ENGLISH

Anybus® Communicator™ - PROFIBUS to Modbus RTU/Serial

USER MANUAL

SCM-1202-177
Version 1.2
Publication date 2021-11-01



Important User Information

Disclaimer

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

HMS 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 Networks and is subject to change without notice. HMS 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 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 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.

Copyright © 2021 HMS Networks

Contact Information

Postal address:
Box 4126
300 04 Halmstad, Sweden

E-Mail: info@hms.se

Table of Contents

1. Preface 1
1.1. About This Document 1
1.2. Document Conventions 1
1.3. Trademarks 2

2. Safety 3
2.1. Intended Use 3
2.2. General Safety 3

3. Preparation 4
3.1. Cabling 4
3.2. System Requirements 4
3.2.1. Supported Operating Systems 4
3.2.2. Supported Web Browsers 4
3.3. Tools 4
3.4. Support and Resources 4
3.5. HMS Software Applications 5
3.6. Third-Party Software Applications 5

4. About Anybus Communicator 6
4.1. Serial Protocol Communication 6
4.1.1. Serial Protocol Types 6
4.1.2. Serial Protocol Building Blocks 7
4.2. How the Communication Works 10
4.3. How the Data Exchange Works 12
4.4. Data Integrity 12

5. Installation 13
5.1. External Parts 13
5.2. DIN Rail Mounting 14
5.3. Connect to PROFIBUS Network 15
5.4. Connect to Serial RS232/RS485 Subnetwork 16
5.5. Rotary Switch Settings 17
5.5.1. Rotary Switches Default Setting 17
5.5.2. Set a Node Address with Rotary Switches 18
5.6. Connect to Power 22
5.7. Security Switch 24
5.8. Lock the Cables 26
5.9. DIN Rail Demount 27

6. Configuration Quick Guide 29
6.1. Prepare Configuration 29
6.2. Setup New Configuration 32
6.3. PLC Configuration 35
6.4. Verify Operation 37

7. Communicator Configuration 39
7.1. Connect the Communicator 39
7.2. Access the Built-In Web Interface From HMS IPconfig 40
7.3. Access the Built-In Web Interface From a Web Browser 42
7.4. Communicator Built-In Web Interface Overview 44
7.5. General Subnetwork Settings 45

7.5.1. Communication Serial Protocol 45
7.5.2. Communication Basic Settings 46
7.5.3. Communication Advanced Settings 48
7.6. About Transaction Templates 49
7.6.1. Transaction Template Example 49
7.6.2. Transaction Template Types 50
7.6.3. Frame Field Types 52
7.7. Build Transaction Template 54
7.7.1. Add Transaction Template 54
7.7.2. Add Frame Fields 58
7.7.3. Configure Frame Fields Settings 60
7.7.4. Data Delimiter and Subnet Delimiter Options 62
7.7.5. Store Transaction Templates 65
7.8. Nodes and Transactions 66
7.8.1. Node and Broadcast Node 66
7.8.2. Add Node 67
7.8.3. Node Settings 68
7.8.4. Add Transactions 69
7.8.5. Transaction Settings 72
7.8.6. Activate/Deactivate Transaction 74
7.8.7. Duplicate Transaction 74
7.8.8. Delete Transaction 75
7.9. High Level Network Settings 76
7.9.1. To Use Automatic I/O Sizes 76
7.9.2. To Configure I/O Sizes Manually 76
7.9.3. PROFIBUS Address Settings 77
7.10. I/O Data Map 78
7.10.1. Optimize the I/O Data Map 79
7.10.2. Map Area Transactions Order 80
7.10.3. Map Area 81
7.10.4. Trigger Byte 81
7.10.5. Endian Swap 82
7.10.6. Offline Option 83
7.10.7. Live List 84
7.10.8. Data Exchange Control 84
7.11. Apply Configuration 86
7.12. To Use an Existing Configuration 88
7.13. To Use an Communicator Classic Configuration 89

8. PLC Configuration 91

8.1. Export I/O Data Map 91
8.2. Export Product GSD File 91
8.3. How to Use GSD File Modules 92

9. Verify Operation 94

9.1. Communicator Status Monitor 94
9.2. Communicator LED Indicators 96

10. Use Cases 97

10.1. Temperature Regulator - Modbus RTU Use Case 97
10.1.1. About the Use Case 97
10.1.2. Before You Begin 97
10.1.3. Choose Serial Protocol Type 98
10.1.4. Setup Serial Communication 98
10.1.5. Set Up the Node 99

10.1.6. Setup the Transactions 100
10.1.7. Check the I/O Data Map 102
10.2. AC Motor Drive - Custom Request/Response Use Case 103
10.2.1. About the Use Case 103
10.2.2. Before You Begin 103
10.2.3. Choose Serial Protocol Type 104
10.2.4. Setup Serial Communication 104
10.2.5. Create Transaction Templates 105
10.2.6. Setup Node and Transactions 109
10.2.7. Check the I/O Data Map 110
10.3. Barcode Scanner - Custom Produce/Consume Use Case 111
10.3.1. About the Use Case 111
10.3.2. Before You Begin 111
10.3.3. Choose Serial Protocol Type 112
10.3.4. Setup Serial Communication 112
10.3.5. Create Transaction Templates 113
10.3.6. Setup Node and Transactions 115
10.3.7. Check the I/O Data Map 116

11. Maintenance 117
11.1. Configuration File Handling 117
11.1.1. Export Configuration 117
11.1.2. Import Configuration 118
11.2. Clear and Revert Configuration 119
11.3. Firmware Management 120
11.3.1. View the Firmware Version 120
11.3.2. Firmware and Configuration Compatibility 120
11.3.3. Firmware File Validation 120
11.3.4. Update Firmware 121

12. Troubleshooting 122
12.1. Diagnostics 122
12.1.1. Serial RS-232/485 Data Monitor 122
12.1.2. Event Log 123
12.1.3. LED Status 124
12.2. Reset to Factory Settings 125
12.3. Firmware Upgrade Error Management 127
12.4. Support 129
12.4.1. Support Package 129

13. Technical Data 130
13.1. Technical Specifications 130

14. Reference Guides 131
14.1. About Input Registers and Holding Registers 131
14.2. Modbus Data Model 131
14.3. Modbus Transactions 131
14.4. Modus Exception Codes 132
14.5. ASCII Table 132
14.6. RS485/RS232 Electrical Connection 133
14.6.1. RS485 Typical Connection 133
14.6.2. RS232 Typical Connection 133

This page is intentionally left blank.

1. Preface

1.1. About This Document

This document describes how to install and configure Anybus® Communicator™.

For additional documentation and software downloads, FAQs, troubleshooting guides and technical support, please visit www.anybus.com/support.

1.2. Document Conventions

Lists

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

User Interaction Elements

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

Program Code and Scripts

```
Program code and script examples
```

Cross-References and Links

Cross-reference within this document: Document Conventions (page 1)

External link (URL): www.anybus.com

Safety Symbols

DANGER
Instructions that must be followed to avoid an imminently hazardous situation which, if not avoided, will result in death or serious injury.

WARNING
Instructions that must be followed to avoid a potential hazardous situation that, if not avoided, could result in death or serious injury.

CAUTION
Instruction that must be followed to avoid a potential hazardous situation that, if not avoided, could result in minor or moderate injury.

IMPORTANT
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.

Information Symbols

NOTE
Additional information which may facilitate installation and/or operation.

TIP
Helpful advice and suggestions.

1.3. Trademarks

Anybus® is a registered trademark of HMS Networks.

All other trademarks are the property of their respective holders.

2. Safety

2.1. Intended Use

The intended use of this equipment is as a communication interface and gateway.

The equipment receives and transmits data on various physical layers and connection types.

If this equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

2.2. General Safety

CAUTION

Ensure that the power supply is turned off before connecting it to the equipment.

CAUTION

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

CAUTION

To avoid system damage, the equipment should be connected to ground.

IMPORTANT

Using the wrong type of power supply can damage the equipment. Ensure that the power supply is connected properly and of the recommended type.

3. Preparation

3.1. Cabling

Have the following cables available:

- Ethernet cable for configuration
- PROFIBUS cable for connecting to the high level network
- Power cable

3.2. System Requirements

3.2.1. Supported Operating Systems

Operating System	Description
Windows 7 SP1, 32-bit	Windows 7 32-bit with Service Pack 1
Windows 7 SP1, 64-bit	Windows 7 64-bit with Service Pack 1
Windows 10 64-bit	Windows 10 64-bit

3.2.2. Supported Web Browsers

The Communicator built-in web interface can be accessed from the following standard web browsers.

- Google Chrome
- Microsoft Edge
- Mozilla Firefox

3.3. Tools

Have the following tools available:

- Flat-head screwdriver, size 5.5 mm
  Needed when removing the Communicator from DIN-rail.
- Flat-head screwdriver, size 3 mm
  Needed when connecting the cables to the 7-pin connector.

3.4. Support and Resources

For additional documentation and software downloads, FAQs, troubleshooting guides and technical support, please visit www.anybus.com/support.



TIP

Have the product article number available, to search for the product specific support web page. You find the product article number on the product cover.

3.5. HMS Software Applications

Download the software installation files and user documentation from www.anybus.com/support.

IPconfig

Use the HMS software application IPconfig and scan your network to discover and change the Communicator IP address and to access the Communicator built-in web interface.



NOTE

As an alternative, you can set a static IP address within the same IP address range as the Communicator IP address on the computer accessing the Communicator built-in web interface.



NOTE

IPconfig is only available for Windows.

3.6. Third-Party Software Applications

Microsoft Excel, or equivalent software application that supports the Office Open XML Workbook (xlsx) file format.

Needed to open and read the I/O data mapping file.

4. About Anybus Communicator

4.1. Serial Protocol Communication

4.1.1. Serial Protocol Types

The gateway features three distinct modes of operation for the subnetwork communication, called Modbus RTU, Custom Request/Response and Custom Produce/Consume.

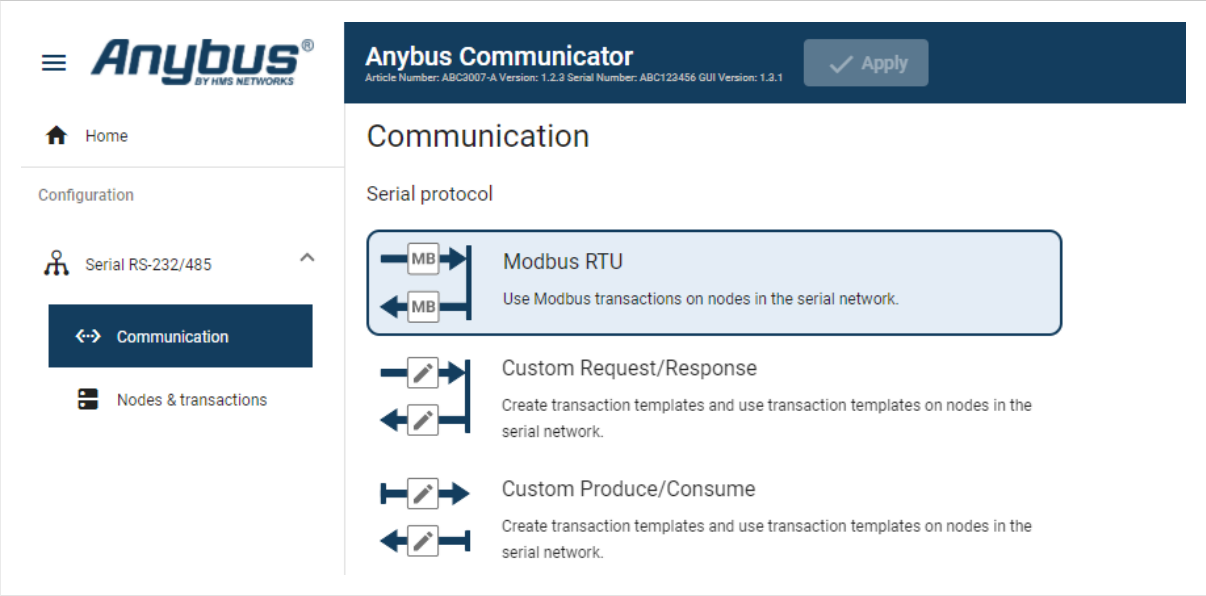


Figure 1. Communication, Serial protocol

Modbus RTU

By default the Communicator uses the Modbus RTU serial protocol.

The Communicator uses Modbus transactions defined by the Modbus standard.

The Communicator acts as a client on the subnetwork, and the serial communication takes place in a request/response fashion.

The nodes on the network are not permitted to issue messages unless they have been addressed by the Communicator first.

Custom Request/Response

In this mode, you can define your own serial transactions to handle a wide range of custom serial protocols.

The Communicator acts as a generic serial client on the subnetwork.

The serial communication takes place in a request/response fashion.

Custom Produce/Consume

In this mode, you can define your own serial transactions to handle a wide range of custom serial protocols.

The Communicator may consume and/or produce messages on the subnetwork.

There is no client-server relationship between the nodes on the network, messages are spontaneously produced or consumed when data is available.

4.1.2. Serial Protocol Building Blocks

The following building blocks are used to describe the subnetwork communication.

Node

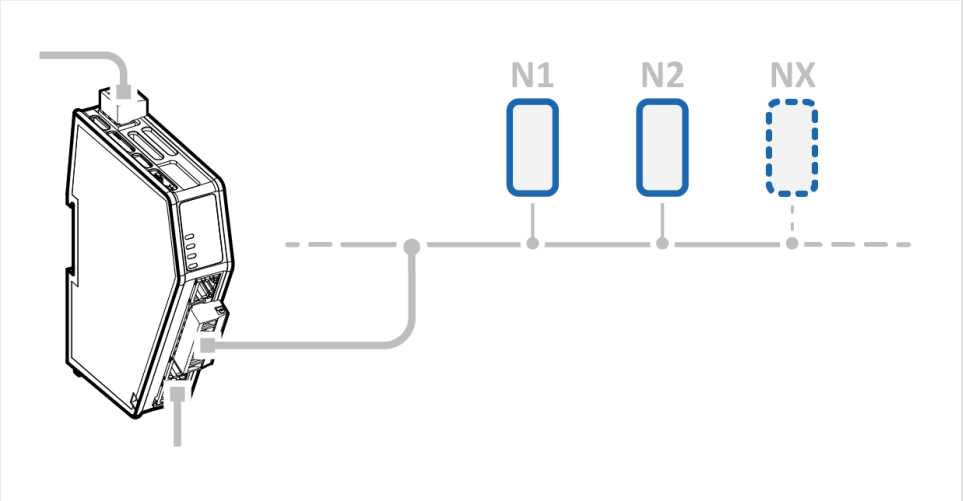


Figure 2. Nodes on a subnetwork

A node represents a single device on the subnetwork.

Each node can be associated with a number of transactions.

Nodes and Transactions

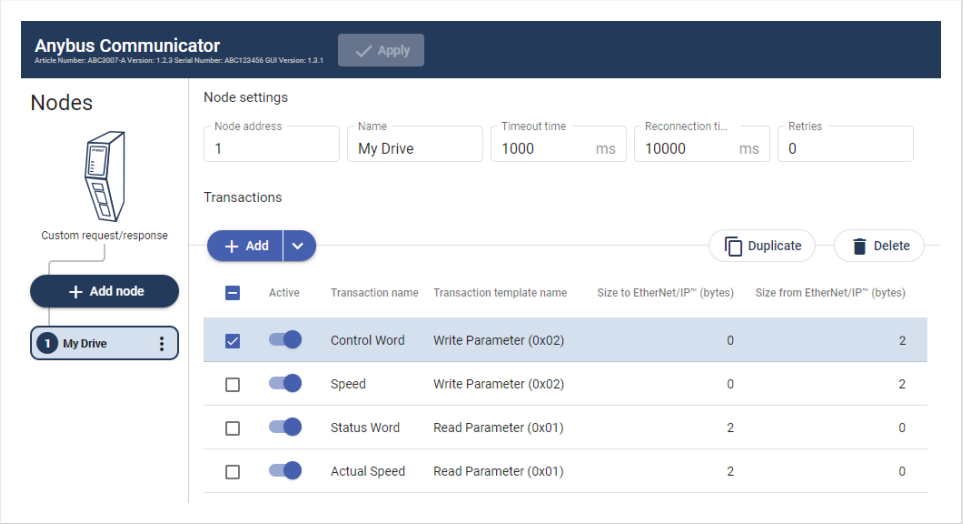


Figure 3. Node with Transactions

Transactions are based on standard Modbus RTU transactions (Modbus RTU serial protocol) or transactions templates (Custom Request/Response or Produce/Consume serial protocol) and define the data to be sent or received.

Each transaction has a number of parameters that need to be configured to define how data is to be sent / received.

Frame Fields

The Frame editor is used to design custom transaction templates.

The Frame editor with Frame fields is available when either the Custom Request/Response or Custom Produce/Consume serial protocol is enabled.

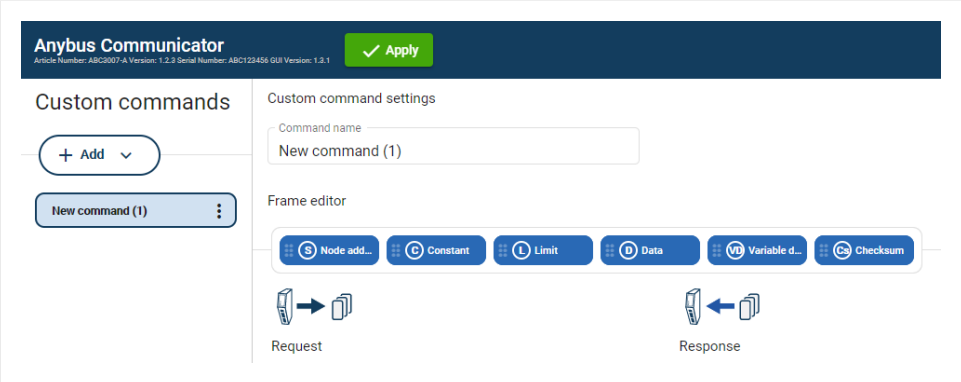


Figure 4. Frame editor

Frame fields are low level entities used to compose transactions.

A frame object can represent a:

- fixed value, a constant
- range of values, limit objects
- block of data or a calculated checksum

Transaction Templates

The Transaction templates are available when either the Custom Request/Response or Custom Produce/ Consume serial protocol is enabled.

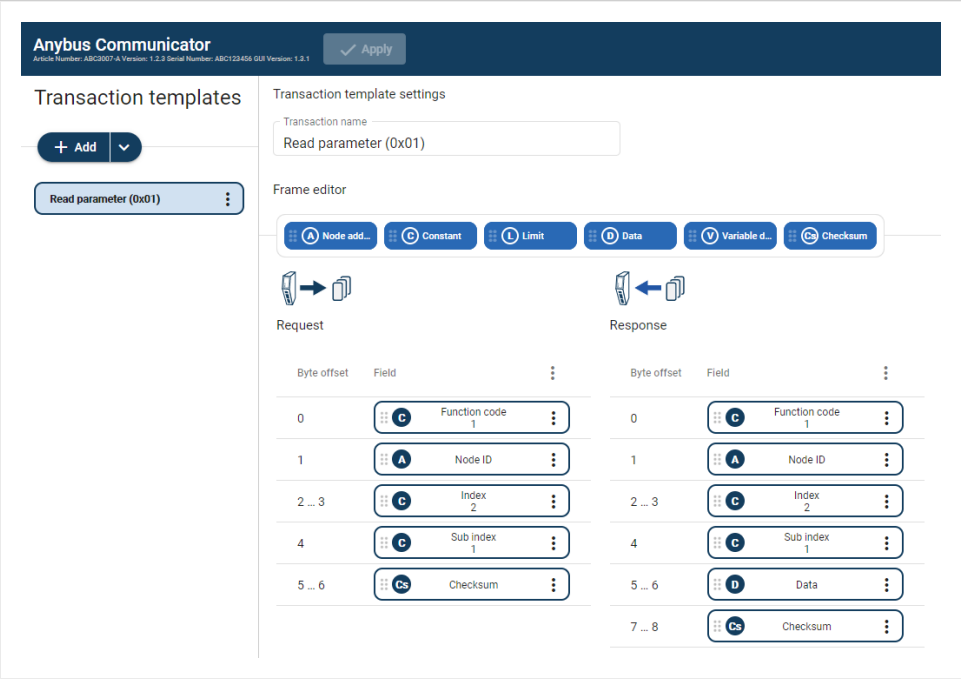


Figure 5. Frame editor, Request and Response

A transaction represents a complete serial telegram, and consists of a number of frame fields.

Each frame field is associated with a set of parameters controlling what is transmitted on the subnetwork.

The transaction templates are stored in the Communicator and can be reused multiple times.

Example 1. Common Read Transaction

If you have a common read transaction. Then you can create one single transaction template for the read transaction and reuse it multiple times times on your node(s).

If you have a function code in your protocol similar to a standard Modbus RTU transaction. Then you can create a transaction template based on the Modbus RTU transaction for the read operation. When you reuse the template on your node(s), you only have to change the function code each time it is used.

4.2. How the Communication Works

The Communicator enables communication, data exchange, between one or more server devices connected to a serial subnetwork and a client device connected to a high level network.

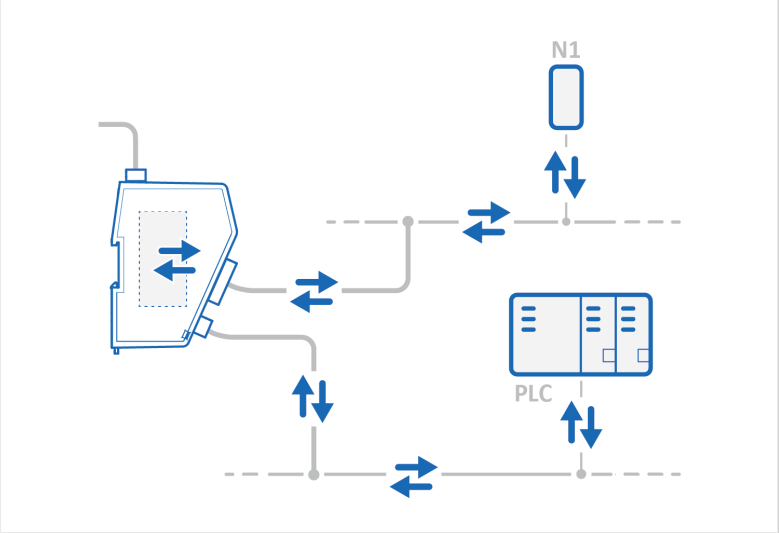


Figure 6. Process data traffic overview

For example:

- The client device can be a PLC controller or a PC.
- A server devices can be a sensor, scanner, industrial robot or sniffer.

The Communicator main task is to cyclically send the transactions that the server(s) are configured to execute, in order to request and transfer process data.

Request Process Data

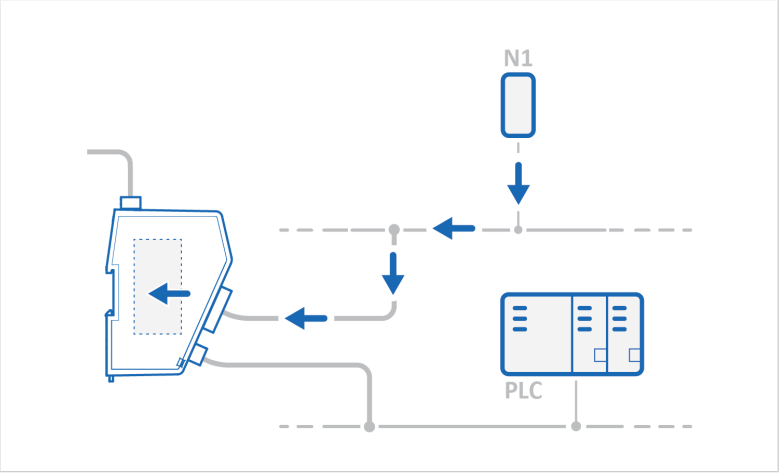


Figure 7. Process data traffic from nodes to client

Request process data from the serial subnetwork nodes, specified in the Communicator configuration, and make the process data available on the server interface and for the high level network client device.

Transfer Process Data

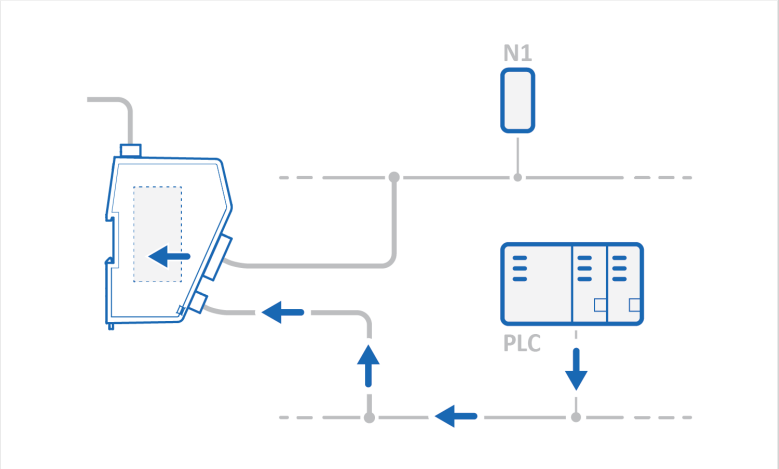


Figure 8. Process data traffic from client to nodes

Transfer process data from the high level network client device and make it available on the server interface and for the serial subnetwork nodes included in the configuration.

4.3. How the Data Exchange Works

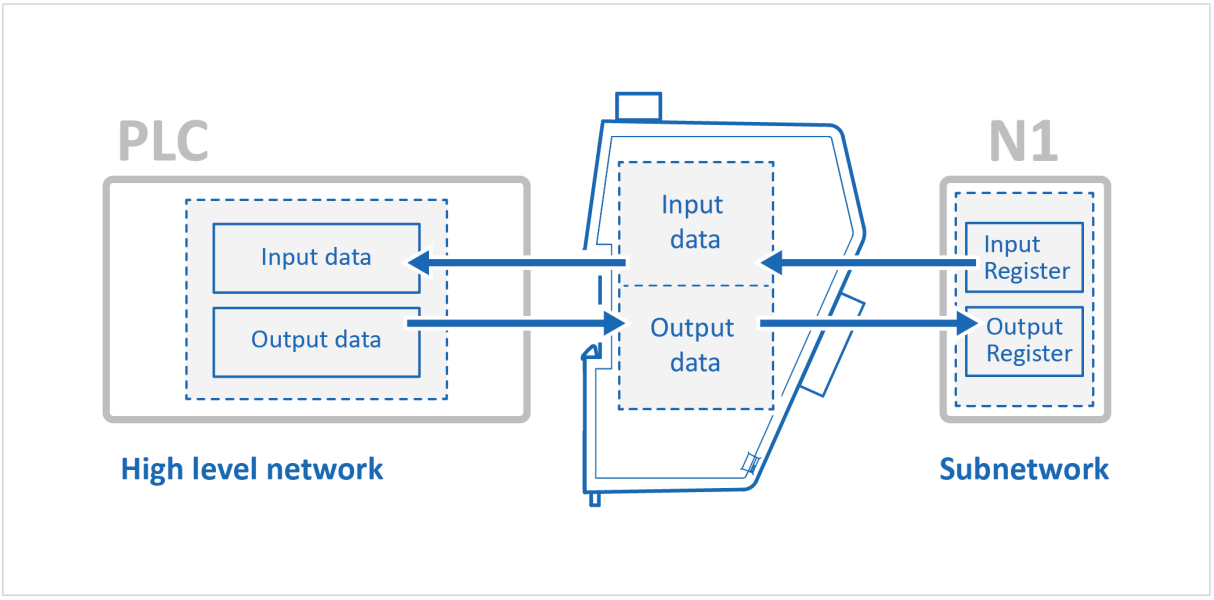


Figure 9. The Communicator internal memory areas

The data exchanged between the Communicator and the serial subnetwork and the high level network resides in the Communicator internal memory buffer.

To exchange data with the serial subnetwork, the high level network reads and writes data to the Communicator internal memory buffer.

The same memory locations are exchanged on the serial subnetwork.

The memory locations are specified when configuring the Communicator, using the Communicator built-in web interface.

The Communicator internal memory buffer is divided into two areas: Input data and Output data.

Input Data

This Input data area is read by the high level network.

The Communicator can handle up to 244 bytes input data.

Output Data

The Output data area is read/written by the high level network.

The Communicator can handle up to 244 bytes output data.

4.4. Data Integrity

A snapshot of the process data buffer between the Modbus Client and the server interface is used during the operation of executing all the transactions within one cycle.

When the cycle is completed, the process data available on the server interface is updated and a new snapshot is created for the next cycle.

5. Installation

5.1. External Parts

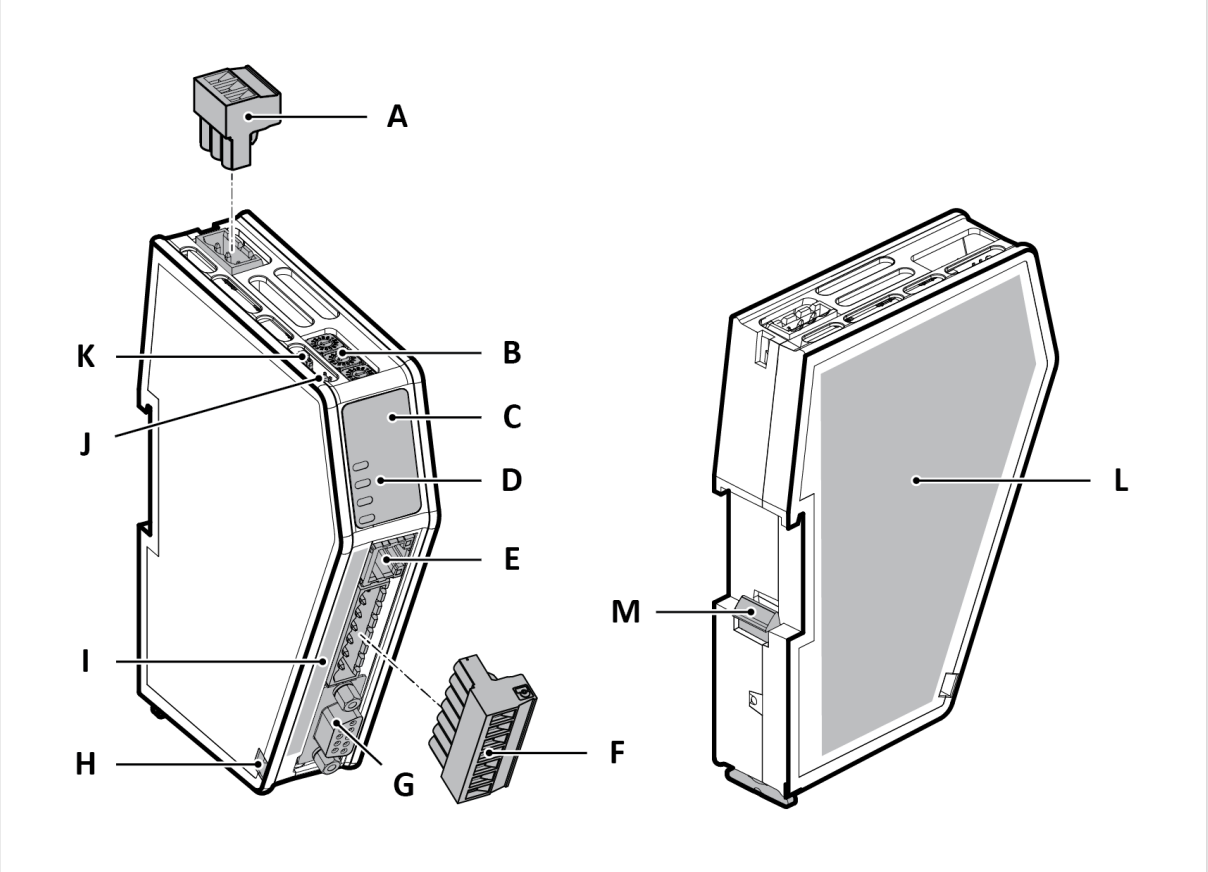


Figure 10. External parts

A. Power connector
B. Rotary Switch x 3
For PROFIBUS address
C. Label with LED designation
D. Status LEDs
E. Configuration port
F. 7-pin connector
G. PROFIBUS DSUB connector
H. Cable tie mount
I. Laser engraved connectors designation
J. Security switch
K. Factory reset button
L. Laser engraved label with product information
M. DIN rail locking mechanism

5.2. DIN Rail Mounting

IMPORTANT

The equipment must be electrically grounded through the DIN rail for EMC compliance. Make sure that the equipment is correctly mounted on the rail and that the rail is properly grounded.

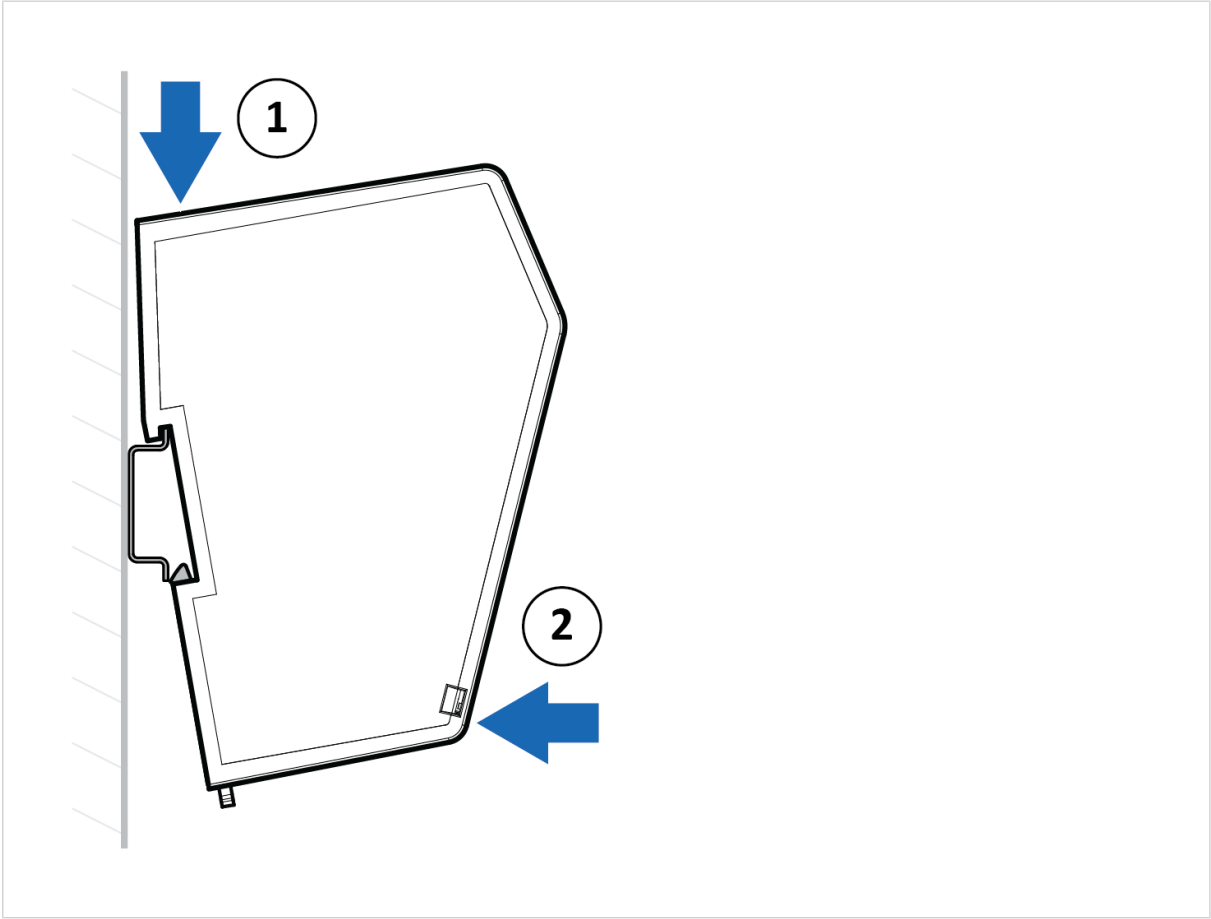


Figure 11. Attach the Communicator on the DIN rail

To attach the Communicator on the DIN rail:

1. Insert the upper end of the DIN rail clip into the DIN rail.
2. Push the bottom of the DIN rail clip into the DIN rail.

5.3. Connect to PROFIBUS Network

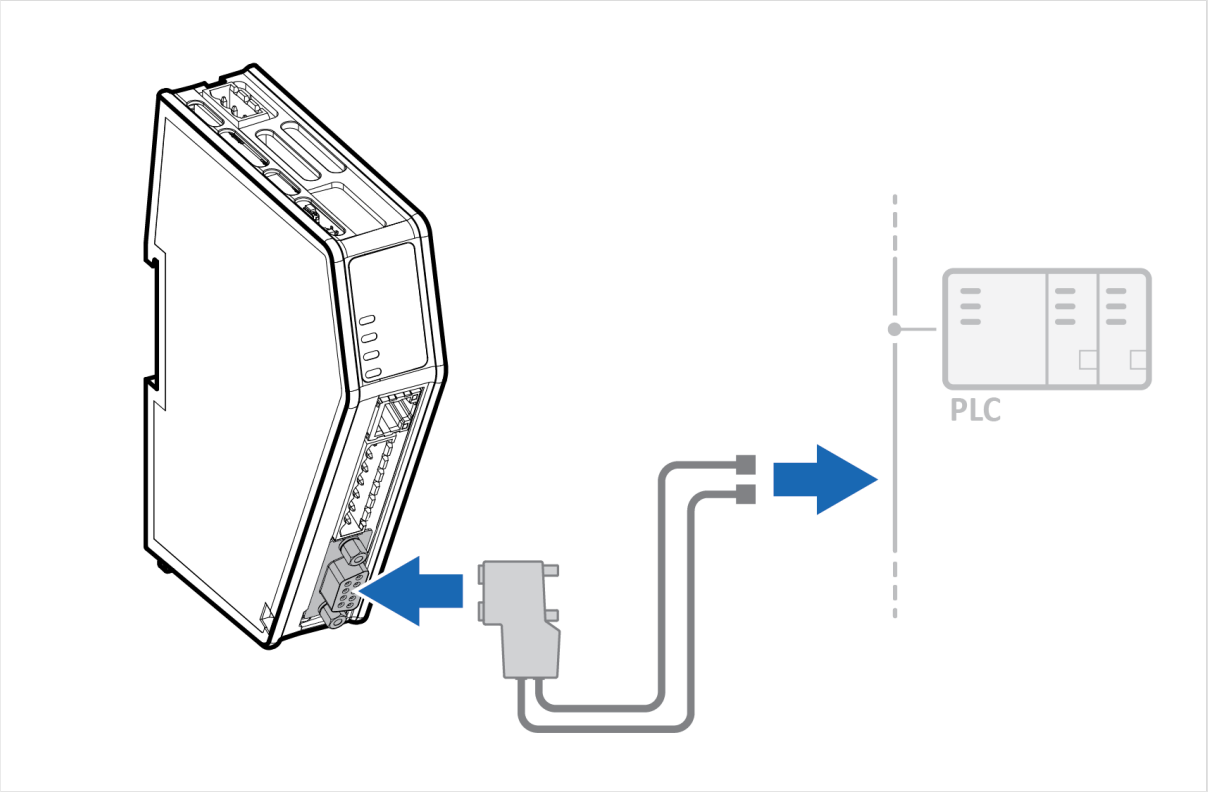
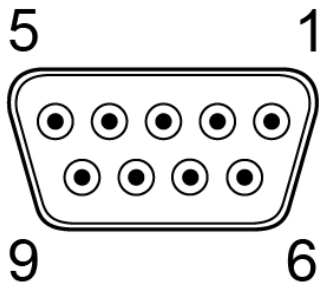


Figure 12. Connect to PROFIBUS network

1. Connect the Communicator to your PROFIBUS network.

PROFIBUS Connector	Pin	Description
	1	Shield
	2	Not used
	3	Line B
	4	RTS
	5	GND Bus
	6	+5 V Bus Out
	7	Not used
	8	Line A
	9	Not used
	Housing	PE

To Do Next

Connect the Communicator to the Serial RS-232/485 network, set the node address on the rotary switches and connect to power.

