Anybus® Communicator™ CAN

PROFINET® IRT (2.32)

USER MANUAL

SCM-1202-035 1.3 en-US ENGLISH

Connecting Devices™

Important User Information

Disclaimer

The information in this document is for informational purposes only. Please inform HMS Industrial Networks of any inaccuracies or omissions found in this document. HMS Industrial Networks disclaims any responsibility or liability for any errors that may appear in this document.

HMS Industrial Networks reserves the right to modify its products in line with its policy of continuous product development. The information in this document shall therefore not be construed as a commitment on the part of HMS Industrial Networks and is subject to change without notice. HMS Industrial Networks makes no commitment to update or keep current the information in this document.

The data, examples and illustrations found in this document are included for illustrative purposes and are only intended to help improve understanding of the functionality and handling of the product. In view of the wide range of possible applications of the product, and because of the many variables and requirements associated with any particular implementation, HMS Industrial Networks cannot assume responsibility or liability for actual use based on the data, examples or illustrations included in this document nor for any damages incurred during installation of the product. Those responsible for the use of the product must acquire sufficient knowledge in order to ensure that the product is used correctly in their specific application and that the application meets all performance and safety requirements including any applicable laws, regulations, codes and standards. Further, HMS Industrial Networks will under no circumstances assume liability or responsibility for any problems that may arise as a result from the use of undocumented features or functional side effects found outside the documented scope of the product. The effects caused by any direct or indirect use of such aspects of the product are undefined and may include e.g. compatibility issues and stability issues.

Table of Contents

Page

1 Preface

1.1 About This Document

This document describes how to install and configure the Anybus Communicator CAN PROFINET IRT (2.32) gateway.

For additional related documentation and file downloads, please visit www.anybus.com/support.

1.2 Document history

Version	Date	Description
1.0	2017-02-23	First release
1.1	2017-11-22	Updated for new firmware
1.2	2019-04-11	Added section about PROFINET Asset Management
1.3	2021-02-12	Added sections about ‘Groups and Transactions’, ‘Configuration of CAN Frames’, ‘Online Menu’ and ‘Anybus Configuration Manager Tools’ DHCP is not supported by this module. This is now reflected in the documentation.

1.3 Document Conventions

Numbered lists indicate tasks that should be carried out in sequence:

1. First do this
2. Then do this

Bulleted lists are used for:

- Tasks that can be carried out in any order
- Itemized information

► An action

→ and a result

User interaction elements (buttons etc.) are indicated with bold text.

```
Program code and script examples
```

Cross-reference within this document: *Document Conventions, p. 4*

External link (URL): www.hms-networks.com

WARNING
Instruction that must be followed to avoid a risk of death or serious injury.

Caution
Instruction that must be followed to avoid a risk of personal injury.

Instruction that must be followed to avoid a risk of reduced functionality and/or damage to the equipment, or to avoid a network security risk.

Additional information which may facilitate installation and/or operation.

2 Description

2.1 Introduction

Anybus Communicator CAN PROFINET IRT (2.32) is designed to exchange data between a CAN subnetwork and a higher level network. The CAN protocol uses individually configurable frames, offering great flexibility. Through the configuration of CAN frames, the Anybus Communicator CAN will adapt to a predefined CAN subnetwork. It will be possible to send data to and receive data from the CAN subnetwork, but also to act as a relay for data on the subnetwork.

The gateway can issue frames cyclically, on change of data, or based on trigger events issued by the control system of the higher level network. It can also monitor communication on the subnetwork and notify the higher level network when data has changed.

Anybus Communicator gateways are configured using *Anybus Configuration Manager*, a family of configuration tools that have an easy to use graphical interface and that do not require programming skills.

Anybus Configuration Manager and additional related software and documentation are available at www.anybus.com/support.

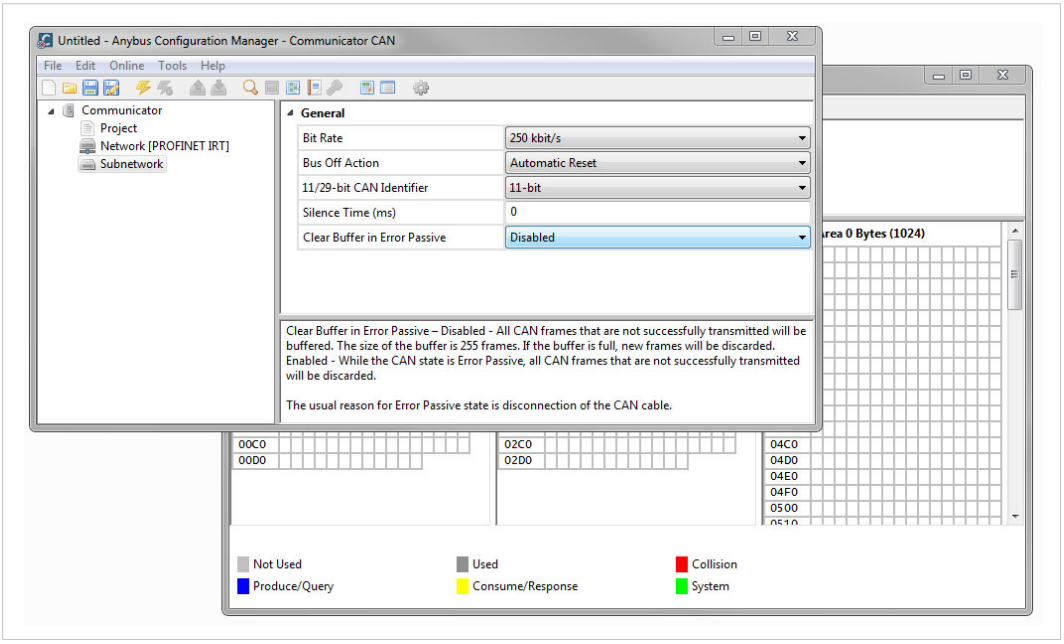


Fig. 1 Anybus Configuration Manager

2.2 Data Exchange Model

2.2.1 Overview

The data exchanged on the CAN subnetwork and the data exchanged on the higher level network reside in the same internal memory in the Anybus Communicator CAN. In order to exchange data with the CAN subnetwork, the higher level network simply reads and writes data to memory locations that have been specified in Anybus Configuration Manager. The same memory locations can then be exchanged on the CAN subnetwork.

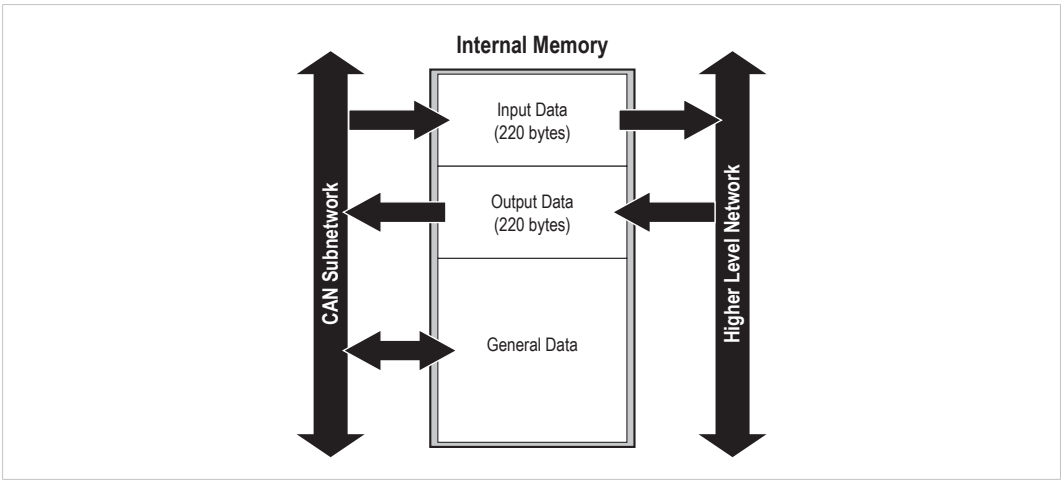


Fig. 2 Memory buffer structure

The internal memory buffer is divided into three areas based on their function:

Input Data (220 bytes)	This area can be read from by the higher level network.
Output Data (220 bytes)	This area can be written to by the higher level network.
General Data (up to 1024 bytes)	This area cannot be accessed by the higher level network but can be used for transfers between individual nodes on the subnetwork, or as a general “scratch pad” for data. The size of the General Data area is 1024 bytes. How much of that area that will be used for communication depends on the configuration.

2.2.2 Memory Map

When building the CAN subnetwork configuration in Anybus Configuration Manager the areas in the memory buffer will be mapped to the following memory locations:

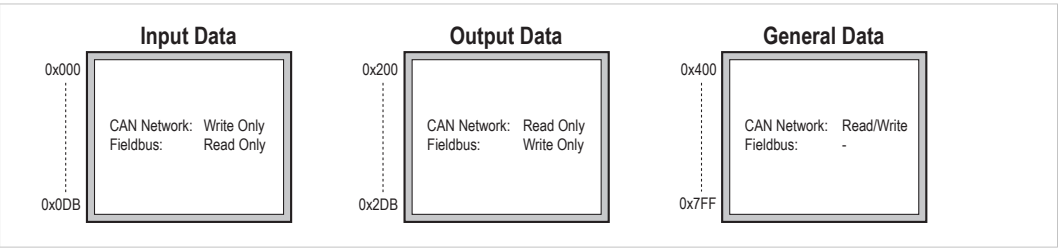


Fig. 3 Memory Map

The illustration shows the maximum available data areas in Anybus Communicator CAN. The actual amount of memory that can be allocated depends on the fieldbus network used.

2.3 PROFINET IRT Protocol

PROFINET is the open Industrial Ethernet standard for automation from PROFIBUS and PROFINET International. The PROFINET IRT device provides PROFINET IO Isochronous Real Time Communication.

PROFINET makes a clear distinction between fast cyclical data, *IO Data*, and acyclical data, *Record Data*. PROFINET IO Data corresponds to what is generally referred to as *I/O Data* in Anybus Communicator CAN PROFINET IRT (2.32). PROFINET Record Data is not supported.

Anybus Communicator CAN PROFINET IRT (2.32) acts as a PROFINET device (slave), which means it can be accessed by a PROFINET controller (master), but will not initiate communication by itself.

PROFINET IO Data (I/O Data)

PROFINET IO Data is exchanged cyclically and is built up by I/O modules. The actual I/O configuration is determined by the PROFINET IO Controller. The modules are mapped to the Input and Output Buffers in the order of their slot number.

The first two bytes of the I/O data area are reserved for the Control Word and the Status Word, which are used by the IO Controller to control and report status on the nodes on the CAN subnetwork. The remainder is available for real-time data transfer using PDOs.

GSD File

All PROFINET devices are associated with an XML-based *GSD* file. This file contains information about the basic capabilities and configuration options of the device.

The latest version of the GSD file for Anybus Communicator CAN PROFINET IRT (2.32) can be downloaded from www.anybus.com/support.

2.4 CAN Network Protocol

2.4.1 General

The CAN protocol is message-based and can exchange up to 8 bytes of data in each message. The protocol only acts as a data carrier, it is up to each application to define and interpret the data content of the messages.

Data is exchanged using *frames*. Each frame has a unique identifier for the data it exchanges, which also represents the message priority. Anybus Communicator CAN supports both 11-bit (CAN 2.0A) and 29-bit (CAN 2.0B) identifiers, selected in the configuration.

CAN is essentially a produce-consume protocol, where all nodes listen to all messages. The devices recognize what data to collect by the identifier in the CAN frame. Anybus Communicator CAN is also able to act as a network master and issue *queries* that demand *responses*. Both methods can be used within the same configuration.

2.4.2 Message Types

Anybus Communicator CAN features three CAN message types: *Query-Response*, *Produce*, and *Consume*. These message types only specify the basic communication model, not the actual CAN protocol. All three message types can be used in the same configuration.

Query-Response

The Anybus Communicator CAN here acts as a master on the CAN subnetwork, and communication takes place in a query-response fashion. The gateway sends a query and expects a response within a specified timeout.

