subnet / Node: Specifies node ID (1 - 127). 0 is not a valid node ID. ... when Broadcast: Specifies an estimate of the channel backlog that would be created by an acknowledged or request/response message broadcast using this address. If unknown, this field can be set to 0, in which case a backlog of 15 is used. ... when Group: Specifies the member ID of this node within the group (0-63). A group of unlimited size has 0 in this field. ... when Not in use / Turnaround: 1 when the this is a turnaround address. 0 if this address table entry is unused.
Repeat Timer / Retries	Bitfield	Note! The encoding of the timer fields can be found in Appendix A in "LonWorks Technology Device Data" by Motorola.
		Bits 4 - 7 Repeat Timer: Specifies the time interval between repetitions of an outgoing message when unacknowledged-repeated service is used.
		Bits 0 - 3 Number of retries: Specifies the number of retries for acknowledged, request/response or unacknowledged-repeated service.
Receive / Tx Timer	Bitfield	Note! The encoding of the timer fields can be found in Appendix A in "LonWorks Technology Device Data" by Motorola.
		Bits 4 - 7 Receive Timer: When the node receives a multicast message, the receive timer is set to the time interval specified by this field. If a message with the same transaction ID is received before the receive timer expires, it is considered to be a retry of the previous message.
		Bits 0 - 3 Tx Timer: Specifies the time interval between retries when acknowledged or request/response service is used. The transaction retry timer is restarted when each attempt is made, and also when any acknowledgement or response is received.
Group / Subnet	The contents of this field depends on which address method that is used.	
	Group	Specifies the ID of this group within this domain. May be in the range 0 - 255.
	Other address methods	Specifies the subnet ID within the specified domain. May be in the range 1-255. 0 is not a valid subnet ID.

4.4.4 Get Address Table (GET_ADDRESS)

This mailbox command gets the information in the specified address table index.

Parameter	Description
Command initiator	Application
Message name	GET_ADDRESS
Message type	0x02
Command number	0x0013
Fragmented	No
Extended header data	No Extended Header data is associated with this command.
Message data	Index to the address table to be read. (See "Set Address Table (SET_ADDRESS)" on page 40)
Response message	Index in the address table, Address type, Domain, Node, Repeat_timer, Retry count, Receive timer, Tx_timer, Group / Subnet. (See "Set Address Table (SET_ADDRESS)" on page 40)

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	Any integer value
Message information	0x4002	0x0002	
Command	0x0013	0x0013	
Data size	0x0001	0x0006	
Frame number	0x0001	0x0001	One frame
Frame count	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	-	Fault information	
Message data byte 1	Address Index	Address Index	(see SET_ADDRESS)
Message data byte 2		Address Type	"
Message data byte 3		Domain ID / Node	"
Message data byte 4		Repeat Timer / Retries	"
Message data byte 5		Receive / Tx Timer	"
Message data byte 6		Group / Subnet	"

4.4.5 Set Network Variable Config Table (SET_NV_CONFIG)

This mailbox command sets the network variable configuration for the specified network variable index.

Parameter	Description
Command initiator	Application
Message name	SET_NV_CONFIG
Message type	0x02
Command number	0x014
Fragmented	No
Extended header data	No Extended Header data is associated with this command.
Message data	Network Variable Index, Priority, Direction, Selector High, Selector Low, Turnaround, Service, Authentication, Address Index.
Response message	The response data is a copy of the command data.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	Any integer value
Message information	0x4002	0x0002	
Command	0x0014	0x0014	
Data size	0x000A	0x000A	
Frame number	0x0001	0x0001	One frame
Frame count	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	-	Fault information	
Message data byte 1	Network Variable Index (Hi)	Network Variable Index (Hi)	
Message data byte 2	Network Variable Index (Lo)	Network Variable Index (Lo)	
Message data byte 3	Priority	Priority	
Message data byte 4	Direction	Direction	
Message data byte 5	Selector (Hi)	Selector (Hi)	
Message data byte 6	Selector (Lo)	Selector (Lo)	
Message data byte 7	Turnaround	Turnaround	
Message data byte 8	Service	Service	
Message data byte 9	Authentication	Authentication	
Message data byte 10	Address Index	Address Index	

Message Data	Value	Explanation
Network Variable Index	Number of NV's declared (Max 512)	Index of the network variable to set configuration for.
Priority	0: No Priority 1: Priority	The network variable should use priority messaging. Note! Priority messaging is not implemented in the present version of the AnyBus-S LON.