Check LED status, refer to Communicator LED Indicators (page 96).

5.4. Connect to Serial RS232/RS485 Subnetwork

NOTE
Use minimum 90 oC copper (Cu) wire only.

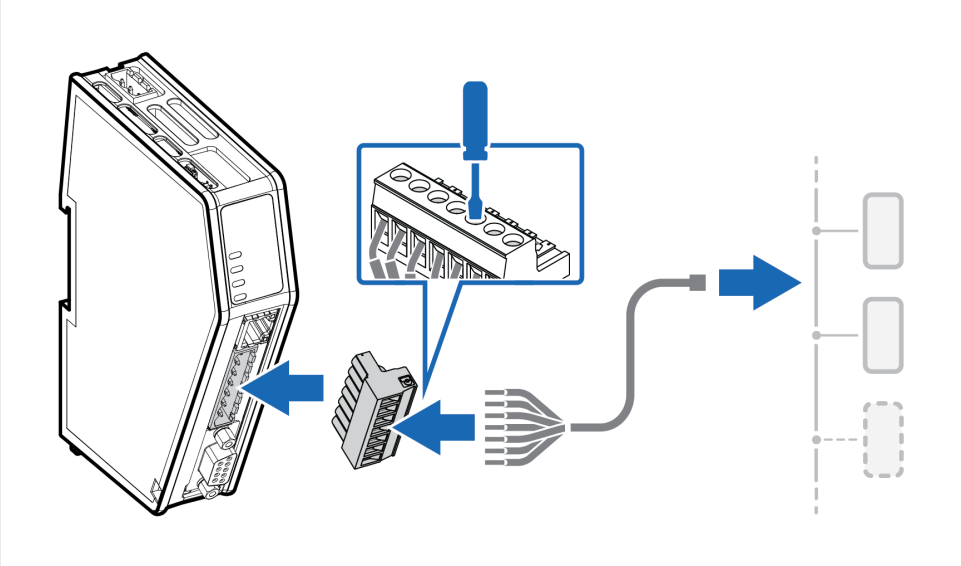
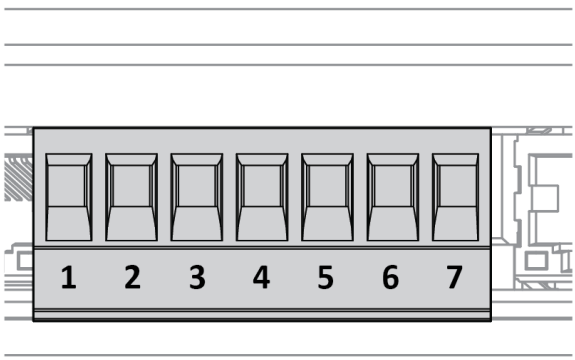


Figure 13. Connect to serial RS232/RS485 subnetwork

1. Insert the cable wires into the 7-pin connector and tighten the wire clamp screws.

7-pin connector	Pin	Signal
	1	RS232 Rx Input
	2	RS232 Tx Output
	3	Functional Earth (FE)
	4	Signal GND
	5	RS485+ B
	6	RS485- A
	7	+5 V OUT

2. Connect the 7-pin connector to the Communicator.
3. Connect the Communicator to your serial subnetwork.

To Do Next

Connect the Communicator to the PROFIBUS network, set the node address on the rotary switches and connect to power.

Check LED status, refer to Communicator LED Indicators (page 96).

5.5. Rotary Switch Settings

5.5.1. Rotary Switches Default Setting

By default, the value on the three rotary switches are set to **000**.

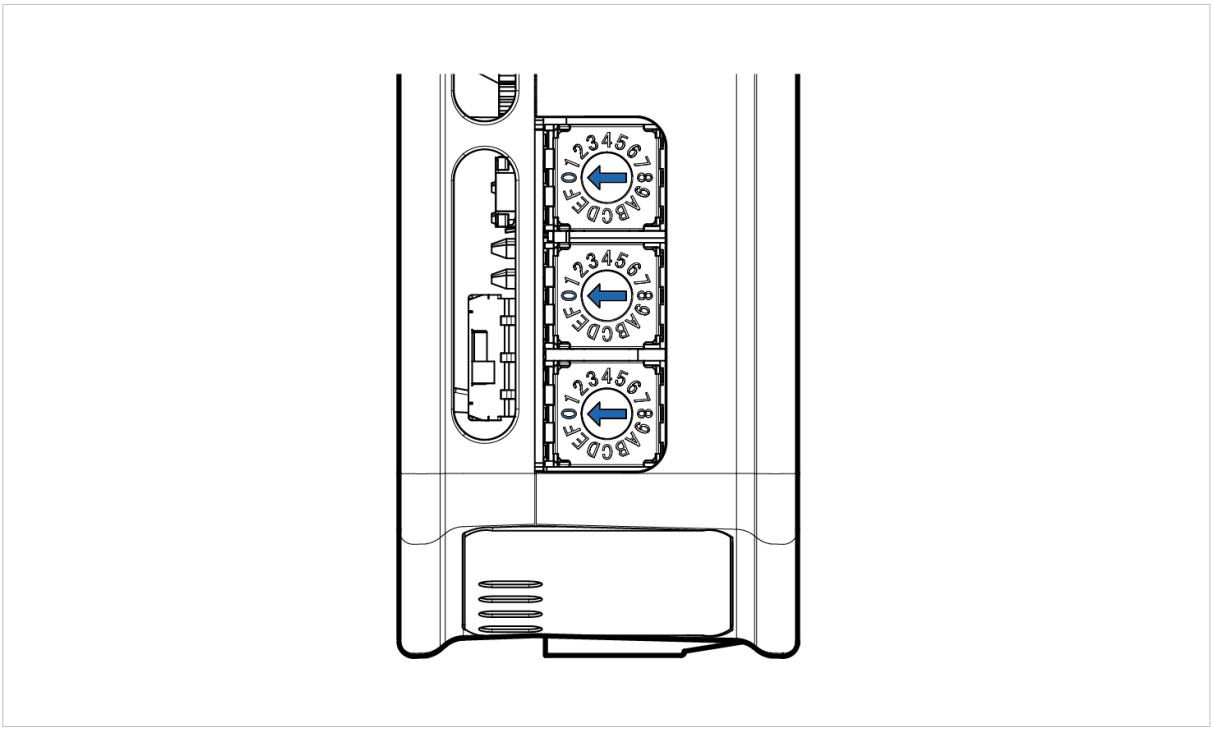


Figure 14. Rotary switches default setting **000**.

5.5.2. Set a Node Address with Rotary Switches

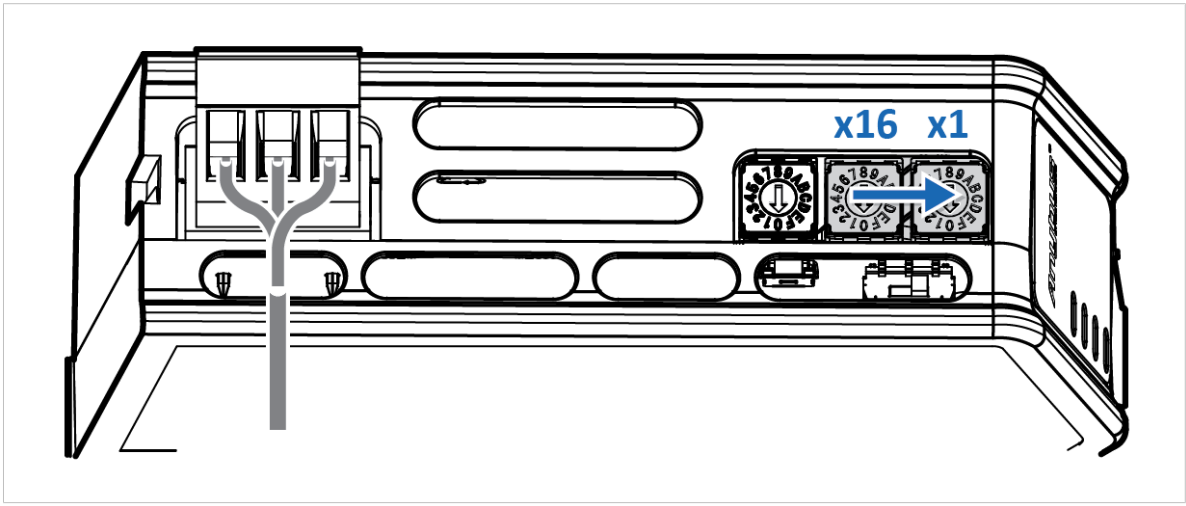
About the Rotary Switch Settings

To set a node address for the PROFIBUS server, use the rotary switches located on the top front of the Communicator .

TIP

Use Windows Calculator (or similar application) to convert between hexadecimal (hex) and decimal (dec).

- The default node address setting is 000.
- When the node address setting is 000, you can sett the node address via the Communicator built-in web interface.
  See PROFIBUS Address Settings (page 77).
- The node address values are set in hexadecimal (hex).
- Minimum value is 00.
- Supports PROFIBUS node addresses 0-126 Dec (0-7E Hex).
- Each node address may only occur once in the network.
- The node address is read from the center rotary switch x16 to the front rotary switch x1.
- The rear rotary switch is not used, ensure that it is set to 0.



Before You Begin

Ensure that the Communicator is disconnected from power.

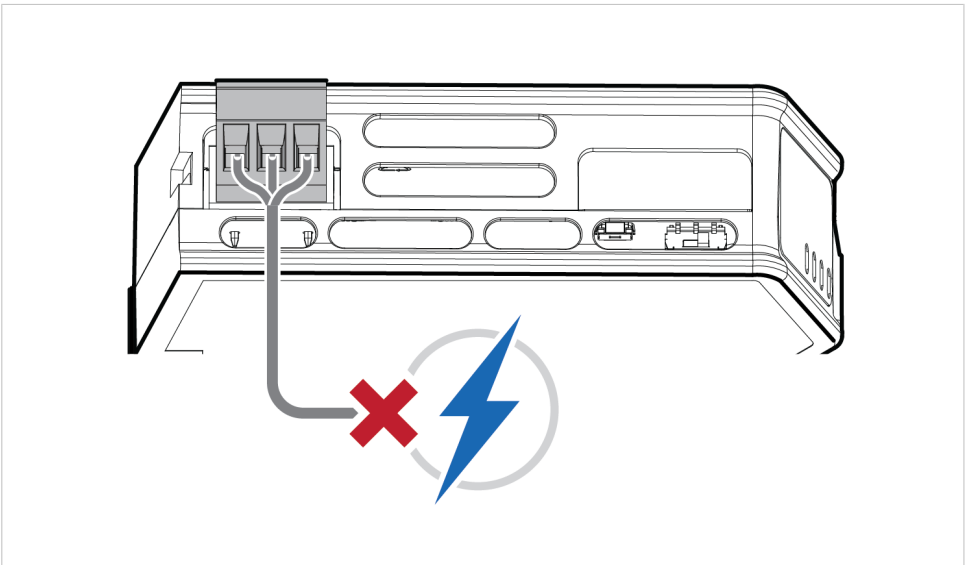
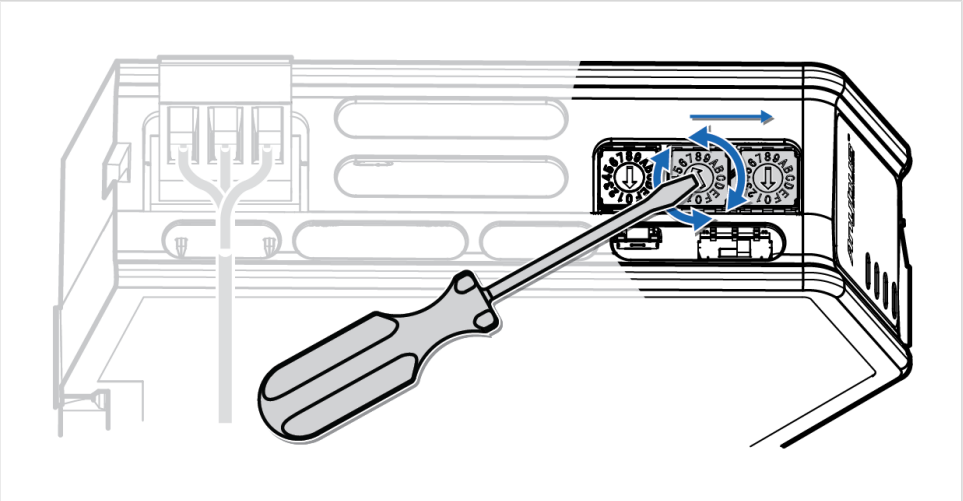


Figure 15. Disconnect Communicator from power

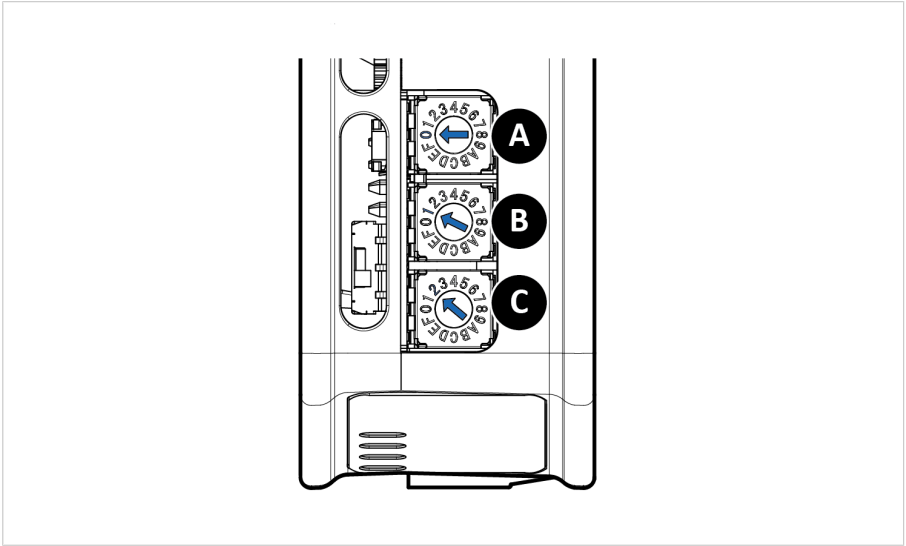
Procedure

Use a screwdriver to change the rotary switch position.

Ensure that the rotary switches engage correctly.



Example 2. To set the node address 12 hex = 18 dec

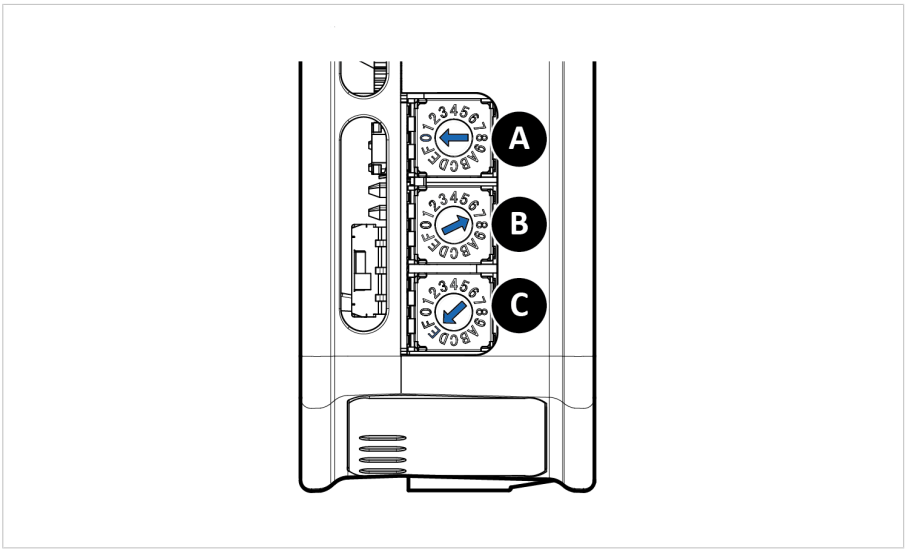


1. The rear rotary switch A is not used, ensure that it is set to 0.
2. Set the center rotary switch B to 1 hex.
3. Set the front rotary switch C to 2 hex.

The center rotary switch B 1 hex = 1 dec and the front rotary switch C 2 hex = 2 dec.

The node address expressed in decimal numbers is therefore 16 x 1 + 2 = 18.

Example 3. To set the node address 7E hex = 126 dec



- The rear rotary switch A is not used, ensure that it is set to 0.
- Set the center rotary switch B to 7.
- Set the front rotary switch C to E.

The center rotary switch B 7 hex = 7 dec and the front rotary switch C E hex = 14 dec.

The node address expressed in decimal numbers is therefore 16 x 7 + 14 = 126.

To Do Next

Connect the Communicator to power. See Connect to Power (page 22).

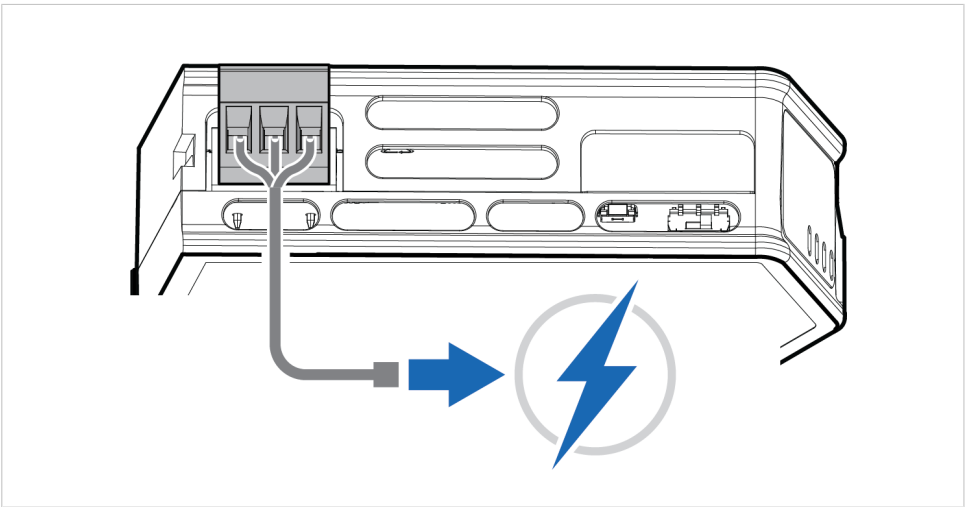


Figure 16. Connect Communicator to power

Result

The set node address is active as soon as the Communicator is powered on.

NOTE

Changing the address settings on the rotary switches during operation is ignored. For a new address to take effect, power cycle the Communicator .

5.6. Connect to Power

CAUTION
Ensure that the power supply is turned off before connecting it to the equipment.

IMPORTANT
Using the wrong type of power supply can damage the equipment. Ensure that the power supply is connected properly and of the recommended type.

IMPORTANT
Before you connect the Communicator to power, ensure that the PROFIBUS address is set on the rotary switches. If the PROFIBUS address is set when the Communicator is operating, you must power cycle the Communicator for the new setting to take effect.

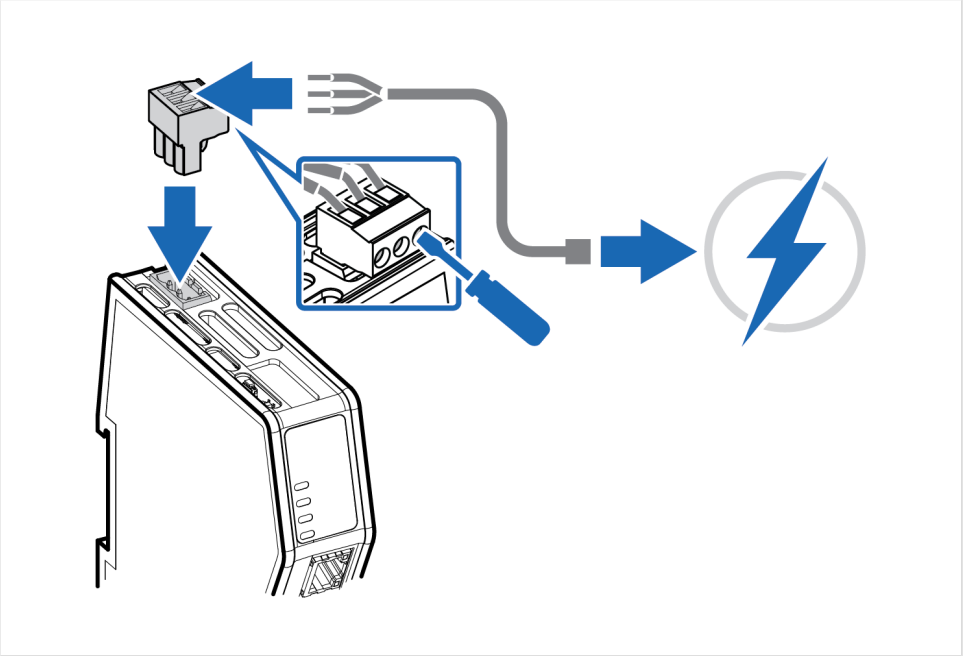
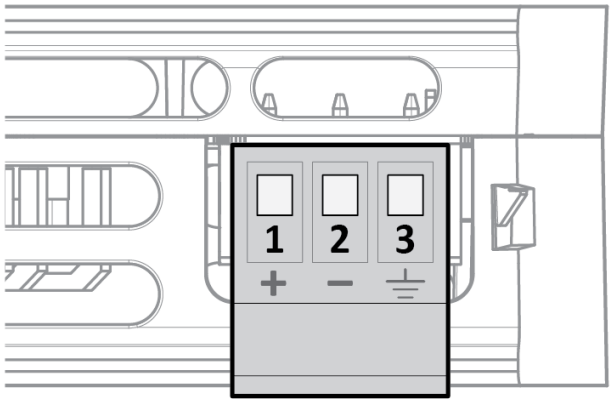


Figure 17. Connect to power

1. Insert the cable wires to the terminal block and tighten the wire clamp screws.

Power port	Pin	Description
	1	12-30 VDC Power Connector
	2	Ground (GND)
	3	Functional Earth (FE)

2. Connect the terminal block to the Communicator.
3. Connect the Communicator to a power supply.
4. Turn on the power supply.

To Do Next

Check LED status, refer to Communicator LED Indicators (page 96).

5.7. Security Switch

NOTE

After completing the configuration of the Communicator, lock the security switch to prevent unauthorized access to the Communicator built-in web interface.

When the security switch is in its locked position, the Communicator built-in web interface can not be accessed and the Communicator can not be configured using the built-in web interface. Network specific parameters, configured via the PLC is still available.

To Lock and Unlock the Security Switch

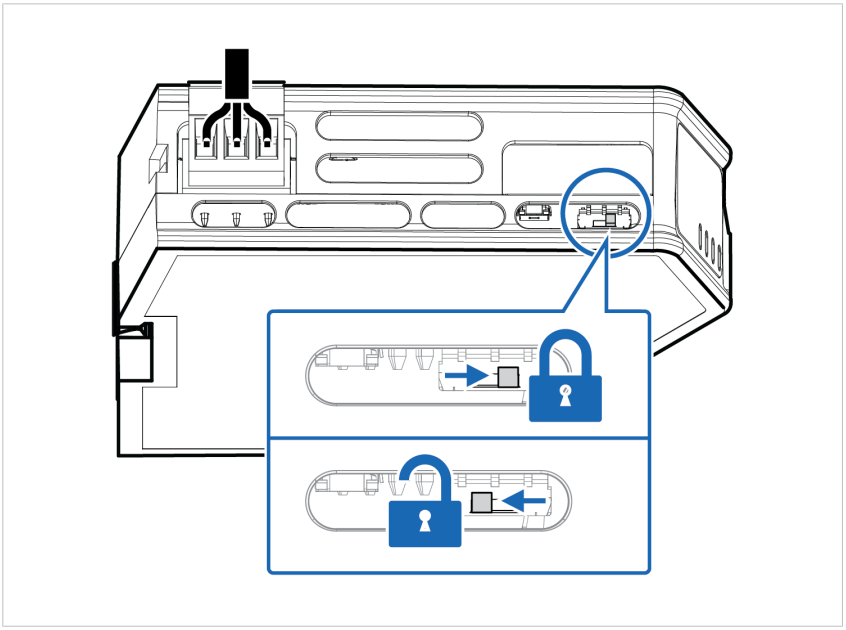


Figure 18. Security switch in locked and unlocked position

Use a pointed object, such as a ballpoint pen.

- To **lock** the security switch, push the toggle towards the **Communicator front**.
- To **unlock** the security switch, push the toggle towards the **Communicator back**.

Security Switch Status LED

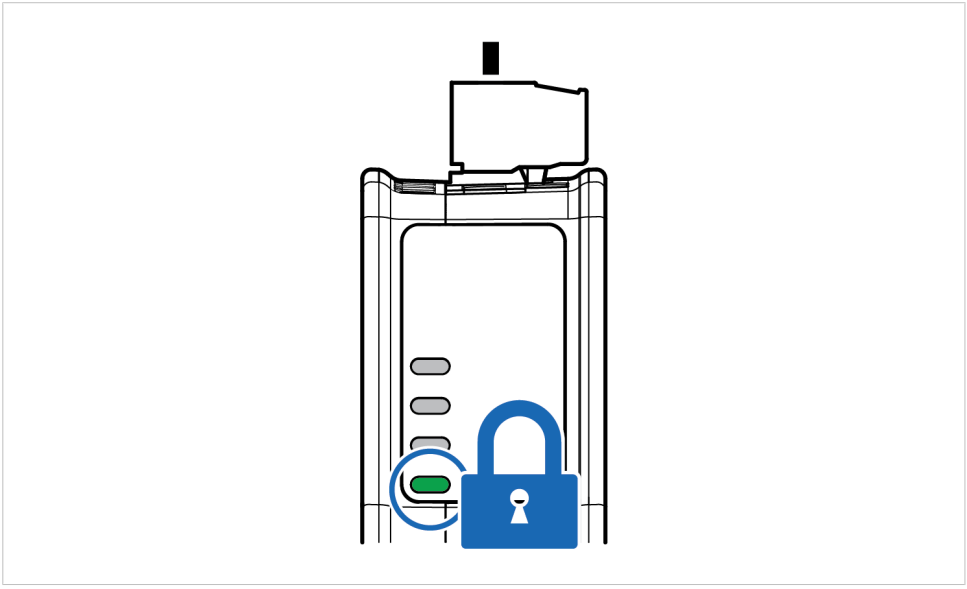


Figure 19. Security switch locked status LED

When the security switch is in its:

- locked position, the security switch status LED turn solid green.
- unlocked position, the security switch status LED is turned off.

5.8. Lock the Cables

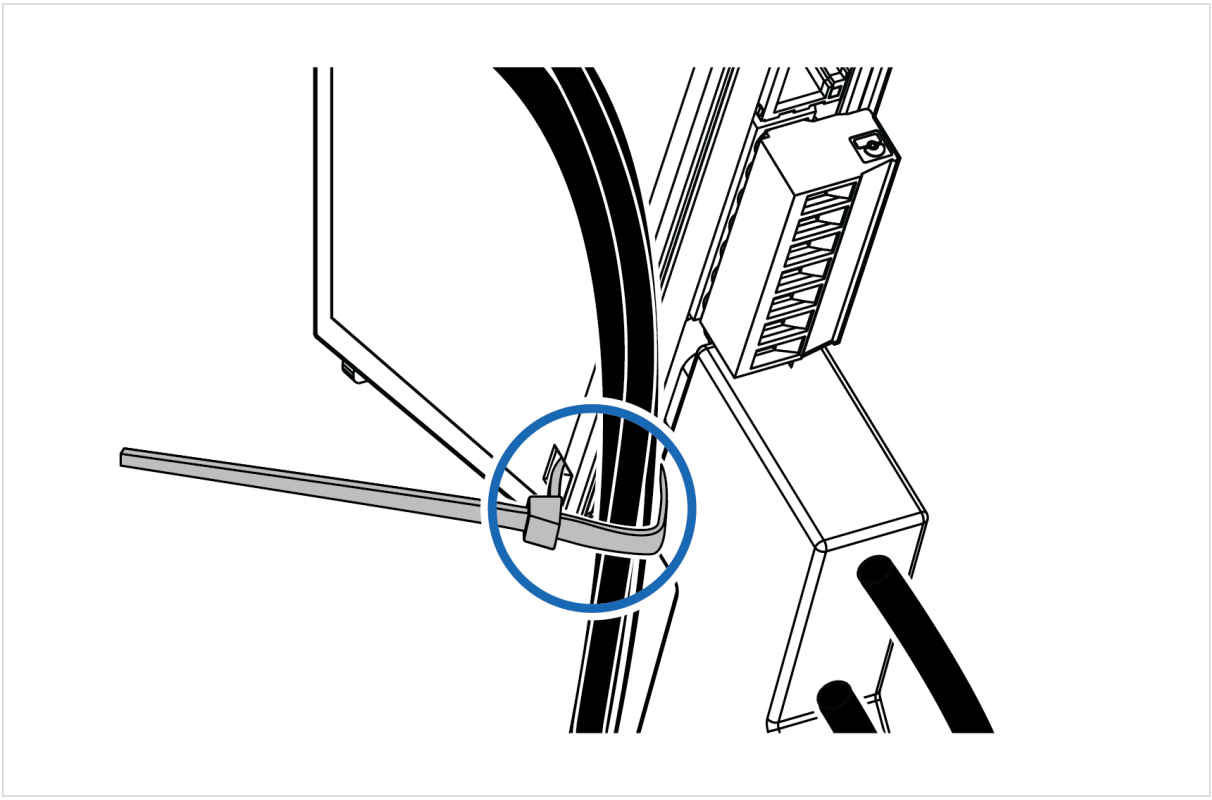


Figure 20. Lock the cables

To strain relieve the cables, place a cable tie in the holder and lock the cables.

5.9. DIN Rail Demount

Before You Begin

IMPORTANT

Be careful when removing the Communicator from the DIN-rail. If not removed properly, the DIN rail locking mechanism and the product cover can break.

Have a flat-blade screwdriver, size 5.5 mm, available.

Procedure

Remove the Communicator from the DIN Rail:

1. Insert the screwdriver into the Communicator DIN rail locking mechanism.
2. To unlock the Communicator DIN rail locking mechanism, turn the screwdriver clockwise.

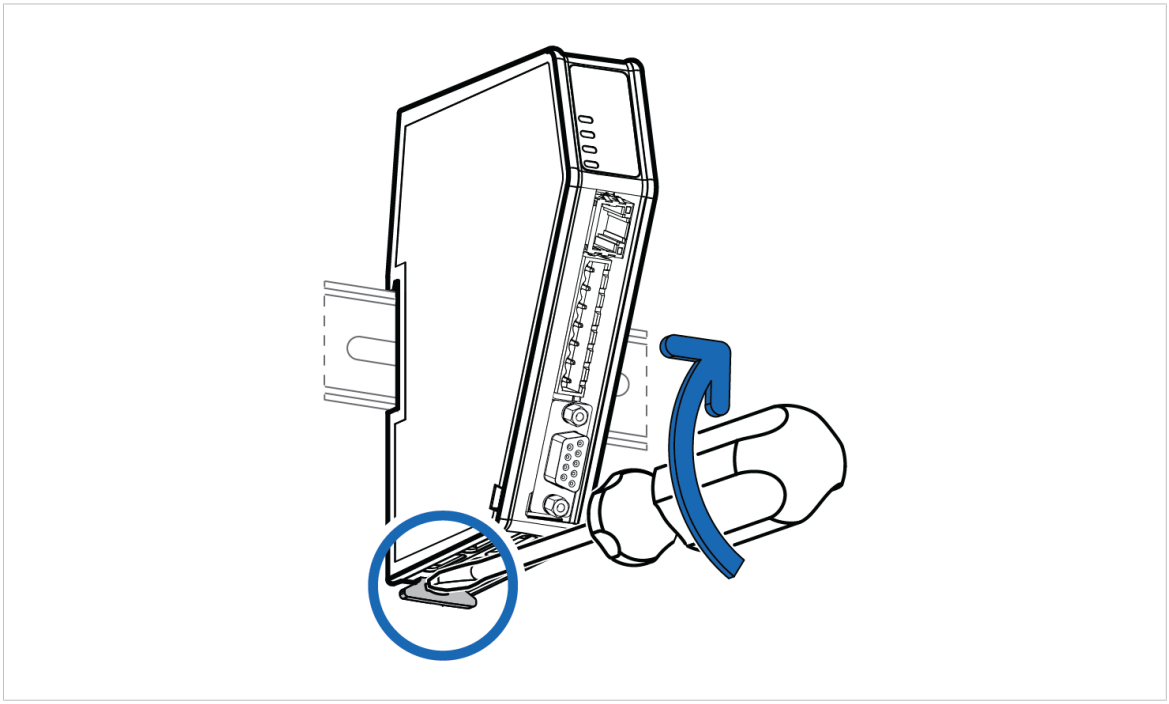


Figure 21. Unlock the Communicator

3. Hold the screwdriver in the DIN rail locking mechanism while you unhook the Communicator from the DIN rail.

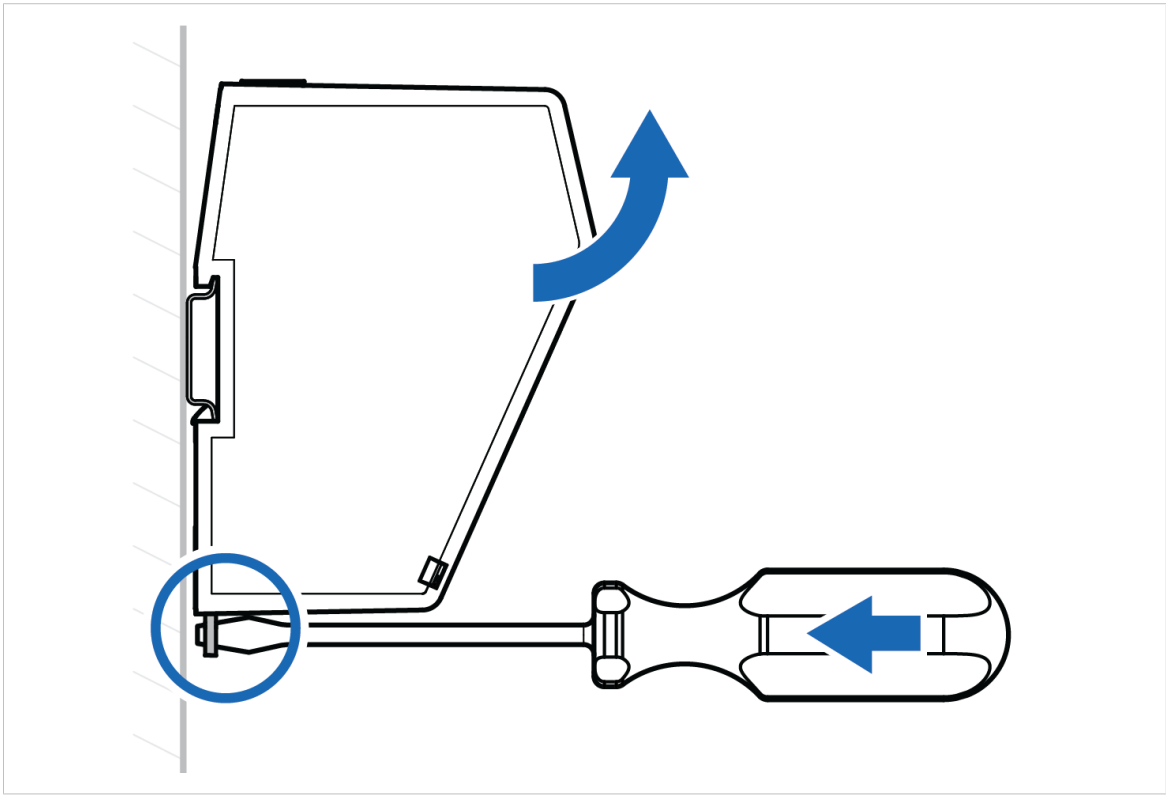


Figure 22. Unhook the Communicator

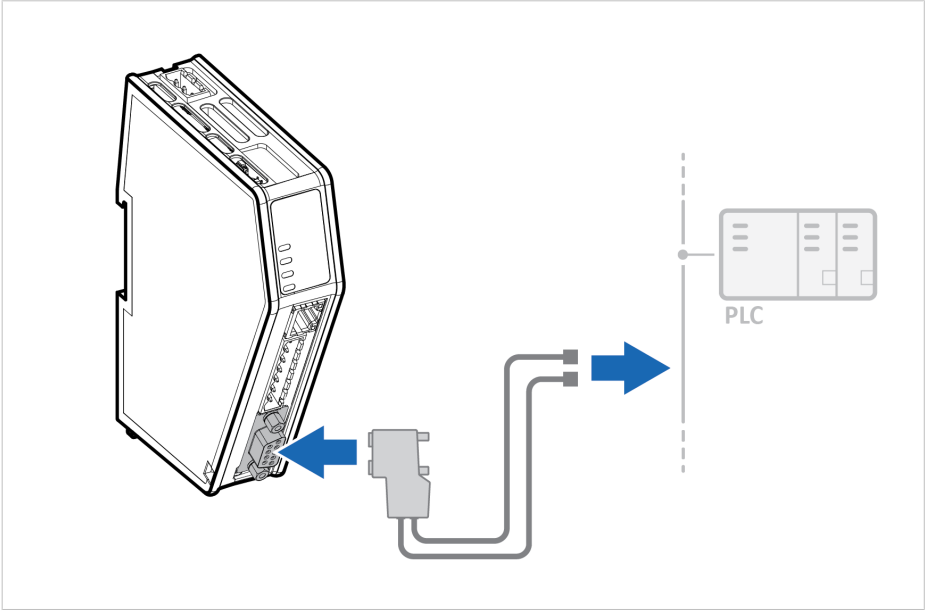
6. Configuration Quick Guide

This section is intended to give you a short overview of the tasks you need to perform to configure the Communicator.