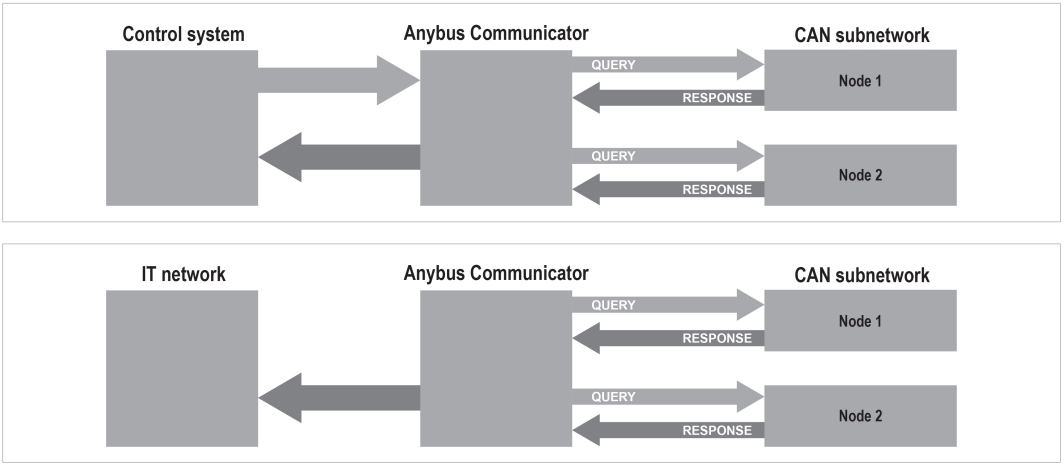


Fig. 4 Query-Response messaging

Produce and Consume

Here there is no master-slave relationship between the Anybus Communicator CAN and the subnetwork nodes. Any node, including the gateway, may both *produce* and *consume* messages. Nodes do not have to respond to messages, or wait for a query in order to send one. The consumed data can be accessed from the higher level network, and vice versa.

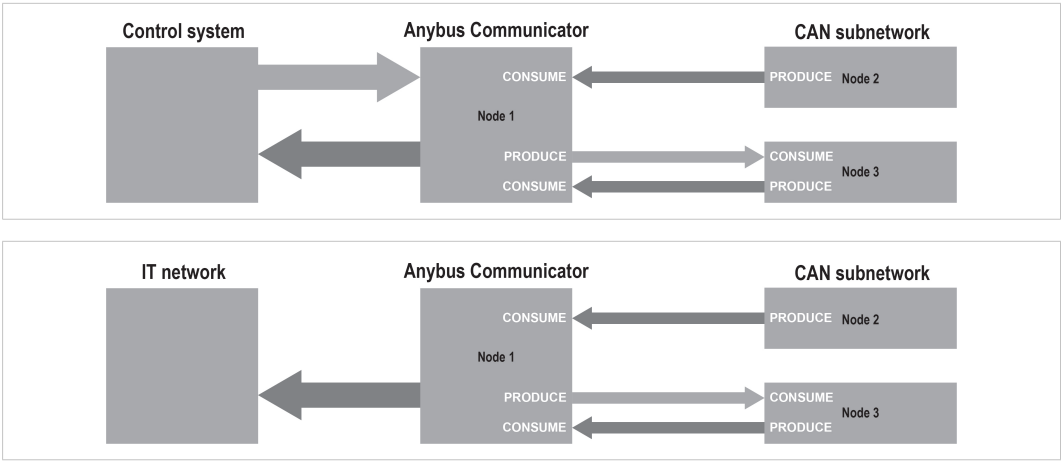


Fig. 5 Produce and Consume messages

2.4.3 Protocol Building Blocks

The following building blocks are used in Anybus Configuration Manager to describe the subnetwork communication.

Group

A *group* does not represent any specific device on the CAN subnetwork, it is only a means to structure the *transactions* that have been defined for the gateway in Anybus Configuration Manager.

Each group can be associated with any number of transactions, however the total number of transactions in a configuration is limited to 128.

Transaction

A *transaction* consists of one or more *CAN frames*. Each transaction is associated with a set of parameters controlling how and when to use it on the subnetwork. There are 5 transaction types: *produce*, *consume*, *query-response*, *dynamic produce* and *dynamic consume*. A group can contain transactions of all three types simultaneously.

A total of 128 transactions can be configured.

Dynamic Transaction

A *dynamic transaction* makes it possible for a network master to change selected parameters during runtime. The parameters are mapped to the Output or General Data areas and cannot be changed using Anybus Configuration Manager.

A dynamic transaction can only consist of a single CAN frame which in turn can only hold one data object.

Only one *dynamic produce* transaction and one *dynamic consume* transaction can be configured.

CAN Frames

CAN frames are low level entities used to compose *transactions*. Each frame carries an 11-bit or 29-bit identifier and can hold up to 8 bytes of data.

A total of 256 CAN frames can be configured.

2.4.4 Control and Status Words

The optional Control and Status Words can be used to control the startup mode of the Anybus Communicator CAN and to read the status of the CAN subnetwork. The Control Word is always mapped to the first two bytes of the output data area, and the Status Word to the first two bytes of the input data area, with the Least Significant Byte in the first byte (byte 0).

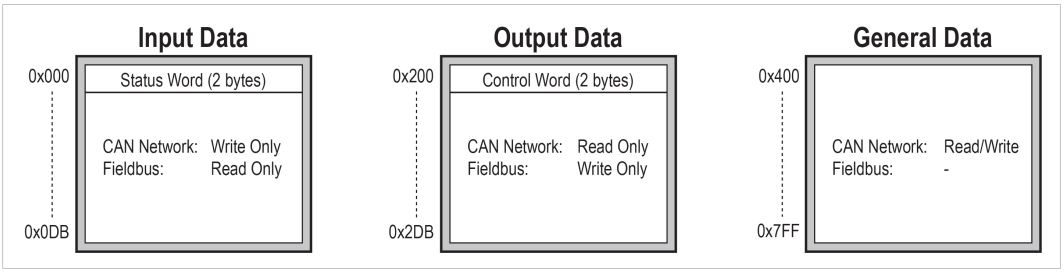


Fig. 6 Memory buffers

The illustration shows the maximum available data areas in Anybus Communicator CAN. The actual amount of memory that can be allocated depends on the fieldbus network used.

Control Word

The Control Word can be used to reset the CAN controller and to select the startup mode and/or reboot the Anybus Communicator CAN.

Bit	Name	Description
15 - 3	(reserved)	
2	Reset CAN	1 - Resets the CAN controller (used when CAN interface is bus off)
1	Reboot module	1 - Reboots the Anybus Communicator CAN (software reset)
0	Operation mode	Sets the start-up operation mode of the Anybus Communicator CAN: 0 - Idle (no new data is issued to the CAN subnetwork. Data received from the CAN subnetwork is sent on to the higher level network.) 1 - Run (data is exchanged between CAN subnetwork and higher level network.)

Status Word

The Status Word holds information from the CAN subnetwork.

Bit	Name	Description
15 - 6	(reserved)	
5	CAN overrun	0 - OK 1 - CAN reception overrun
4	Error passive	0 - CAN interface is NOT in error passive state 1 - CAN interface is in error passive state
3	Bus off	0 - Bus runnning 1 - Bus off
2	Reset CAN complete	If set, the CAN controller has been reset (used when CAN interface is bus off)
1	(reserved)	
0	Operation mode	0 - Idle 1 - Run

2.4.5 Transaction Live List

The optional *Transaction Live List* consists of a bit array where each bit corresponds to a transaction on the CAN subnetwork (bit 0 corresponds to transaction 1 etc.). A set bit indicates normal functionality. The bit is not set if the transaction is non-working or non-existent.

The live list is mapped in the Input data area of the memory, either at the start of the area or directly after the Status word. From 8 transactions up to 128 transactions in steps of 8 can be monitored using the live list. This means that up to 16 bytes of the input data area of the memory can be occupied by the live list.

The latest live list can always be accessed in Anybus Configuration Manager, regardless of whether the live list is mapped in the input data area or not.

3 Installation

> This product contains parts that can be damaged by electrostatic discharge (ESD). Use ESD prevention measures to avoid damage.

3.1 Installation Overview

These are the basic steps for installing Anybus Communicator CAN gateways.

Depending on the fieldbus network type there may also be configuration switches on the Anybus Communicator CAN that need setting. See the following sections for more information.

Basic installation steps

1. Mount the Anybus Communicator CAN on the DIN rail.
2. Connect the CAN network.
3. Connect the fieldbus network.
4. Configure the fieldbus network interface (if applicable).
5. Connect the power cable and apply power.
6. Connect the USB cable between the gateway and a PC.
7. Download Anybus Configuration Manager from www.anybus.com/support and install it on the PC following the instructions in the installer.

   (Anybus Configuration Manager requires Microsoft® Windows XP or later)
8. Continue to *Configuration, p. 18*

3.2 Connectors and Indicators

3.2.1 External Parts

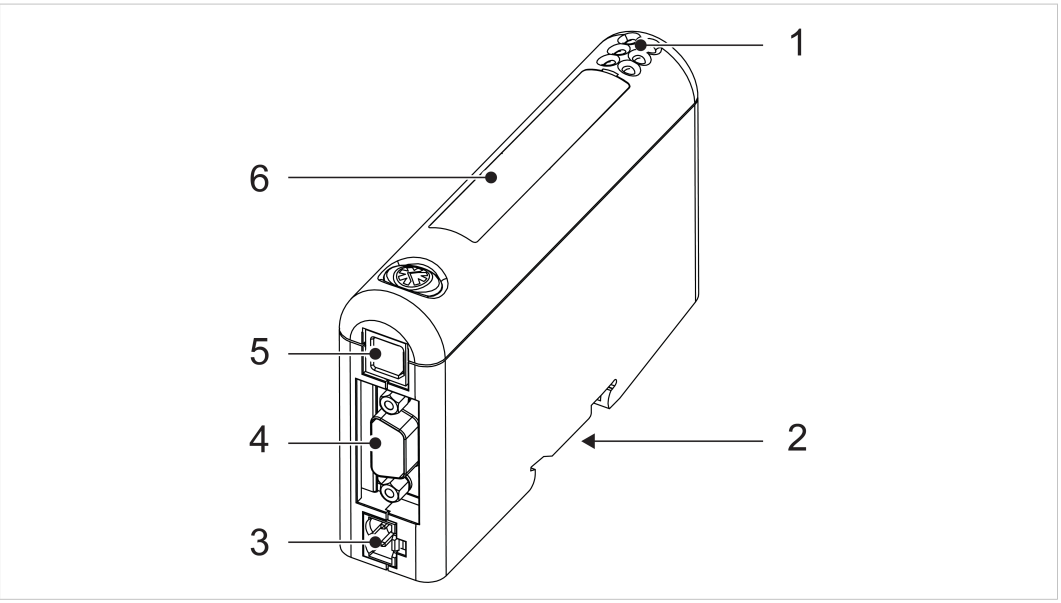


Fig. 7 Overview

1	LED indicators
2	DIN rail mount
3	Power connector
4	CAN connector
5	USB connector
6	PROFINET IRT network interface

3.3 DIN Rail Mounting

> **!** The unit must be electrically grounded through the DIN rail for EMC compliance.

Mount on DIN rail

1. Hook the unit onto the upper lip of the rail and push gently downwards.
2. Push the unit towards the rail until it snaps into place.

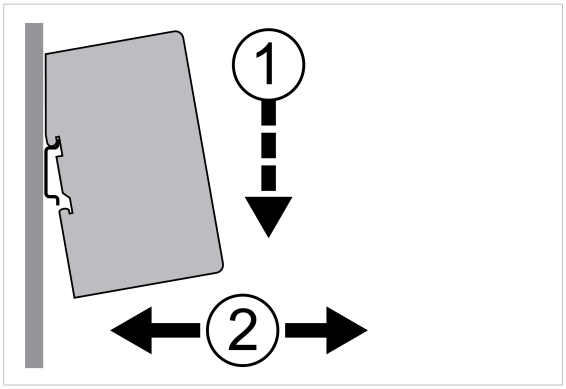


Fig. 8 Push down to mount or remove

Remove from DIN rail

1. Push the unit gently downwards on the rail.

2. Pull the bottom end of the unit free of the rail and remove it.

3.4 CAN Interface

The CAN network connector is located on the bottom of the unit.

Pin	Signal
2	CAN_L
3	CAN_GND
5	Shield
6	CAN_GND
7	CAN_H
1, 4, 8, 9	(reserved)

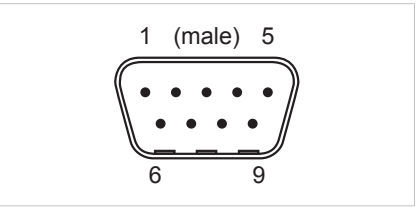


Fig. 9 CAN connector

3.5 PROFINET Interface

The PROFINET IRT interface contains a dual port Ethernet switch with RJ45 type connectors. The two ports are labeled **LAN 1** and **LAN 2**.

Pin	Function
1	TD+
2	TD-
3	RD+
6	RD-
4, 5, 7, 8	(reserved)

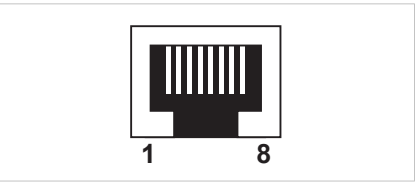


Fig. 10 Ethernet connector (RJ45)

3.6 Power Connector

See also *Technical Data, p. 55* regarding power supply requirements.