Message Data	Value	Explanation
Direction	0: NV is an input variable 1: NV is an output variable	Sets the direction of the network variable from the network point of view. Note! This field is currently not used.
Selector	0 - 0x3FFF	This is the network variable selector value that is used when binding two NV's together. Selector values 0x3000 - 0x3FFF are reserved for unbound network variables, with the selector value equal to 0x3FFF minus the network variable index. Values 0 - 0x2FFF are available for bound network variables.
Turnaround	0: Not Turnaround 1: Turnaround	Specifies if a network variable has a turnaround connection (bound to a variable on the same node)
Service	0: Acknowledged 1: Unacknowledged - Repeated 2: Unacknowledged	Specifies the service used to deliver this network variable.
Authentication	0: Not Authenticated 1: Authenticated	Specifies if the variable uses authenticated transactions.
Address Index	0 - 14 0x0F means not set	Specifies the index into the address table for this network variable. Multiple network variables may use the same index.

4.4.6 Get Network Variable Config Table (GET_NV_CONFIG)

This mailbox command gets the network variable configuration for the specified table index.

Parameter	Description
Command initiator	Application
Message name	GET_NV_CONFIG
Message type	0x02
Command number	0x015
Fragmented	No
Extended header data	No Extended Header data is associated with this command.
Message data	Network Variable Index to get config from.
Response message	The response data contains: Network Variable Index, Priority, Direction, Selector High, Selector Low, Turnaround, Service, Authentication, Address Index.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	Any integer value
Message information	0x4002	0x0002	
Command	0x0015	0x0015	
Data size	0x0002	0x000A	
Frame number	0x0001	0x0001	One frame
Frame count	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	-	Fault information	
Message data byte 1	Network Variable Index (Hi)	Network Variable Index (Hi)	
Message data byte 2	Network Variable Index (Lo)	Network Variable Index (Lo)	
Message data byte 3		Priority	
Message data byte 4		Direction	
Message data byte 5		Selector (Hi)	
Message data byte 6		Selector (Lo)	
Message data byte 7		Turnaround	
Message data byte 8		Service	
Message data byte 9		Authentication	
Message data byte 10		Address Index	

Message Data Information

See section 4.4.5 'Set Network Variable Config Table (SET_NV_CONFIG)'

4.5 Advanced Configuration

IMPORTANT! These commands should only be used if you have good knowledge in how network variables and documentation strings are declared in a LonWorks node. Usually the AnyBus module comes with a pre-defined set of network variables.

These commands will destroy all previous network variable information in the module.

4.5.1 Start Network Variable Change (START_NV_CHANGE)

IMPORTANT! This command destroys the network variables on the node. Only use this command if you have a good knowledge in how network variables is declared in a LonWorks node. This mailbox command must be sent before the ADD_NV commands are sent.

Note: This command can only be sent during module initialization.

Parameter	Description
Command initiator	Application
Message name	START_NV_CHANGE
Message type	0x02
Command number	0x0020
Fragmented	No
Extended header data	No Extended Header data is associated with this command.
Message data	-
Response message	-

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	Any integer value
Message information	0x4002	0x0002	
Command	0x0020	0x0020	
Data size	0x0000	0x0000	
Frame number	0x0001	0x0001	One frame
Frame count	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	-	-	"
		Fault information	

4.5.2 Add a network variable (ADD_NV)

IMPORTANT! This command destroys the present network variables on the node. Only use this command if you have a good knowledge in how network variables is declared in a LonWorks node.

This command adds a network variable to the node. The variables will have the same index as the order they are created, starting from 0.

Note: This command can only be sent during module initialization.

Parameter	Description
Command initiator	Application
Message name	ADD_NV
Message type	0x02
Command number	0x0025
Fragmented	No
Extended header data	No Extended Header data is associated with this command.
Message data	Descriptor, SNVT Type, Type Size, Direction, Extension Record, Default data, Documentation string
Response message	The response data is a copy of the command data.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	Any integer value
Message information	0x4002	0x0002	
Command	0x0025	0x0025	
Data size	0x0005 + m + n + o	0x0005 + m + n + o	
Frame number	0x0001	0x0001	One frame
Frame count	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	-	Fault information	
Message data byte 1	Descriptor	Descriptor	
Message data byte 2	SNVT Type	SNVT Type	
Message data byte 3	TypeSize	TypeSize	
Message data byte 4	Direction	Direction	
Message data byte 5	Extension Record	Extension Record	
Msg. data	Default Data [0] ... [m]	Default Data [0] ... [m]	String, null terminated
Msg. data	NV Doc String [0] ... [n]	NV Doc String [0] ... [n]	String, null terminated
Msg. data	NV Name String [0] ... [o]	NV Name String [0] ... [o]	String, null terminated