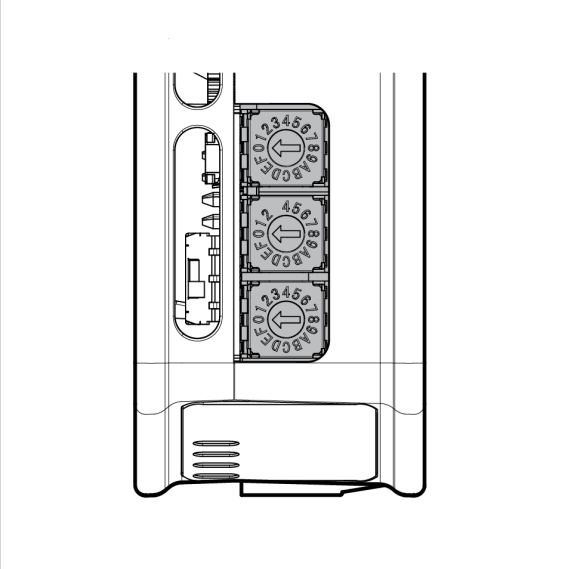
For detailed information, please refer to Communicator Configuration (page 39) .

6.1. Prepare Configuration

1. **Connect Communicator to the PROFIBUS network.**



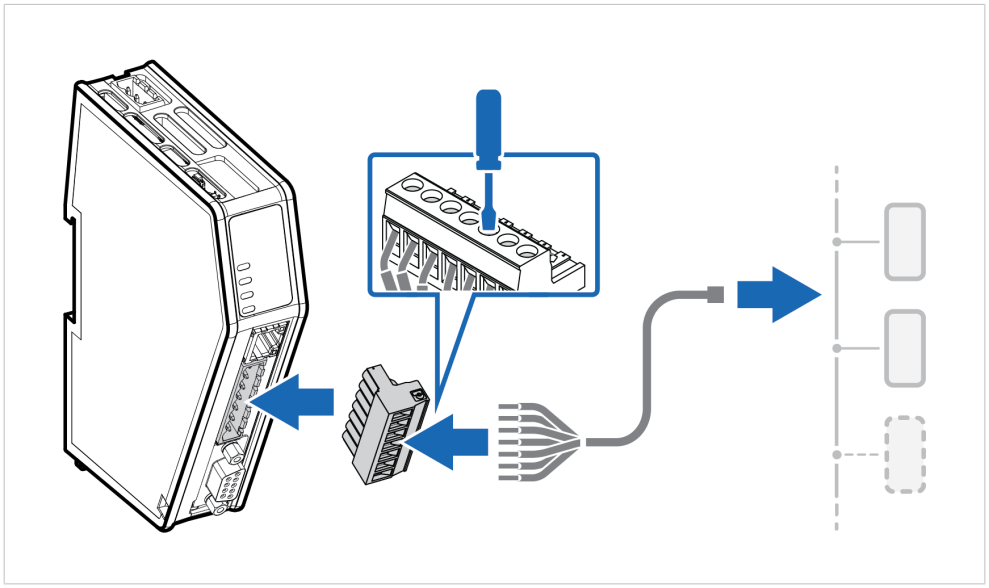
2. **Set the desired PROFIBUS address on the Rotary Switches.**



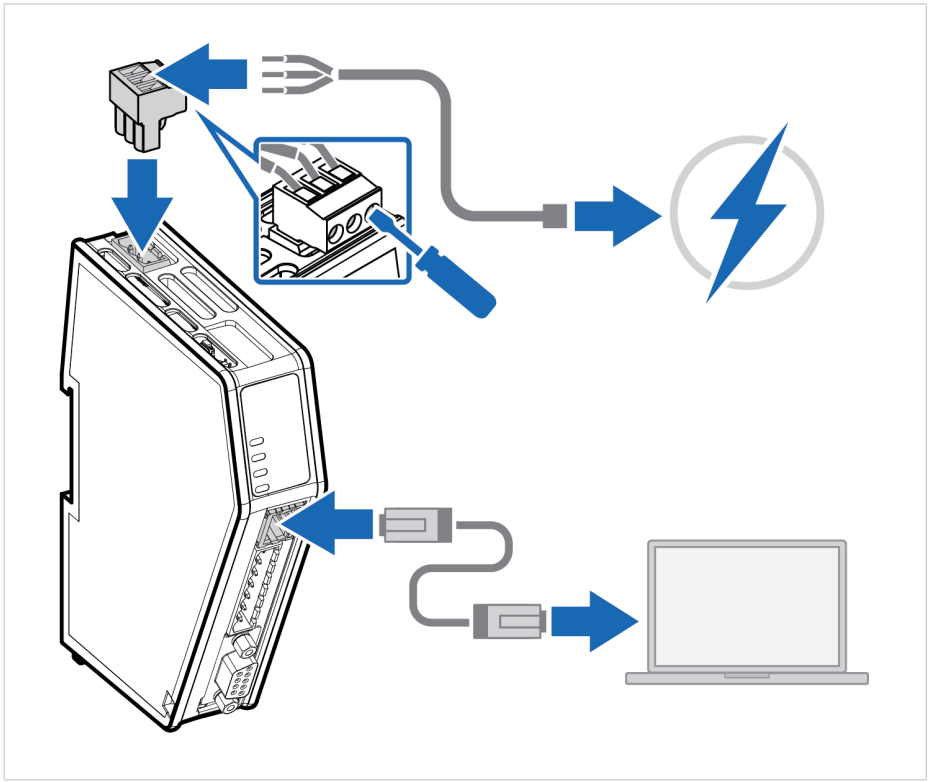
The default setting is 0.

For more info see Rotary Switch Settings (page 17).

3. **Connect the Communicator to the serial RS232/RS485 subnetwork**



4. **Connect to PC and power**



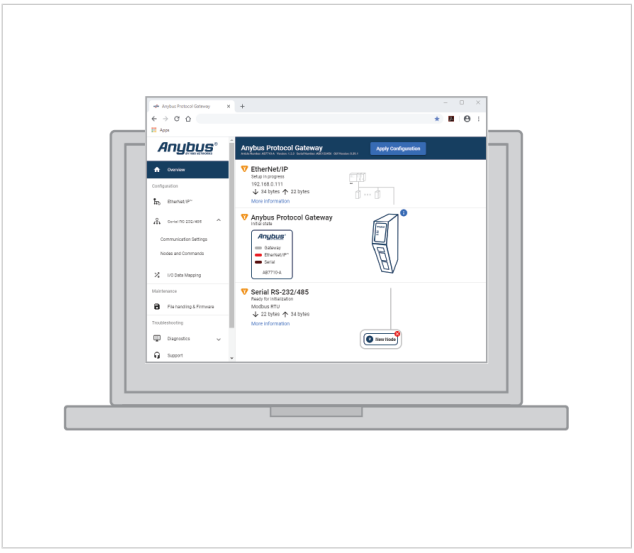
a. Connect an Ethernet cable between the Communicator configuration port and your PC.
b. Connect the Communicator to a power supply.

5. **Find the Communicator on your PC**

The Communicator default IP address is 192.168.0.10.

Option 1	Option 2
 On the PC accessing the Communicator built-in web interface, set a static IP address within the same IP address range as the Communicator IP address.	 Change the IP address on the Communicator configuration port to one within the same IP address range as your PC. Use the software application HMS IPconfig to find the Communicator default IP address on your PC. Download the installation files and user documentation from www.anybus.com/support.

6. **Access the Communicator built-in web interface**



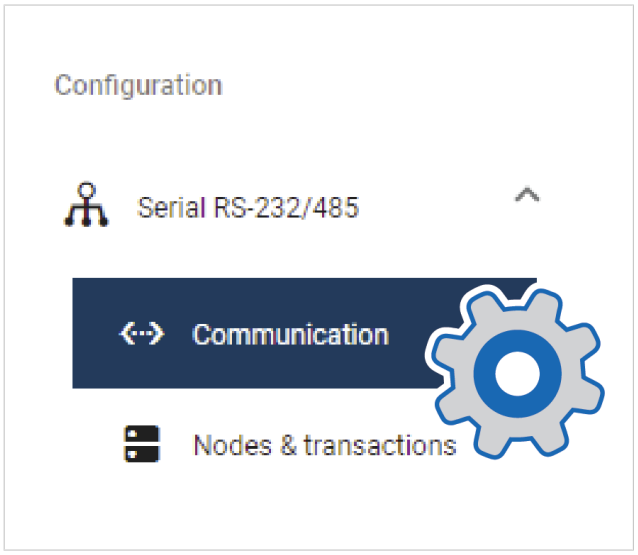
Open the Communicator built-in web interface in HMS IPconfig or enter the Communicator IP address in your web browser.

The Communicator built-in web interface overview page opens in your browser.

6.2. Setup New Configuration

Follow these steps to setup a new Communicator configuration.

1. **Subnetwork configuration**

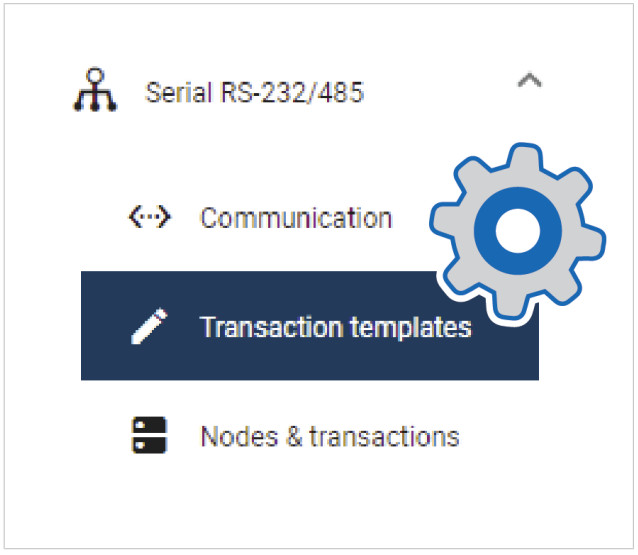


On the **Communication** page:

a. Select a serial protocol: **Modbus RTU** (default), **Custom Request/Response** or **Custom Produce/Consume**.
For information about the serial protocol types, see Serial Protocol Types (page 6).

b. Configure the basic settings Physical standard:, Baud rate, Data bits, Parity and Stop bits.

2. **Create Transaction Templates**

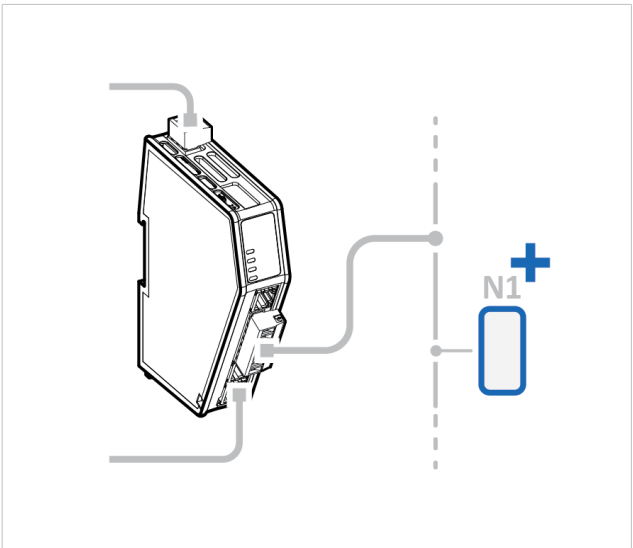


This step applies when the serial protocol **Custom request/response** is selected.

On the **Transaction templates** page: Add a transaction template and configure the template settings.

Repeat until you have added and configured all your transaction templates.

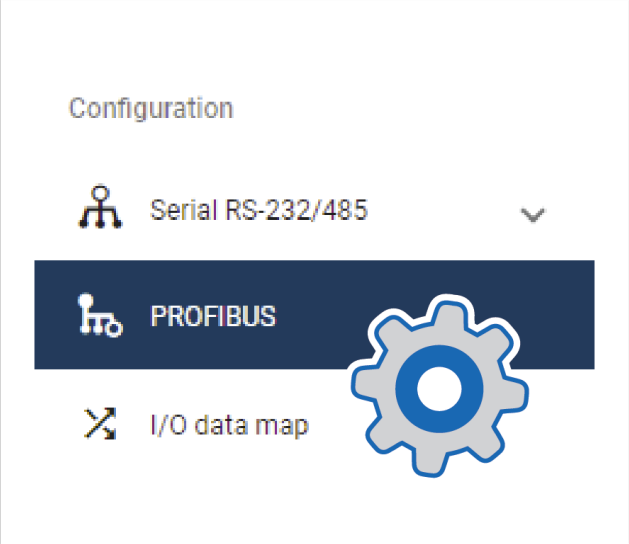
3. **Add Nodes and Commands**



On the **Nodes & commands** page:

a. Add a node and configure the node settings.
b. Add commands to the node and configure the command settings.
c. Repeat until you have added and configured all your nodes.

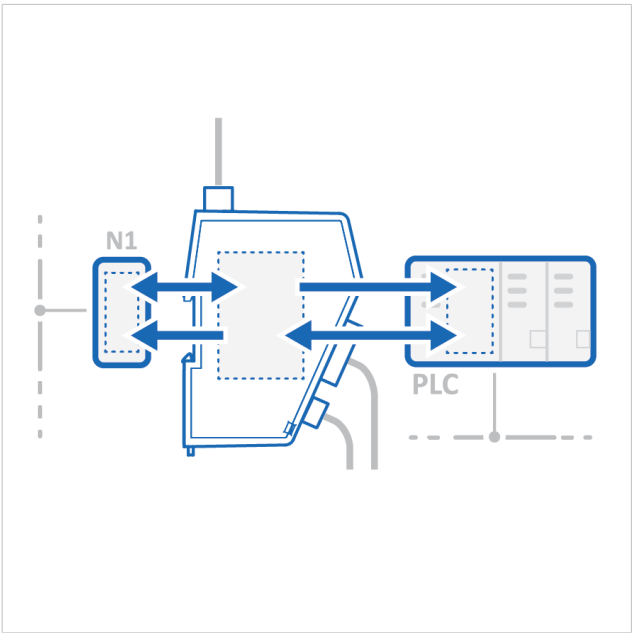
4. **High level network configuration**



On the **PROFIBUS** page:

1. Use Automatic I/O sizes provided by the subnetwork or choose to set them manually.
2. If the rotary switches are set to 0, set the PROFIBUS address.
3. Apply the PROFIBUS settings.

5. **I/O Data Mapping**

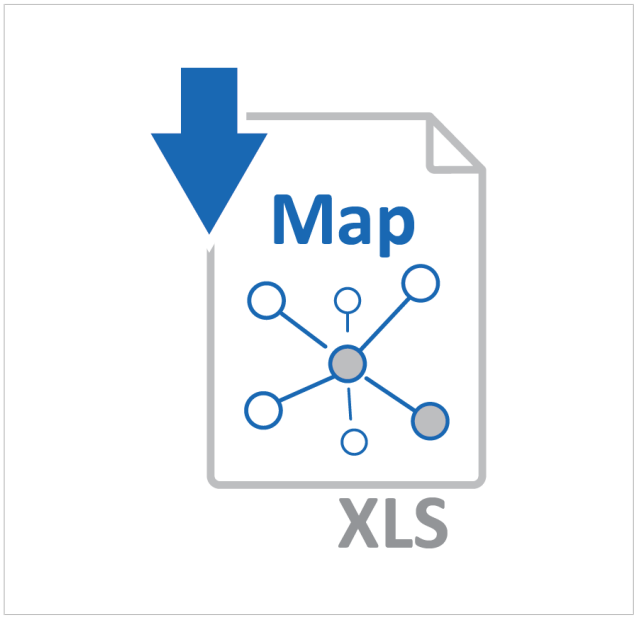


The commands you added to the nodes are automatically mapped to the Communicator internal memory area.
View the added nodes and commands on the **I/O data map** page.

6.3. PLC Configuration

In the Communicator built-in web interface:

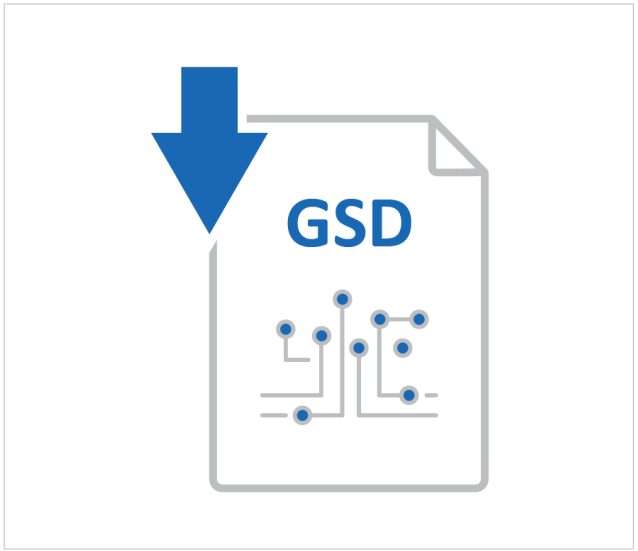
1. **Export I/O data map**



When you configure the communication between the Communicator and the PLC, you can use the I/O data map as a specification to ensure that the commands match.

On the **I/O data map** page: You can download the I/O data mapping in a spreadsheet to your PC.

2. **Download GSD File**

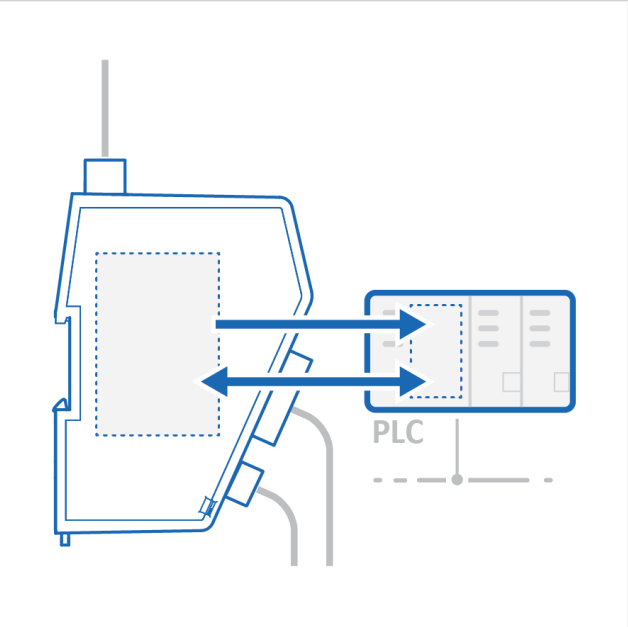


Option if the PLC program requires a GSD (General Station Description) file.

On the **PROFBUS** page: Download the GSD file to your PC.

In the PLC program:

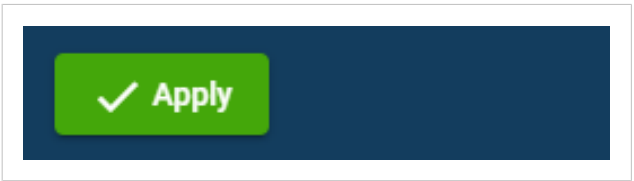
1. **Import product file**
   Option if the PLC program requires a GSD (General Station Description) file.
   Import the GSD file into your PLC project.
2. **Configure the communication**



Configure the PLC to communicate with the Communicator according to the I/O data map created in the Communicator.

6.4. Verify Operation

1. **Apply the configuration**



When you have completed and verified the configuration, click **Apply** for the settings to take effect.

2. **Verify status and LED indications**



On the **Home** page:
Monitor the Communicator, network and node status.
You can also view the Communicator LED indications remotely.

3. **Verify and monitor communication**



In **Diagnostics**, use the:

- **Serial RS-232/485** page to verify that the serial commands are sent and received by the Communicator.
- **Event log** page to detect failures and unexpected behavior over time.

7. Communicator Configuration

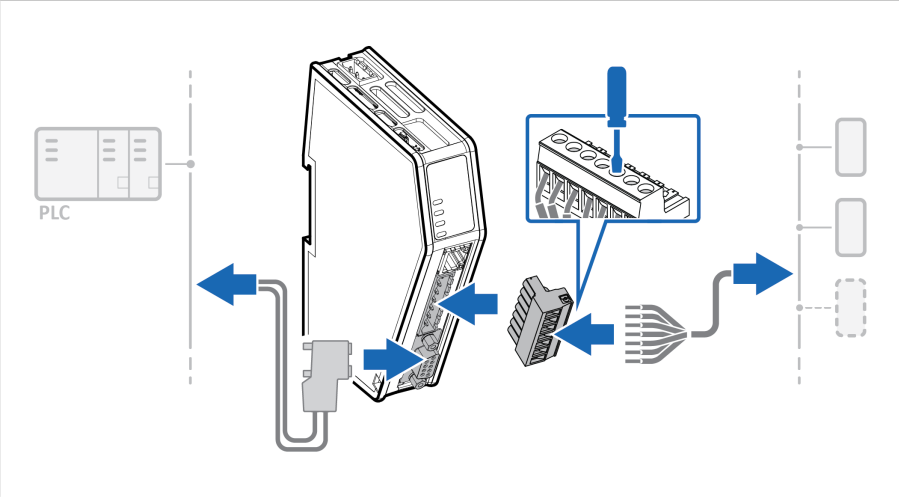
This section is intended to give you detailed information about the tasks you need to perform to setup a new Communicator configure.

For a more brief overview of the configuration steps, please refer to the Configuration Quick Guide (page 29).

7.1. Connect the Communicator

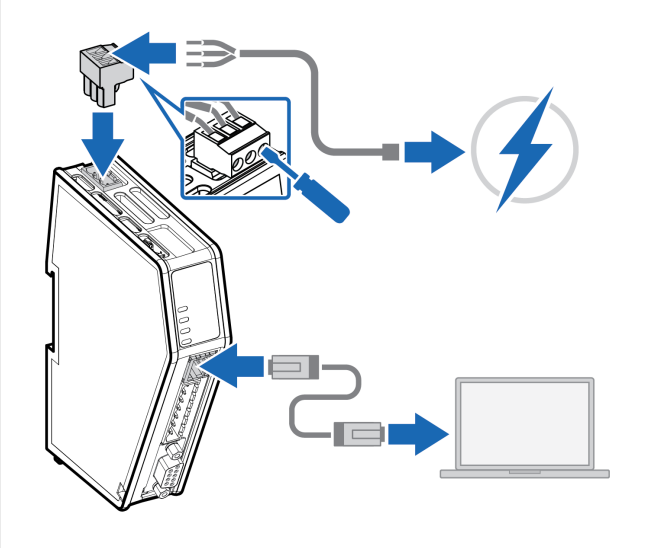
Procedure

Connect to High Level Network and Subnetwork



- Connect the Communicator to the subnetwork.

Connect to PC and Power



1. Connect an Ethernet cable between the Communicator and your PC.
2. Connect the Communicator to a power supply.

7.2. Access the Built-In Web Interface From HMS IPconfig

Before You Begin

Download the software application HMS IPconfig installation files and user documentation from www.anybus.com/support.

NOTE
The Communicator default IP address is 192.168.0.10.

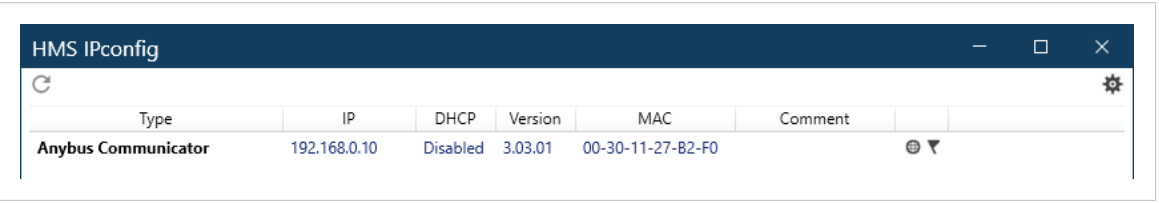
NOTE
To access the Communicator built-in web interface, ensure that Port 80 TCP is open in your Firewall. This applies to any Firewall between the web browser and the gateway.

NOTE
To access the Communicator built-in web interface from HMS IPconfig, ensure that port Port 3250 UDP is open in your PC Windows Firewall.

NOTE
Ensure that the security switch is unlocked. HMS IPconfig cannot configure the Communicator if the security switch is locked.

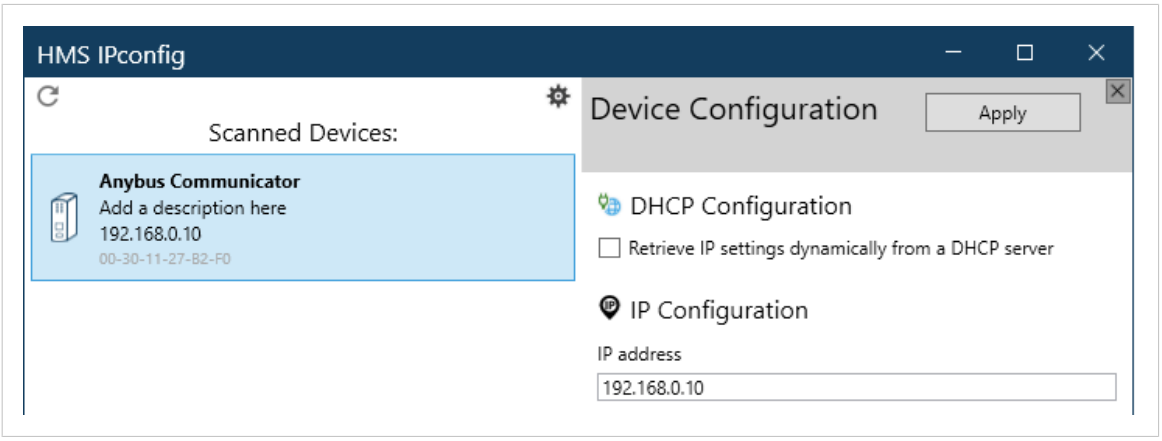
Procedure

1. Install HMS IPconfig on your PC.
2. Open HMS IPconfig.

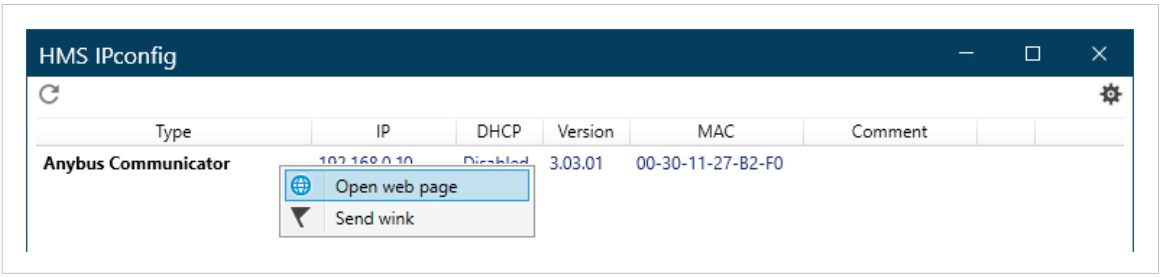


   - HMS IPconfig automatically starts scanning for compatible and active HMS devices.
   - Found HMS devices are added to the device list.
3. To open the settings pane, click on the Communicator in the device list.

4. Change the Communicator IP address to one within the same IP address range as your PC.

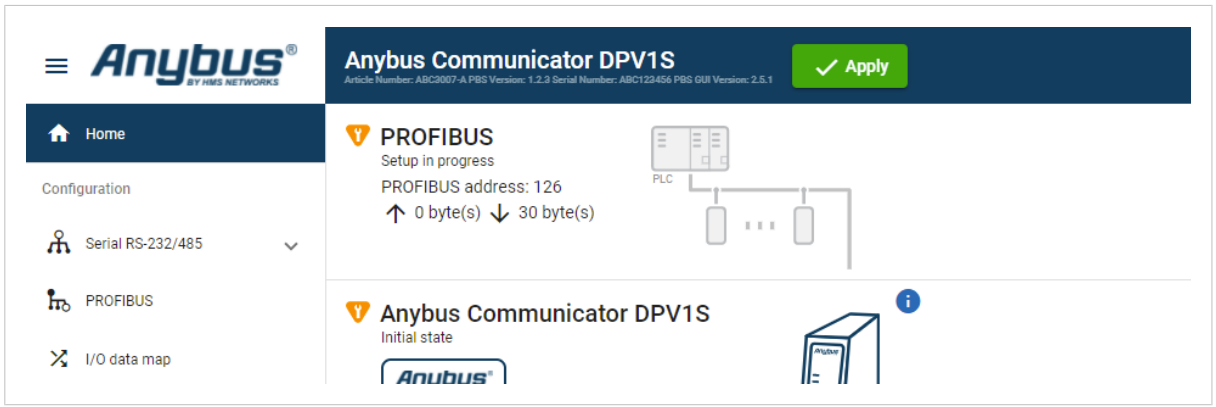


5. To open the **Open web page** built-in web interface, click Communicator.



Result

You are redirected to the Communicator built-in web interface Home page.



7.3. Access the Built-In Web Interface From a Web Browser

Before You Begin

NOTE
The Communicator default IP address is **192.168.0.10**.

NOTE
To access the Communicator built-in web interface, ensure that Port 80 TCP is open in your Firewall. This applies to any Firewall between the web browser and the gateway.

NOTE
When you change to a static IP address on your computer, internet access may be lost.

Procedure

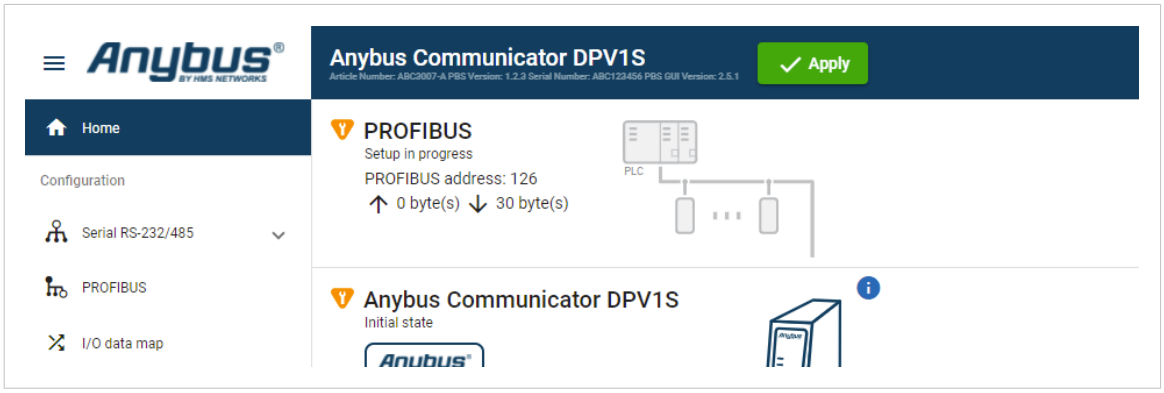
1. On the PC accessing the Communicator built-in web interface, set a static IP address within the same IP address range as the Communicator IP address.



2. Open a web browser.
3. Click to select the **Address bar** and enter the Communicator IP address.



4. To open the built-in web interface Home page, press **Enter**.



7.4. Communicator Built-In Web Interface Overview

Use the Communicator built-in web interface to configure, maintain and troubleshoot the Communicator.

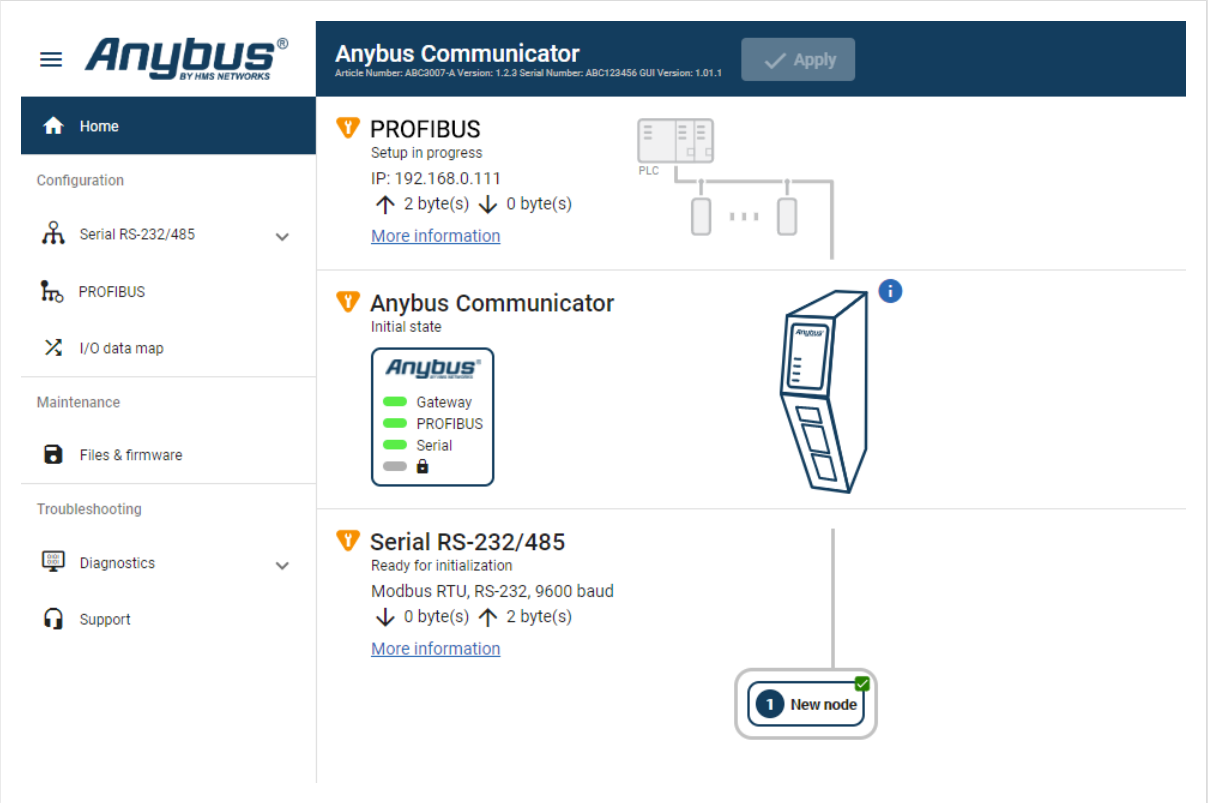


Figure 23. The Communicator built-in web interface Home page

Home	View the Communicator, network and node status.
Apply	After configuration changes are made and verified, press Apply to make the settings take effect.
Serial RS-232/485	Serial Subnetwork with Nodes. Select a Serial protocol, use Modbus RTU standard transactions or create your own transaction templates. Configure communication and add nodes and commands.
PROFIBUS	High Level Network with Client. Configure I/O Size and PROFIBUS address.
I/O data map	View the added commands mapped to the Communicator internal memory area.
Files & firmware	Save settings in a configuration files, upload configuration files and upgrade firmware.
Diagnostics	Monitor and troubleshoot the Communicator.
Support	Contains Communicator product information, Anybus contact information, link to Anybus support website, and product file for download. Here you can generate a support package with product information, to send to your Anybus support technician.

7.5. General Subnetwork Settings

7.5.1. Communication Serial Protocol

Before You Begin

Before starting the configuration, select the Serial protocol you want to use:

- **Modbus RTU**, Default setting: Use for serial devices that conform to the Modbus communication specification.
- **Custom Request/Response**: Create your own custom request/response transactions.
  The transactions can be based on the Modbus communication specification or fully customized.
- **Custom Produce/Consume**: Create your own custom produce/consume transactions.

IMPORTANT

When changing the serial protocol, all settings are reset to default and all added nodes, transactions, and transaction templates are deleted.

Procedure

On the **Communication** page, General settings:

1. To choose a **Serial protocol**, select **Modbus RTU**, **Custom Request/Response** or **Custom Produce/Consume**.

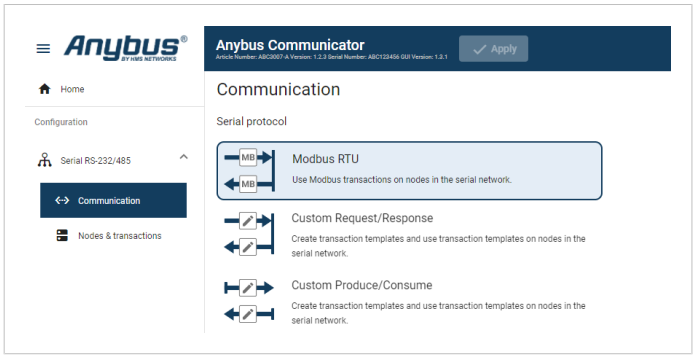


Figure 24. Communication, Serial protocol

2. To confirm the selected protocol, click **Change serial protocol**.

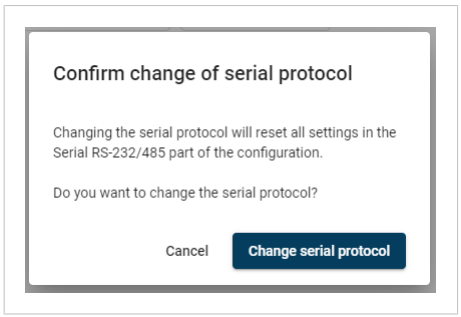


Figure 25. Confirm change of serial protocol

Apply Configuration

To apply the settings, click **Apply** in the web-interface header, and follow the instructions.