Pin	Signal
1	+24 VDC
2	Power Ground

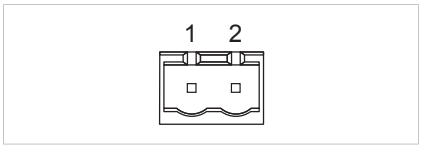


Fig. 11 Power connector

3.7 USB Connector

The USB connector is used for connecting the Anybus Communicator CAN to a computer for uploading and downloading configurations. The USB cable should be removed when not in use.

Pin	Signal
1	+5 V input
2	USBDM (USB communication)
3	USBDP (USB communication)
4	Signal ground
Housing	Cable shield

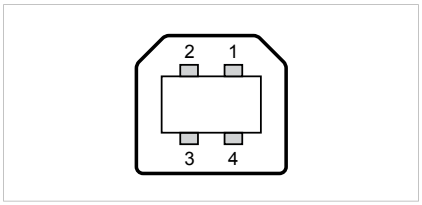
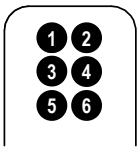


Fig. 12 USB type B connector

3.8 LED Indicators

The LED indicators provide diagnostic information about data communication and status of the network interfaces as well as general device status.



LED 1 to 4 PROFINET IRT network status

LED 5 CAN network status

LED 6 Device status

LED	Indication	Meaning
1 - Network Status	Off	Offline – No power – No connection to IO Controller
	Green	Online (RUN) – Connection to IO Controller
	Green, 1 flash	Online (STOP) – Connection to IO Controller – IO Controller in STOP state or IO data bad – RT synchronization not finished
	Red	Fatal error
	Red, 1 flash	Station name error
	Red, 2 flashes	IP address error
	Red, 3 flashes	Configuration error
	Alternating red/green	Firmware update in progress
2 - Module Status	Off	No power or initializing
	Green	Normal operation
	Green, 1 flash	Diagnostic event present
	Red	Fatal error
	Alternating red/green	Firmware update in progress
3 - Link/Activity 1 4 - Link/Activity 2	Off	No power or no link detected
	Green	Link OK
	Green, flickering	Transmitting/receiving data

LED	Indication	Meaning
5 - CAN Subnet Status	Off	No power or no CAN communication
	Green	Running, no errors or timeout
	Red. flashing	Transaction error, timeout, or CAN subnet stopped
	Red	Fatal error
Device Status	Off	No power or initializing
	Green	Running
	Green, flashing	Idle
	Red	Fatal error
	Alternating red/green	Configuration error

4 Configuration

4.1 Configuration Overview

Device Description Files

A device description file contains a description of a network device, its functions, object dictionary implementations, etc., and is used when configuring the network interface. The device description file can be referred to as a DDF, EDS, GSD, etc., depending on the type of network.

The latest versions of the device description files can be downloaded from www.anybus.com/support.

Basic steps when configuring Anybus Communicator CAN PROFINET IRT (2.32)

1. Install, connect and power up the Anybus Communicator CAN gateway (if you have not already done so). See also *Installation, p. 13*.
2. Download Anybus Configuration Manager from www.anybus.com/support and install it.
3. Download the latest device description file for Anybus Communicator CAN PROFINET IRT (2.32) from www.anybus.com/support.
4. Build your configuration using Anybus Configuration Manager and download it to the gateway.See *Anybus Configuration Manager, p. 24*.
5. Install the appropriate device description file in the PROFINET IRT configuration tool.
6. Configure the PROFINET IRT network as required. Make sure that the configuration matches the configuration present in the Anybus Communicator CAN.

4.2 Network Configuration

To be able to communicate over Ethernet, the PROFINET network interface needs a valid TCP/IP configuration. This section explains some basic concepts and describes how to configure the TCP/IP settings using the *IPconfig* software tool.

When Ethernet communication has been established the TCP/IP settings can also be changed from the web interface. See *Web Pages, p. 23*.

4.2.1 Basic TCP/IP Concepts

IP Address

The IP address is used to identify each node on a TCP/IP network. IP addresses are written as four decimal integers (0–255) separated by dots, where each integer represents the binary value of one byte of the IP address. This is known as *dot-decimal notation*.

Example: `10000000 00001010 00000010 00011110` is written as `128.10.2.30`

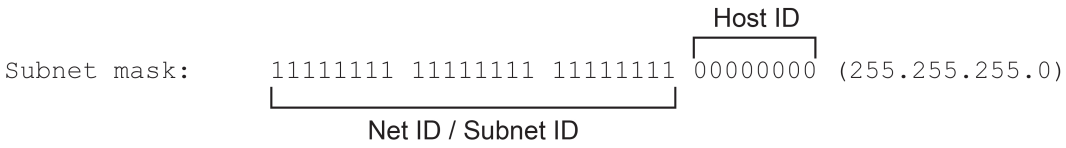
The following IP addresses are reserved for special purposes and cannot be used:

0.n.n.n	First byte zero — used for broadcast messages
127.n.n.n	First byte 127 — used for loopback addresses to the local host
n.n.n.0	Last byte zero — identifies a whole network/subnet
n.n.n.255	Last byte 255 — used for broadcast messages

Subnet Mask

The IP address is divided into three parts: *Net ID*, *Subnet ID* and *Host ID*. A subnet mask is a 32-bit binary pattern, where a set bit allocates a bit for Network/Subnet ID, and a cleared bit allocates a bit for the Host ID. The subnet mask is usually written in dot-decimal notation.

Example: To make the IP address `128.10.2.30` belong to subnet `128.10.2`, the subnet mask must be `255.255.255.0`.



Default Gateway

For devices to be able to communicate over Ethernet they must either belong to the same subnet or communicate via a gateway or router.

A gateway or router routes communication between networks, i.e. it enables the nodes on one network to access the nodes on another. The *default gateway* address in the TCP/IP settings of your product specifies the IP address of the gateway or router on the local network.

4.2.2 TCP/IP Configuration

Installing the IPconfig Utility

IPconfig is a Windows-based tool for configuration of TCP/IP settings in HMS devices. The tool will detect all compatible and active HMS devices on the local network.

1. Download IPconfig from www.anybus.com/support.
2. Unpack the contents of the zip archive and run the installer program.

Scanning for Connected Devices

When IPconfig is started it will automatically scan all available local networks for HMS devices. Detected devices will be listed in the main window. To refresh the list, click on **Scan**.

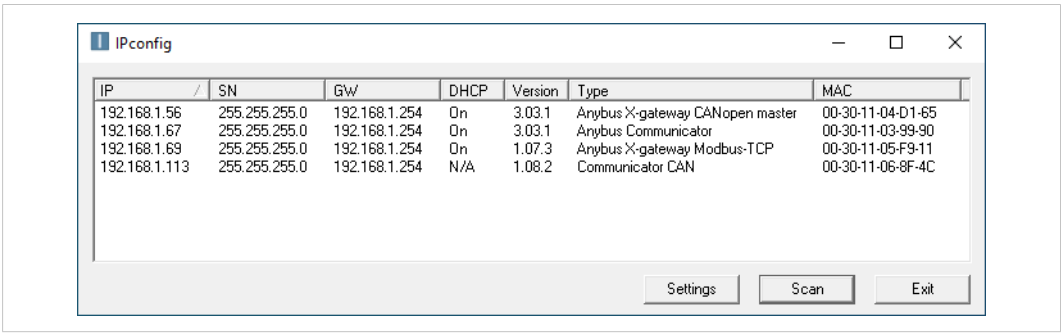


Fig. 13 IPconfig main window

IP	IP address of the device
SN	Subnet mask
GW	Default gateway
DHCP	Automatically managed IP configuration
Version	Firmware version
Type	Product name
MAC	Ethernet MAC address (System ID)

Ethernet Configuration

To change the IP settings for a device, double-click on the entry in the main window or right-click on it and select **Configuration**.

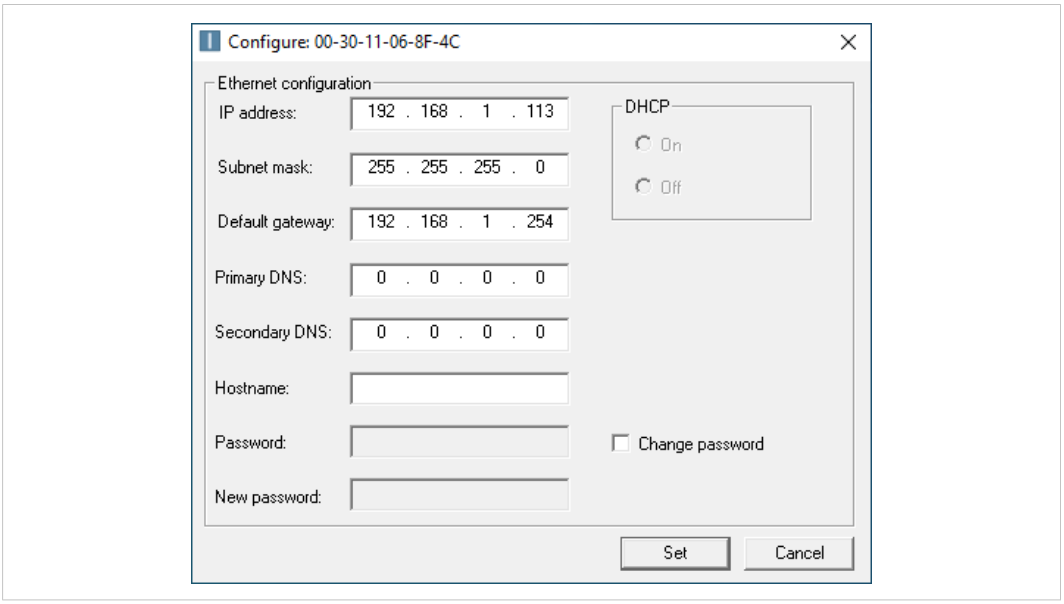


Fig. 14 Ethernet configuration

Enter static IP settings as required. DHCP is not supported.

You can add a name for the device in the **Hostname** field. Only characters a–z, A–Z, 0–9 and _ (underscore) are allowed.

The default password for changing IP settings is blank (no password). If a password has been set for the device you must enter it to be able to change the settings.

To set a new password, check the **Change password** box and enter the current password in the **Password** field, then enter the new password in the **New password** field.

> **!** For security reasons the default password should always be changed.

Click on **Set** to save the new settings. The device will reboot automatically.

IPconfig Settings

Additional settings for IPconfig can be accessed by clicking on **Settings**.

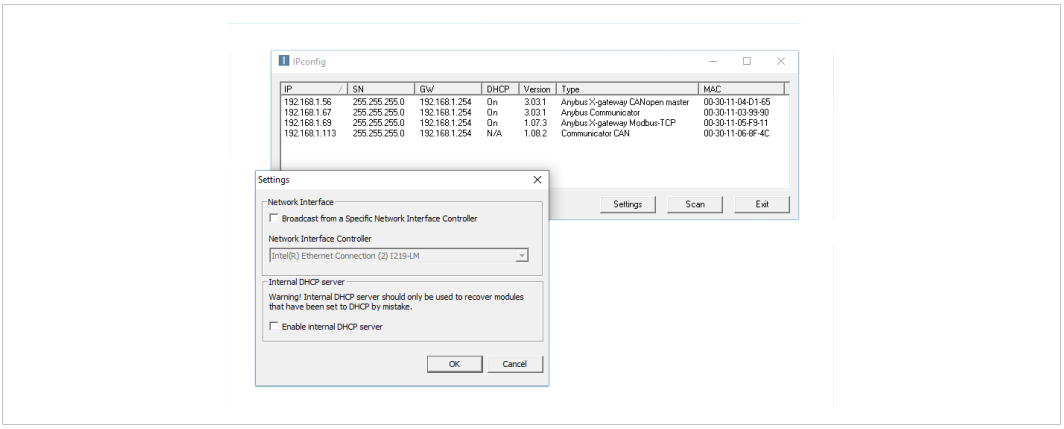


Fig. 15 IPconfig settings

Network Interface

Check this option to select a specific network interface to use when scanning for devices from a computer which has more than one interface. If this option is left unchecked, all available networks will be scanned.

For this product, enabling the internal DHCP server option will have no effect.

4.2.3 DCP (Discovery and Control Protocol)

Anybus Communicator CAN PROFINET IRT (2.32) PROFINET IRT supports the DCP protocol, which allows a PROFINET IO Controller/Supervisor to change the network settings during runtime.

4.3 Web Pages

Network configuration settings and status of the PROFINET IRT network interface can be accessed by pointing a web browser to the IP address of the interface.