Message Data	Value		Explanation
Descriptor	Bitfield	Bit 0	Set if configuration class variable (EEPROM)
		Bit 1	Set if authentication may be modified by installation tool.
		Bit 2	Set if priority may be modified by installation tool.
		Bit 3	Set if service type may be modified by installation tool
		Bit 4	Set if installation tool should take the node offline before variable is updated.
		Bit 5	Set if variable is polled
		Bit 6	Set if the variable is synchronous
		Bit 7	Set if the variable has an extension record following the nod's documentation string.
SNVT Type	0 - 250		Specifies the network variable type according to The SNVT Master List and Programmer's Guide. If this field is 0, the network variable is of a non-standard type.
Type Size	1 - 31		The size in bytes of the network variable type
Direction	0: Input Variable		Specifies the direction of the variable
	1: Output Variable		
Extension Record	Bitfield	Bits 0 - 2	Unused
		Bit 3	Set if the extension record contains a 16-bit count of the number of network variables of this type. (Not supported)
		Bit 4	Set if the extension record contains the self-documentation string for the network variable.
		Bit 5	Set if the extension record contains the name of the network variable.
		Bit 6	Set if extension record contains an estimate of the average rate at which this network variable is updated. (Not supported)
		Bit 7	Set if extension record contains an estimate of the maximum rate at which this network variable is updated. (Not supported)
Default Data	Depending on type size		Contains the initial data this variable should have when the module is powered on.
NV Doc String	ASCII String, null terminated		The network variable documentation string, null terminated. (Max 30 characters including string termination)
NV Name String	ASCII String, null terminated		The network variable name string, null terminated. (Max 17 characters including string termination)

4.5.3 End Network Variable Change (END_NV_CHANGE)

Important! This command destroys the present network variables on the node. Only use this command if you have a good knowledge in how network variables is declared in a LonWorks node. This mailbox command is sent after all ADD_NV commands are sent.

Note: This command can only be sent during module initialization.

Parameter	Description
Command initiator	Application
Message name	END_NV_CHANGE
Message type	0x02
Command number	0x0029
Fragmented	No
Extended header data	No Extended Header data is associated with this command.
Message data	Program ID and Node Self documentation String (Null terminated), see below.
Response message	The response data is a copy of the command data.

Command and response layout

	Command	Expected response	
Message ID	(ID)	(ID)	Any integer value
Message information	0x4002	0x0002	
Command	0x0029	0x0029	
Data size	0x0009 + n	0x0009 + n	
Frame number	0x0001	0x0001	One frame
Frame count	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	-	Fault information	
Message data byte 1 - 8	Program ID [0] .. [8]	Program ID [0] .. [8]	
Msg. data	Node Selfdoc String [0] ... [n]	Node Selfdoc String [0] ... [n]	String, null terminated
Msg. data byte 9 + n	Number of Alias	Number of Alias	

Message Data	Value	Explanation
Program ID	6-byte string	Program ID of the node
Node Selfdoc String	ASCII string	Self-documentation string of the node, null terminated.
Number of Alias	0 - 255	The number of alias network variables that should be used. Alias network variables will prevent network variable constraints and are required on a node when a single network variable output on the node must be connected to two or more network variable inputs on another node. For example a single switch connected to a node containing 4 actuators where all four of the actuator inputs must be controlled by the single switch.

5 Connector Pinouts

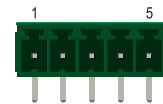
5.1 Application Connector

The application connector features a standard AnyBus-S 2kbyte parallel DPRAM itnerface. (Consult the general AnyBus-S Parallel Design Guide for more information)

5.2 Fieldbus Connector

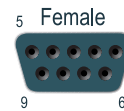
5.2.1 5-pin Pluggable Screw Terminal (3.84mm and 5.08mm)

Pin	Name
5	NET A
4	NET B
3	Service pin
2	GND
1	Shield



5.2.2 9-Pin D-SUB connector (Female)

DSUB pin	Name
1	NET A
2	NET B
3	Service pin
4	GND
5	-
6	-
7	-
8	-
9	-
Housing	Shield