7.5.2. Communication Basic Settings

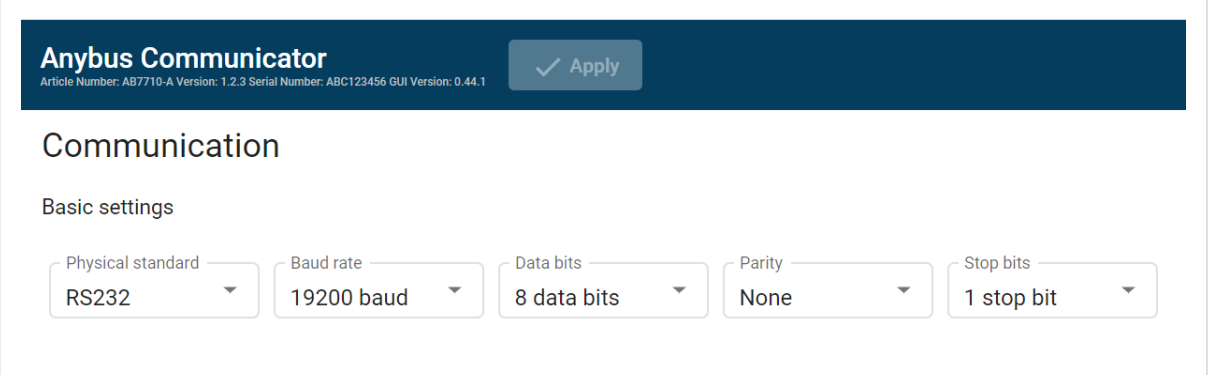
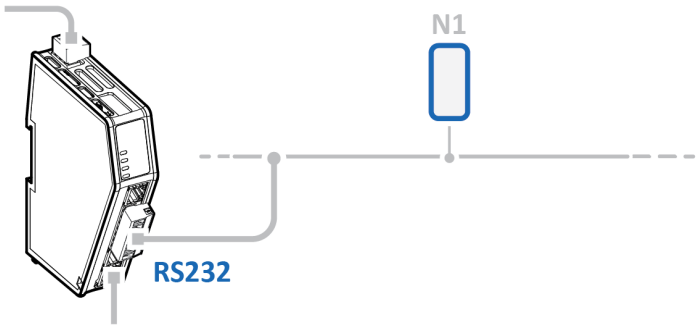
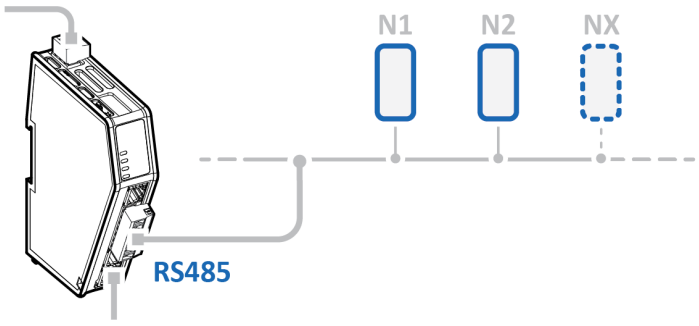


Figure 26. Communication, Basic settings

Physical standard

Specify the physical interface type for the device connected to the Communicator.

- Select a physical standard from the **Physical standard** drop-down menu.

Setting	Value	Description
Physical standard	RS-232, Default standard	Use RS-232 when one single node is connected to the subnetwork. 
	RS-485	Use RS-485 when multiple nodes are connected to the subnetwork. 

Baud rate

Specify the baud rate; the serial transfer speed, maximum bits per second.

Select a baud rate value from the **Baud rate** drop-down menu.

Data bits

Data bits is the number of bits used in the data representation of characters in the telegrams.

The rate for Modbus RTU is 7 data bits or 8 data bits. The default setting is 8 data bits.

Setting	Value
Baud rate	1200 baud
	1800 baud
	2400 baud
	4800 baud
	9600 baud, Default value
	19200 baud
	35700 baud
	38400 baud
	57600 baud
	115200 baud
	128000 baud

Parity

Specify if parity should be used to detect errors in the data.

Select parity value from the **Parity** drop-down menu.

Setting	Value	Description
Parity	None, Default value	No parity checking Parity bit is not transmitted
	Odd	Odd parity checking
	Even	Even parity checking

Stop bits

Specify the number of stop bits used to indicate the end of data transmission.

Select a stop bits value from the **Stop bits** drop-down menu.

Setting	Value
Stop bits	1 stop bit, Default value
	2 stop bit

Apply Configuration

To apply the settings, click **Apply** in the web-interface header, and follow the instructions.

7.5.3. Communication Advanced Settings

Inter-Telegram Timeout Mode Settings

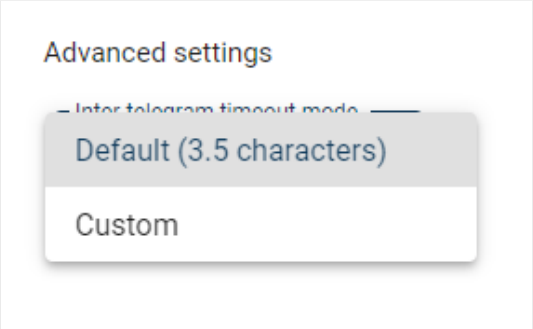


Figure 27. Advansed settings, Default (3.5 characters)

By default, Inter-telegram timeout mode Default (3.5 characters) is used.

This is according the Modbus RTU standard, which advocates the use of a silent period equivalent to 3.5 characters between each message. The silent period is used to find out where one message ends and the next begins.

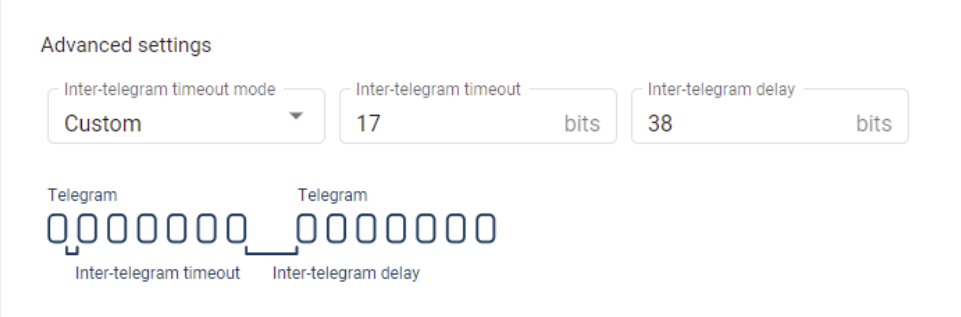


Figure 28. Custom settings, Inter-telegram timeout and Inter-telegram delay

You can use Custom settings to set the desired Inter-telegram timeout and Inter-telegram delay.

The following must be applied on all nodes:

- The time between two adjacent characters in the same telegram must be less than Intertelegram timeout.
- The time between two characters in two different telegrams the same or more than Intertelegram delay.

7.6. About Transaction Templates

This section applies when the Custom Request/Response or Custom Produce/Consume serial protocol is applied, refer to Communication Serial Protocol (page 45).

7.6.1. Transaction Template Example

Custom Request/Response

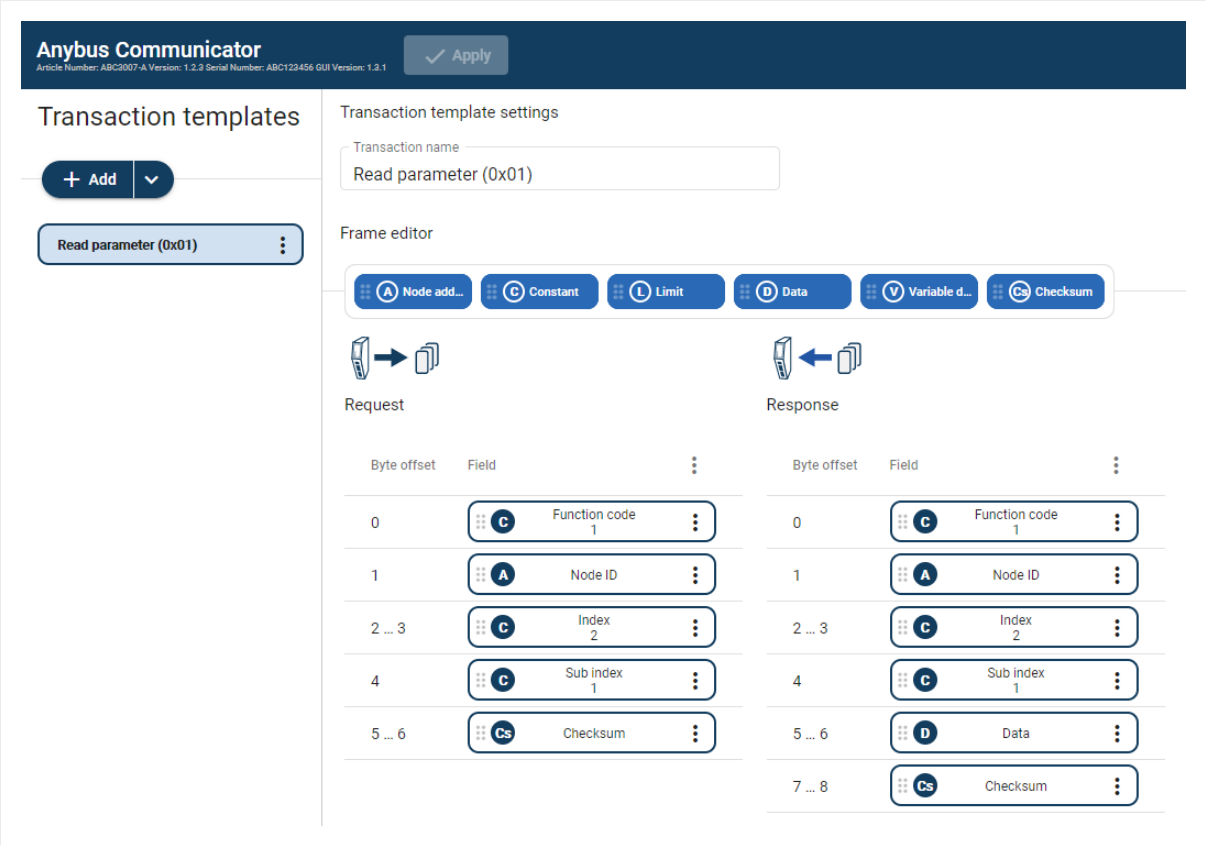


Figure 29. Request/Response transaction template example

The transaction named Read parameter (0x01) consists of a number of frame fields.

In the Request field there are three Constants, a Node address and a Checksum field.

In the Response field there are three Constants, a Node address, a Data field and a Checksum field.

Custom Produce/Consume

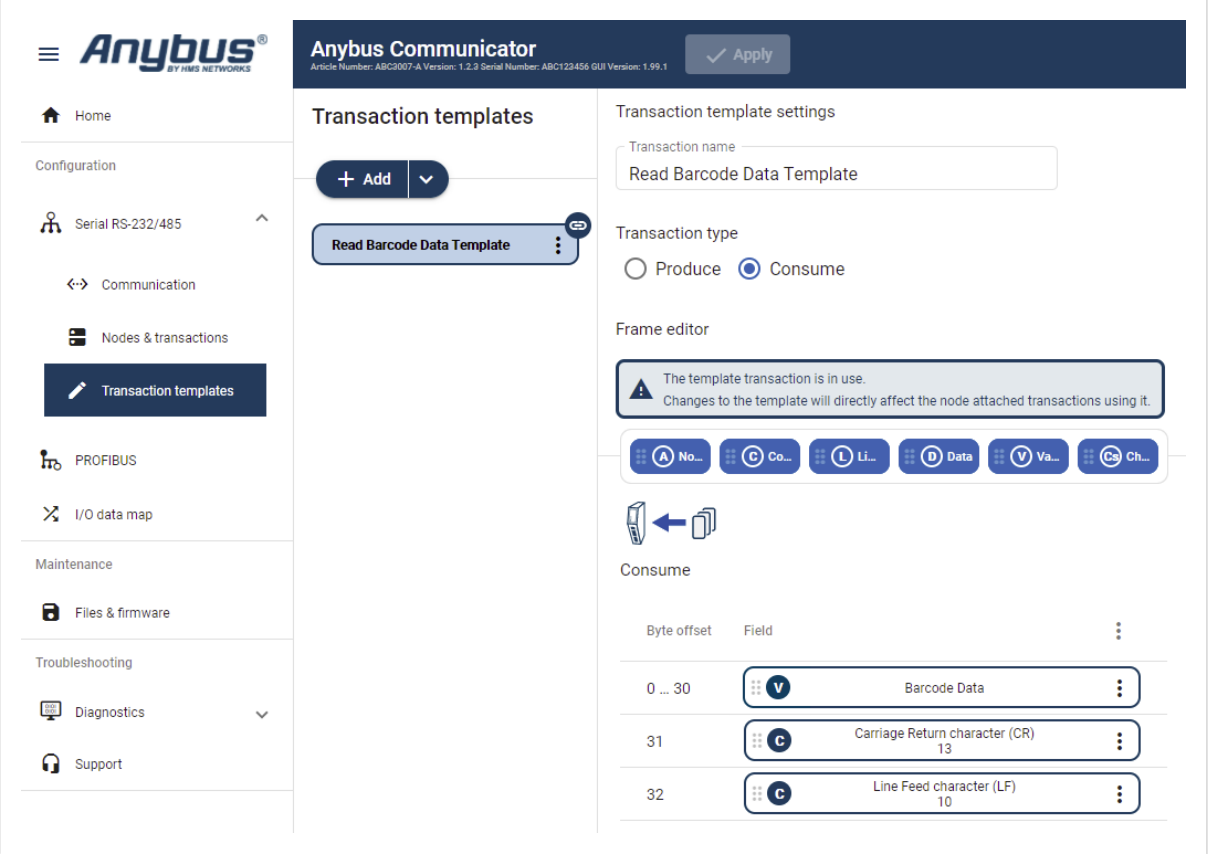


Figure 30. Produce transaction template example

The transaction named Read Barcode Data Template consists of a number of frame fields.

The Transaction type can be Produce or Consume. In this example the Transaction type Consume is selected.

In the frame field we have added one Variable data field and two Constant fields.

7.6.2. Transaction Template Types

There are two types of transaction templates, Empty template and Modbus template.

Empty Template

When using the **Empty template**, you start with an empty transaction and build a desired structure by adding and arranging frame fields.

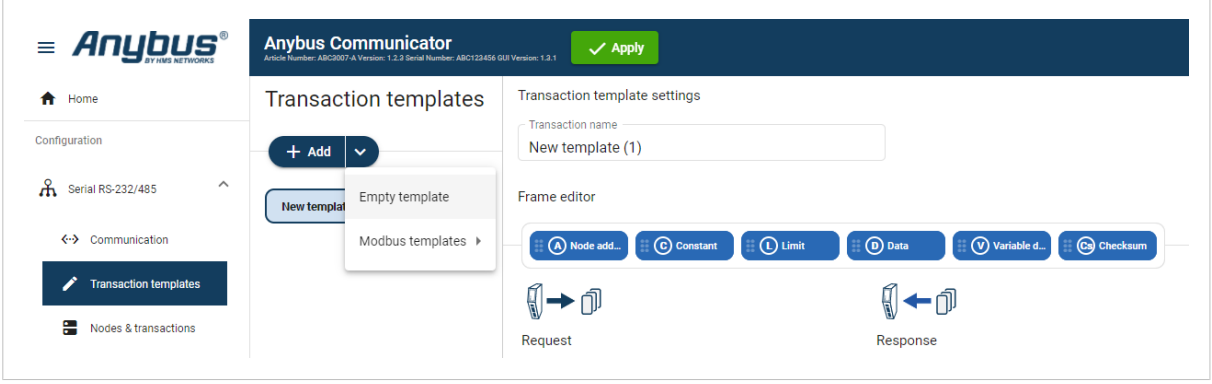


Figure 31. Transaction template, Empty template

For the produce/consume transactions you select; **Empty produce template** or **Empty consume template**.

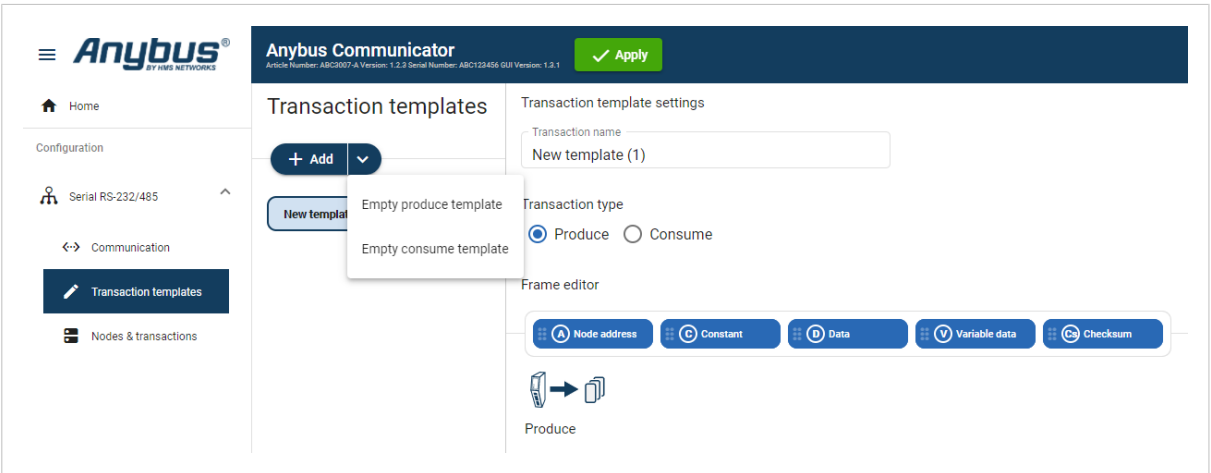


Figure 32. Transaction template, Empty Produce and Consume templates

Modbus Templates

Modbus templates are available for Custom Request/Response and Modbus RTU transactions.

When using the Modbus template, you first select the **Modbus template** from which you want to start.

You can then restructure the transaction by rearranging, adding or removing frame fields.

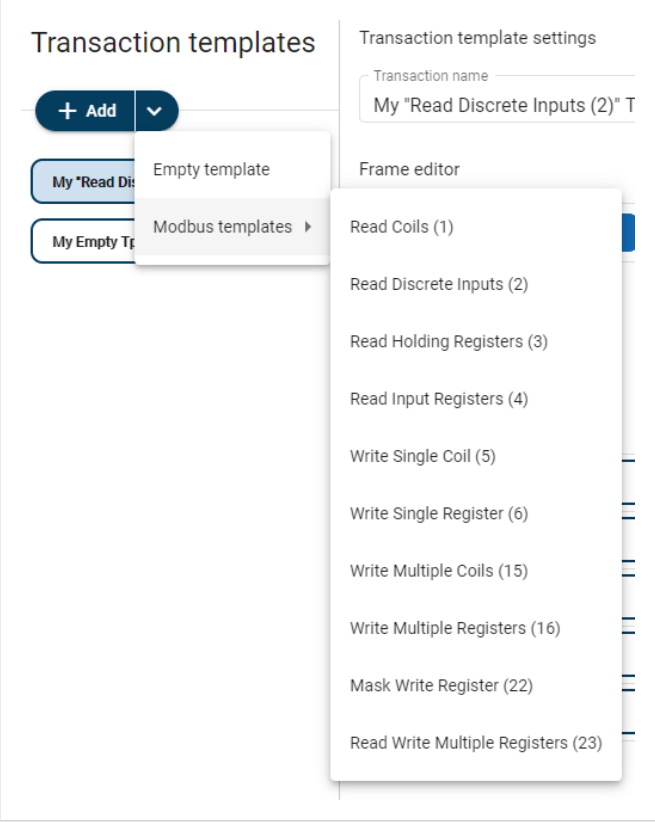
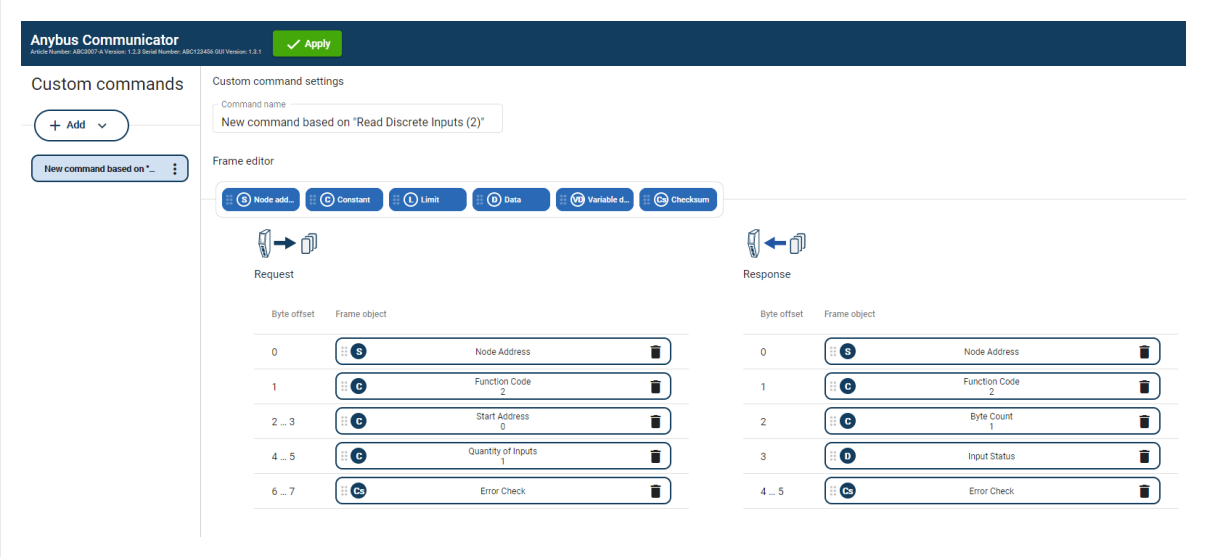


Figure 33. Transaction template, Modbus template

Example 4. New transaction template based on the Modbus template Read Discrete Inputs



7.6.3. Frame Field Types

Each transaction consists of frame fields which makes up the serial telegram frame.

Each frame field specifies how the Communicator shall interpret or generate a particular part of the telegram.

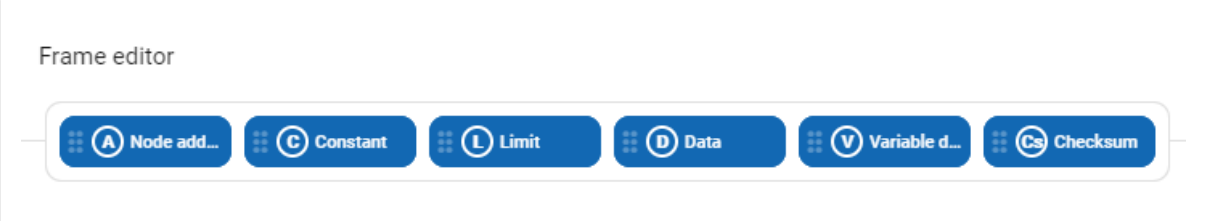


Figure 34. The following frame objects are available

Node address

Frame field representing the Node address of the Node.

A constant byte that holds a copy of the nodes address when the transaction is used by a node.

Constant

Constant frame fields are handled differently depending on the direction of the transaction.

- Produce/Request Transactions: The Communicator sends the value as it is without processing it.
- Consume/Response Transactions: The Communicator checks if the received byte/word/dword matches the specified value. If the message does not fit, it is discarded.

Limit

NOTE

Limit is not available for the Transaction Type Produce.

- Consume/Response Transactions: The Communicator checks if the received byte/word/dword fits inside the specified boundaries. If the message does not fit, it is discarded.

Data
Data frame fields are used to represent raw data as follows.

- Produce/Request Transactions: The specified data block is forwarded from the higher level network to the subnetwork.
- Consume/Response Transactions: The specified data block is forwarded from the sub-network to the high level network.

Variable data
Produce/Request Transactions:

- The specified data block will be forwarded from the higher level network to the sub-network.
- The control system must supply an End or Length character in order for the Communicator to know the size of the data block.
- The End- or Length-character itself may either be forwarded to the sub-network or discarded.

Consume/Response Transactions:

- The specified data block is forwarded from the sub-network to the higher level network.
- The End- or Length-character will be generated by the Communicator automatically (if applicable).
- The End- or Length-character itself may either be forwarded to the higher level network or discarded.

Checksum
Most serial protocols features some way of verifying that the data has not been corrupted during transfer.

The checksum frame field calculates and includes a checksum in a transaction.

7.7. Build Transaction Template

Before You Begin

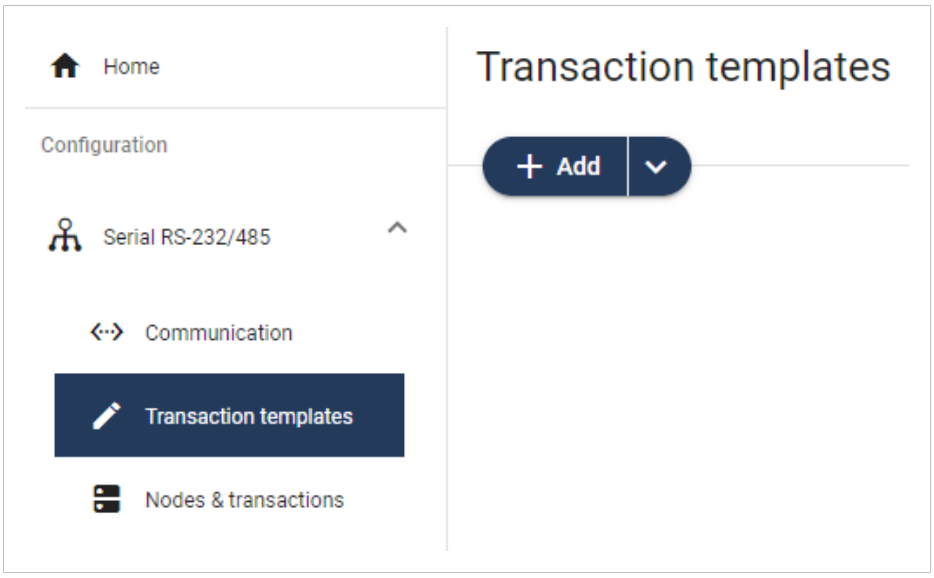
Ensure that you have applied the Custom Request/Response or Custom Produce/Consume serial protocol, refer to Communication Serial Protocol (page 45).

7.7.1. Add Transaction Template

Procedure

Add a transaction template:

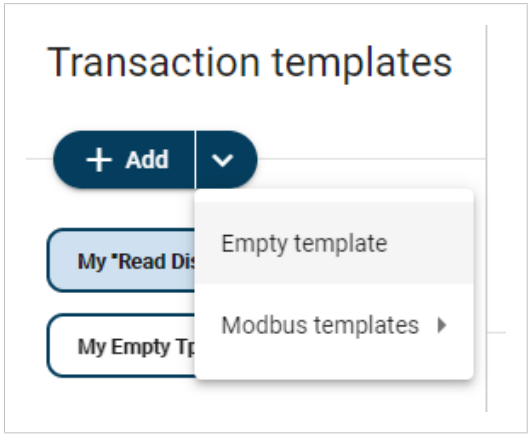
1. In the web-interface left sidebar menu, click **Transaction templates**.



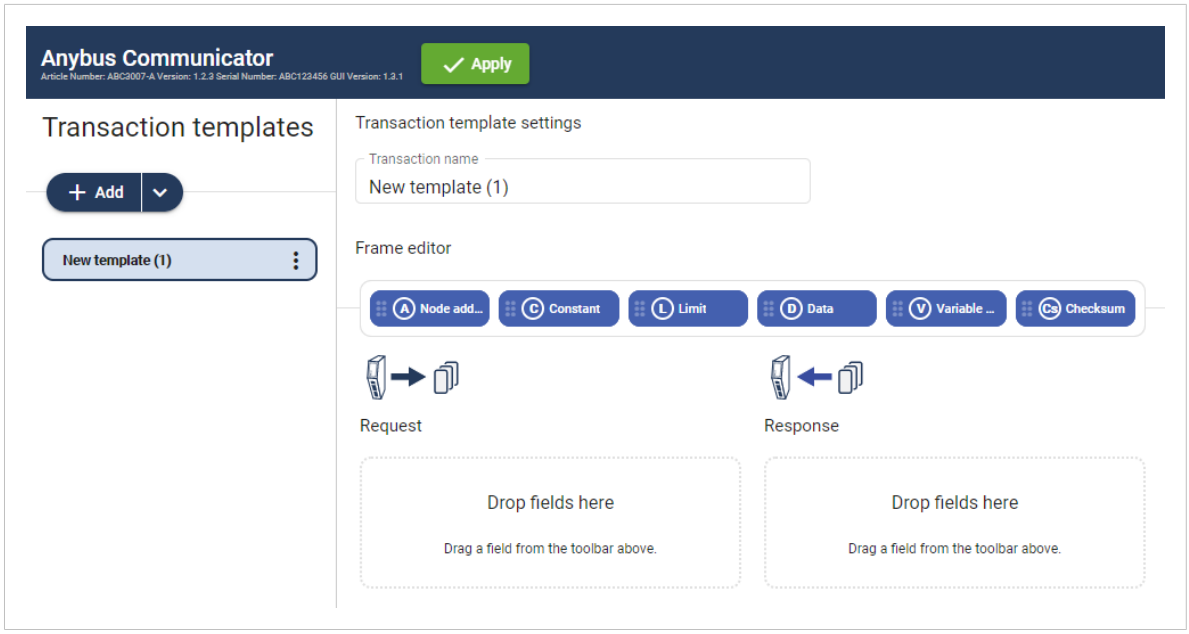
2. To select the template you want to use, click the **Add** drop-down button.

Options for the Custom Request/Response Protocol

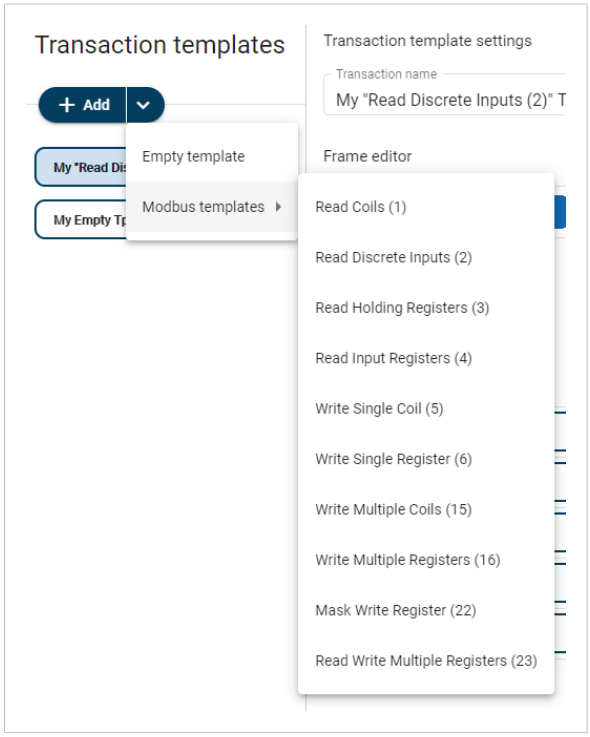
- To add a new empty template without any frame fields, select **Empty template**.



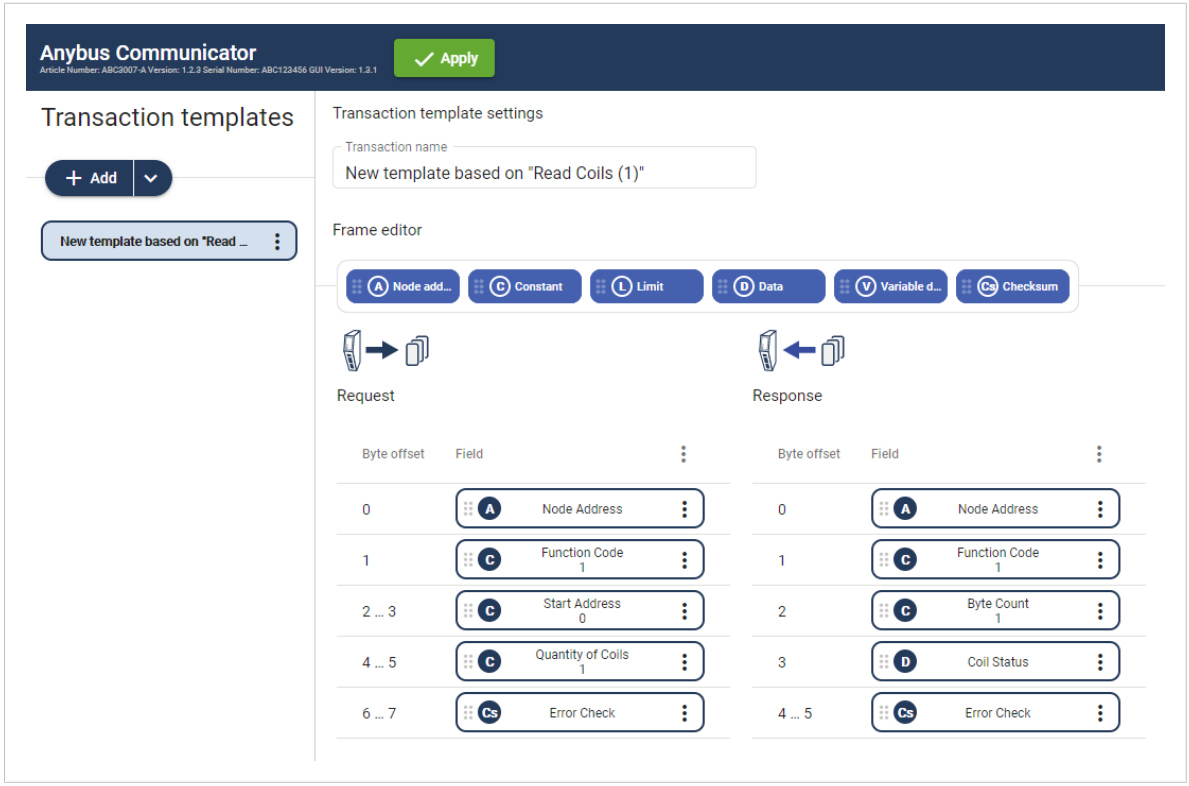
Example 5. A new empty request/response template is added to the transaction template list



- To add a new template based on a standard Modbus transaction, select **Modbus templates** and then the desired Modbus transaction.

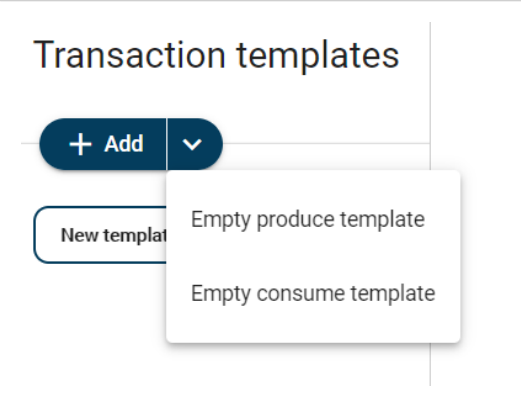


Example 6. A new request/response template based on “Read Coils (1)” is added to the transaction template list

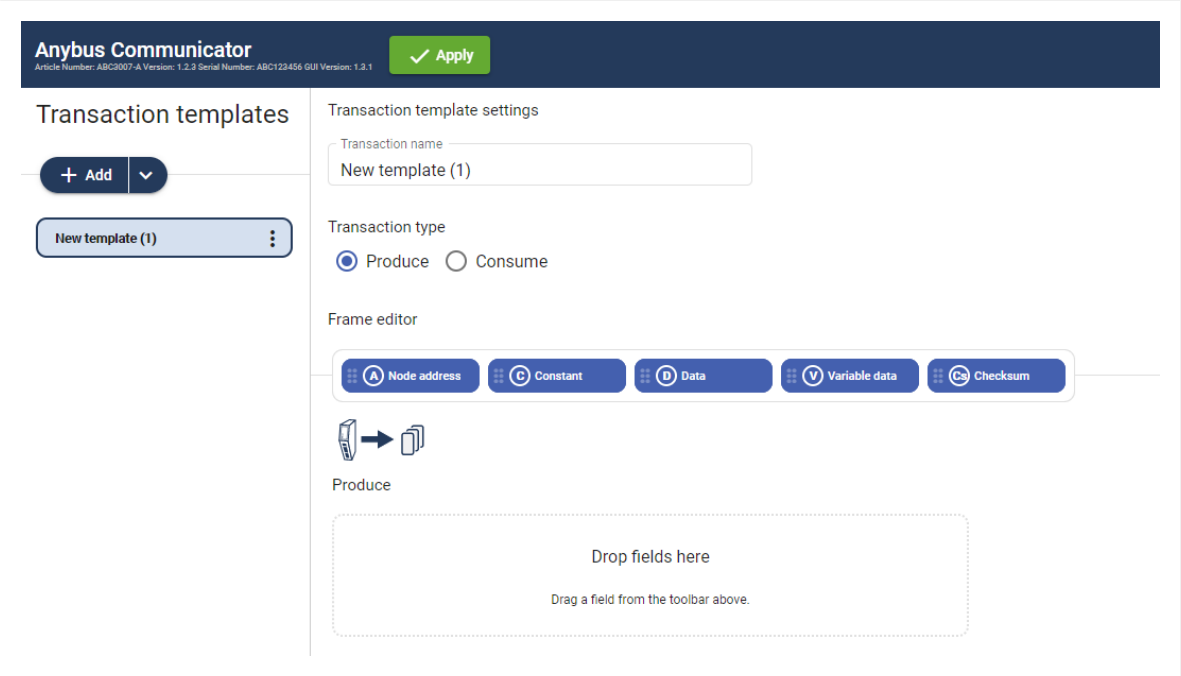


Option for the Custom Produce/Consume Protocol

- Select **Empty produce templates** or **Empty consume templates**.
  You can change the Transaction type after the transaction template is added.