Module Overview

Fig. 16 Overview tab

Provides basic information about the Anybus Communicator CAN including the serial number and the installed firmware version.

Network Status

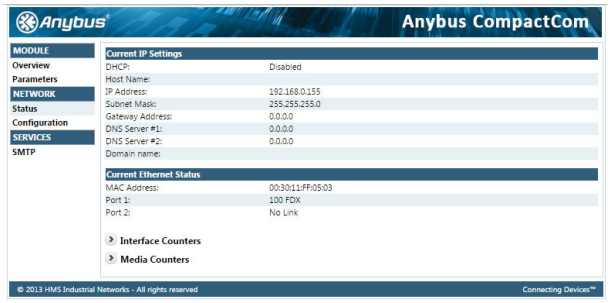


Fig. 17 Status tab

Displays an overview of the current network status.

Network Configuration

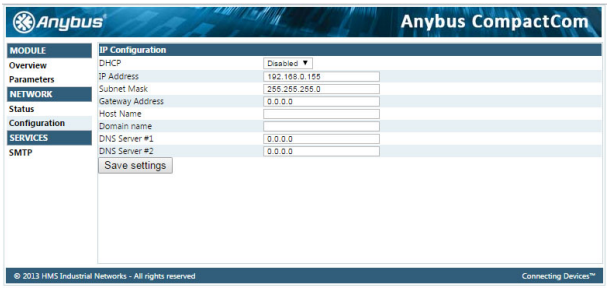


Fig. 18 Configuration tab

Provides access to the TCP/IP network settings. These parameters can also be configured using the *IPconfig* tool.

5 Anybus Configuration Manager

5.1 Main Window

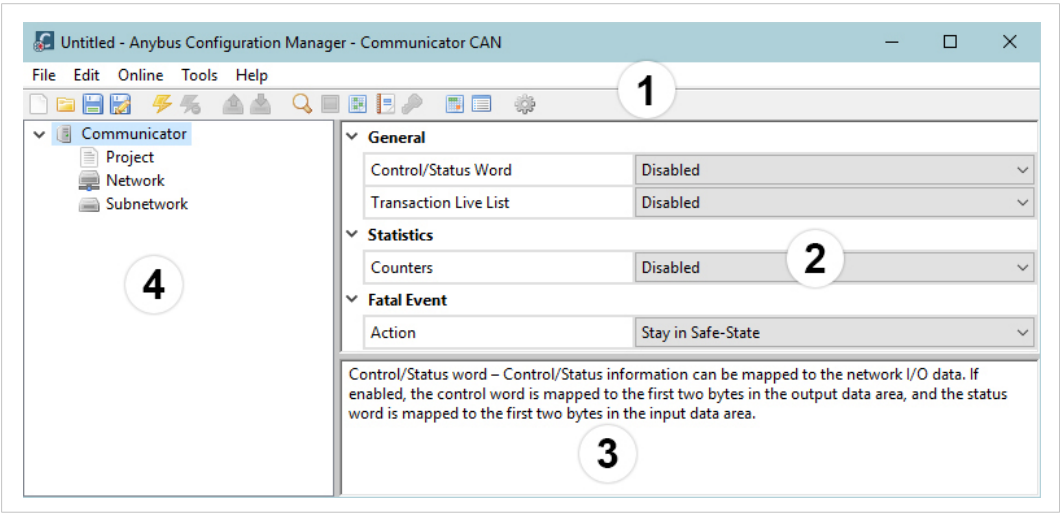


Fig. 19 Anybus Configuration Manager - Communicator CAN

1: Menus and Toolbar

The most common menu commands can also be carried out by clicking on a button in the toolbar. Moving the mouse cursor over a toolbar button will display a tooltip explaining its function.

2: Parameter List

Lists the parameters or options related to the currently selected entry in the Navigation Tree.

Values can be selected from a dropdown menu or entered manually depending on the parameter. Values can be specified in either decimal or hexadecimal format.

Example: The decimal value `42` can also be entered as `0x2A`.

Moving the mouse cursor over a parameter in this window will show a help text in the Information Window explaining how to use the parameter.

3: Information Window

Displays a help text describing the current parameter.

4: Navigation Tree

A hierarchic tree view of the configuration, divided into three main sections:

Project	Information about the current configuration project
Network	Settings for the higher level network
Subnetwork	Settings for the CAN subnetwork

Select an entry to display its available parameters in the Parameter List. Right-click on the entry to show additional options.

5.2 Basic Settings

For more detailed explanations of each configuration setting, see the help texts in the Information Window.

5.2.1 Project

Used to store project information such as project name, project creator, version and description.

5.2.2 Network Settings

General

During startup the fieldbus interface is initialized to fit the configuration created in Anybus Configuration Manager. Some initialization parameters can optionally be set manually to provide better control over how the data shall be treated by the Anybus Communicator CAN.

Network Type

The higher level network type must be selected here in order to create a valid configuration.

5.2.3 Communicator Settings

Additional settings for the Anybus Communicator CAN.

General	
Parameter	**Comment**
Control/Status Word	If enabled, the Control/Status Word will occupy the first two bytes of the Output/Input areas of the memory. To avoid memory address collisions this parameter should be enabled **before** adding frames to the configuration.
Start-up Operation Mode	Decides the start-up mode of the CAN subnetwork if the Control Word is enabled.
Transaction Live List	If the Transaction Live List is enabled it is mapped from the beginning of the input area or, if the Control/Status Word is enabled, after the Status Word. It is possible to map from 8 to 128 transactions, in steps of 8. Each transaction is represented by a bit that tells the system whether the transaction is alive or not.

Statistics	
Parameter	**Comment**
Counters	The receive counter and the transmit counter count successful CAN messages on the subnetwork. If enabled, the counters can be mapped to the input data area. The first free address in the input data area is selected by default. The counters can be disabled and enabled separately. To avoid memory address collisions this parameter should be enabled **before** adding frames to the configuration. The messages are counted only if they have been configured in Anybus Configuration Manager.
Receive Counter Address	Enter the address in the input data area where the receive counter shall be mapped. The receive counter occupies 2 bytes.
Transmit Counter Address	Enter the address in the input data area where the transmit counter shall be mapped. The transmit counter occupies 2 bytes.

Fatal Event		
Parameter	**Values**	**Comment**
Action	Stay in Safe-State	In case of a fatal software event, the Anybus Communicator CAN will be locked in the safe state.
	Software Reset	In case of a fatal software event, the software will be reset and the Anybus Communicator CAN will be restarted.

5.2.4 Subnetwork Settings

Settings for the CAN subnetwork.

Parameter	Values	Comment
Bit Rate	20 kbit/s 50 kbit/s 100 kbit/s 125 kbit/s 200 kbit/s 250 kbit/s 500 kbit/s 800 kbit/s 1000 kbit/s	Select the bit rate on the CAN subnetwork.
Bus Off Action	No Action Automatic Reset	Select what will happen to the CAN controller when the CAN subnetwork goes bus off. When the Control/Status Word is enabled this parameter will be disabled (No Action).
11/29-bit CAN Identifier	11 bit 29 bit	Select CAN identifier size on the subnetwork If there are configured transactions when this parameter is changed, the following will happen: • A change from 11 bit to 29 bit identifier will cause the identifier to be padded with zeroes up to 29 bits, keeping the 11 bits at the same location. • A change from 29 bit to 11 bit identifier will cause the upper 18 bits to be deleted and the lower 11 bits kept. **This may in some cases cause faulty CAN identifiers.**
Silence Time (ms)	0 - 65535	Default = 0 (disabled) The minimum time that must elapse between the end of a message and the beginning of the next message. If a device on the CAN subnetwork is slow and/or does not have a message queue, it may be necessary to enter a delay between the messages to ensure that they are handled correctly. The delay is set in milliseconds.
Clear Buffer in Error Passive	Disabled/Enabled	**Disabled:** All CAN frames that are not successfully transmitted will be buffered. The size of the buffer is 255 frames. If the buffer is full, new frames will be discarded. **Enabled:** While the CAN state is Error Passive, all CAN frames that are not successfully transmitted will be discarded. The usual reason for Error Passive state is disconnection of the CAN cable.

5.3 Groups and Transactions

5.3.1 General

The configuration of the Communicator is set up in groups, each containing one or more transactions. Please note that the groups do not represent a physical device on the CAN network. They are a means for structuring the application, and maintaining an overview of it. The maximal number of groups is 128.

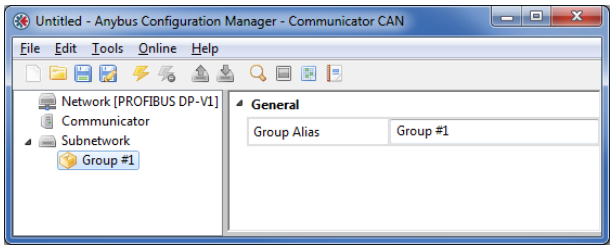
A transaction can be either a Produce, a Consume or a Query/Response transaction. Each transaction holds one or more CAN frames, which transport the data on the network. A total of 128 transactions is allowed, and a total of 256 CAN frames.

Each CAN frame can hold up to 8 bytes of data.

Groups and transactions as well as frames and objects (described in the next section) can be copied and pasted in the configuration tree, but only at the same level as they were copied from, or their parent.

5.3.2 Groups

To create a group, right click on ‘Subnetwork’ and select ‘Add Group’. The name of the group can be changed by selecting ‘Group’ and then entering a new name at ‘Group Alias‘.

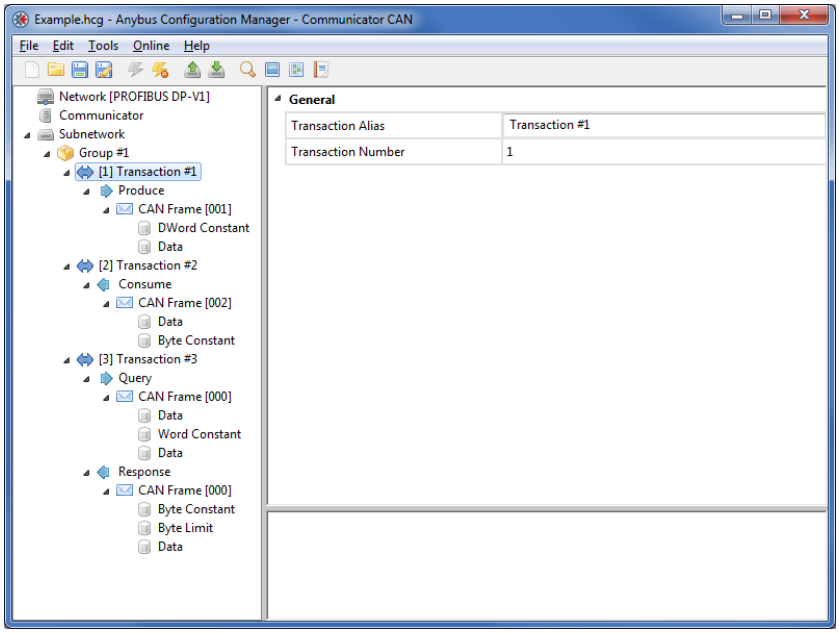


If you want to insert another group, right click on ‘Subnetwork’ once more. The new group will be added to the end of the list of groups. If you right click on a group and select ‘Insert Group’, the new group will be inserted before the selected group.

It is recommended to change the group name, to better present the configuration.

5.3.3 Transactions

There are five kinds of transactions: Produce, Consume, Query/Response, Dynamic Produce and Dynamic Consume.



Selecting the transaction will give the option to give the transaction an alias. The order of the transactions in the tree is given as the transaction number in the parameter section. Each transaction number corresponds to a bit in the transaction live list that can be mapped to the input data area.

The transaction live list is always available in the Diagnostics/Status window, even when it is not mapped to the input data area in the memory.

To add a transaction to the group, right click on the group and select either Add Produce, Add Consume, Add Query/Response, Add Dynamic Produce or Add Dynamic Consume. Each transaction holds one CAN frame by default when added to a group. The dynamic transactions can not hold more than one CAN frame.

Produce

A produce transaction transmits CAN frames on the CAN network for all devices on the network to listen to. A CAN device on the network will use the identifier of the produce transaction to decide if the data is meant for it or not. The Communicator operates as any other device on the CAN network, that produces and transmits data on the network. Selecting ‘Produce’ gives access to the following parameters:

Parameter	Value	Comment
Produce Alias	-	An alias for the produce transaction (max 16 characters).
Offline Options	Clear Data	Select what will happen to the output if the PROFINET network goes offline.
	Freeze Data	
	Stop Transaction	
Update Mode	Cyclically	Defines how the transmission of the transaction is triggered.
	On Data Change	
	Single Shot	
	Trigger Byte	
Update on RTR	Disabled	If a message on the configured CAN identifier for a produce transaction is received with the RTR (Remote Transmission Request) bit set, the produce transaction is triggered to be sent. Only available if only one CAN frame is configured in the transaction.
	Enabled	
Transmission Complete Byte	Disabled	When enabled, the Transmission Complete Byte is incremented each time a produce transmission is completed.
	Enabled	
Transmission Complete Address	First available address (default)	If the Transmission Complete Byte is enabled, enter the address here.
Update Time (ms)	1000 (default)	When Update Mode ‘Cyclically’ is selected, this parameter defines the time interval (ms) between two transmissions. Valid range: 5 - 65535.
Trigger Byte Address	First available address (default)	When Update Mode ‘Trigger Byte’ is selected, this parameter specifies the address of the trigger byte. The transaction will be triggered on a change in this byte.