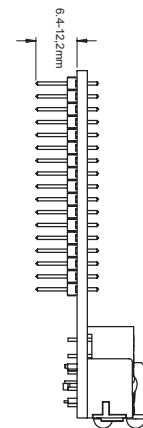
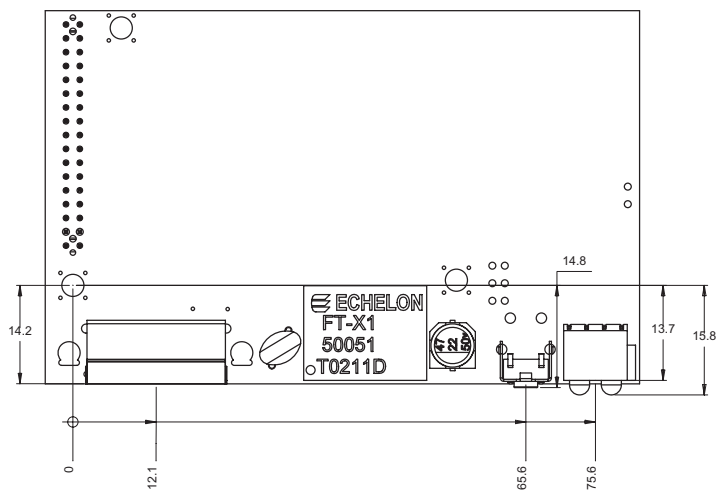
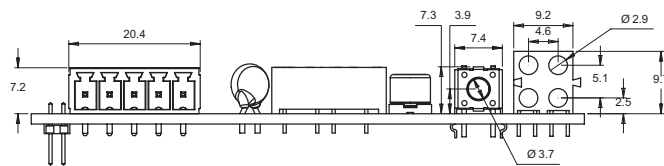
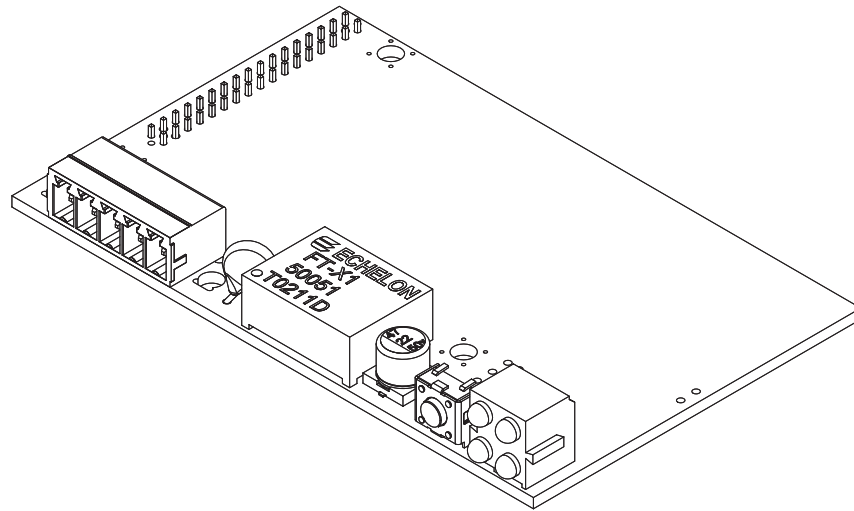


5.2.3 10-pin Board to Board Connector (2mm)

Pin	Name
1	Shield
2	-
3	GND
4	-
5	-
6	Service pin
7	-
8	-
9	NET B
10	NET A



6.2 Standard Configuration



7 Electrical Characteristics

7.1 Supply Voltage

Both the module electronics and the fieldbus interface requires a regulated 5V DC power supply. For more information regarding power requirements, consult the AnyBus-S Design Guide

7.2 Power Consumption

The total maximum power consumption from the application side is 160mA.

7.3 Protective Earth

A PE-connection is available on one of the mounting holes according to the AnyBus-S specification.

8 Environmental Specification

8.1 Temperature

Operating

+0 to +70 degrees Celsius

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

Non Operating

-15 to +85 degrees Celsius

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

8.2 Relative Humidity

The product is designed for a relative humidity of 5 to 95% non-condensing.

Test performed according to IEC 68-2-30.

8.3 EMC Compliance

Emission

According to EN 50 081-2:1993

Tested per 55022:1997, class A, radiated

Immunity

According to EN 61000-6-2:1999

Tested per EN 61000-4-2:1995

EN 61000-4-3:1996

EN 61000-4-4:1995

EN 61000-4-5:1995

EN 61000-4-6:1996