Example 7. A new produce template is added to the transaction template list



3. To apply the settings, click **Apply** in the web-interface header, and follow the instructions.

To Do Next

- Add frame fields to the transaction template, refer to Add Frame Fields (page 58).

7.7.2. Add Frame Fields

Procedure

1. In the transaction template list, select a transaction template to add frame fields to.

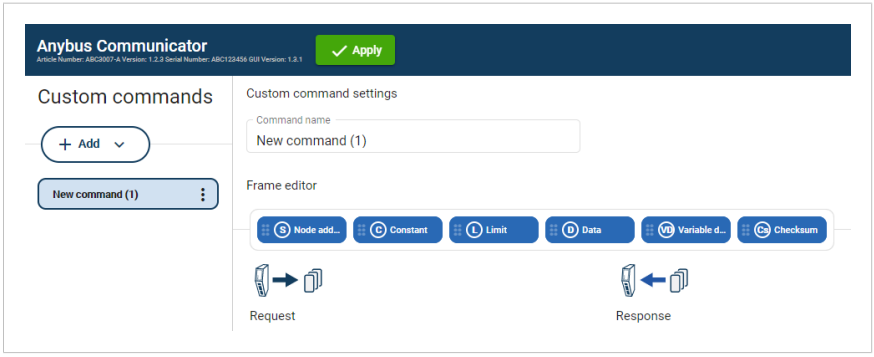


Figure 35. Select transaction template

2. Build the transactions.

- To add frame fields: In the Frame editor frame fields menu, drag and drop the desired frame fields into the drag and drop fields.

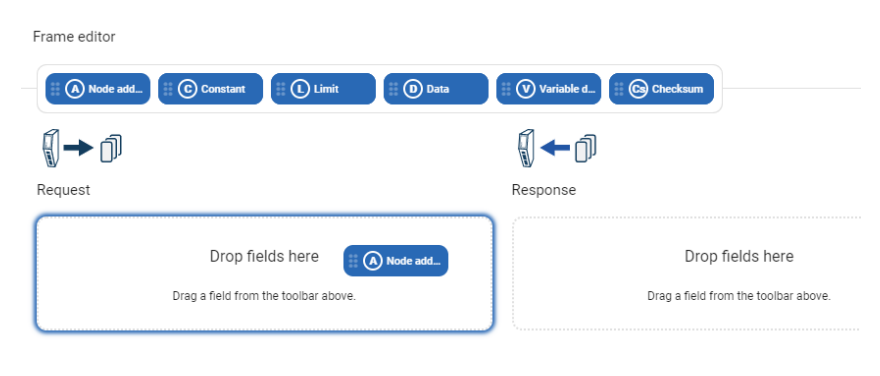


Figure 36. Add frame fields

- To duplicate a frame field: On the frame field that you want to delete, click the three dots icon and then click Duplicate.



Figure 37. Duplicate frame fields

- To change the order of the frame fields: Drag and drop the frame fields in the list to change the order.



Figure 38. Change frame fields order

- To delete a frame field: On the frame field that you want to delete, click the **three dots icon**. Click **Delete** and then **Yes** to confirm.

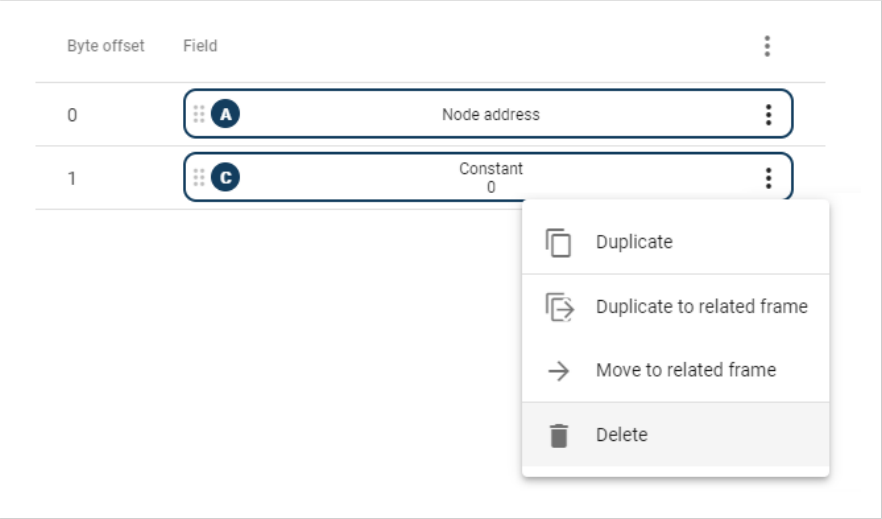


Figure 39. Delete frame field

3. To apply the settings, click **Apply** in the web-interface header, and follow the instructions.

To Do Next

- Configure the frame field settings, see Configure Frame Fields Settings (page 60).

7.7.3. Configure Frame Fields Settings

Procedure

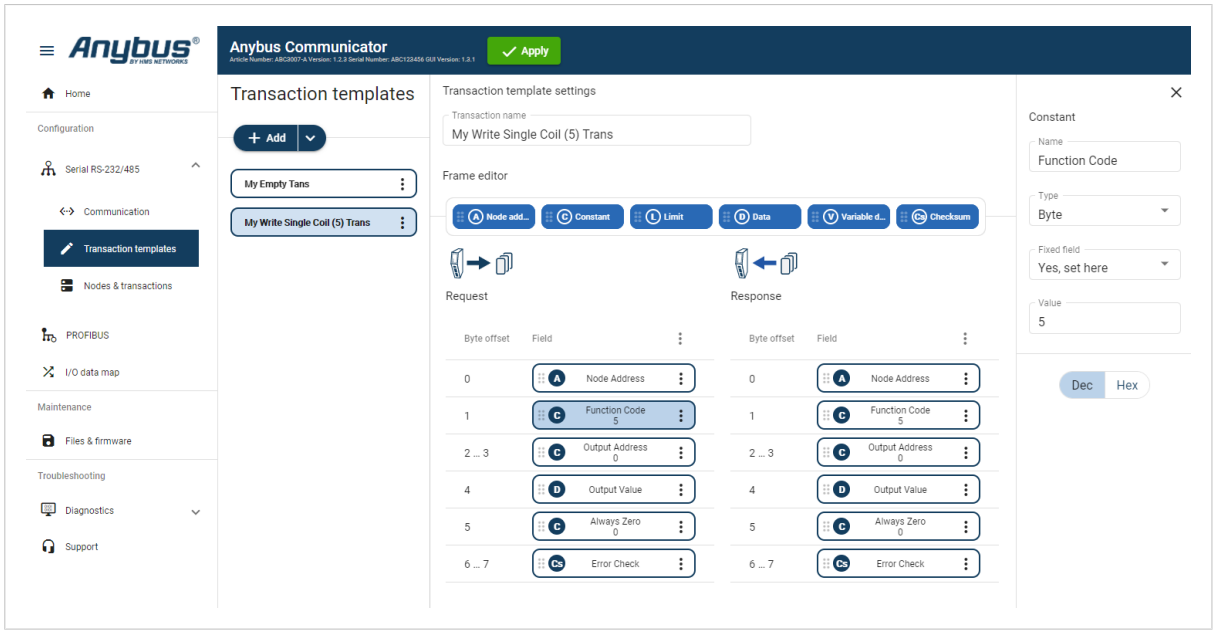


Figure 40. Frame Fields Settings

1. In the Transaction templates list, select a transaction template to configure.
2. In the Transaction template settings select a Field to configure.
   The Field sidebar opens, on the right side of the screen.
3. Configure the **Field** settings.

Node address

- Frame field representing the Node address of the node. A constant byte that holds a copy of the nodes address when the transaction is used by a node.
- When the transaction template is used by a node, the Node address field will automatically be replaced with the actual node address of the node.

Constant

- **Name**: You can name the Frame Field to make it easier to identify.
- **Type**: Specify the number of bytes in the frame.
  Select Byte (1 byte) (Default), Word (2 bytes), Double word (4 bytes), Array of bytes or String.
- **Endianess**: Select Big-endian (Default) or Little-endian.
- **Fixed field***: Select Yes, set here (Default) or No, set when used.
- **Length**: Valid for Array of bytes. Enter a byte offset value between 0 and 32 byte.
  Default value is 1 byte.
  Enter a Value for each Byte (0–31).
- **Value**: The value of the Constant in the frame.
  Enter a value between 0 (Default) and 255.
- **Min value**: Specify the minimum value that can be set when the template is used.
- **Max value**: Specify the maximum value that can be set when the template is used.
- **Default value**: Default value set when the template is used.

Limit

> **NOTE**
> Limit can only be added as a Response frame field.

- **Name**: You can name the Frame Field to make it easier to identify.
- **Type**: Specify the number of bytes in the frame. Select Byte (1 byte) (Default), Word (2 bytes), Double word (4 bytes).
- **Endianess**: Select Big-endian (Default) or Little-endian.
- **Min value**: The lowest value of the limit range.
- **Max value**: The highest value of the limit range.
- **Base number system**: Select Decimal Dec (Default) or Hexadecimal Hex.

Data

- **Name**: You can name the Frame Field to make it easier to identify.
- **Fixed field***: Select Yes, set here (Default) or No, set when used.
- **Length**: Enter a value between 1 (Default) and 512 bytes.
- **Min length**: Specify the minimum length that can be set when the template is used.
- **Max length**: Specify the maximum length that can be set when the template is used.
- **Default length**: Specify the default length that can be set when the template is used.

Variable data

- **Name**: You can name the Frame Field to make it easier to identify.
- **Fixed field***: Select Yes, set here (Default) or No, set when used.
- **Minimum payload length**: Specify the minimum payload length that can be set when the template is used.
- **Maximum payload length**: Specify the maximum payload length that can be set when the template is used.
- **Default max payload length**: Specify the default payload length that can be set when the template is used.
- **Data delimiter**: Specify how to detect/define the length of the variable data of the high level network. Select Byte counter, End pattern or None (Default).
  For information about End- and Length character, see Data Delimiter and Subnet Delimiter Options (page 62).
- **Subnet delimiter**: Specify how to detect/define the length of the variable data of the serial subnetwork. Select Byte counter, End pattern or None (Default).
  For information about End- and Length character, see Data Delimiter and Subnet Delimiter Options (page 62).
- **End pattern**: Specify the value defining the end of the payload, when a delimiter is set to end pattern.
- **Fill padding**: Fill up unused data mapped to the high level network or the general area with a field padding value.
  To deactivate/activate Fill padding, click the slide toggle. When Fill padding is activated, enter a Fill padding value between 0 and 255.
- **Base number system**: Select Decimal Dec (Default) or Hexadecimal Hex.

Checksum

- **Name**: You can name the Frame Field to make it easier to identify.
- **Checksum type**: Specify the algorithm used to calculate the checksum. Select CRC (CRC-16-IBM) (Default), LRC (ISO 1155:1978), XOR or ADD.
- **Start offset**: Specify the offset from where to start the checksum calculation. Enter a value between 0 (Default) and 511.

- **Error check type**: Specify how the checksum is converted. Select None (Default), One's complement or Select None, Two's complement.
- **Representation**: Specify how the checksum is represented. Select Binary (Default) or ASCII.

About Fixed field*

- **Yes, set here**: The Value set here is fixed and cannot be changed when the transaction is used on a node. The value must be updated in the transaction template.
- **No, set when used**: The Default value set here can be edited when the transaction is used on a node. The allowed range is the min/max values.

Total size including delimiters:

- High Level Network: 1 byte(s)
- Subnetwork: 1 byte(s).

4. Repeat step 1 to 3 until you have configured all the desired frame objects.

Apply Configuration
To apply the settings, click **Apply** in the web-interface header, and follow the instructions.

7.7.4. Data Delimiter and Subnet Delimiter Options

In a variable data object, the length of the data field may vary depending on the type of data being read in a specific case.

In order to present the variable data correctly on the corresponding network, the length of the data field must be identified.

In a Variable data object, there are three ways to identify the data length; by length character, end character or length of message.

Data delimiter - Data is forwarded From the Communicator to the PLC
The Communicator can be configured to forward data as process data.

Different Data delimiter options can be used for data sent from the subnetwork to the Communicator and for data forwarded from the Communicator to the high level network, to fit the requirements in the PLC.

In most cases, when a stream of data is sent from the Communicator to the PLC the Byte counter (length character) or End pattern (end character) format is used.

Subnet delimiter - Incoming Data From a Serial Node to the Communicator
The Communicator can be configured to expect data from one of the three Subnet delimiter options; Byte counter, End pattern or None.

If the incoming data match the Subnet delimiter format the data is captured and the data section is forwarded to the high level network.

If the incoming data do not match the Subnet delimiter format, the data is ignored and will be matched with the next consume transaction.

Transaction Template Variable Data Settings

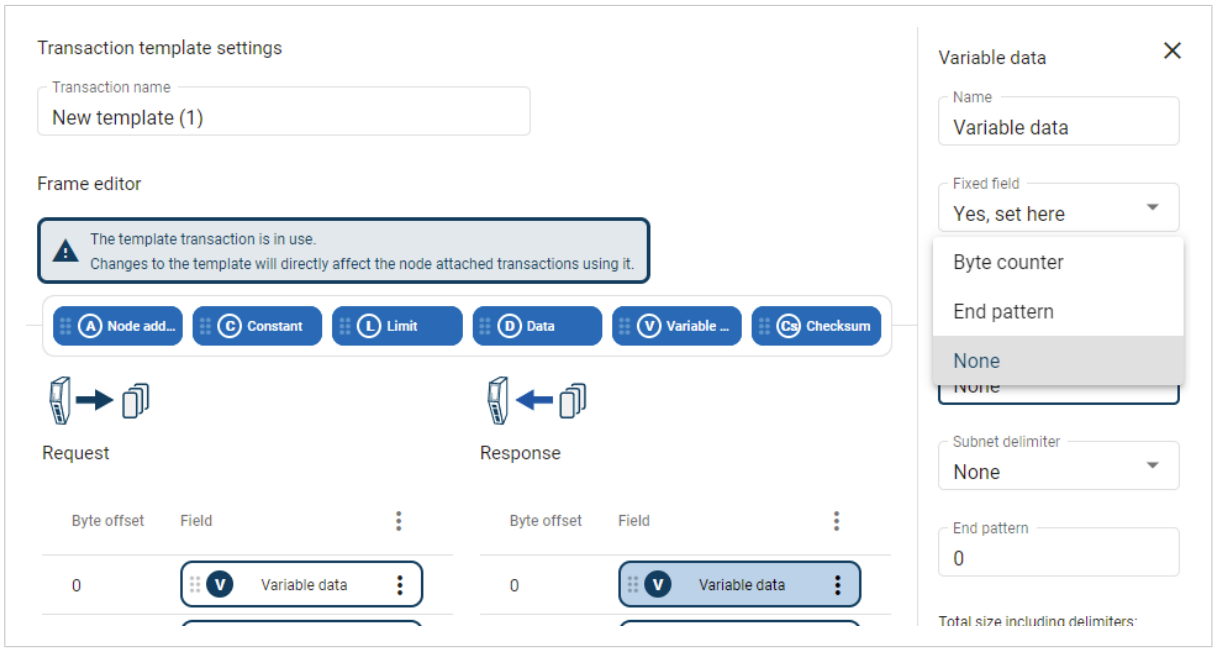
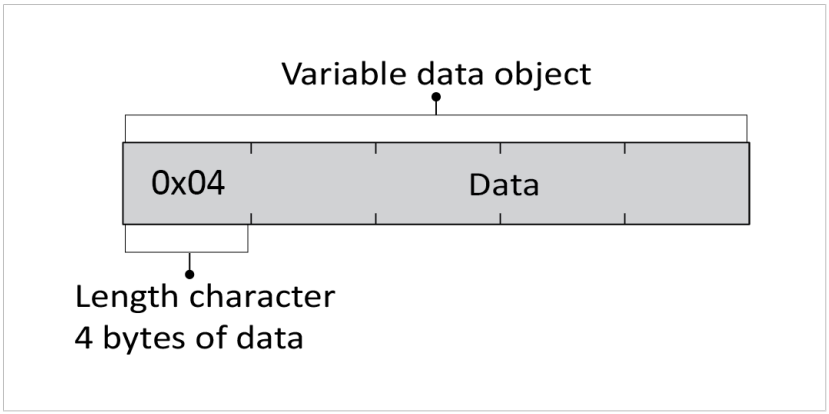


Figure 41. Transaction template variable data settings

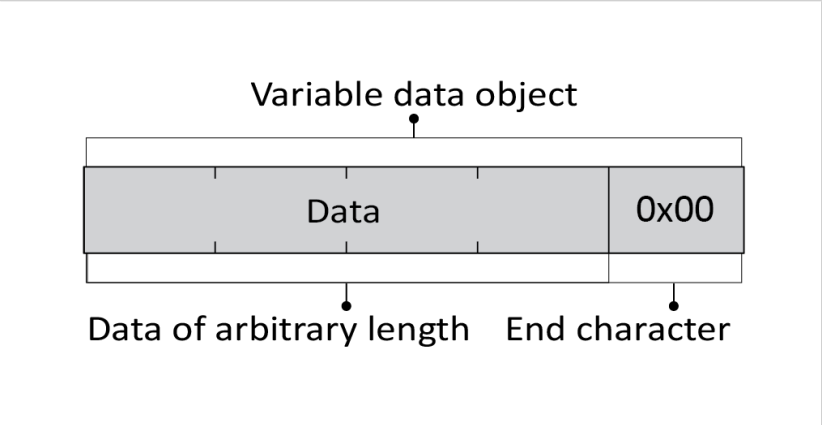
1. Select a desired **Variable data** object.
2. In the **Data delimiter** and/or **Subnet delimiter** drop down menu, select one of the following options.

Data delimiter and/or Subnet delimiter options

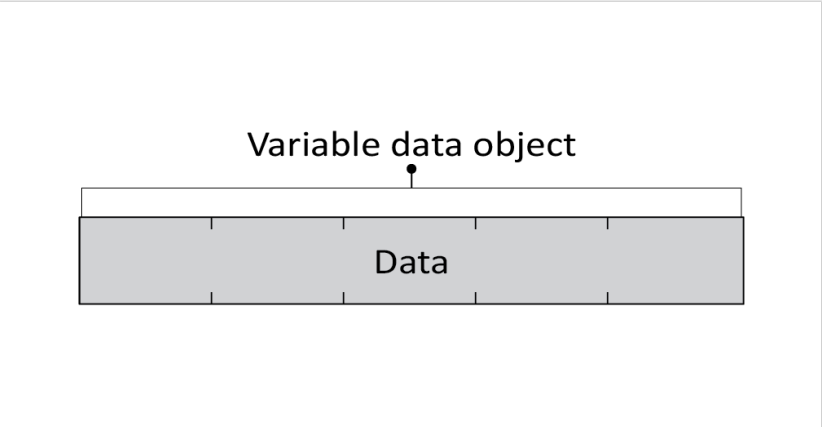
- Byte counter
  The data packet consists of a length character, indicating the length of the data section, followed by the variable data object itself. In order to copy the exact data size from the transaction message, the length of the variable data object is first identified.



- End pattern
  The package consists of a data section followed by an end character to indicate where the data section ends. End pattern is used to define whether the delimiter is an end character or an end pattern, which depends on whether the message is forwarded from the subnetwork or sent as process data.



- None
  The package contains only the data section. By measuring the total length of the message, the length of the data section can be calculated.



3. When a delimiter is set to End pattern: In the **End pattern** field, enter the value that will define the end of the payload.

7.7.5. Store Transaction Templates

The transaction templates are stored on the **Transaction templates** page.

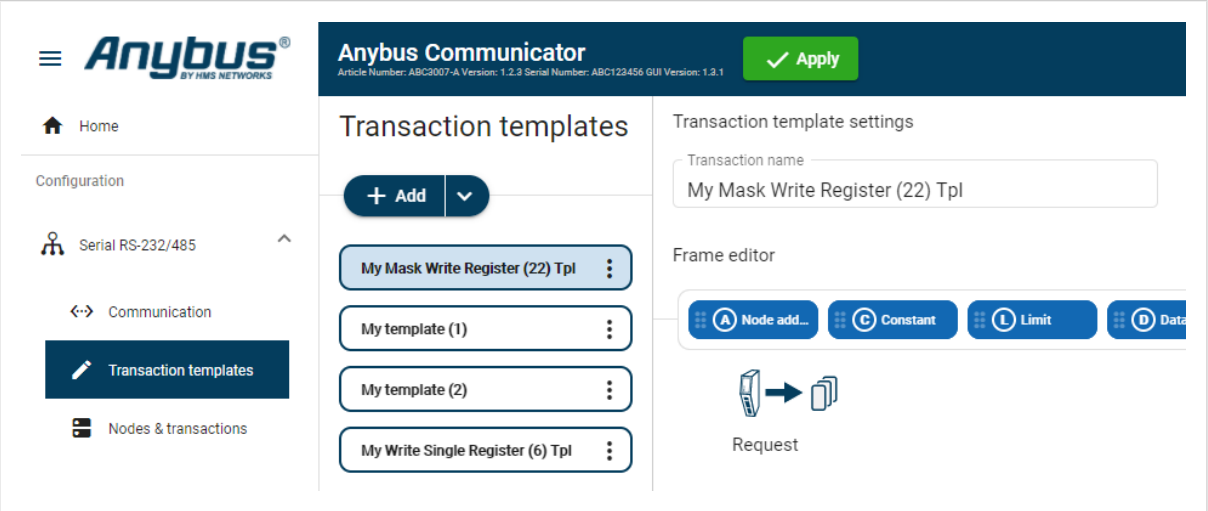


Figure 42. **Transaction templates** page

The transaction templates are available for use on the **Nodes & transaction** page, when you add transactions to a node.

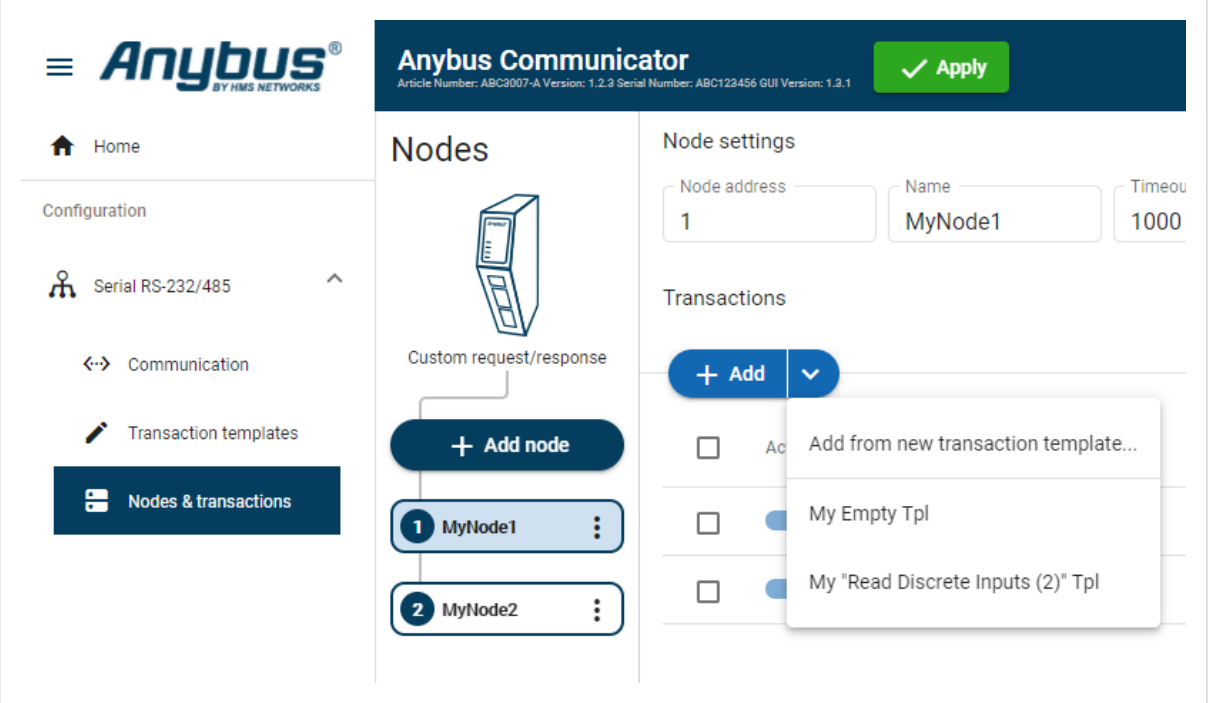


Figure 43. **Nodes & transaction** page

For information on how to add the transaction templates to the nodes, refer to Transaction Settings (page 72).

7.8. Nodes and Transactions

A node represents a single device on the serial subnetwork.

Add nodes and set up the communication between the nodes and the client.

Before You Begin

Obtain user documentation, from the manufacturers of the devices to communicate with, describing available registers and how to address them.

7.8.1. Node and Broadcast Node

You can add two types of nodes, Node and Broadcast Node.

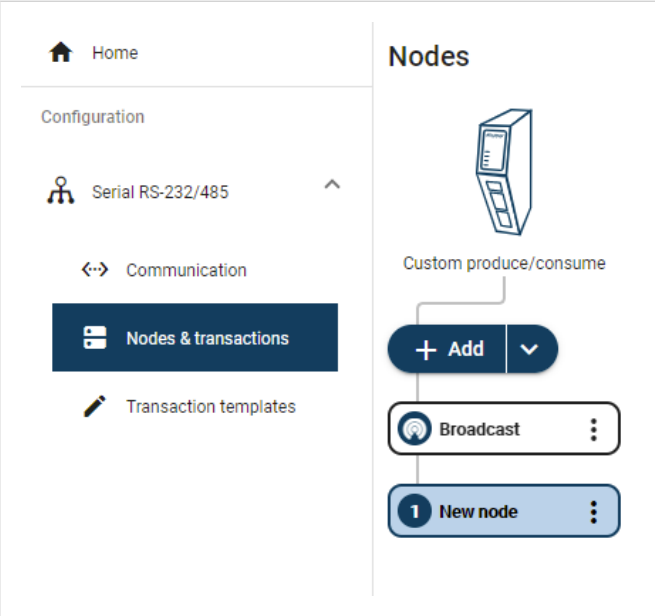


Figure 44. Add Node or Broadcast Node

Broadcast node

- You can add one single Broadcast node.
- The Broadcast node can only hold produce transactions.

Node

- You can add up to 31 Nodes.
- The type of transactions a node can hold depends on the serial protocol used, refer to About Transaction Templates (page 49).

7.8.2. Add Node

NOTE
You can add one single Broadcast node.

The maximum number of Nodes that can be added is 31.

1. In the web-interface left sidebar menu, click **Nodes & transactions**.
2. Click **Add node**.
3. Select **Add broadcast node** or **Add node**.

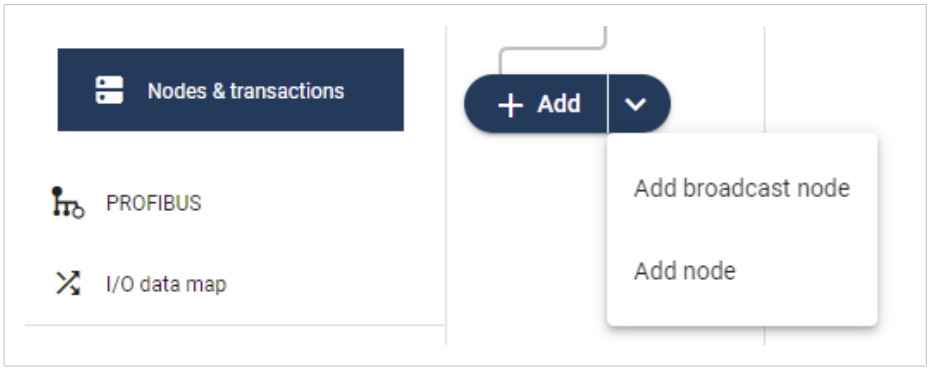


Figure 45. Add node

A new node/brodcats node is added to the nodes list.

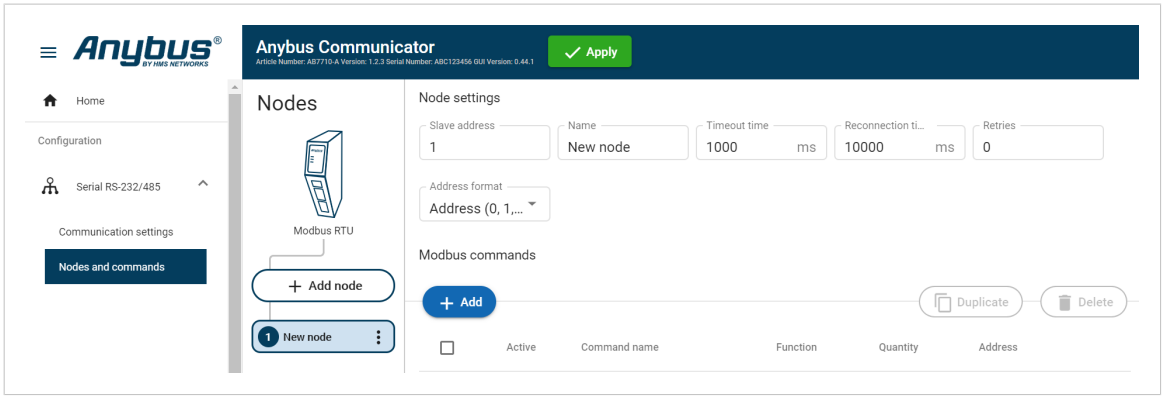


Figure 46. A new node is added to the node list

To Do Next
Configure the Node Settings, Node Settings (page 68).

7.8.3. Node Settings

Before You Begin

Ensure that the Communicator Basic settings, on the **Communication** page, match the Node settings.

There are no Node settings for the Broadcast node, except Name.

Procedure

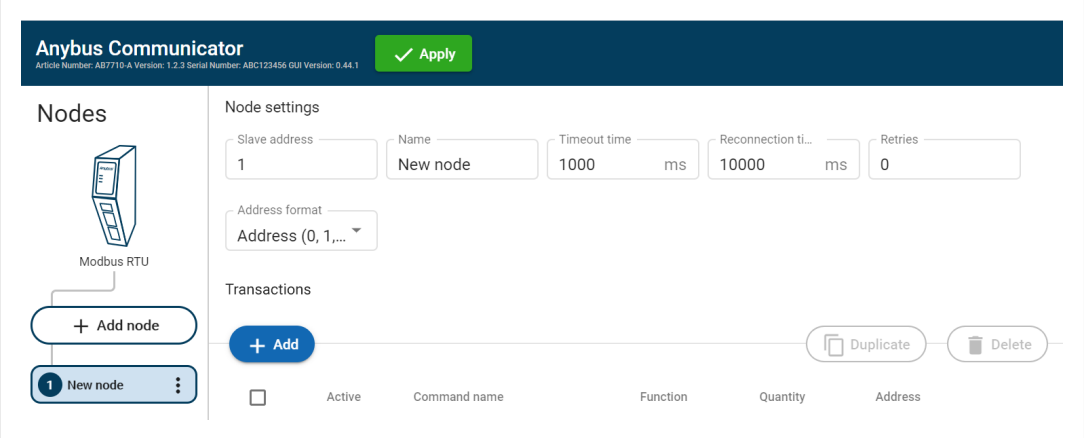


Figure 47. Nodes page, Node settings

1. In the node list, select a node to configure.
2. Configure the **Node settings**.

Setting	Value	Description
Node address	1 to 247	Node ID, also called node address, is the node's identity on the subnetwork. The node id is a number between 1 and 247. By default, the node is assigned the next available number. The same node id cannot be used on multiple nodes.
Name	N/A	By default, the node is assigned the name New node and the corresponding Server address. The node name can be changed.
Timeout time	10 ms to 10 000 ms Default 1000 ms	If a transaction in a transaction fulfills the specified timeout time value for all specified retries, the remaining transactions defined for the node will be skipped in the current cycle. The maximum addition to the cycle length is only one instance of the timout setting. Specify how long the Communicator should wait before sending the message again, when no response is received from the node. If the timeout time is exceeded, the Communicator continues to send the message until the maximum number of retries has been reached.
Reconnection time	Min 10 ms Max 60 0000 ms Default 1000 ms	Specify for how long the Communicator should wait before attempting to reconnect, if the node is disconnected. The default value is 10 000 ms. Reconnect time (10 ms) is not applicable for the broadcast node, that hold transactions destined to all nodes.
Retries	0 to 10 Default 0	Specify the number of attempts the Communicator should make, when no response is received from the node.
Address format	Default format: Address Register Modicon Modicon extended	Available for the Modbus TCP serial protocol. Specify the address format for the node. Address: 0, 1, 2, ... Register: 1, 2, 3, ... Modicon: 00001/10001/30001/40001 Modicon extended: 000001/100001/...

3. To apply the settings, click **Apply** in the web-interface header, and follow the instructions.

To Do Next
Add Transactions, Add Transactions (page 69).

7.8.4. Add Transactions

> **NOTE**
> The maximum number of transactions that can be added to a node is 150.

1. In the node list, select a node to configure.
2. In the transaction list, click **Add**.
3. Choose one of the following alternative:

When using the Modbus RTU Serial Protocol

- Click **Add** and select a transaction from the list of standard Modbus RTU transactions.

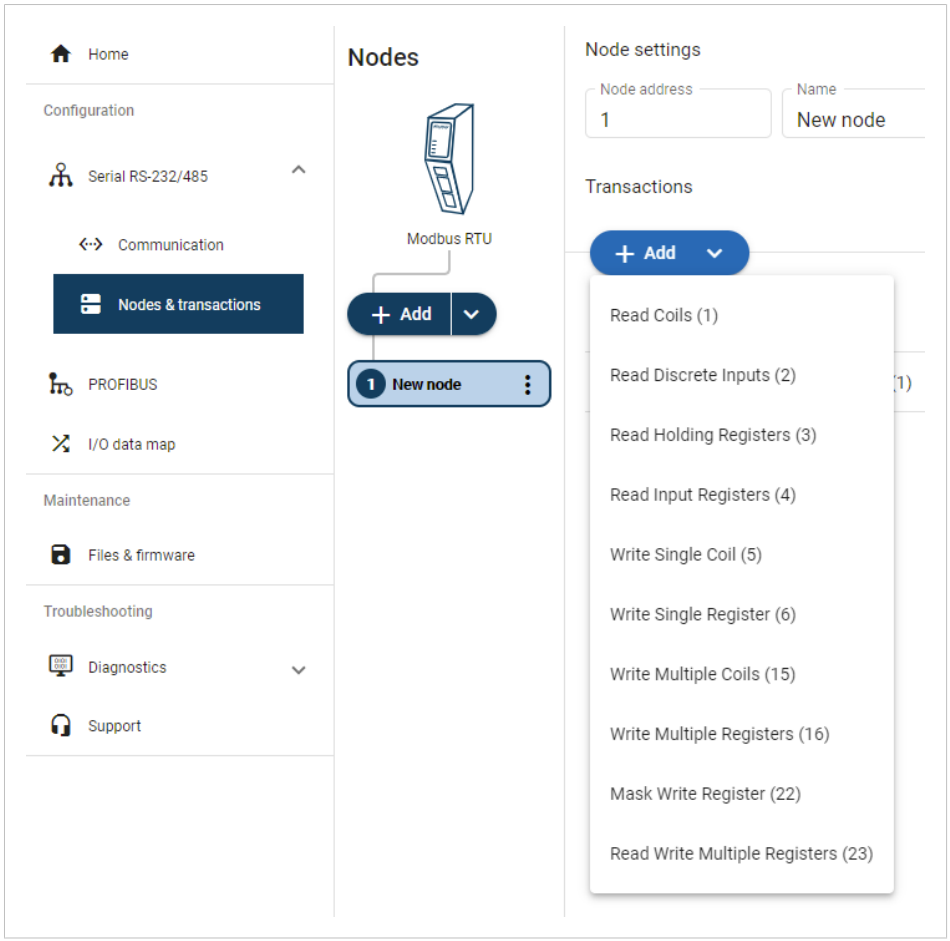


Figure 48. Add Modbus RTU transactions

When using the Request/Response or Produce/Consume Serial Protocol

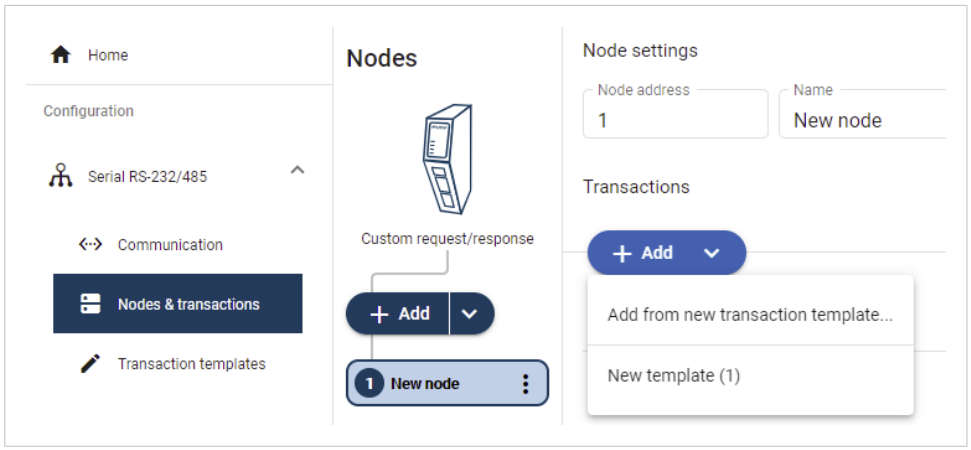


Figure 49. Add new empty template

- Click **Add** and select **Add from new transaction template**.
  You are redirected to the **Transaction template** page.
  A new empty template is added to the Transaction templates list.

> **NOTE**
> You must build the transactions before you can use the template, refer to Build Transaction Template (page 54).