Right click on ‘Produce’ to add another CAN frame. For the setup of CAN frames see *Configuration of CAN Frames, p. 33*.

Consume

A consume transaction listens to CAN frames on the CAN network and collects data from a frame with a matching CAN identifier. The Communicator operates as any other device on the CAN network that listens to all data that is available on the network. Selecting ‘Consume’ gives access to the following parameters:

Parameter	Value	Comment
Consumer Alias	-	An alias for the consume transaction (max 16 characters).
Offline Options	Clear Data	Select what will happen to the input data if the CAN subnetwork goes offline.
	Freeze Data	
Consistency Check	Disabled	When enabled, all frames in the transaction must be received before evaluation. The frames are verified to contain expected data according to the configuration. Once verified, the fieldbus process data is updated with the received data. When disabled, all frames will be evaluated individually and the fieldbus process data is updated directly. The Offline Timeout will be set to 0.
	Enabled	
Offline Timeout	0 (default)	The maximum time before the transaction is considered to be lost. Use 0 to disable the timeout. Valid range: 0, 10 - 65535
Reception Trigger Byte	Disabled	When enabled, the Reception Trigger Byte is incremented each time a consume transmission is received.
	Enabled	
Reception Trigger Address	First available address (default)	If the Reception Trigger Byte is enabled, enter the address here.
Transaction Status Byte	Disabled	When enabled, the Transaction Status Byte is updated every time the status of the transaction is changed.
	Enabled	
Transaction Status Address	First available address (default)	If the Transaction Status Byte is enabled, enter the address here.

Right click on ‘Consume’ to add another CAN frame. For the setup of CAN frames see *Configuration of CAN Frames, p. 33*.

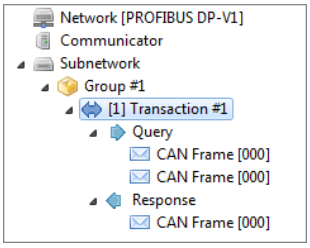
Transaction Status Byte

If enabled, the transaction status byte holds the following status information for each separate transaction.

Bit	Name	Description
0	Timeout	0 - The transaction has not timed out. 1 - The transaction has timed out.
1	Data error	0 - The transaction does not have data errors. 1 - The transaction has data errors. (Data errors that can be detected are ‘data out of range’, ‘invalid data size’ and ‘non-matching constant bytes’ (in Constant object)).
2	Not executed	0 - The transaction has been executed at least once. 1 - The transaction has not been executed yet.
3-7	(reserved)	(Always 0).

Query/Response

In Query/Response mode the Communicator operates as a master and issues queries to the CAN network. The Communicator will then expect a response within the specified timeout. A Query/Response transaction includes both query CAN frames and response CAN frames.



Selecting Query will give the same options as selecting Produce, except 'Update on RTR', see:

- *Produce, p. 28*.

Selecting Response will give the same options as selecting Consume, see:

- *Consume, p. 29*.

Please note that the Offline Timeout value indicates the maximum time that the Communicator will wait for an answer before an error is issued. For a cyclic query, the offline timeout must be lower than the update time.

Right click on either 'Query' or 'Response' to add a new CAN frame. For the setup of CAN frames see *Configuration of CAN Frames, p. 33*.

Dynamic Produce

Only one dynamic produce transaction can be added to a configuration. The function and parameters are similar to a produce transaction with the exception that some parameters can be accessed by the network master in the output data area or the general data area. At the same time, these parameters are not accessible via the Anybus Configuration Manager. Parameters that are available in the Anybus Configuration Manager are given in the table below.

Parameter	Value	Comment
Produce Alias	-	An alias for the dynamic produce transaction (max 16 characters).
Offline Options	Clear Data	Select what will happen to the output data if the PROFINET network goes offline.
	Freeze Data	
	Stop Transaction	
Update Mode	Cyclically	Defines how the transmission of the transaction is triggered.
	On Data Change	
	Single Shot	
	Trigger Byte	
Transmission Complete Byte	Disabled	When enabled, the Transmission Complete Byte is incremented each time a produce transmission is completed.
	Enabled	
Transmission Complete Address	First available address (default)	If the Transmission Complete Byte is enabled, enter the address here.
Trigger Byte Address	First available address (default)	When Update Mode ‘Trigger Byte’ is selected, this parameter specifies the address of the trigger byte. The transaction will be triggered on a change in this byte.
Dynamic Config Address	First available address (default)	This parameter specifies the memory address for the dynamically configurable parameters.

Parameters that can be changed dynamically are stored at the specified memory address in the order given in the table below. These parameters are initialized at 0.

Parameter	Size	Comment
CAN-ID	2 or 4 bytes	11 bit or 29 bit CAN identifier
Update time	2 bytes	When Update Mode ‘Cyclically’ is selected, this parameter defines the time interval (ms) between two transmissions. Valid range: 5 - 65535. Cyclic update is stopped if this parameter is set to 0.
Data length (bit 0-3)	1 byte	The data length is given in bits 0-3 in this byte. At initialization this value is set to 0 and can later be changed up to the maximal data length entered for the data object in the Anybus Configuration Manager.
RTR bit (bit 4)		Signals a remote transmission request.
Reserved (bit 5-7)		-

Dynamic Consume

Only one dynamic consume transaction can be added to a configuration. The function and parameters are similar to a consume transaction with the exception that some parameters can be accessed by the network master in the output data area or the general data area. At the same time, these parameters are not accessible via the Anybus Configuration Manager. Parameters that are available in the Anybus Configuration Manager are given in the table below

Parameter	Value	Comment
Consumer Alias	-	An alias for the consume transaction (max 16 characters).
Offline Options	Clear Data	Select what will happen to the input data if the CAN subnetwork goes offline.
	Freeze Data	
Reception Trigger Byte	Disabled	When enabled, the Reception Trigger Byte is incremented each time a consume transmission is received.
	Enabled	
Reception Trigger Address	First available address (default)	If the Reception Trigger Byte is enabled, enter the address here.
Transaction Status Byte	Disabled	When enabled, the Transaction Status Byte is updated every time the status of the transaction is changed.
	Enabled	
Transaction Status Address	First available address (default)	If the Transaction Status Byte is enabled, enter the address here.
Dynamic Config Address	First available address (default)	This parameter specifies the memory address for the dynamically configurable parameters.

Parameters that can be changed dynamically are stored at the specified memory address in the order given in the table below. These parameters are initialized at 0.

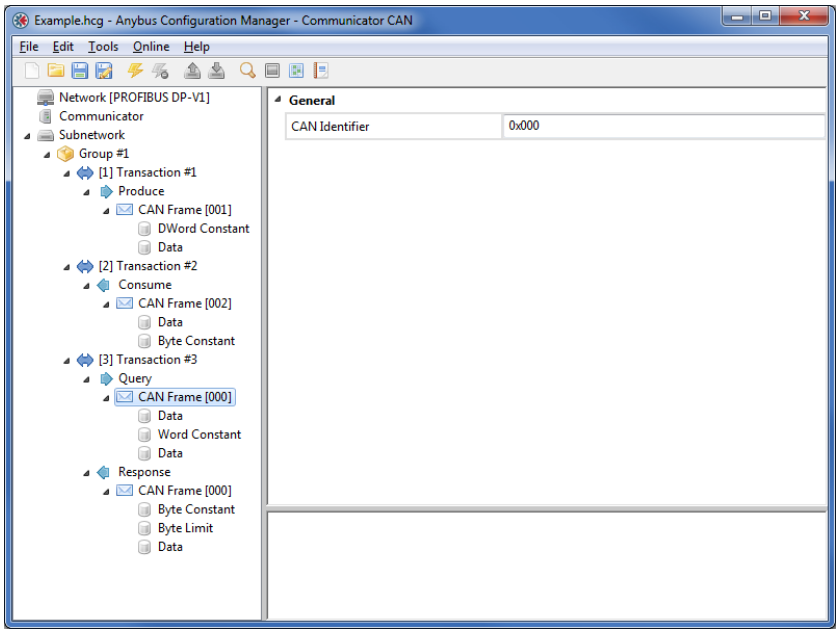
Parameter	Size	Comment
CAN-ID	2 or 4 bytes	11 bit or 29 bit CAN identifier
Offline Timeout	2 bytes	The maximum time before the transaction is considered to be lost. Use 0 to disable the timeout. Valid range: 0, 10 - 65535
Data length (bit 0-3)	1 byte	The data length is given in bits 0-3 in this byte. At initialization this value is set to 0 and can later be changed up to the maximal data length entered for the data object in the Anybus Configuration Manager.
Reserved (bit 4-7)		-

5.4 Configuration of CAN Frames

5.4.1 General

Each transaction includes one or more CAN frames. A total of 256 CAN frames is allowed. Right-clicking on a transaction will give the opportunity to add another frame to the transaction.

The Anybus Configuration Manager makes it possible to decide the configuration of the 8 bytes of data that can be included in each frame. The configuration manager automatically allocates memory space in the input and output areas of the Communicator for the data objects that are configured in the frames. The result can be seen in the Address Overview, see page 44. Any address conflicts will turn up red in this view.



A CAN frame can not contain more than 8 bytes of data. It is possible to configure the data area in each frame, but the size of the combination of objects must not exceed 8 bytes.

CAN Identifiers

Each frame has a CAN identifier, to make it possible for each node on the CAN network to recognize data meant for it. Default identifier is ‘0’. It can be changed by selecting the CAN Frame and enter the new CAN Identifier in the Parameter window.

The CAN frame has either a 11-bit identifier or a 29-bit identifier. If the size of the identifier is changed, an 11-bit identifier will have the 11 original bits padded with zeroes in front. A 29 bit identifier will have its 18 highest bits cut, which may cause a not valid 11-bit identifier.

It is possible to have several frames in one transaction. The first frame in a Consume or Response transaction must have a CAN identifier that does not appear in any other Consume or Response transaction. Consecutive frames within a received transaction may have the same identifier, on two conditions:

- The first part of the data area in the frame is a byte, word or Dword constant with a unique value compared to other frames with the same identifier within the transaction.
- If any frame with another identifier is added to the transaction, it must not break the sequence of frames with identical identifiers.

5.4.2 Produce/Query CAN Frame

The following objects and parameters are configurable in a CAN frame in a produce transaction, or when used in the query part of a query/response transaction. To add objects to the 8 byte data area of the frame, right-click on CAN Frame.

Object	Parameter	Comment
Data	Data Length (bytes)	A data object can occupy 1 - 8 bytes (default =1).
	Data Address	Address in the data area where the object shall be mapped. Default: The first available position is used.
	Swap	Values: Result (original value = 0102 0304): No Swapping (default): 0102 0304 Word Swap: 0201 0403 Double Word Swap: 0403 0201
Byte Constant	Value (1 byte, valid range: 0x00 - 0xFF)	Constant value to be transmitted (little endian).
Word Constant	Value (2 bytes, valid range: 0x0000 - 0xFFFF)	
Dword Constant	Value (4 bytes, valid range: 0x00000000 - 0xFFFFFFFF)	

5.4.3 Consume/Response CAN Frame

The following objects and parameters are configurable in a CAN frame in a consume transaction, or when used in the response part of a query/response transaction. To add objects to the 8 byte data area of the frame, right-click on CAN Frame.