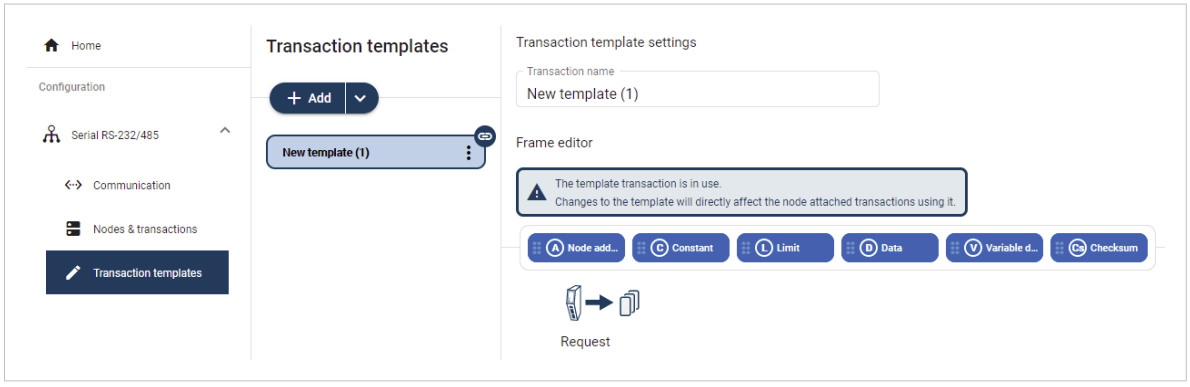


Figure 50. Add new empty template

- If you already have created Transaction templates, click **Add** and select the desired template from the list. A new transaction is added to the transactions list.

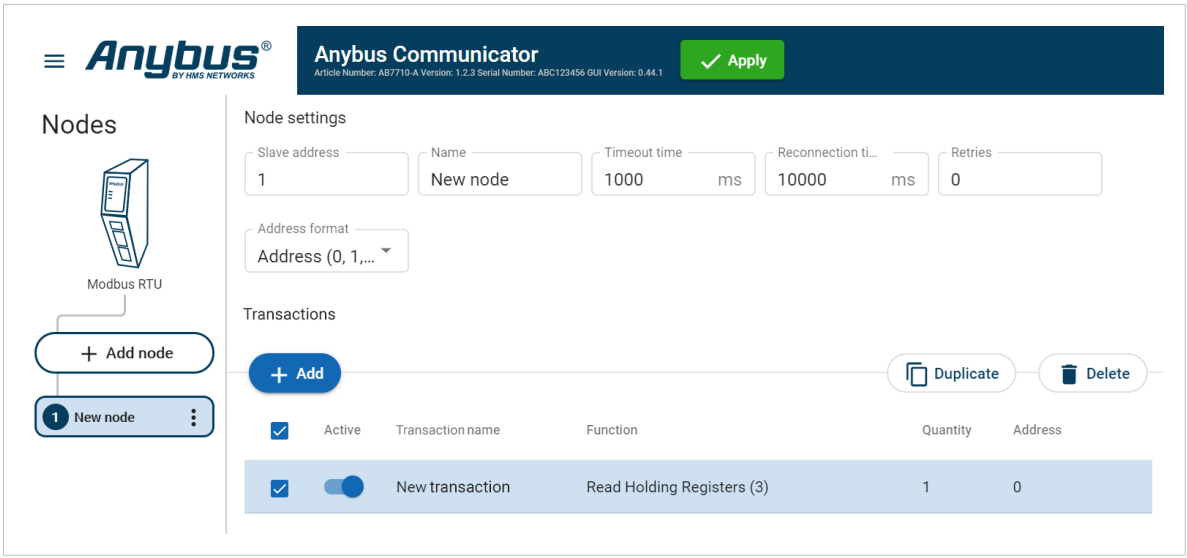


Figure 51. Add new transaction

To Do Next

Configure the Transactions settings, Transaction Settings (page 72).

7.8.5. Transaction Settings

Before You Begin

NOTE
When a custom transaction is selected, the custom transaction template is locked for editing.

For Modbus transaction reference guide, refer to Modbus Transactions (page 131).

Procedure

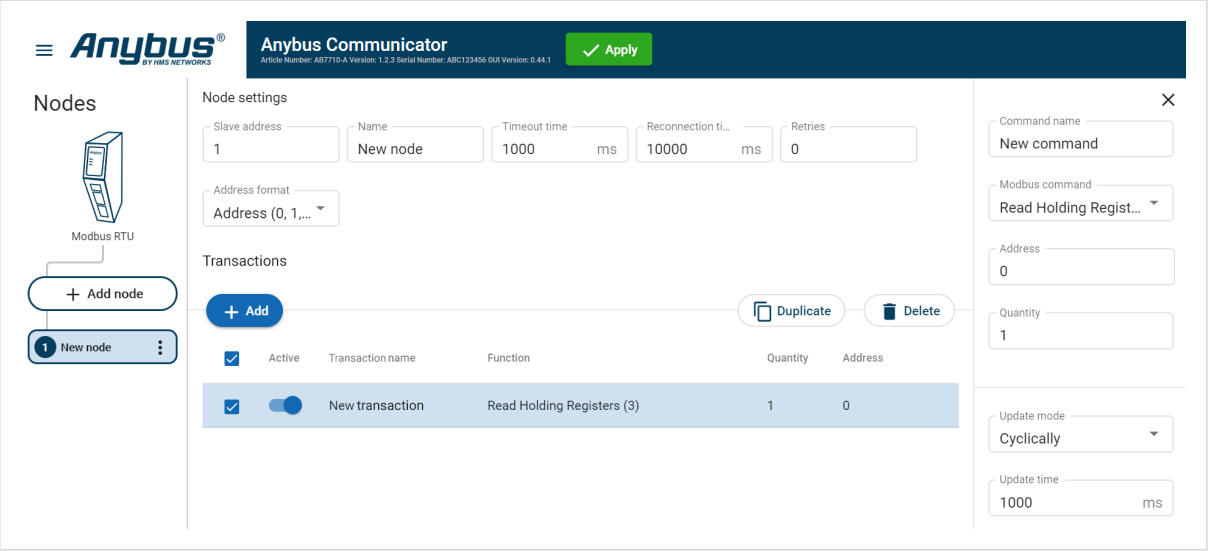


Figure 52. Modbus TCP Protocol

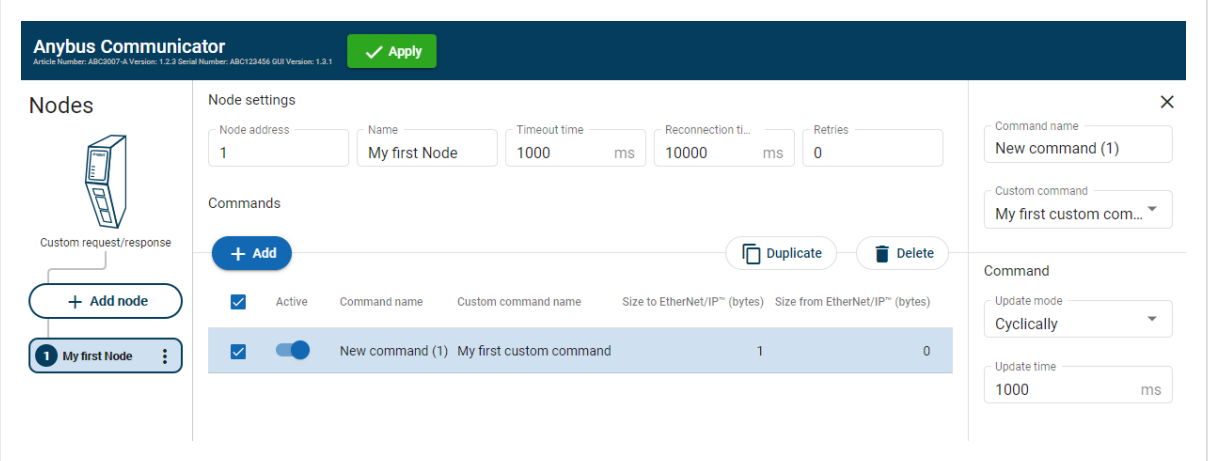


Figure 53. Custom request/response Protocol

1. In the node list, select a node to configure.
2. In the transactions list, select a transaction to configure.
   The transactions sidebar opens, on the right side of the screen.
3. Enter a transaction name.
   By default, the node is assigned the name New transaction.

4. Select a transaction type from the **Modbus transaction/Custom transaction** drop-down menu.
   The transaction type defines what the node should perform when the transaction is executed.
5. Configure the Command settings.

Setting	Value	Description
Transaction name	N/A	You can name the transaction to make it easier to identify.
Read quantity	1 to 125	Specifies the number of registers to read to follow in the read data field. Appear when Modbus transaction Read Write Multiple Registers (23) is selected.
Address	0 to 65 535	Specify the start address for the read/write transaction. The address acts as an address to the data position, where the data is read from or written to. Modbus holding register addresses starts at 0. Modbus address 0 = Register 1
Write quantity	Read Write Multiple Registers (23), 1 to 123	Specifies the quantity of registers to follow in the write data field. Appear when Modbus transaction Read Write Multiple Registers (23) is selected.
Quantity	Read Holding Registers (3) Read Input Registers (4), 1 to 125 Write Multiple Coils (15), 1 to 1968 Write Multiple Registers (16), 1 to 123 Read Coils (1) Read Discrete Inputs (2), 1 to 2000	The Quantity parameter appear when you select a Modbus transaction that can address more than one data object. Example when Quantity is set: For the Modbus Transaction Read Input Registers (4) you need to set the Quantity in order to define the array of data. Example when no Quantity is set: For the Modbus Transaction Write Single Coil (5) you do not need to set the Quantity parameter because there can not be an array of data. The transaction is used to write a single output to either ON or OFF in a remote device. For Write Single Coil (5), Write Single Register (6) and Mask Write Register (22) Quantity cannot be set.
Constant	0 to 255	The value of the Constant in the frame.
Data	0 to 512	The length of the data field.
Variable data	0 to 255	The maximum payload length of the variable data field.
Update mode	Cyclically On data change Single shot Change of state on trigger	Specify when a transaction shall be sent to the server. The transaction is issued cyclically, at the interval specified in the Update time parameter. Cyclically: The transaction is sent cyclically, at the interval specified in the Update time parameter. On data change: The transaction is sent when the data is changed. Single shot: The transaction is issued once at start up. Change of state on trigger: The transaction is triggered when the content of a specified byte changes. In the I/O data map, the node will be marked with a flash icon.In the I/O data map you can also configure the area map and the trigger byte address. See Trigger Byte (page 81).
Update time	3 ms to 60 000 ms	Update mode parameter must be set to Cyclically. The Update time parameter appear when Cyclically is select. Specify how often, in steps of 10 ms, the transaction are going to be issued.
Positive ack	N/A	When Positive Acknowledgement is enabled, the positive ack data byte in the I/O data map is incremented each time this transaction succeeds.
Negative ack	N/A	When Negative Acknowledgement is enabled, the negative ack data byte in the I/O data map is incremented each time this transaction fails.

6. To apply the settings, click **Apply** in the web-interface header, and follow the instructions..

7.8.6. Activate/Deactivate Transaction

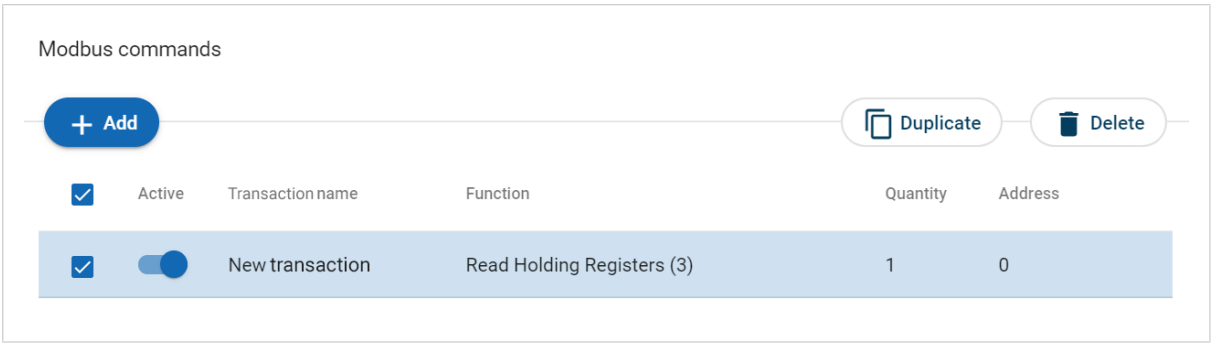


Figure 54. Activate/Deactivate Transaction

The transaction default status is **Active**.

To deactivate/activate a transaction, select the transaction and click the **slide toggle**.

7.8.7. Duplicate Transaction

When you duplicate a transaction, all settings are preserved.

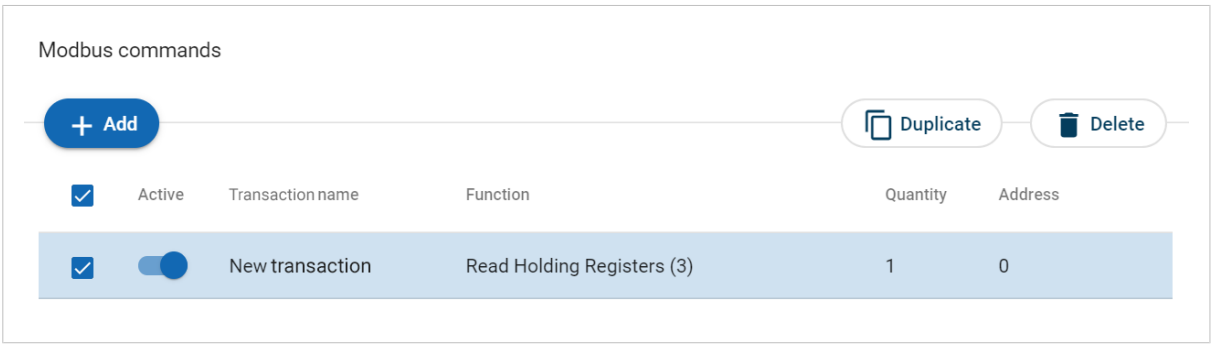


Figure 55. Duplicate transaction

To duplicate:

- One transaction, select the command and click **Duplicate**.
- Multiple transactions, select the checkbox in front of each transaction you want to duplicate and click **Duplicate**.
  The duplicated transaction is added at the bottom of the transactions list.

7.8.8. Delete Transaction

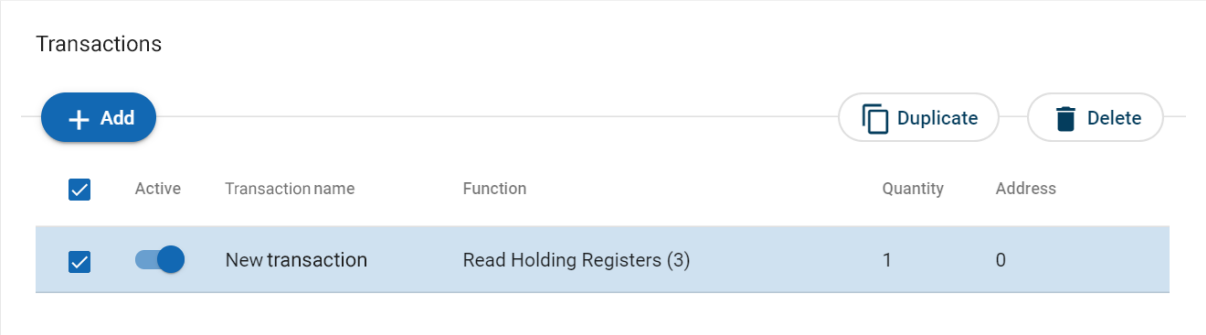


Figure 56. Delete transaction

1. To delete:
   - One transaction, select the transaction and click **Delete**.
   - Multiple transactions, select the checkbox in front of each transaction and click **Delete**.
2. To confirm, click **Yes**.

7.9. High Level Network Settings

Configure the PROFIBUS network settings.

7.9.1. To Use Automatic I/O Sizes

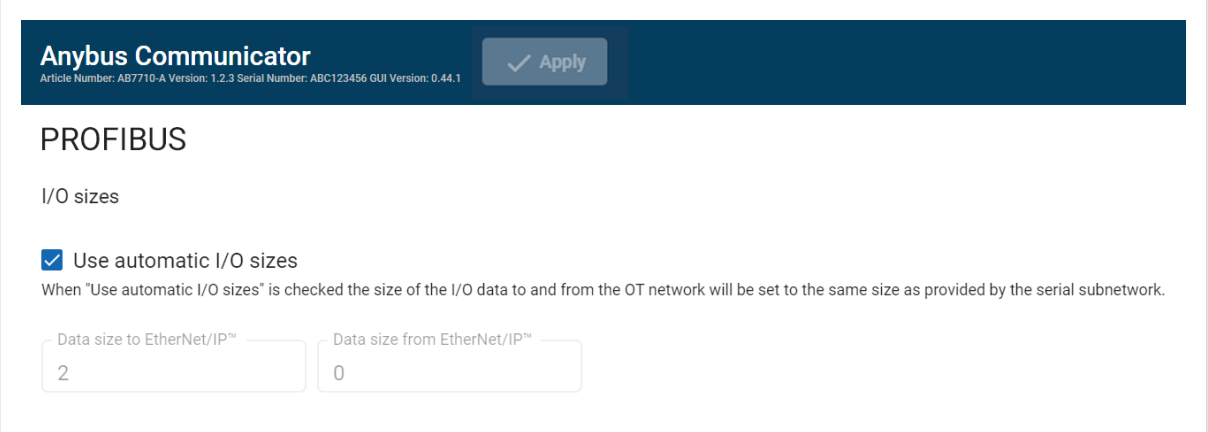


Figure 57. PROFIBUS, I/O sizes

By default, the Communicator is set to use automatic I/O sizes.

The size of the input data, Data Size to PROFIBUS, and the output data, Data Size from PROFIBUS, is determined by the subnetwork configuration.

In the Communicator built-in web interface, the **Use Automatic I/O Sizes** checkbox is selected.

7.9.2. To Configure I/O Sizes Manually

NOTE
The maximum data size in each direction is 244 bytes.

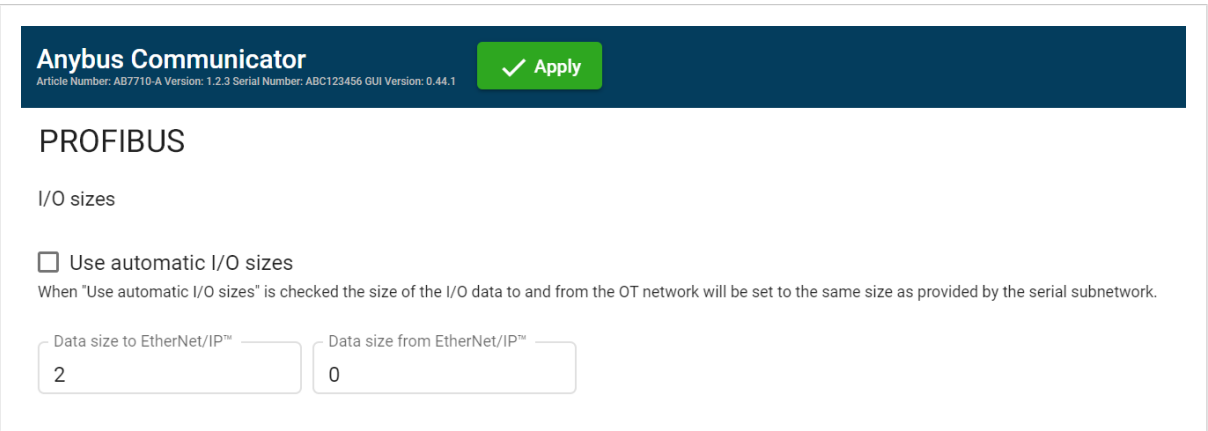


Figure 58. PROFIBUS, I/O sizes

1. Deselect the **Use Automatic I/O Sizes** checkbox.
2. Enter a value for Data Size to PROFIBUS and a value for Data Size from PROFIBUS.

7.9.3. PROFIBUS Address Settings

NOTE
Ensure that the Communicator rotary switches are sett to 000.

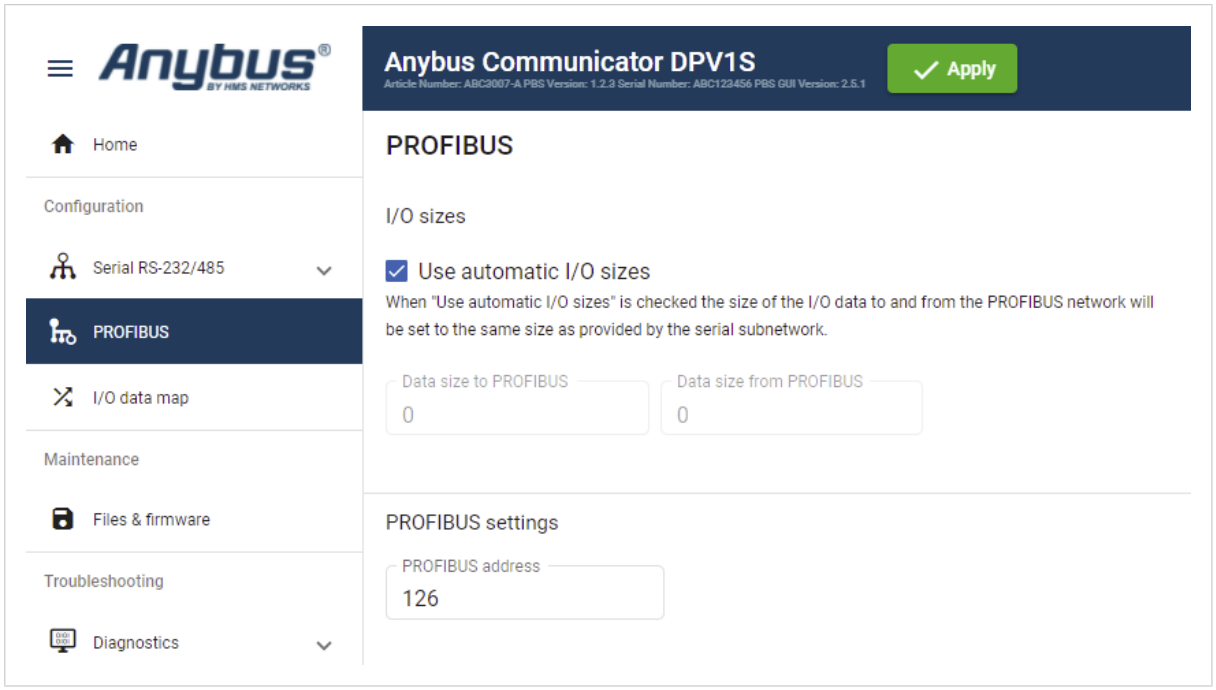


Figure 59. PROFIBUS address setting

In the PROFIBUS address field, enter the Communicator node address.

7.10. I/O Data Map

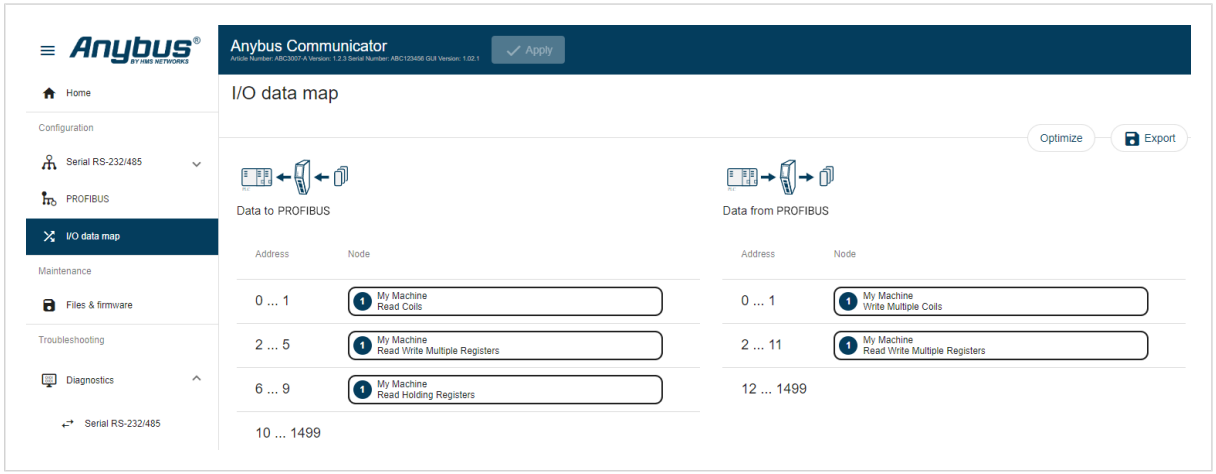


Figure 60. **I/O data map** page

On the **I/O data map** page the data communication between the subnetwork (Node) and the high level network (PLC) is mapped.

The allocated I/O area is auto-generated based on how the settings on the **Serial communication** page and the **Nodes and transactions** page are configured.

It is possible to set the I/O area manually, if you want to pro-actively allocate more I/O for future expansions without re-configuring the PLC. See To Configure I/O Sizes Manually (page 76).

There are three areas: **Data from PROFIBUS**, **Data to PROFIBUS**and **General area**. See Map Area (page 81).

7.10.1. Optimize the I/O Data Map

The optimize function is used to automatically remove gaps between the mapping.

IMPORTANT

Optimize remove gaps between the data objects in the map and should be used with care on already commissioned systems. Expected mapping in the PLC may change.

NOTE

If you optimize the I/O data map, the current I/O data map will be overwritten.

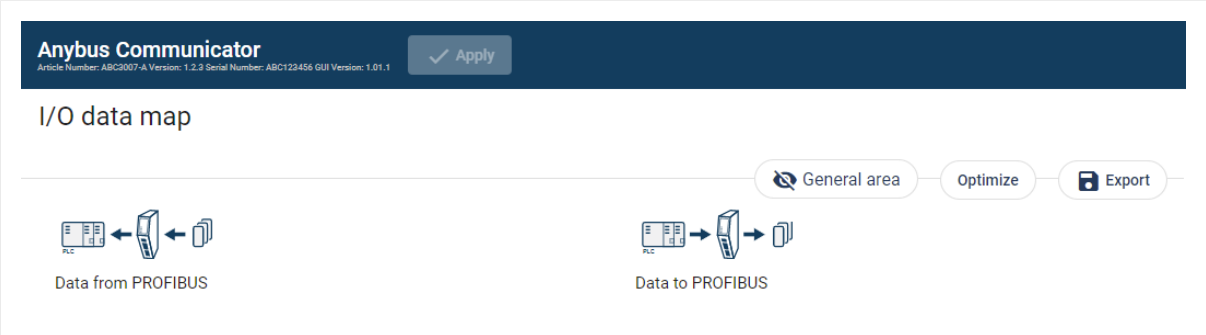


Figure 61. I/O data map page, Optimize

To optimize the map:

1. Click **Optimize**.
2. To confirm, click **OK**.

7.10.2. Map Area Transactions Order

To change the order of the transactions in a map area, drag and drop the desired transaction to a new location.

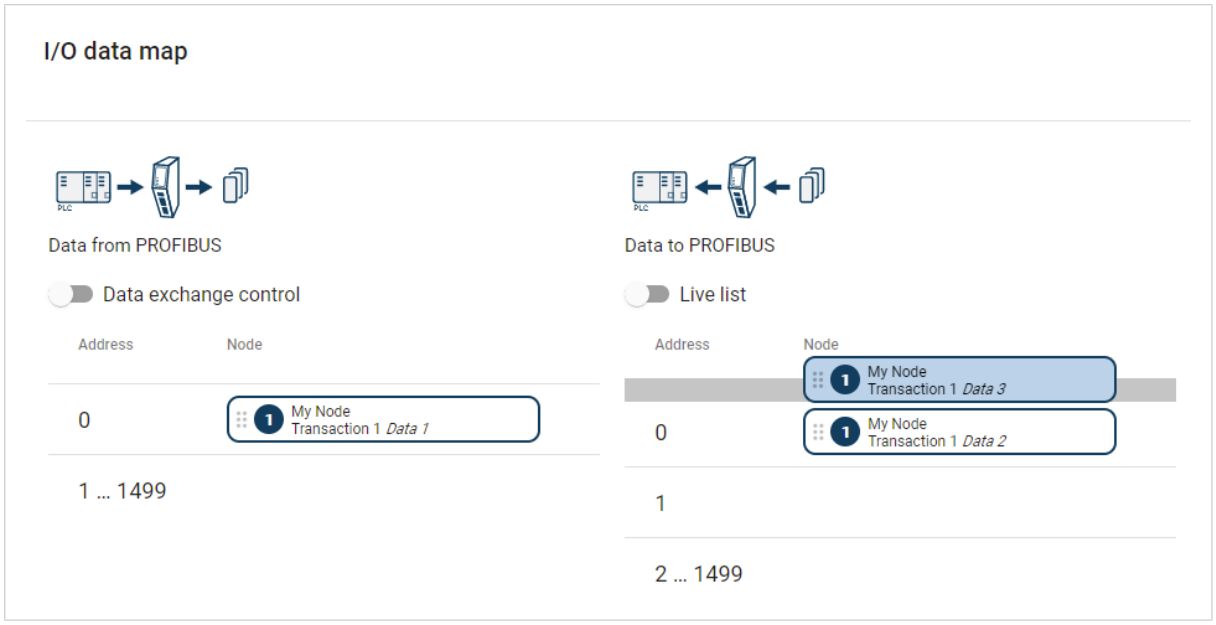


Figure 62. I/O data map, change the order of transactions

Transactions can not share the same I/O are.

If multiple transactions are placed in the same I/O area, the area is highlighted.

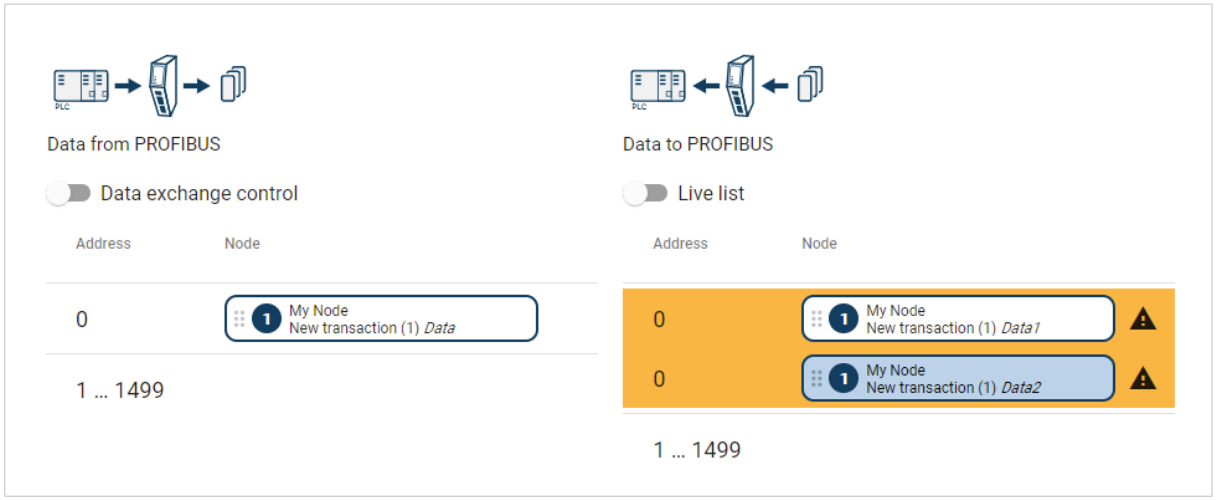


Figure 63. Highlighted I/O area

7.10.3. Map Area

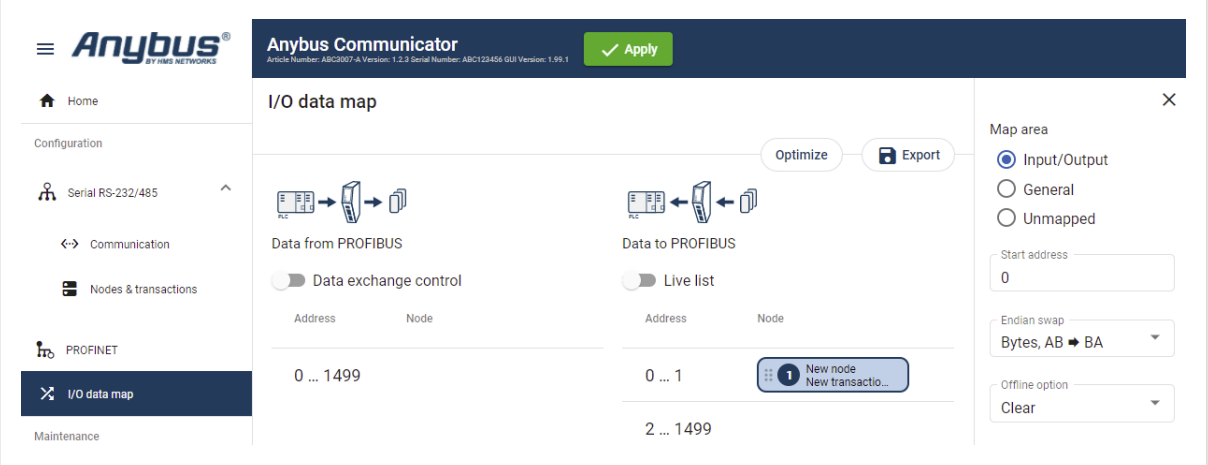


Figure 64. I/O data map page, Map area options

Map area options
You must specify the map area to use for each transaction in the I/O data map.

Select one of the following **Map area** options:

- **Input/Output**: The transaction data is sent/recieved to/from the high level network.
- **General**: This area is used for transferring transaction data between individual nodes on the subnetwork. When General is selected, the transaction data cannot be accessed from the high level network.
- **Unmapped**: The transaction data is not used.

Start address
For Input/Output and General, you can enter a start address for the transaction data.

7.10.4. Trigger Byte

Trigger byte is used to enable/disable the trigger functionality for the response.

When Trigger byte is enabled, the Communicator increases the trigger byte by one when the Communicator receives new data from the subnetwork.

The Trigger byte is stored in the **Data from PROFIBUS** area or the **General** area.

The location of the trigger byte is specified by the address.

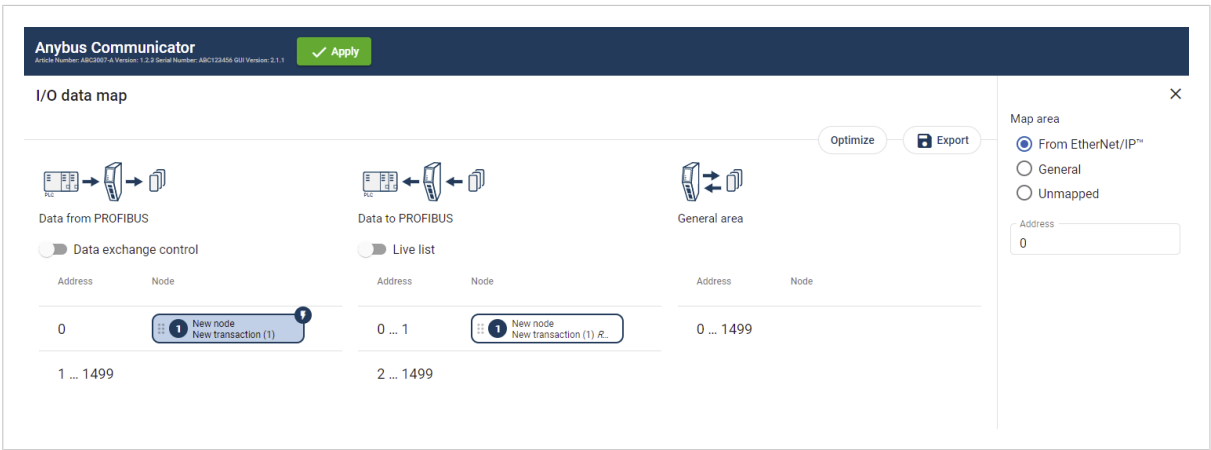


Figure 65. I/O data map

How to Enable Trigger Byte on a Node
Procedure

1. Navigate to the **Nodes & transactions** page.
2. Select the decried node and transaction.
3. In the transaction sidebar **Update mode** menu, select **Change of state on trigger**.
4. Navigate to the **I/O data map** page.
5. The transaction with the trigger byte enabled is marked with a flash icon.
   To open the **Map Area** sidebar, click on the flash icon.
6. In the **Map Area** sidebar, specify the map area to use and the trigger byte address:
   Map area options

   - **From PROFIBUS**: The trigger byte is stored in the I/O data map Data from PROFIBUS area.
   - **General**: The trigger byte is stored in the I/O data map General area.
   - **Unmapped**: The transaction data is not used.

   Address

   - Enter an Address, the location in the specified Map area (From PROFIBUS or General) where the trigger byte will be saved.
     Value: 0 (default) to 1499

7.10.5. Endian Swap

By default PROFIBUS uses the big-endian format.

Big-endian
The big-endian format places the most significant byte of the data at the byte with the lowest memory address.

Little-endian
The little-endian format places the least significant byte of the data at the byte with the lowest memory address.

To convert between big-endian and little-endian you must reverse the byte order.

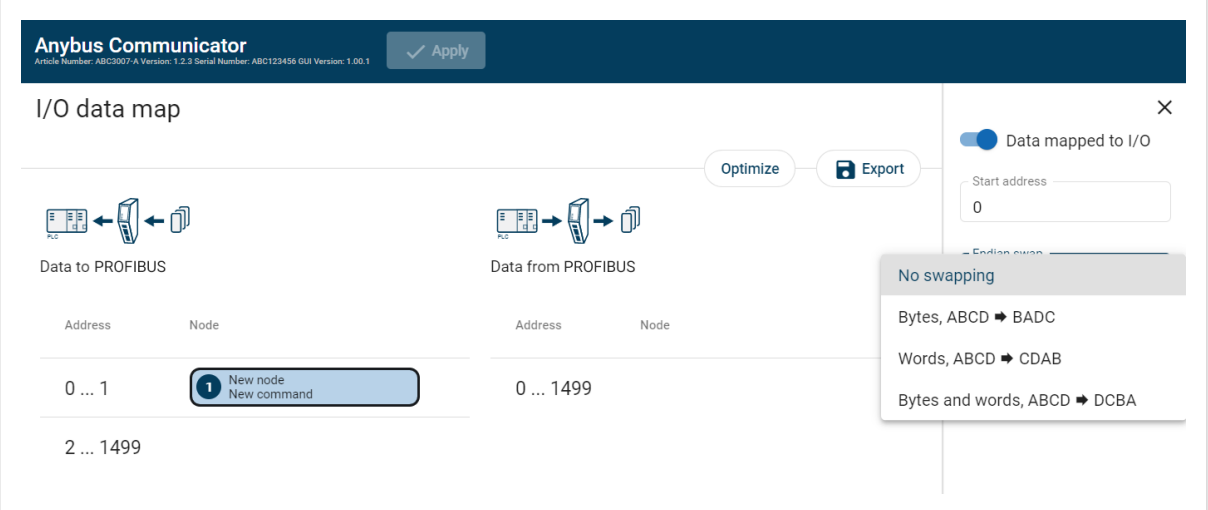


Figure 66. **I/O data map**, Endian swap

To reverse the byte order:

1. In the web-interface left sidebar menu, click **I/O data map**.
2. In the data map, select the transaction for which you want to do swap the byte order.