Object	Parameter	Comment
Data	Data Length (bytes)	A data object can occupy 1 - 8 bytes (default =1).
	Data Address	Address in the data area where the object shall be mapped. Default: The first available position is used.
	Swap	Values: Result (original value = 0102 0304): No Swapping (default): 0102 0304 Word Swap: 0201 0403 Double Word Swap: 0403 0201
Byte Constant	Value (1 byte, valid range: 0x00 - 0xFF)	When receiving a message with a constant, the received value will be checked against this value. If the values differ, the message will be ignored (little endian).
Word Constant	Value (2 bytes, valid range: 0x0000 - 0xFFFF)	
Dword Constant	Value (4 bytes, valid range: 0x00000000 - 0xFFFFFFFF)	
Byte Limit (1 byte, valid range: 0x00 - 0xFF)	Minimum Value	When receiving a message with a limit object, the received value will be checked against the minimum value. If the received value is lower than the minimum value, the message will be ignored.
	Maximum Value	When receiving a message with a limit object, the received value will be checked against the maximum value. If the received value is larger than the maximum value, the message will be ignored.
Word Limit (2 bytes, valid range: 0x0000 - 0xFFFF)	Minimum Value	When receiving a message with a limit object, the received value will be checked against the minimum value. If the received value is lower than the minimum value, the message will be ignored.
	Maximum Value	When receiving a message with a limit object, the received value will be checked against the maximum value. If the received value is larger than the maximum value, the message will be ignored.
Dword Limit (4 bytes, valid range: 0x00000000 - 0xFFFFFFFF)	Minimum Value	When receiving a message with a limit object, the received value will be checked against the minimum value. If the received value is lower than the minimum value, the message will be ignored.
	Maximum Value	When receiving a message with a limit object, the received value will be checked against the maximum value. If the received value is larger than the maximum value, the message will be ignored.

5.4.4 CAN Frames in Dynamic Transactions

A dynamic transaction must consist of one frame. Only one object, a data object, can be added to the 8 byte data area of this frame.

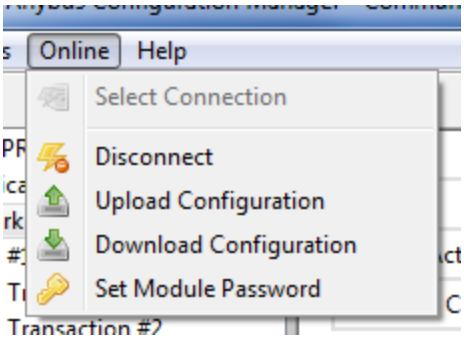
The CAN Identifier of the CAN frame in a dynamic transaction can not be set in the Anybus Configuration Manager. The identifier is stored in the output data area or the general data area.

The Max Data Length of the data object is entered in the Configuration Manager, but the actual data length is given by the parameter mapped to the output data area or the general data area.

5.5 Online Menu

5.5.1 Select Connection

To be able to access the module, start by choosing ‘Select Connection’.



The module supports Ethernet and USB connections.

Although the Anybus Ethernet (UCOM) connection is available in the connection list, it can in practice only be used if the module has an RJ45 connector for Ethernet.

General

When a connections is selected, the PC running the ACM will lock to that specific Communicator. If the USB connection is used, the Communicator will be identified by its serial number. If the Ethernet connection is used, the IP address will be used for identification. If the configuration is to be downloaded to another module, using the same PC, the process of selecting connection will have to be repeated for that specific module.

It is recommended to select a specific Communicator for the connection, as this will diminish the risk of downloading the wrong configuration.

Anybus Ethernet (UCOM) Connection

Selecting ‘Anybus Ethernet (UCOM)’ and pressing ‘Configure’ opens a window where available Anybus Communicator CAN modules are listed.

To scan the network for further modules, press the ‘Scan’ button at the bottom of the window.

If the IP settings for a module are not set, it is possible to set these by pressing the ‘Set IP’ button. The module can be identified by the MAC Id listed in the rightmost column.

These IP settings will be overwritten at the next power up of the module, if the settings are changed within the configuration.

Anybus USB (UCOM) Connection

To use a USB connection, select ‘Anybus USB (UCOM)’. Continue by pressing ‘Configure’ to open the ACM USB Connection window. The dropdown menu in this window shows available Anybus Communicator CAN modules.

There are also the options to either manually enter the serial number of a desired device or to select ‘first available device’ to download a configuration to.

5.5.2 Connect/Disconnect

The Communicator is connected/disconnected using this entry in the menu.

5.5.3 Download and Upload Configuration

Selecting “Download Configuration” downloads the configuration to the Communicator. Any configuration previously present in the ACM will be overwritten.

Selecting ‘Upload Configuration’ will fetch the configuration in the connected Communicator to the Anybus Configuration Manager.

If the configuration is to be downloaded to another Communicator, change the connection, see *Select Connection, p. 36*.

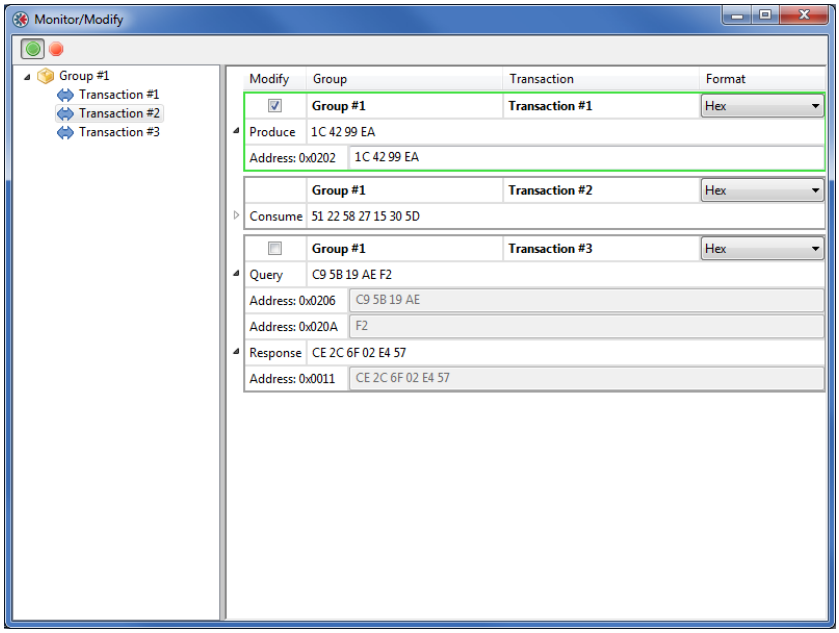
5.6 Anybus Configuration Manager Tools

The Anybus Configuration Manager (ACM) gives access to different tools for monitoring and controlling the module and the CAN subnetwork:

- Monitor/Modify
- CAN Line Listener
- Address Overview
- Diagnostics/Status
- Reassign Addresses
- Project Summary
- Password
- Options

5.6.1 Monitor/Modify

Selecting this option in the Tools menu opens this window, where the data areas of the transactions can be monitored. If the configuration downloaded to the Communicator is the same as is open in the ACM, it is possible to monitor and modify the transactions. Pressing the green button on the left starts the monitoring/modifying:



If Modify is enabled, it is possible to change the data values during runtime in Produce transactions and in the Query part of Query/Response transactions, i.e. only the out area of the Communicator can be modified. This will inhibit any data from the industrial network (EtherCAT), but input data from the CAN network will still be updated.

Addresses in the general area range can not be modified. If a transaction only has addresses in the general area, the Modify check box will be disabled.

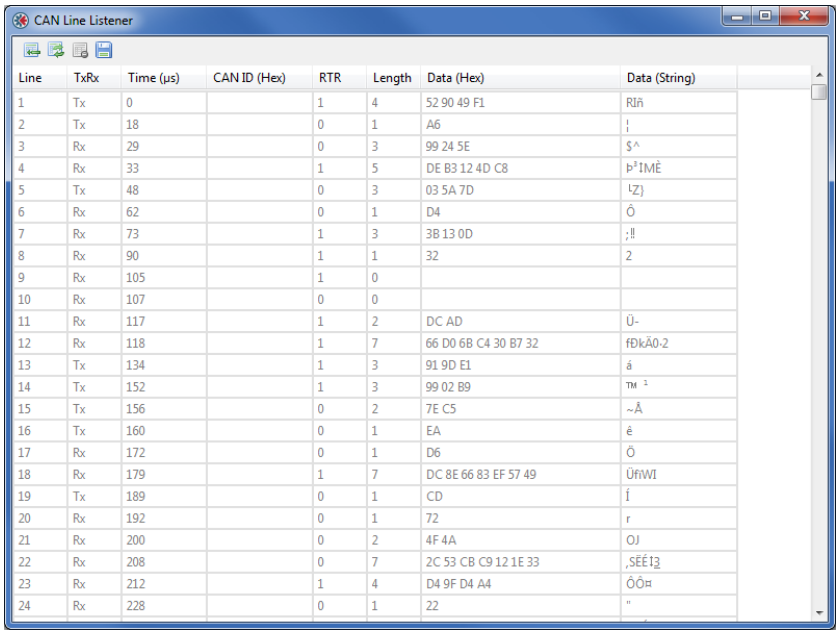
If a transaction is defined to transmit on a change of state in a trigger byte, this transaction can not be modified by this tool.

5.6.2 CAN Line Listener

The CAN Line Listener gives the opportunity to log the traffic on the CAN network. Any log can be saved for later use. The 5000 latest frames are logged. This is done continuously, or it is possible to stop logging after 5000 frames from a defined time.

The CAN Line Listener shows all CAN frames present on the CAN network, not only those sent or received by the Communicator. Information about CAN frames, that have identifiers present in the configuration, that is downloaded to the Communicator, is shown in black text. Information about all other frames is shown in gray text. Clicking on the save icon will save the log at the location entered in the Tools/Options dialogue, see *Options, p. 42*.

Please note that the configuration in the ACM and the configuration in the Communicator have to match.

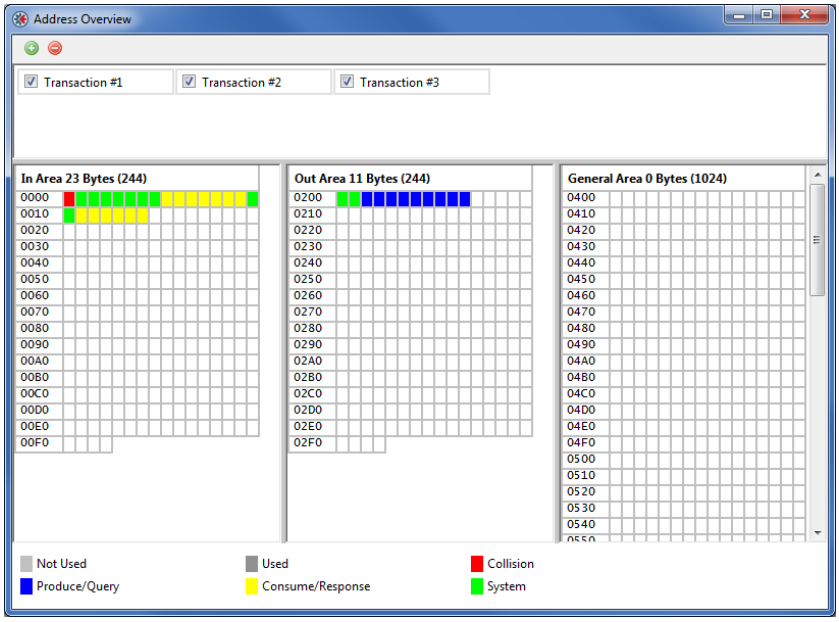


Line	TxRx	Time (µs)	CAN ID (Hex)	RTR	Length	Data (Hex)	Data (String)
1	Tx	0		1	4	52 90 49 F1	RIñ
2	Tx	18		0	1	A6	¦
3	Rx	29		0	3	99 24 5E	$^
4	Rx	33		1	5	DE B3 12 4D C8	Þ³1MÈ
5	Tx	48		0	3	03 5A 7D	Z}
6	Rx	62		0	1	D4	Ô
7	Rx	73		1	3	3B 13 0D	;‼
8	Rx	90		1	1	32	2
9	Rx	105		1	0		
10	Rx	107		0	0		
11	Rx	117		1	2	DC AD	Ü-
12	Rx	118		1	7	66 D0 6B C4 30 B7 32	fÐkÄ0·2
13	Tx	134		1	3	91 9D E1	á
14	Tx	152		1	3	99 02 B9	™ ¹
15	Tx	156		0	2	7E C5	~Å
16	Tx	160		0	1	EA	ê
17	Rx	172		0	1	D6	Ö
18	Rx	179		1	7	DC 8E 66 83 EF 57 49	ÜfƒWI
19	Tx	189		0	1	CD	Í
20	Rx	192		0	1	72	r
21	Rx	200		0	2	4F 4A	OJ
22	Rx	208		0	7	2C 53 CB C9 12 1E 33	,SËÉ13
23	Rx	212		1	4	D4 9F D4 A4	ÔŸÔ¤
24	Rx	228		0	1	22	"

The CAN Line Listener will only display data if the RTR bit is NOT set.

5.6.3 Address Overview

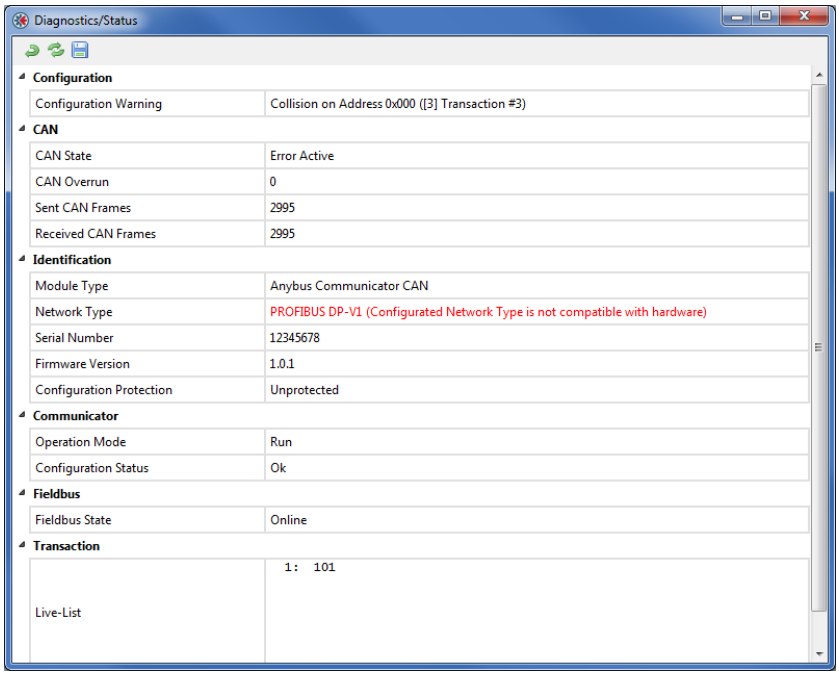
The Address Overview tool shows the usage of the different memory areas in the module. It gives an easy view of any collisions of data that are present in the different memory areas. If needed, the memory location for the data of one transaction at a time, can be shown.



The Address Overview is an offline tool with no reference to the module. It shows the memory usage of the the configuration that is present in the Anybus Configuration Manager at the moment.

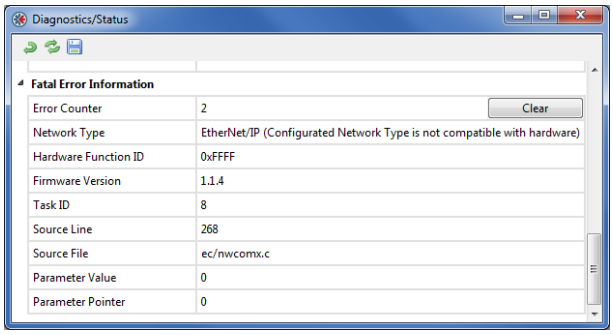
5.6.4 Diagnostics/Status

The Diagnostics/Status tool gives access to diagnostics and status information.



Item	Description
Configuration Tool	The configuration is validated by the ACM and any errors will be reported here, e.g. if the some address has been used for several transactions or if the same CAN identifier is used for more than one transaction. This is the only section of the Diagnostic/Status window that can be used when the configuration tool is not connected to a Communicator *This information is valid for the configuration in the tool only, and does not relate to any configuration stored in the module.*
CAN	Information on the status of the CAN subnetwork
Identification	Information on the module.
Communicator	This item gives the operation mode and the configuration status of the Communicator.
Network	Network state.
Transaction	The live list will be shown here. It can also be kept in the input memory area.
Fatal Error Information	If the Communicator is subject to a fatal error, this information is used by HMS support when troubleshooting the module. Please contact HMS support if a fatal error occurs.

The figure below shows an example of what the section Fatal Error Information may look like.



The information in the Diagnostics/Status window can be saved by clicking on the save icon (file format CSV). The file will be saved at the location entered in the Tools/Options dialogue, see *Options, p. 42*.

5.6.5 Reassign Addresses

This tool sorts all assigned data and puts it in order, from the beginning of the memory area and on. It also removes any collisions. The result can be seen using the tool “Address Overview”.

There will be no confirmation notice after clicking the “Reassign Addresses” button.

5.6.6 Project Summary

Project information and a summary of the configuration is saved as a html file and can be read in any browser. The file is saved as the location entered in the Tools/Options dialogue, see *Options, p. 42*.

Selecting ‘Project Summary’ will open a browser window that displays the summary.

5.6.7 Password

It is possible to password protect a configuration. Passwords can be set both for uploading and downloading a configuration.

Passwords are set in the Options window, see *Options, p. 42*. The same password can be used for uploading a configuration and for downloading a configuration. If the “Set Module Password on Download” parameter is enabled, the password will be downloaded to the module along with the configuration. When a configuration is protected by passwords you can still use the tools that are listed in this chapter. It is only the configuration by itself that is protected.

The passwords in a connected module can be changed directly from the ‘Change Module Password’ entry in the Tools menu. If no password has been set previously, the “Old Password” box should be left empty

5.6.8 Options

Selecting this entry gives access to more settings, that can be used to adapt the behavior of the Communicator.

Item	Subitem	Comment
General	Language	-
	Monitor/Modify Update Time (ms)	Enter the time between monitor/modify updates in milliseconds. Valid range: 1000 to 60000 Default: 2000
Security	Module Upload Password	-
	Module Download Password	-
	Set Module Password on Download	Default: disabled
File Paths	Diagnostics & Line Listener	By default, the logs and the project summary are saved in the user catalog in Windows (\My Documents\HMS\ACM Communicator CAN\). To change this, browse to or enter the name of the folder where the logs shall be saved. If the folder entered does not exist, the ACM will use the default address.

6 PROFINET Asset Management

6.1 Asset Management Record

With the *asset management record* functionality data about the assets available on a non PROFINET network can be recorded and read out over a PROFINET network.

Together with the *Identification & Maintenance data* functionality an extensive registration of devices and machines is possible, even in facilities where the devices are not installed in the PROFINET environment.

Factory owners and system integrators can collect data about devices installed beyond the *Anybus gateway*.

The recorded data can be used as basis for the design of easier maintenance and operation processes, despite the increasing complexity of processes and associated machines.

6.2 Recording and Reading Data

An *asset management* file containing all the *assets* and their corresponding data on the non PROFINET network is created and uploaded via an *FTP* server to the Gateway *file system*.

The *asset management* file can be transferred from a computer connected to a PROFINET network.

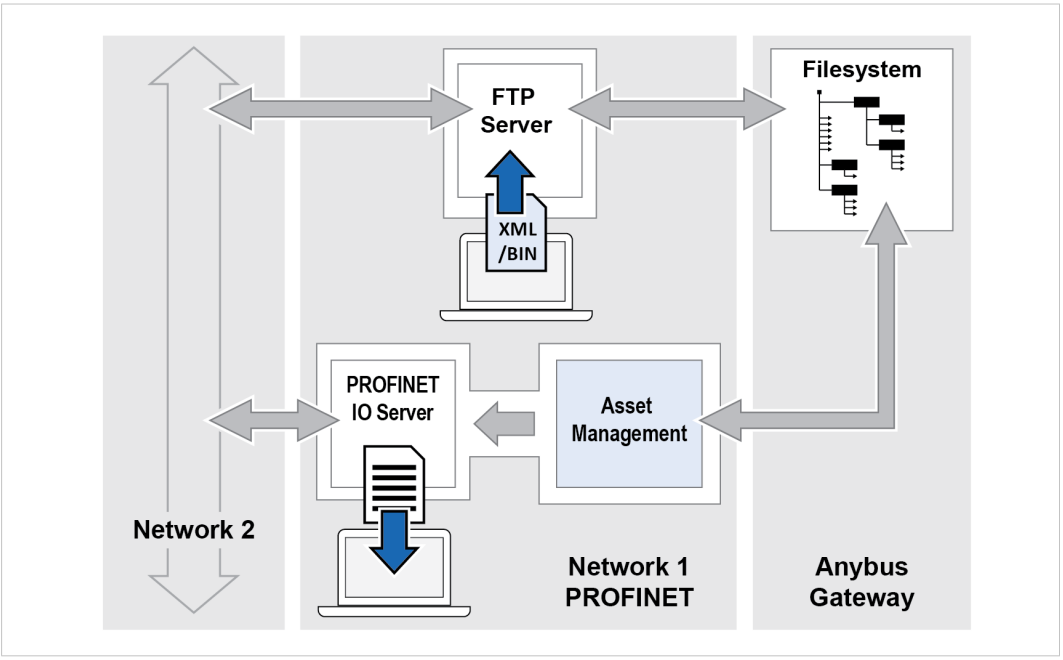


Fig. 20 The Asset Management Default Mode

By using the *superposed parameter channel* mode it is also possible to transfer the *asset management* file from a PLC connected to a non PROFINET network.

For further details about the *superposed parameter channel* mode, please refer to www.anybus.com/support.

Record Data

Data about the *assets* on the non PROFINET network is recorded and stored in an *XML* file or an *binary* file.

Read Data

Each time an *instance* is requested the *asset management* data is read out over the PROFINET network.

The recorded *asset management* data can be downloaded to a computer connected to the PROFINET network.

6.3 Supported File Formats

The following file formats are supported for the *asset management* file.

Format	Version
XML	XML Version 1.0
Binary file Little-endian	N/A

6.4 Supported Asset Management Records

Supported *asset management records*:

- Unique ID
- Location
- Hardware Revision
- Annotation
- Order ID
- Serial Number
- Software Revision
- Serial Number

6.5 XML Based Asset Management

6.5.1 Creating the Asset Management XML File

Creating the *asset management* XML file:

1. List all assets and their corresponding data on the non PROFINET network.
2. Create an XML file that include one *asset management record* for each asset.

   Repeat all the *attributes* after each other.
3. When all *attributes* are listed, close the *element* by using a *closing entry.*
4. Name the XML file *asset_mgmt*.

6.5.2 XML File Size Limitation

The size of the asset management file may not exceed 95 kb.

Up to 32 *instances* can be added.

In order to keep the file size small, consider the following:

- Keep strings as short as possible.
- Do not pad with empty spaces for strings.
- Try to use as few spaces as possible for indentation in the file.
- The number of white-space also affects the file size.
- Avoid using *optional name strings*.

6.5.3 XML Attribute Name and Data Format

The order of the elements is significant for the XML schema to work with the Anybus Gateways.
If the XML schema is incorrect, the XML file will not work and no data will be recorded.

When creating the XML file, add the *elements* and their *attributes* in the same order as the *attribute names* are listed in the table below.

Each *element* consists of a series of *attributes* and their various data.

Each *attribute* is described by one *entry.*

The supported *attribute names* are specified in the table.

Example 1: XML *element* including an *attribute* with the *location* record.

```
<AbccAttribute>
<Name Value="Location Type"/>
<Attribute Value="3"/>
<Data Value="1"/>
</AbccAttribute>
```

Attribute Name and Data Format		
Attribute Name	**Data Format**	**Description**
AM info Type Location Type	Unsigned 8	The value can be set in either of two formats, `0x12` or `18`.
AM Type Identification IM Hardware Revision	Unsigned 16	The value can be set in either of two formats, `0x1234` or `4660`.
IM Annotation	String of length `X`	Maximum number of elements in array: 64.
IM Order ID	String of length `X`	Maximum number of elements in array: 64.
IM Serial Number	String of length `X`	Maximum number of elements in array: 16.
AM Software Revision	String of length `X`	Maximum number of elements in array: 64.
AM Hardware Revision	String of length `X`	Maximum number of elements in array: 64.
IM Software Revision	String	Format of the string shall be `C.X.Y.Z`. `C` is one character. `X`, `Y` and `Z` represent a value between 0 and 255. `X` – Major version `Y` – Minor version `Z` – Internal
IM Unique Identifier	Array of Unsigned 8 Length is 16	Format of the value shall be `0xXX;0xYY…0xZZ`. 16 values in hex-format, where each value is separated by a ";".
Location LT	Array of Unsigned 16 Length is up to 12 elements.	Format of the value shall be `0xXXXX;0xYYYY…0xZZZZ`. Up to 12 values in hex-format, where each value is separated by a ";".
Location SS AM Device Identification	Array of Unsigned 16 Length is 4.	Format of the value shall be `0xXXXX;0xYYYY…0xZZZZ`. 4 values in hex-format, where each value is separated by a ";".