3. Select the swapping type from the **Endian swap** drop-down menu.

Setting	Description
No swapping	Default setting No swapping is performed on the data.
Byte swap	Swap 2 bytes A B C D becomes B A D C
Word swap	Swap 4 bytes A B C D becomes C D A B
Byte and Word swap	A B C D becomes D C B A

4. To apply the settings, click **Apply** in the web-interface header, and follow the instructions.

7.10.6. Offline Option

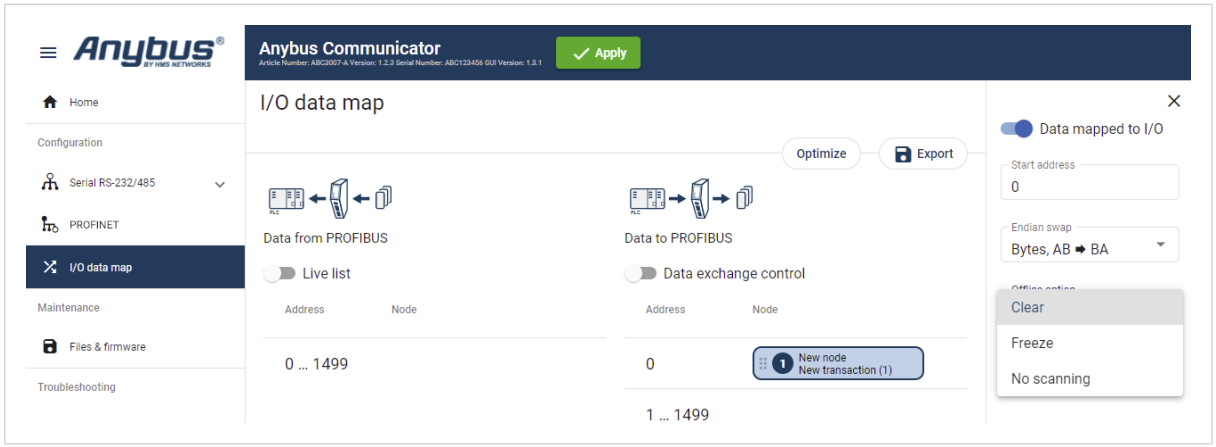


Figure 67. I/O data map page, Offline options

Offline mode is used to define what data to send if the network connection or connection with a specific node is lost.

You must specify the offline mode to use for each transaction on the I/O data map.

Select one of the following **Offline options**:

- **Clear (Default)**: The data is cleared and the value 0 is sent.
- **Freeze**: The Communicator holds the value until the connection is restored.
- **No scanning**: Stop sending this transaction on the sub-network. This option is only valid for produce and request transactions.

7.10.7. Live List

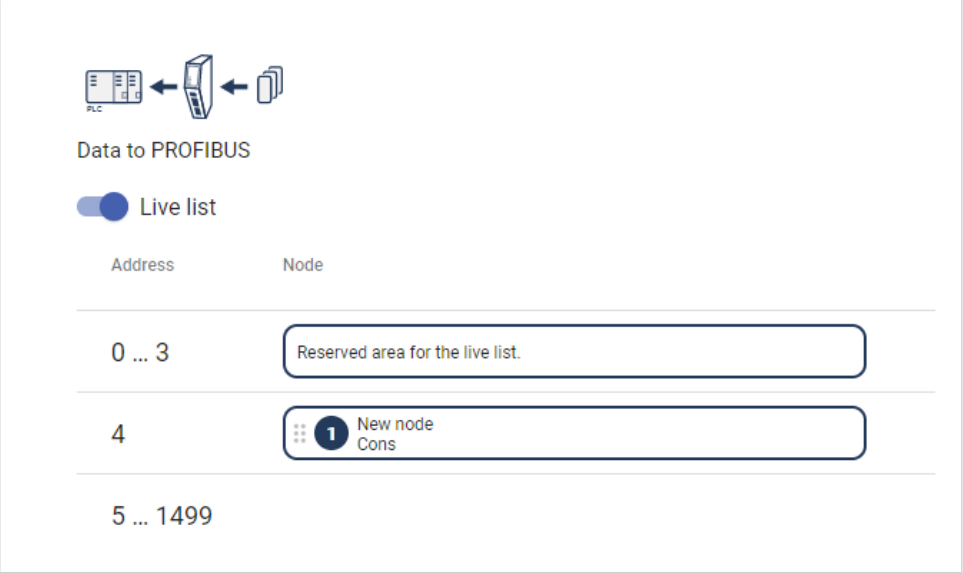


Figure 68. I/O data map page, Live list enabled

By default **Live list** is disabled.

When **Live list** is enabled, the first four bytes of process data on the PROFIBUS network contain the live list.

The **Live list** holds 32 bits.

Each bit in the **Live list** can hold the status for a total of 32 nodes connected to the Communicator.

The bit is 0 when the bit does not correspond to the configured node. For example, this occurs when the number of configured nodes is less than 32.

Each bit is 1 when the corresponding node is online.

7.10.8. Data Exchange Control

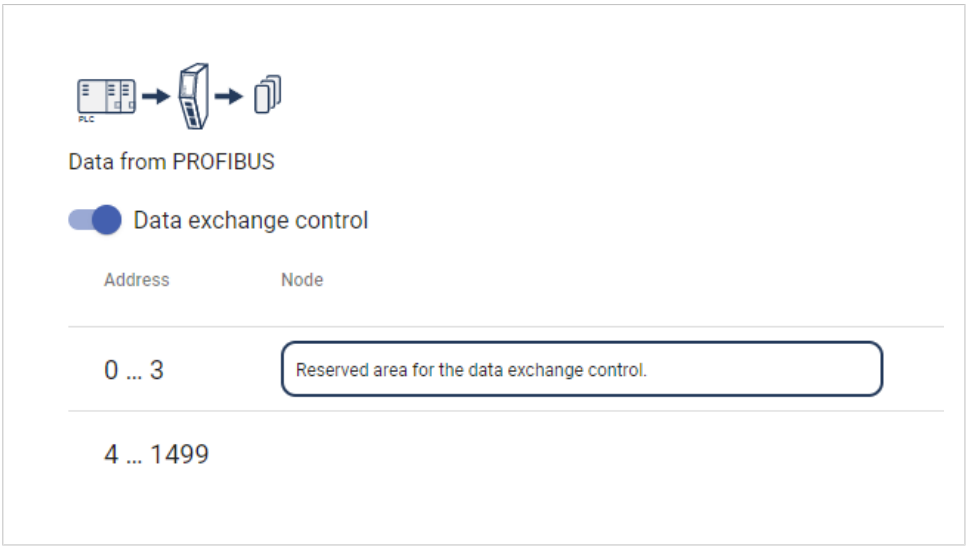


Figure 69. I/O data map, **Data exchange control** enabled

By default **Data exchange control** is disabled.

When **Data exchange control** is enabled, the first four bytes of process data on the PROFIBUS network contain the data exchange control.

The **Data exchange control** holds 32 bits.

Each bit in the **Data exchange control** can be used to enable/disable data exchange for individual nodes on the subnetwork.

If control bit does not correspond to a configured node, the control bit is ignored. For example, this occurs when the number of configured nodes is less than 32.

The node order in the **Data exchange control** 32 bit array always matches the Live List.

When the data exchange is enabled for the corresponding node, the control bit is 1.

7.11. Apply Configuration

Before You Begin

NOTE
When you apply the configuration, any existing configuration is overwritten.

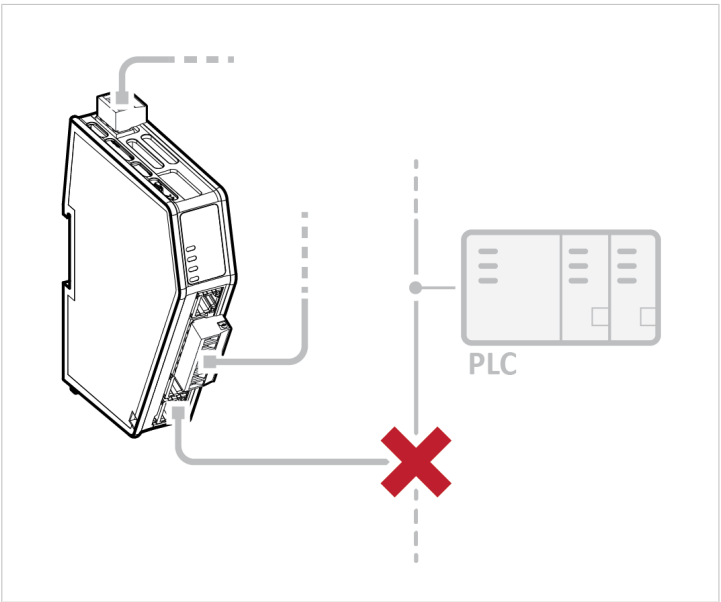


Figure 70. Disconnect the Communicator from the PROFIBUS network

Before you can apply the configuration, ensure that there is no active communication on the PROFIBUS network where the Communicator is connected.

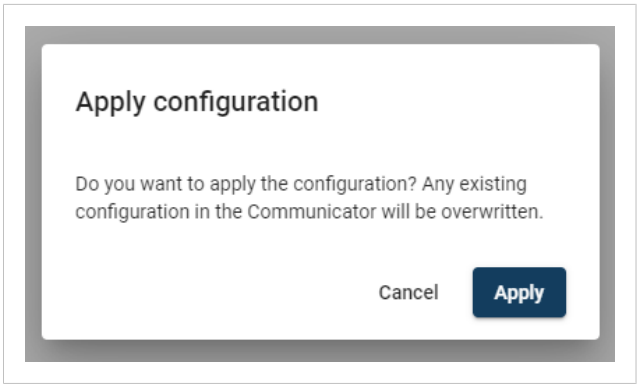
Procedure

To make the settings take effect, download the configuration to the Communicator:

1. In the web-interface header, click **Apply**



2. To confirm download, click **Apply**.



- The configured settings are downloaded and applied to the system.

7.12. To Use an Existing Configuration

When you have configured a Communicator and want to use the same settings to configure additional Communicator, do the following.

Procedure

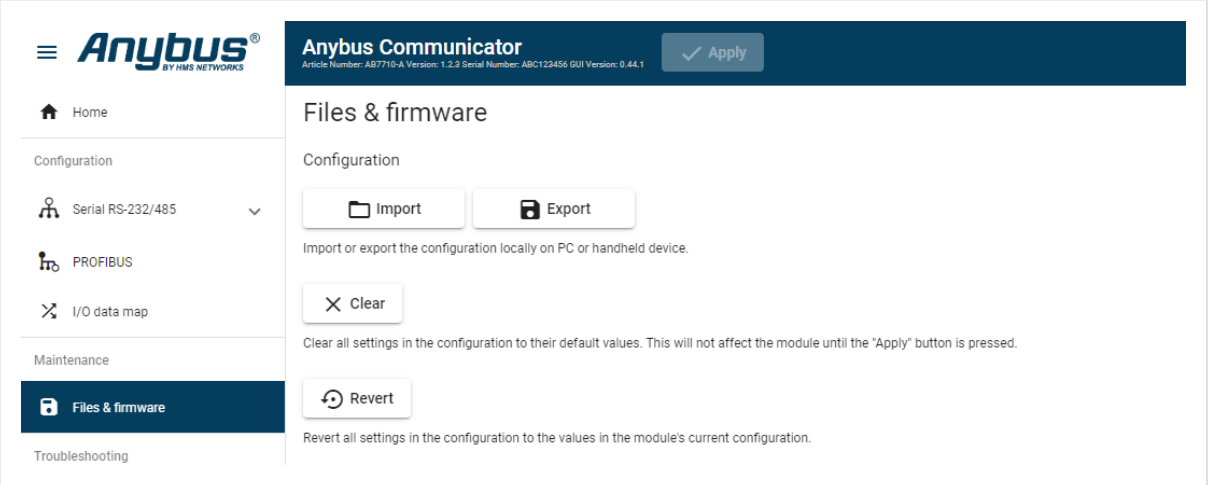


Figure 71. **Files & firmware** page

In the built-in web-interface of the Communicator with the configuration you want to use:

1. On the **Files & firmware** page, click **Export**
   The configuration is saved in an conf file and downloaded to your PC.

In the built-in web-interface of the new Communicator to be configured:

2. On the **Files & firmware** page, click **Import**
3. In the Import configuration window, click **Select file (.conf)**.
4. In the Open dialog box, browse to and select the configuration file and click **Open**.
5. To import the configuration file, click **Import**.

Result

All the configuration settings are imported.

To apply the settings, click **Apply** in the web-interface header, and follow the instructions.

7.13. To Use an Communicator Classic Configuration

Before You Begin

NOTE

Only the Communicator Classic serial configuration settings can be imported.

The I/O data map and high-level network settings are not supported and must be set manually in the Communicator built-in web interface.

Communicator Classic Classic configuration intended use

The intended use of the Communicator Classic configuration import is to:

- convert custom protocols.
- get a new Communicator unit up and running quickly and then complete the configuration in the Communicator built-in web interface.

Configuration files containing standard Modbus RTU commands

- It is not recommended to import Communicator Classic configuration files containing standard Modbus RTU commands if you need to be able to easily make change to the imported configuration.
- When standard Modbus RTU commands are imported, they are converted to Custom Request/Response transactions. See Communication Serial Protocol (page 45).
- The behavior of the imported standard Modbus RTU commands is preserved, but adding or changing Modbus RTU commands in Custom Request/Response mode is more difficult than in Modbus RTU mode.

Procedure

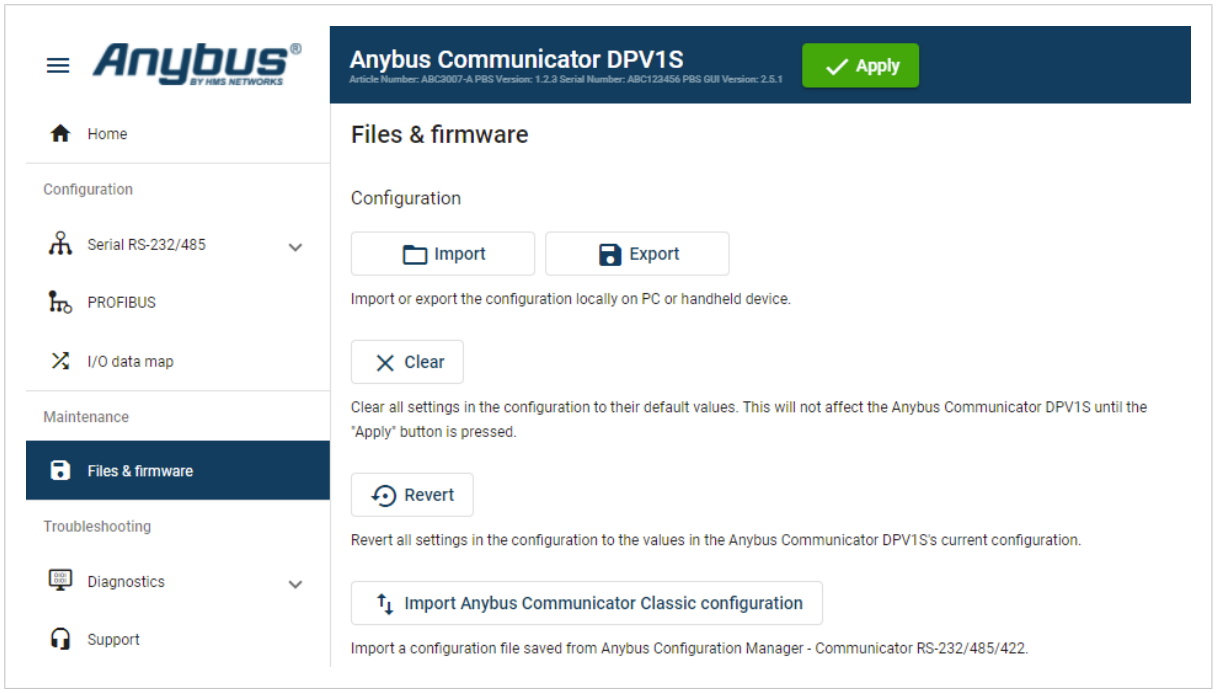


Figure 72. **Files & firmware** page

1. On the **Files & firmware** page, click **Import Anybus Communicator Classic configuration**
2. In the **Import Anybus Communicator Classic configuration** window, click **Select file (.cfg)**.

3. In the Open dialog box, browse to and select the configuration .cfg file and click **Open**.
4. If you want to import a name file, click **Select name file (.cfx)**.
5. In the Open dialog box, browse to and select the configuration .cfx file and click **Open**.
6. To import the configuration, click **Import**.

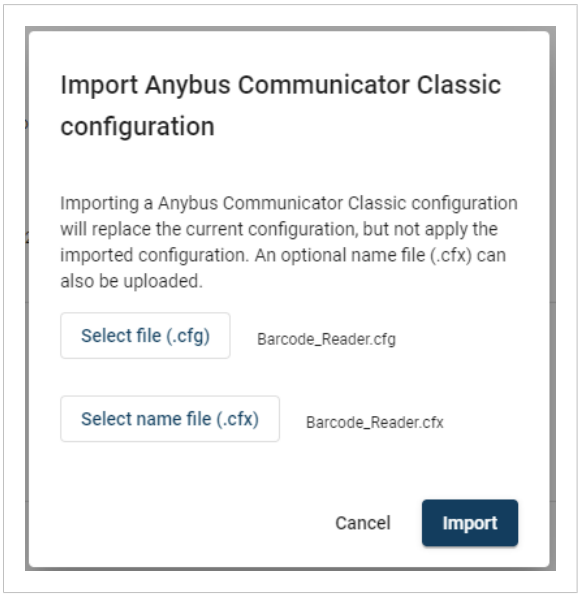
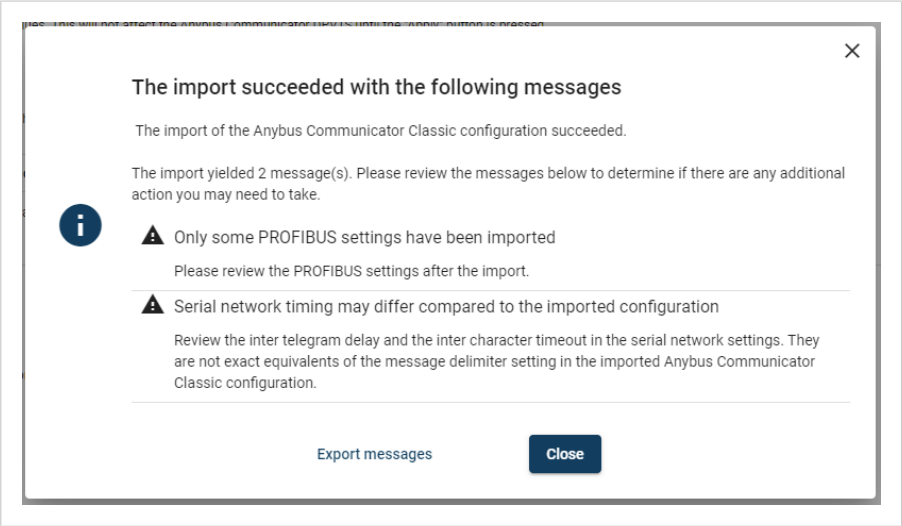


Figure 73. Example, selected .cfg and .cfx files

Result

The Communicator Classic serial configuration settings are imported.



A window with messages about the imported configuration appear.

In the list you can view the settings that are fully supported or adjusted to work with Communicator and which settings that are not supported and must be set manually in the built-in Communicator interface.

To export the messages in an Excel XLS file, click **Export Messages**.

Figure 74. Example, list with messages about the import

To apply the settings, click **Apply** in the web-interface header, and follow the instructions.

8. PLC Configuration

8.1. Export I/O Data Map

When configuring the communication between the PLC and the nodes on the subnetwork, use the I/O data map as a specification to ensure that the transactions match.

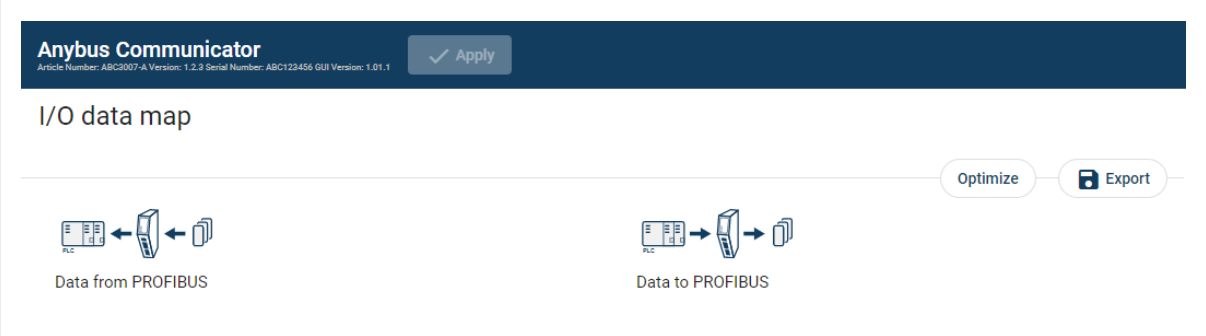


Figure 75. I/O data map page

On the I/O data map page you can export the I/O data map in an Excel XLS file, where all the nodes and transactions are listed.

To export the I/O data map:

1. Click **Export**.
   An Excel XLS file with the mapping is downloaded to your PC.

8.2. Export Product GSD File

Option if the PLC program requires a product file, GSD (General Station Description) file, describing how the Communicator can be used on the high level network.

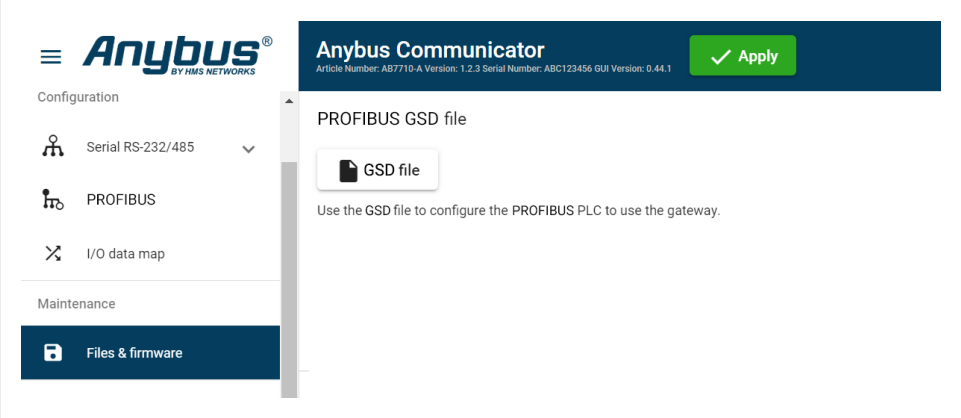


Figure 76. Export Product GSD File

You find the *PROFIBUS GSD* file on the Communicator built-in web interface **PROFIBUS** page, **Files & firmware** page and on the **Support** page.

To export the GSD file:

1. Click **GSD file**.
   The GSD file is downloaded to your PC.

8.3. How to Use GSD File Modules

When configuring the PROFIBUS communication, ensure that the I/O sizes in the PROFIBUS master match the I/O sizes sett on the Communicator **PROFIBUS** page, see High Level Network Settings (page 76).

Import the Communicator GSD file into TIA portal and combine the modules to get the correct I/O size.



NOTE

Fill unused module slots with the value Empty.

Table 1. Modules available in the Communicator GSD file

Mod_name	Config	Module-Reference	In/Out	Size
Empty module	0x00	1	N/A	0
1 byte input module	0x90	2	In	1 byte
2 bytes input module	0xD0	3	In	1 word (2 bytes)
4 bytes input module	0xD1	4	In	2 words (4 bytes)
8 bytes input module	0xD3	5	In	4 words (8 bytes)
16 bytes input module	0xD7	6	In	8 words (16 bytes)
32 bytes input module	0xDF	7	In	16 words (32 bytes)
64 bytes input module	0x40, 0xDF	8	In	32 words (64 bytes)
128 bytes input module	0x40, 0xFF	9	In	64 words (128 bytes)
1 byte output module	0xA0	10	Out	1 byte
2 bytes output module	0xE0	11	Out	1 word (2 bytes)
4 bytes output module	0xE1	12	Out	2 words (4 bytes)
8 bytes output module	0xE3	13	Out	4 words (8 bytes)
16 bytes output module	0xE7	14	Out	8 words (16 bytes)
32 bytes output module	0xEF	15	Out	16 words (32 bytes)
64 bytes output module	0x80, 0xDF	16	Out	32 words (64 bytes)
128 bytes output module	0x80, 0xFF	17	Out	64 words (128 bytes)

Example 8. How to combine modules

In this example we configure the following sizes:

- Data size to PROFIBUS: 95 bytes
  Combine modes: 64 + 16 + 8 + 4 + 2 + 1
  Size: [0xDF + 0xD7 + 0xD3 + 0xD1 + 0xD0 + 0x90]
- Data size from PROFIBUS: 58 bytes
  Combine modes: 32 + 16 + 8 + 2
  Size: [0xEF + 0xE7 + 0xE3 + 0xE0]

Example 9. Fill unused module slots

The gateway cannot move positions and it expects data of a certain size at each position between the combine modes.

Therefore, unused module slots needs to be filled with the value Empty.

In this example we use an I/O size of 35 bytes.

Combine modes: 1 + 2 + 32

To fill the positions, we add Empty where the Combine modes 4, 8, 16, 64 and 128 are expected.

1 byte + 2 bytes + Empty + Empty + Empty + 32 bytes + Empty + Empty

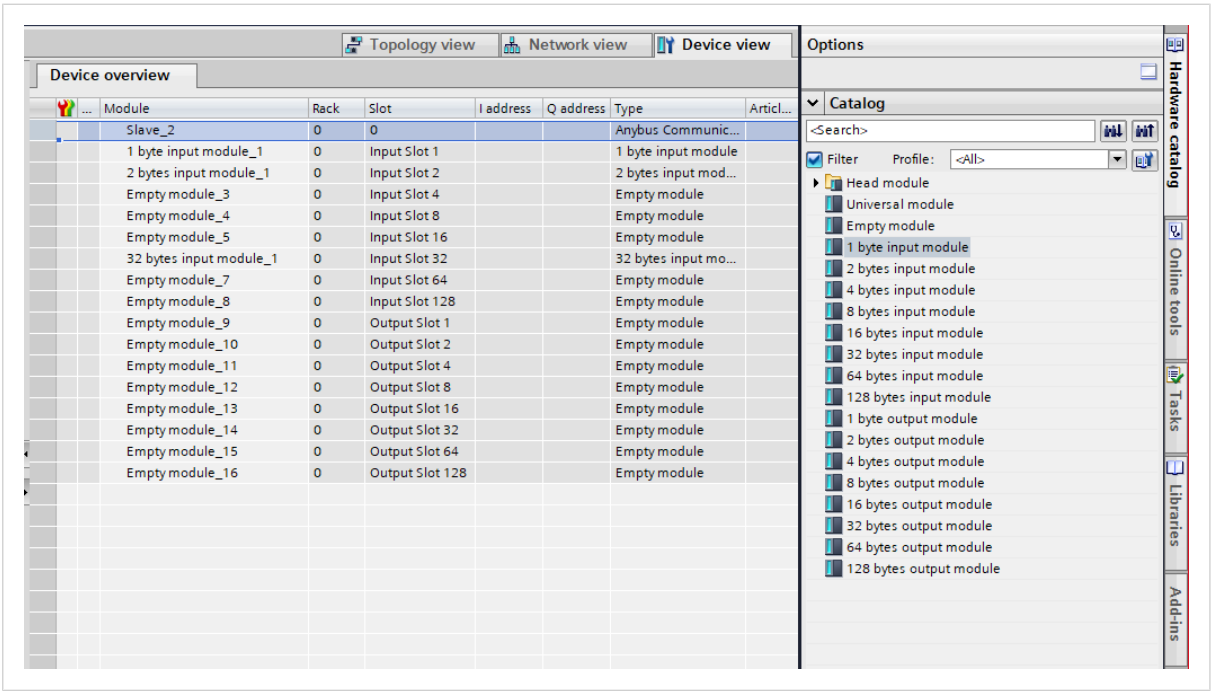


Figure 77. Example, fill unused module slots in TIA Portal

9. Verify Operation

Before You Begin

Ensure that the Communicator is connected to your PC, to a power supply and to the OT network.

See Installation (page 13).

9.1. Communicator Status Monitor

On the Home page, you can get a quick overview of the network and the Communicator operating status.

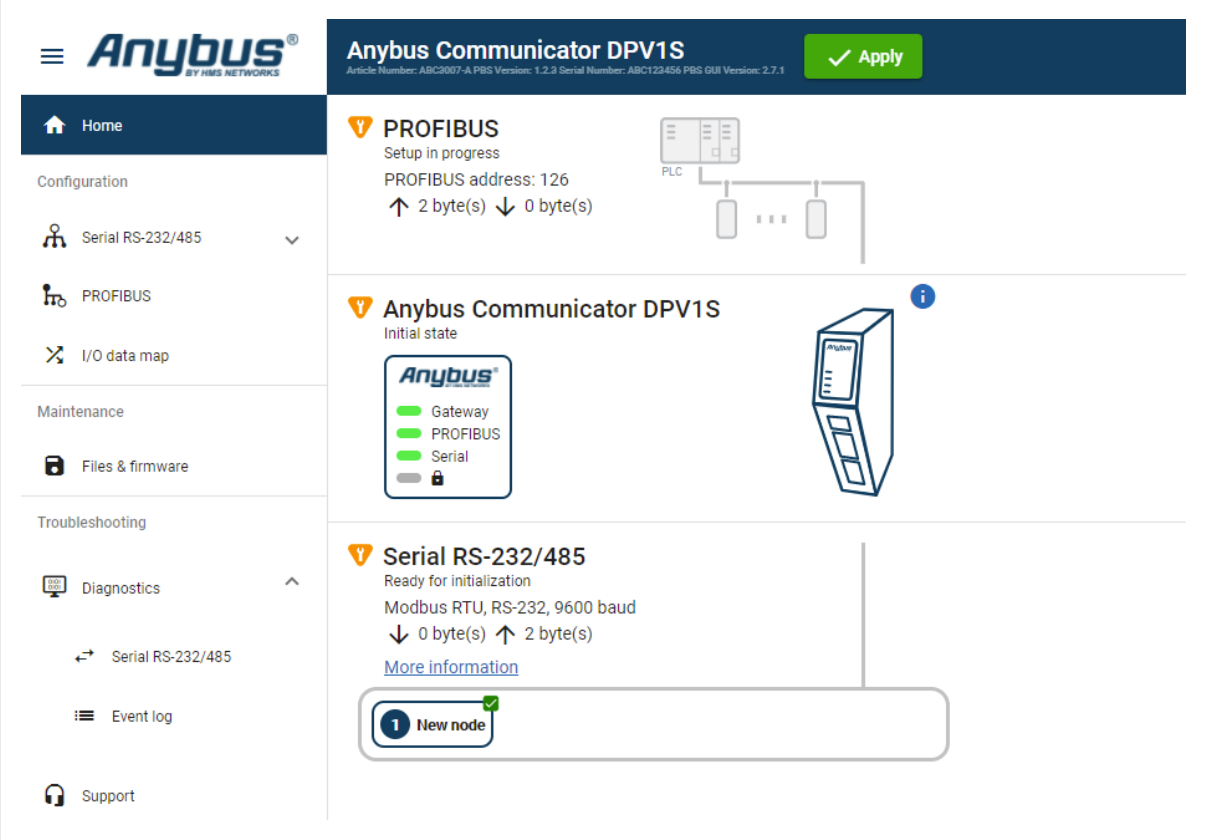


Figure 78. Home page

Gateway status

Overview the Communicator LED indications remotely.

Refer to Communicator LED Indicators (page 96).

Node Status

Overview the status for each node added to the subnetwork.

Network Status and Settings

Overview communication status and the current networks settings.

Status Symbols

Symbol	Description
	Internal error has occurred and operation cannot be guaranteed.
	Out of Specification.
	Check Function: • Initial state where non network components are started and configured. • Network startup in progress. • Invalid configuration detected.
	Normal operation.

9.2. Communicator LED Indicators

NOTE
Before you can verify operation you must configure the Communicator.

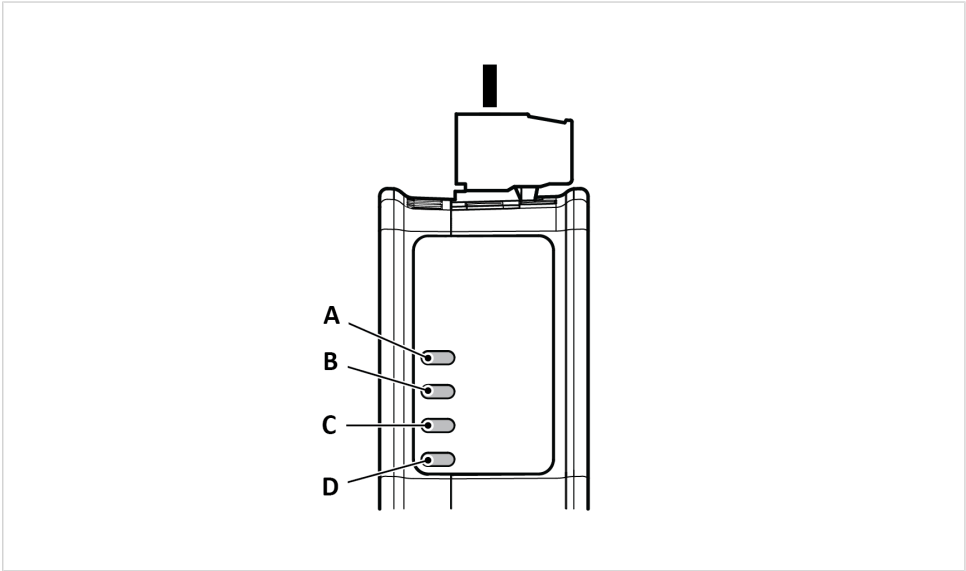


Figure 79. Communicator status (A), High level Network/Client (B), Subnetwork 2 (C) and (D) Security Switch

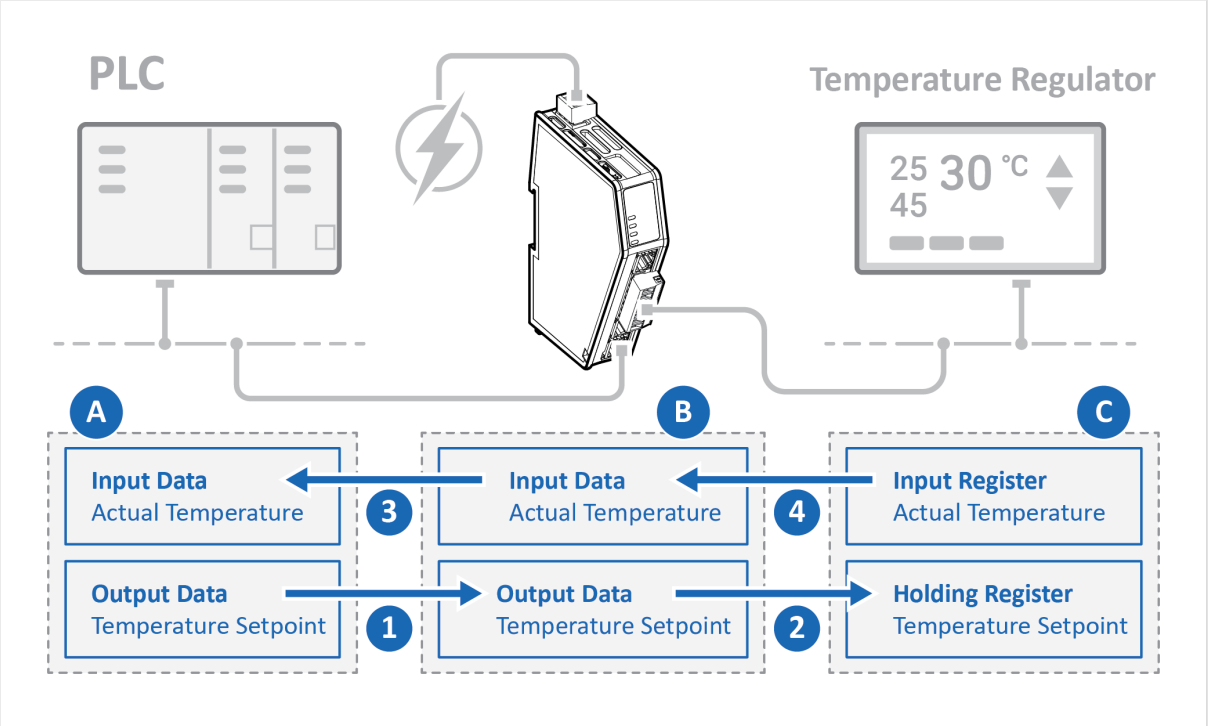
	LED A	LED B	LED C	LED D
Operation Status	**Gateway status**	**PROFIBUS**	**Subnetwork**	**Security switch**
Off	No power	No data exchange	No power/Exception/ Subnetwork not running/Node is switched off via a control word	No power/Security switch is unlocked/ Exception/Fatal error
Green, flashing	Startup phase	Clear, data exchange	Running, one or more nodes are offline	N/A
Green	Operational	Operate, data exchange	Running	Security switch is locked
Red	Exception/Fatal error	FATAL event	Fatal error	N/A
Red, one flash	Exception/Fatal error	Parameterization error	Fatal error	N/A
Red, two flash	Exception/Fatal error	PROFIBUS configuration error	Fatal error	N/A
Green/Red, flashing	Power up self-test/ Firmware update/ Firmware recovery	N/A	N/A	N/A

10. Use Cases

10.1. Temperature Regulator - Modbus RTU Use Case

10.1.1. About the Use Case

The purpose of this use case is to explain how to use the **Modbus RTU** serial protocol.



In this use case we use the Communicator to enable data exchange between an Temperature Regulator and a PLC.