6.5.4 Asset Management XML File Structure Example

The code example presented below can be used as a guide when creating the *asset management* XML file.

```
<AssetManagement Created="2017-01-01 01:01:01">
    <AbccObject>
        <Data Value="0xE5"/>
            <AbccInstance>
                <Data Value="1"/>
                    <AbccAttribute>
                        <Name Value="AM info Type"/>
                        <Attribute Value="1"/>
                        <Data Value="0"/>
                    </AbccAttribute>

                    <AbccAttribute>
                        <Name Value="IM Unique Identifier"/>
                        <Attribute Value="2"/>
                        <Data Value="0x01;0x02;0x03;0x04;0x05;0x06;0x07;0x08;0x09;0x0A;0x0B;0x0C;0x0D;0x0E;0x0F;0x10;"/>
                    </AbccAttribute>

                    <AbccAttribute>
                        <Name Value="Location Type"/>
                        <Attribute Value="3"/>
                        <Data Value="1"/>
                    </AbccAttribute>

                    <AbccAttribute>
                        <Name Value="Location LT"/>
                        <Attribute Value="4"/>
                        <Data Value="0x0001;0x0002;0x0003;0x0004;0x0005;0x0006;0x0007;0x0008;0x0009;0x000A;0x000B;0x000C;"/>
                    </AbccAttribute>

                    <AbccAttribute>
                        <Name Value="Location SS"/>
                        <Attribute Value="5"/>
                        <Data Value="0x0001;0x0002;0x0003;0x0004;"/>
                    </AbccAttribute>

                    <AbccAttribute>
                        <Name Value="IM Annotation"/>
                        <Attribute Value="6"/>
                        <Data Value="64 characters max"/>
                    </AbccAttribute>

                    <AbccAttribute>
                        <Name Value="IM Order ID"/>
                        <Attribute Value="7"/>
                        <Data Value="64 characters max"/>
                    </AbccAttribute>
```

```
                    <AbccAttribute>
                        <Name Value="IM Serial Number"/>
                        <Attribute Value="8"/>
                        <Data Value="16 chars max    "/>
                    </AbccAttribute>

                    <AbccAttribute>
                        <Name Value="AM Device Identification"/>
                        <Attribute Value="9"/>
                        <Data Value="0x0001;0x0002;0x0003;0x0004;"/>
                    </AbccAttribute>

                    <AbccAttribute>
                        <Name Value="AM Type Identification"/>
                        <Attribute Value="10"/>
                        <Data Value="0x0000"/>
                    </AbccAttribute>

                    <AbccAttribute>
                        <Name Value="AM Software Revision"/>
                        <Attribute Value="11"/>
                        <Data Value="64 characters max"/>
                    </AbccAttribute>

                    <AbccAttribute>
                        <Name Value="IM Software Revision"/>
                        <Attribute Value="12"/>
                        <Data Value="V.1.02.03"/>
                    </AbccAttribute>

                    <AbccAttribute>
                        <Name Value="AM Hardware Revision"/>
                        <Attribute Value="13"/>
                        <Data Value="64 characters max"/>
                    </AbccAttribute>

                     <AbccAttribute>
                        <Name Value="IM Hardware Revision"/>
                        <Attribute Value="14"/>
                        <Data Value="0x0000"/>
                    </AbccAttribute>
            </AbccInstance>

    </AbccObject>
</AssetManagement>
```

Fig. 21 Asset management XML file structure example

6.6 Binary Based Asset Management

6.6.1 Creating the Asset Management Binary File

Creating the *asset management* binary file:

1. List all assets and their corresponding data on the non PROFINET network.
2. Create an Binary file that include a *asset management record* for each asset.

   Repeat all the *attributes* after each other.
3. When all *attributes* are listed, close the *element* by using a *closing entry.*
4. Name the bin file *asset_mgmt*.

6.6.2 Binary File Size Limitation

The size of the asset management file may not exceed 12 kb.

32 instances can be added, instance 1 to 32.

In order to keep the file size small, consider the following:

- Keep strings as short as possible.
- Do not pad with empty spaces for strings.

6.6.3 Binary File Header

Omitted attributes are disabled or set to their default value.

The size of the file header is 70 bytes.

The supported *file headers* are specified in the table.

Supported File Headers

File Header	Byte Number	Data Type	Comment
File format version	0-1	UINT16	Version number of the file format. Set to 0.
File checksum	2-5	UINT32	Used for version control of the file. Not used by the gateway. If not used, the field must be set to zero.
Byte offset to Instance 1	6-7	UINT16	Byte offset to the start of the data describing Asset management Instance X. Set to zero if instance is not used.
Byte offset to Instance 2	8-9		
Byte offset to Instance 32	68-69		
Instance data	70-x	N/A	Data for the instance(s), as specified below.

6.6.4 Binary Instance Data

Each *instance* consists of a series of *attributes* and their respective data.

Attribute Description

Each *attribute* is described by one entry.

Attribute Description	Byte number	Data type	Comment
Attribute number	0	UINT8	Attribute number of the data being described.
Data length	1	UINT8	Optional checksum. Shall represent the number of data bytes following. Not used by the gateway.
Attribute data	2–x	Depends on the attribute being described.	Data for the attribute. Format shall be as described for the data-type. Not needed for strings padding or termination.

Attribute Closure Description

Use a *closing entry* to close the instance data.

Attribute Description	Byte number	Data type	Comment
Closure	0–1	UINT16	Data-field which tell that there will not follow any more attributes for this instance. Set to value 0xFFFF.

Attribute Name and Data Format

Supported *attribute names* and *data formats*.

Attribute Name and Data Format

Attribute Name	Data Format	Description
AM info Type Location Type	Unsigned 8	The value is set as one byte value.
AM Type Identification IM Hardware Revision	Unsigned 16	The value is set with two bytes, *little-endian* format.
IM Annotation	String of length X	Maximum number of elements in array: 64.
IM Order ID	String of length X	Maximum number of elements in array: 64.
IM Serial Number	String of length X	Maximum number of elements in array: 16.
AM Software Revision	String of length X	Maximum number of elements in array: 64.
AM Hardware Revision	String of length X	Maximum number of elements in array: 64
IM Software Revision	Array of Unsigned 8 Length is 4	First byte is a character. Bytes 2, 3 and 4 represent the version in the format X.Y.Z where X, Y and Z represent a value between 0 and 255. C is one character. X, Y and Z represent a value between 0 and 255. X – Major version Y – Minor version Z – Internal
IM Unique Identifier	Array of Unsigned 8 Length is 16	Format is 16 bytes.
Location LT	Array of Unsigned 16 Length is up to 12 elements.	Each Unsigned 16 comprises two bytes, where each two bytes form an Unsigned 16 in *little-endian* format. The number of Unsigned 16's can be up to 12, placed directly after each other
Location SS AM Device Identification	Array of Unsigned 16 Length is 4.	Each Unsigned 16 comprises two bytes, where each two bytes form an Unsigned 16 in *little-endian* format. The number of Unsigned 16's shall be 4, placed directly after each other.

6.6.5 Asset Management Binary File Example

The binary file structure example presented below can be used as a guide when creating the *asset management* binary file.

Only *instance 1* is supported.

For *instance 1*, only attribute 1 and 2 are defined.

	0	1	2	3	4	5	6	7
0	0x00	0x00	0x01	0x02	0x03	0x04	0x46	0x00
8	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
16	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
24	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
32	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
40	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
48	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
56	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
64	0x00	0x00	0x00	0x00	0x00	0x00	0x01	0x01
72	0x01	0x02	0x10	0x01	0x02	0x03	0x04	0x05
80	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D
88	0x0E	0x0F	0x10	0xFF	0xFF			

Fig. 22 Binary file example

6.7 Uploading the Asset Management File to the FTP Server

Use *Windows Explorer* or a standard *FTP client* to transfer the *asset management file* to the *FTP server.*

When the *superposed parameter channel* function is enabled, transfer the *asset management file* via a PLC connected to the network where the gateway is installed.

6.7.1 Transferring the Asset Management File from Windows Explorer

Transfer the *asset management file*, XML or binary file, to the *FTP server* using *Windows Explorer.*

Before You Begin

Use only one of the file formats, XML format or binary format.

Only upload one single file on the FTP server.

- Name the *asset management file*: *asset_mgmt*
- The default port is FTP port 21.
- Make sure that the gateway and your computer are connected to the PROFINET network to be used.

Procedure

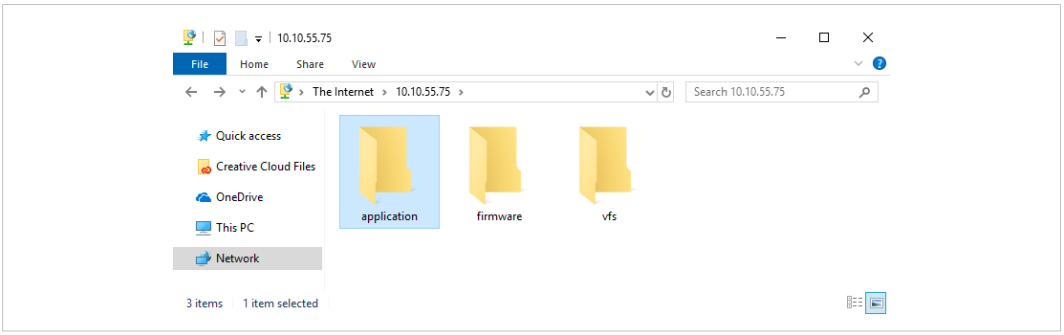


Fig. 23 The FTP Server root folder

1. Open an **Windows Explorer Window**.
2. Click to select the Address bar.
3. Enter **ftp://Username:Password@IPaddress**.
   - Replace “Username” and “Password” with a valid username and password combination.
   - Replace ‘IPaddress’ with the IP address of the PROFINET interface.
4. Press **Enter**.

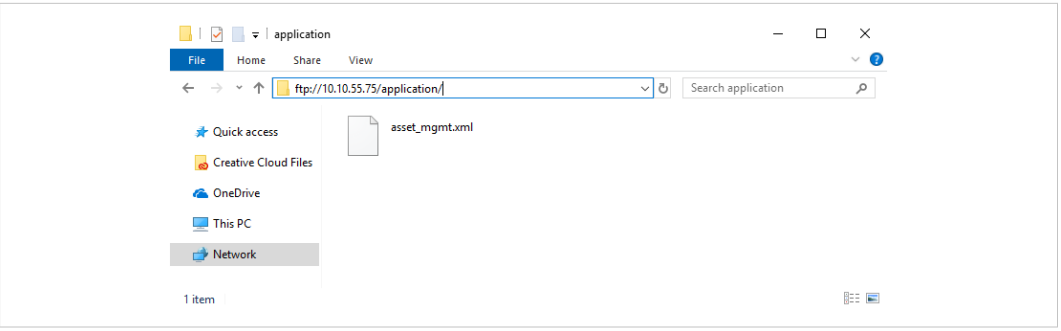


Fig. 24 Application folder with an asset_mgmt.xml file

5. Open the *application* folder and save the *asset management file*, XML or Binary file, in the folder.

This page intentionally left blank

A Technical Data

A.1 General Specifications

Model name	**Anybus Communicator CAN PROFINET IRT (2.32)**
Order code	AB7328
Dimensions (L x W x H)	120 x 75 x 27 mm
Weight	150 g
Operating temperature	-25 to +55 °C (IEC 60068-2-1 and IEC 60068-2-2)
Storage temperature	-40 to +85 °C (IEC 60068-2-1 and IEC 60068-2-2)
Humidity range	5–95 % RH, non-condensing (IEC 60068-2-30)
Power supply	24 V ±10 % DC regulated power source
Current consumption	Typical: 100 mA @ 24 VDC Maximum: 250 mA @ 24 VDC
Galvanic isolation	Yes, on both network sides
Mechanical rating	IP20, NEMA rating 1
Mounting	DIN rail (EN 50022) Network shield conductance via DIN rail
Certifications	CE

A.2 CAN Interface

Maximum baud rate	1 Mbit/s
CAN connector	D-sub 9 Male (included)
CAN specification	CAN 1.0, 2.0A and 2.0B

A.3 PROFINET IRT Interface

PROFINET specification	2.32
PROFINET functionality	Isochronous Real-Time (IRT) communication Conformance supporting Class A, B and C Media Redundancy Protocol (MRP) support Discovery and Configuration Protocol (DCP) support Acyclic Data exchange (Record Data Requests) Asset Management
Isochronous cycle times	0.250 ms to 16 ms
Maximum I/O data	Up to 512 byte in each direction
Ethernet	100 Mbit/s, full duplex (fixed) Dual port cut-through switch, RJ45 connectors Ethernet Transport Provider support

B Licenses

This product includes software developed by Carnegie Mellon, the Massachusetts Institute of Technology, the University of California, and RSA Data Security:

This code is derived from software contributed to Berkeley by Stephen Deering of Stanford University.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

****

License to copy and use this software is granted provided that it is identified as the “RSA Data Security, Inc. MD4 Message-Digest Algorithm” in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as “derived from the RSA Data Security, Inc. MD4 Message-Digest Algorithm” in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided “as is” without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

****

License to copy and use this software is granted provided that it is identified as the “RSA Data Security, Inc. MD5 Message-Digest Algorithm” in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as “derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm” in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided “as is” without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

This page intentionally left blank

Box 4126
300 04 Halmstad, Sweden

info@hms.se