The use case describes how to map the communication in the Communicator.

The Temperature Regulator is connected to the serial subnetwork via a custom RS-232 protocol.

The PLC is connected to an PROFIBUS network (high level network).

Figure 80. Temperature Regulator - Modbus RTU Use Case

10.1.2. Before You Begin

- Connect the Communicator configuration port to your computer.
- Power on the Communicator.
- Ensure that your computer can find the Communicator IP address.
- Enter the Communicator built-in web interface of the.

For more information refer to Communicator Configuration (page 39).

10.1.3. Choose Serial Protocol Type

The Temperature Regulator is using a request/response protocol to access parameters addressed with index and sub index.

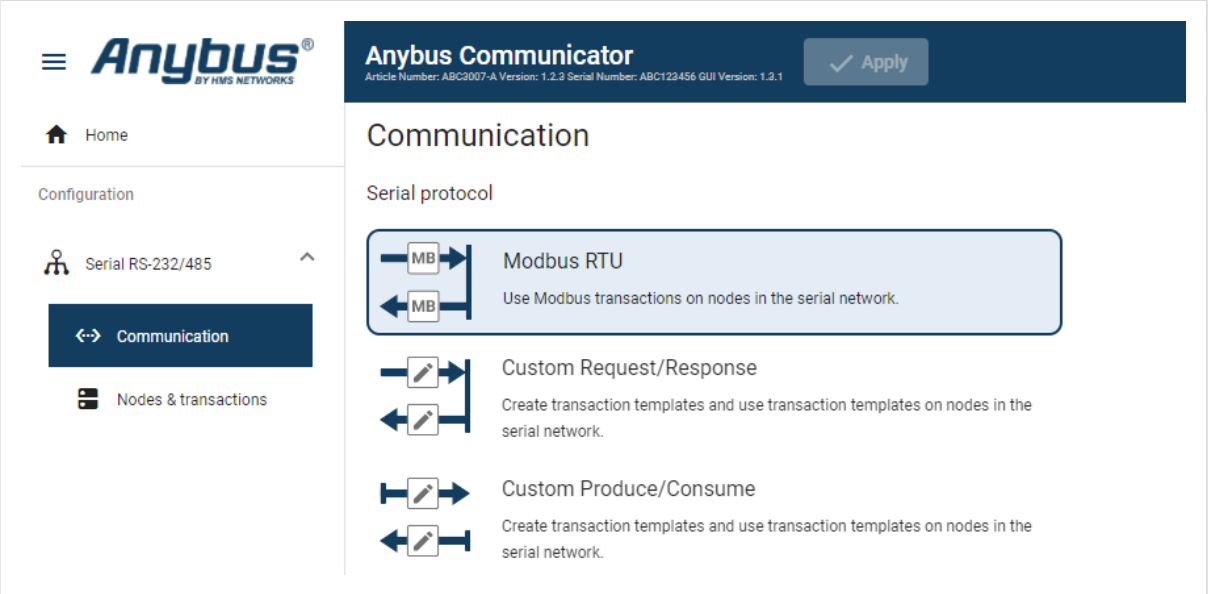


Figure 81. Communication page, **Modbus RTU**

- On the **Serial RS232/485** page, select **Modbus RTU**.

10.1.4. Setup Serial Communication

Set up the communication between the Communicator and the Temperature Regulator.

In the **Serial RS232/485** page, configure the **Communication** settings.

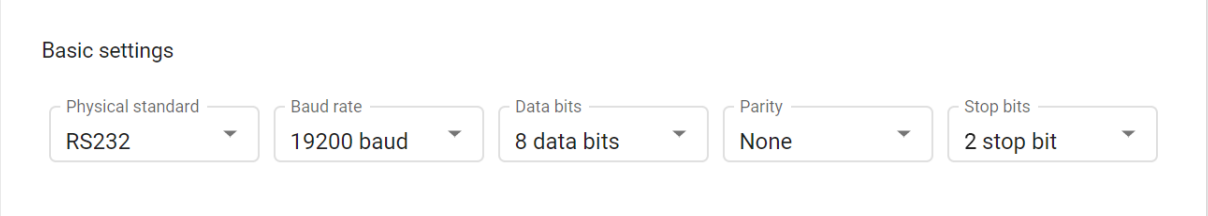


Figure 82. Serial RS232/485, Basic settings

Table 2. Used the following settings:

Frame objects	Value
Physical standard	RS-232
Baud rate	19200 baud
Data bits	8 bits
Parity	None
Stop bits	2 stop bit

10.1.5. Set Up the Node

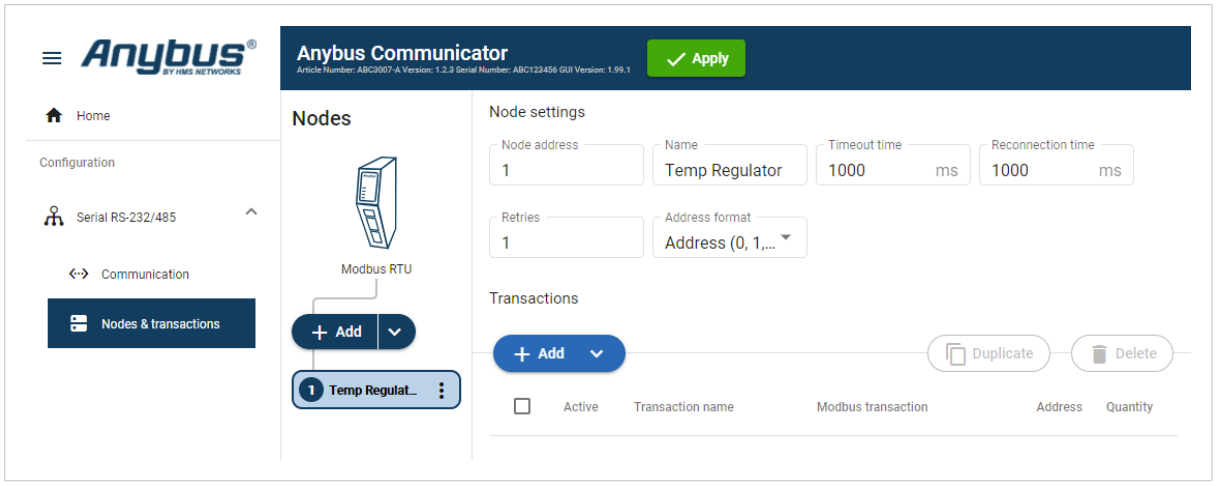


Figure 83. Add the Temperature Regulator node

1. Add a node and select it.
2. In Node settings, configure the node with the following settings:

Node settings	Value
Slave address	240
Name	Temp Regulator
Timeout time	1000 ms
Reconnection time	1000 ms
Retries	1
Address format	Register

10.1.6. Setup the Transactions

Set up the communication between the node and the master.

In this example, the communication between the Temperature Regulator and the PLC.

The Temperature Regulator has two Modbus transactions:

- One registers holding the setpoint temperature.
- One registers holding the actual temperature.

Configure the temperature setpoint transaction

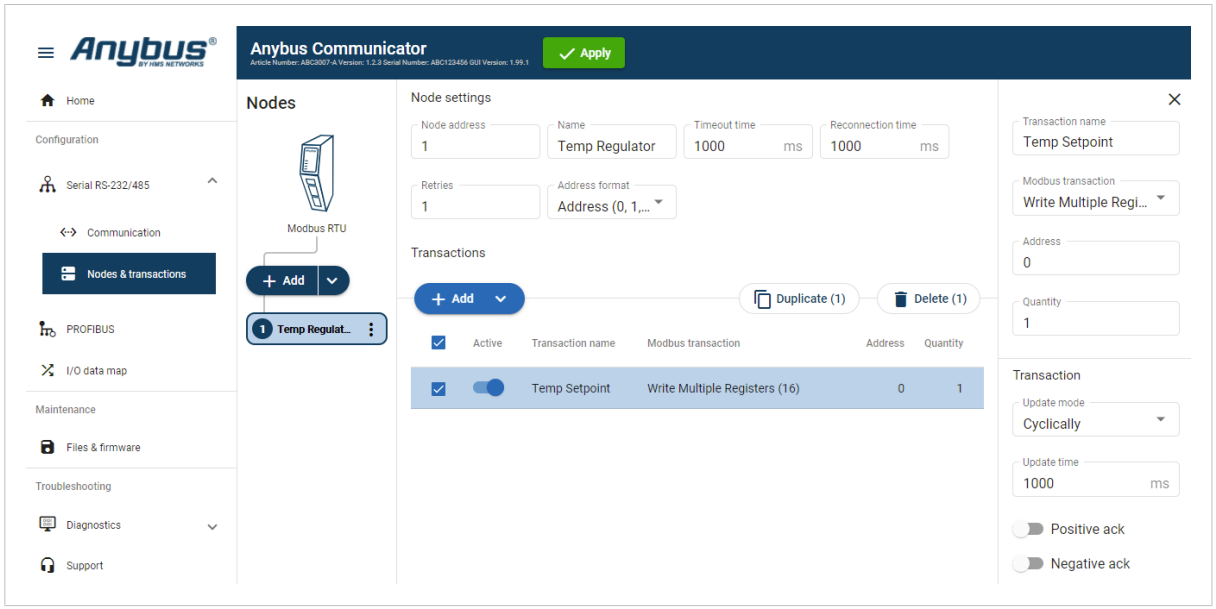


Figure 84. Temperature setpoint transaction

1. Select the **Temp Regulator** node.
2. To add a transaction, click **Add**.
3. Select the transaction to configure.
4. In the transaction sidebar, on the right side of the screen.
   Enter values for the transaction settings.

   Table 3. Setpoint temperature transaction settings

Setting	Value	Description
Transaction name	Temp Setpoint	Give the transaction a name.
Modbus transaction	Write Multiple Registers (16)	The PLC writes a block of contiguous registers to the temperature regulator.
Address/ Register	Address: 0 Register: 1	Address 0 is Register 1.
Quantity	1	The transaction will address one data object.
Update mode	Cyclically	The temperature regulator sends a new message cyclically, every 1000 ms.
Update time	1000 ms	The update cycle is 1000 ms.

Configure the actual temperature transaction

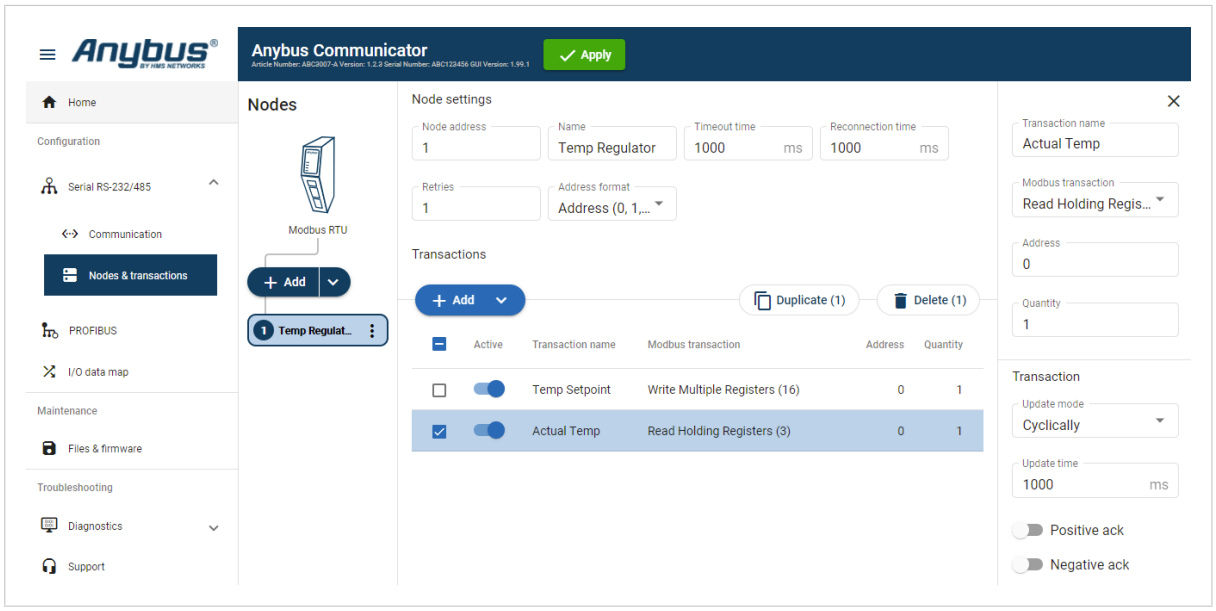


Figure 85. Actual temperature transaction

1. To add a second transaction, click **Add**.
2. Select the transaction to configure.
3. In the transaction sidebar, on the right side of the screen.
   Enter values for the transaction settings.

   Table 4. Actual temperature transaction settings:

Setting	Value	Description
Transactio name	Actual Temp	Give the transaction a name.
Modbus transaction	Read Holding Registers (3)	This register read the actual temperature from the temperature regulator to the PLC.
Address	Address: 0 Register: 1	Address 0 is Register 1.
Quantity	1	The transaction will address one data object.
Update mode	Cyclically	Default value, can not be changed.
Update time	1000 ms	The update cycle is 1000 ms.

10.1.7. Check the I/O Data Map

The transactions to and from the Temperature Regulator are mapped as follows in the **I/O data map** page.

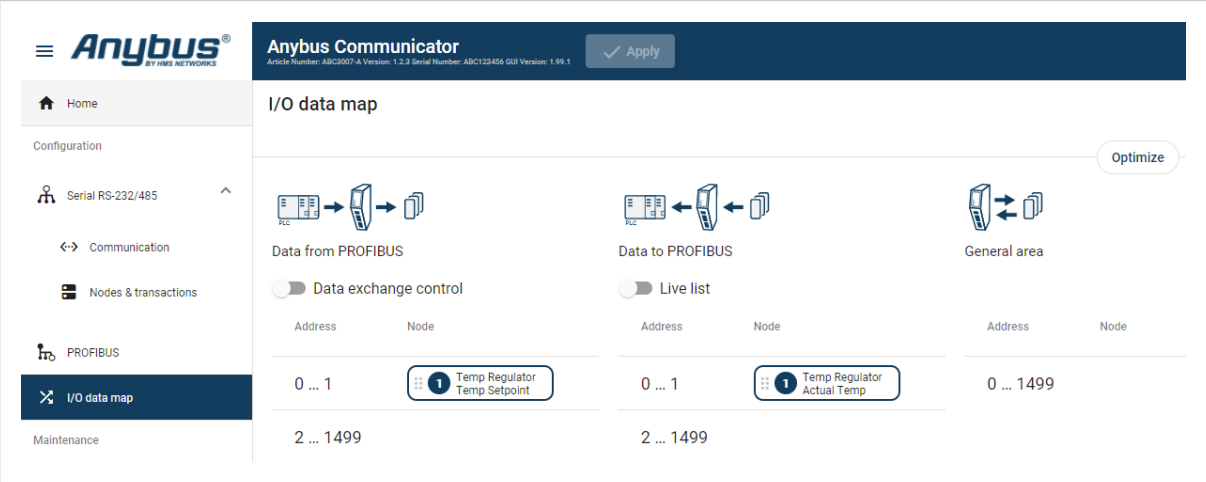


Figure 86. I/O data map

Address	Data to PROFIBUS
0-1	Setpoint temperature from PROFIBUS to the Temperature Regulator.

Address	Data from PROFIBUS
0-1	Actual temperature speed from the Temperature Regulator to PROFIBUS.

10.2. AC Motor Drive - Custom Request/Response Use Case

10.2.1. About the Use Case

The purpose of this use case is to explain how to use the **Custom Request/Response** serial protocol.

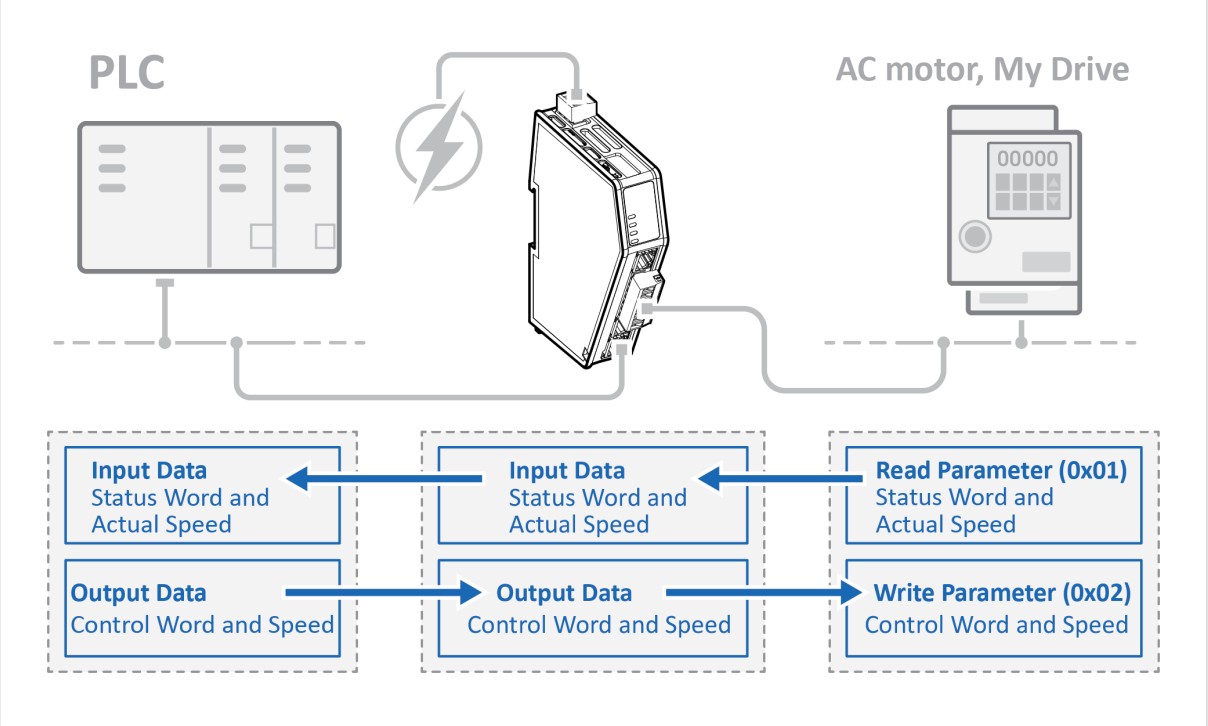


Figure 87. AC Motor Drive - Custom Request/Response Use Case

In this use case we use the Communicator to enable data exchange between an AC motor, of the type My Drive, and a PLC.

The use case describes how to map the communication in the Communicator.

My Drive is connected to the serial subnetwork via a custom RS-485 protocol.

The PLC is connected to an PROFIBUS network (high level network).

We use the Custom Request/Response serial protocol and create customized transaction template to map up:

- Status word and actual speed from My Drive to the PROFIBUS network.
- Control word and speed from the PROFIBUS network to My Drive.

10.2.2. Before You Begin

- Connect the Communicator configuration port to your computer.
- Power on the Communicator.
- Ensure that your computer can find the Communicator IP address.
- Enter the Communicator built-in web interface of the.

For more information refer to Communicator Configuration (page 39).

10.2.3. Choose Serial Protocol Type

My Drive is using a request/response protocol to access parameters addressed with index and sub index.

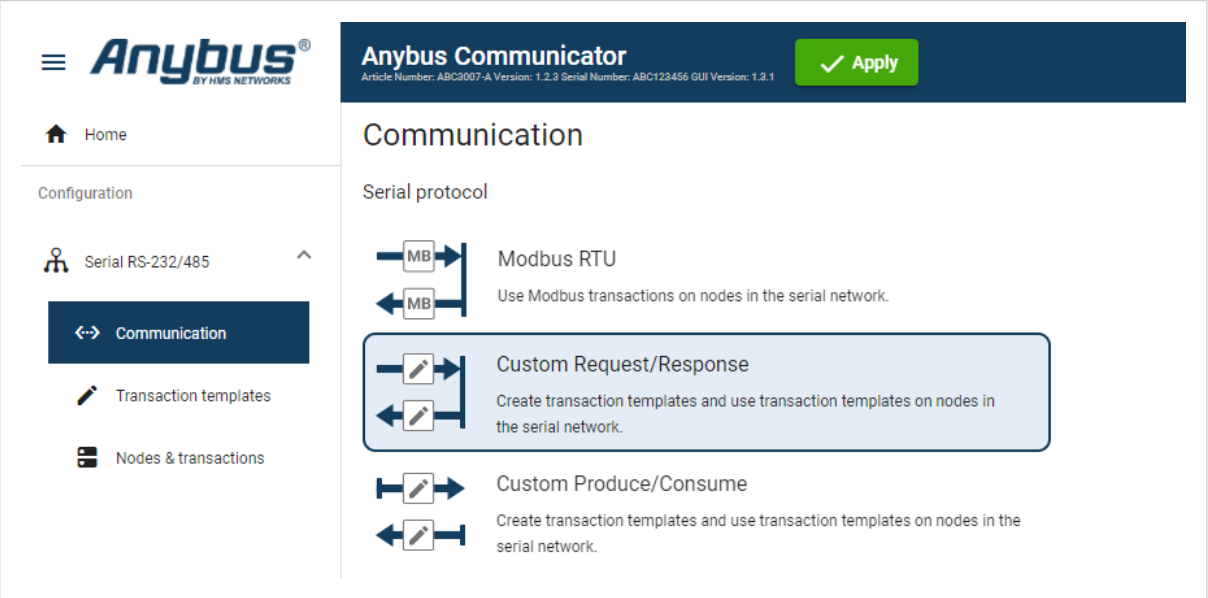


Figure 88. Communication page, **Custom Request/Response**

- On the **Serial RS232/485** page, select **Custom Request/Response**.

10.2.4. Setup Serial Communication

Set up the communication between the Communicator and My Drive.

In the **Serial RS232/485** page, configure the **Communication** settings.

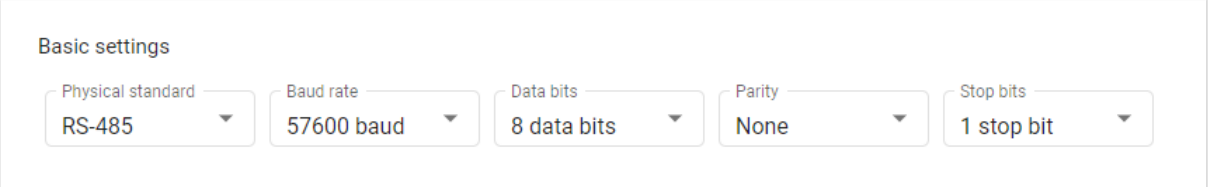


Figure 89. Serial RS232/485, Basic settings

Table 5. Used the following settings:

Frame objects	Value
Physical standard	RS-485
Baud rate	57600 baud
Data bits	8 bits
Parity	None
Stop bits	1 stop bit

10.2.5. Create Transaction Templates

All frames are verified using a CRC-16-IBM checksum.

My Drive is using a request/response protocol to access parameters addressed with index and sub index.

Map up control word, speed from PROFIBUS to My Drive and status word and actual speed from the drive to PROFIBUS.

Create Read Parameter (0x01)

The Communicator reads values delivered from to the My Drive node on to the PLC.

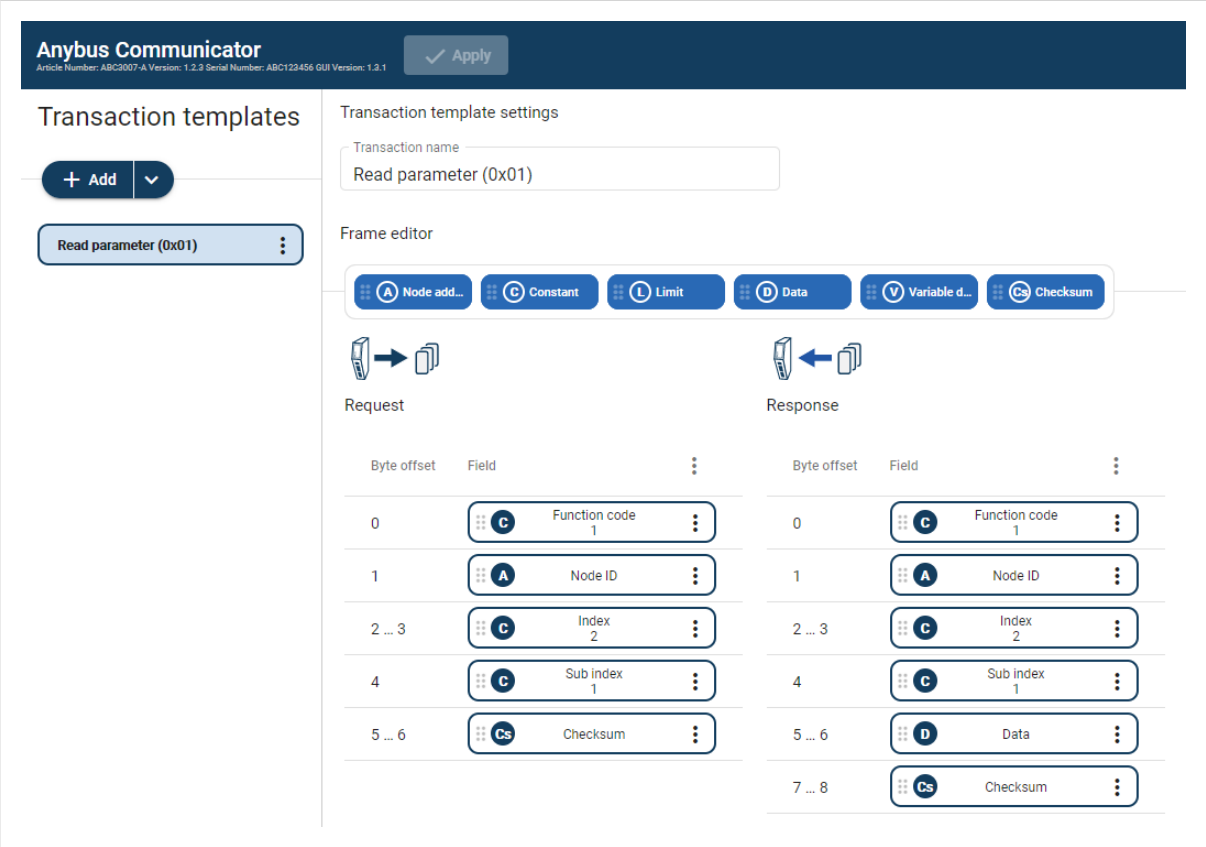


Figure 90. Read Parameter (0x01)

1. Add an **Empty template** and select it.
2. Name the template **Read parameter (0x01)**.

3. In the Frame editor **Request** area, add five **frame objects** with the following settings:

Table 6. Request frame objects

Frame objects	Name	Bytes/Length	Type/ Checksum type	Endianess	Fixed field	Value
Constant	Function code	1	Byte	N/A	Yes	N/A
Node address	Node ID	1	Byte	N/A	N/A	N/A
Constant	Index	2	Word (two bytes)	Big-endian	No	Min 0 Max 1000
Constant	Sub index	1	Byte	N/A	No	Min 0 Max 255
Checksum	Checksum	2	CRC	N/A	N/A	N/A

4. In the Frame editor **Response** area, add six **frame objects** with the following settings:

Table 7. Response frame objects

Frame object	Name	Bytes/Length	Type/ Checksum type	Endianess	Fixed field	Value
Constant	Function code	1	Byte	N/A	Yes	N/A
Node address	Node ID	1	Byte	N/A	N/A	N/A
Constant	Index	2	Word (two bytes)	Big-endian	No	Min 0 Max 1000
Constant	Sub index	1	Byte	N/A	No	Min 0 Max 255
Data	Data	2	Byte	N/A	Yes	N/A
Checksum	Checksum	2	CRC	N/A	N/A	N/A

Create Write Parameter (0x02)

The Communicator writes values delivered from the PLC to the My Drive node.

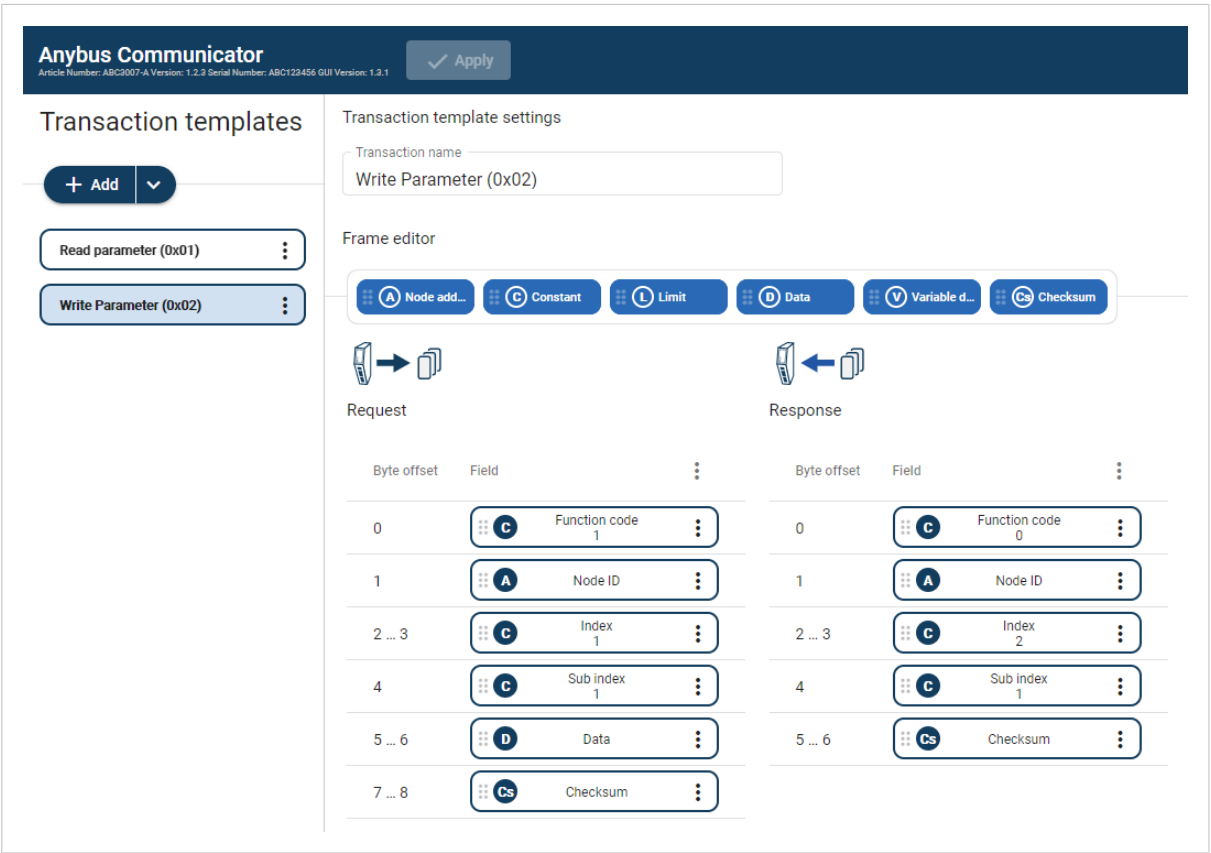


Figure 91. Write Parameter (0x02)

1. Add an **Empty template** and select it.
2. Name the template **Write parameter (0x02)**.
3. In the Frame editor **Request** area, add six **frame objects** with the following settings:

Table 8. Request frame objects

Frame object	Name	Bytes/Length	Type/ Checksum type	Endianess	Fixed field	Value (Hex)
Constant	Function code	1	Byte	N/A	Yes	N/A
Node address	Node ID	1	Byte	N/A	N/A	N/A
Constant	Index	2	Word (two bytes)	Big-endian	No	Min 0 Max 1000
Constant	Sub index	1	Byte	N/A	No	Min 0 Max 255
Data	Data	2	Byte	N/A	Yes	N/A
Checksum	Checksum	2	CRC	N/A	N/A	N/A

4. In the Frame editor **Response** area, add five **frame objects** with the following settings:

Table 9. Response frame objects

Frame objects	Name	Bytes	Type/ Checksum type	Endianess	Fixed field	Value (Hex)
Constant	Function code	1	Byte	N/A	Yes	N/A
Node address	Node ID	1	Byte	N/A	N/A	N/A
Constant	Index	2	Word (two bytes)	N/A	No	Min 0 Max 1000
Constant	Sub index	1	Byte	N/A	No	Min 0 Max 255
Checksum	Checksum	2	CRC	N/A	N/A	N/A

10.2.6. Setup Node and Transactions

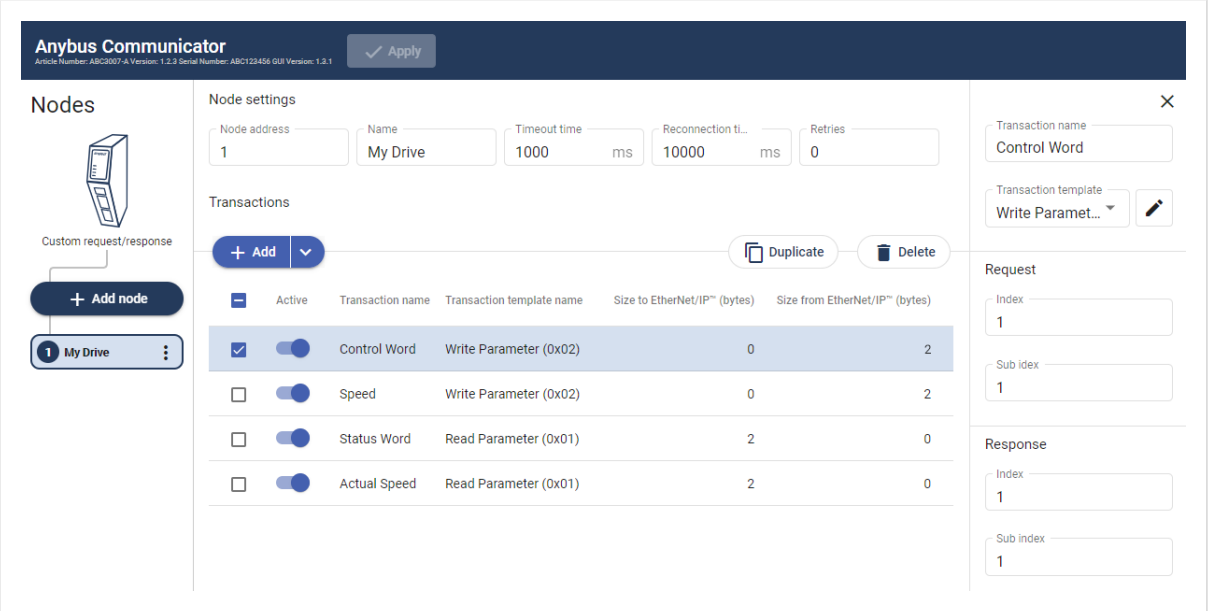


Figure 92. My Drive node with transactions

1. Add a node and select it.
2. In Node settings configure the node with the following settings:

Node settings	Value
Node address	1 My Drive is set up as a node with Node address 1.
Name	My Drive
Timeout time	1000 ms (default)
Reconnecting time	1000 ms (default)
Retries	0 (default)

3. Add four transactions to the My Drive node and configure them with the following settings:

Table 10. My Drive contains the following parameters

Transaction name	Transaction template	Index	Sub index
Control Word	Write Parameter (0x02)	1	1
Speed	Write Parameter (0x02)	1	2
Status Word	Read Parameter (0x01)	2	1
Actual Speed	Read Parameter (0x01)	2	2

10.2.7. Check the I/O Data Map

The control word, speed from PROFIBUS to My Drive and status word and actual speed from My Drive to PROFIBUS are mapped as follows in the **I/O data map** page.

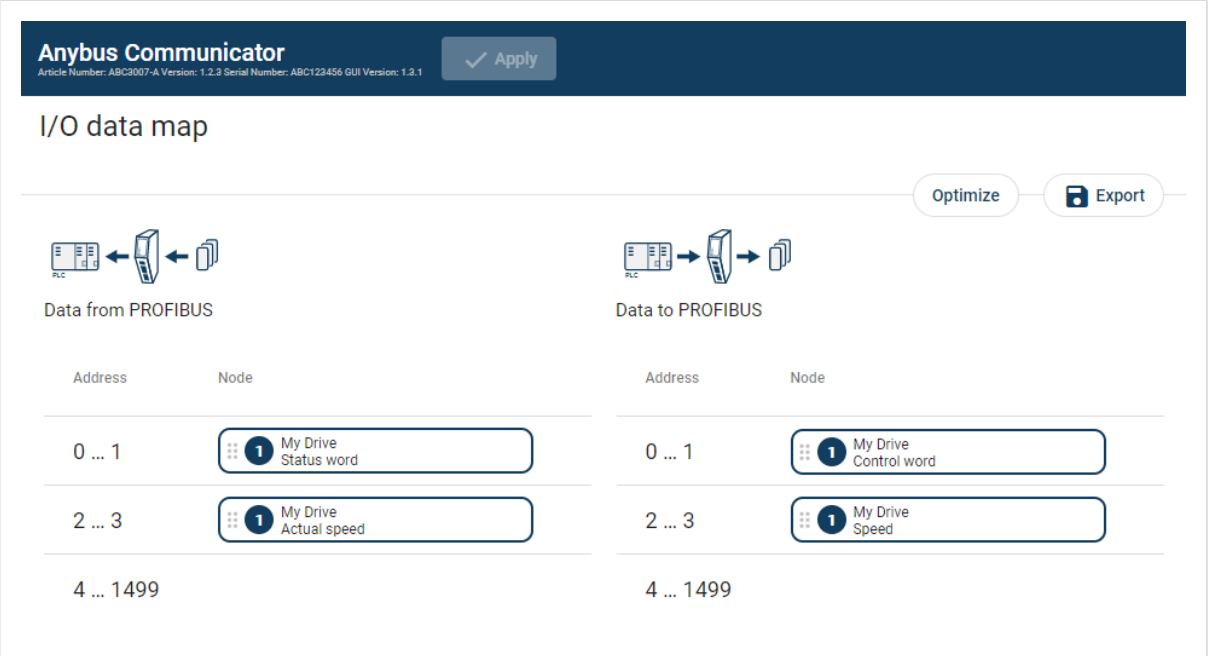


Figure 93. I/O data map

Table 11. Control word and speed from PROFIBUS to My Drive

Address	Drive Parameter
0-1	Control Word
2-3	Speed

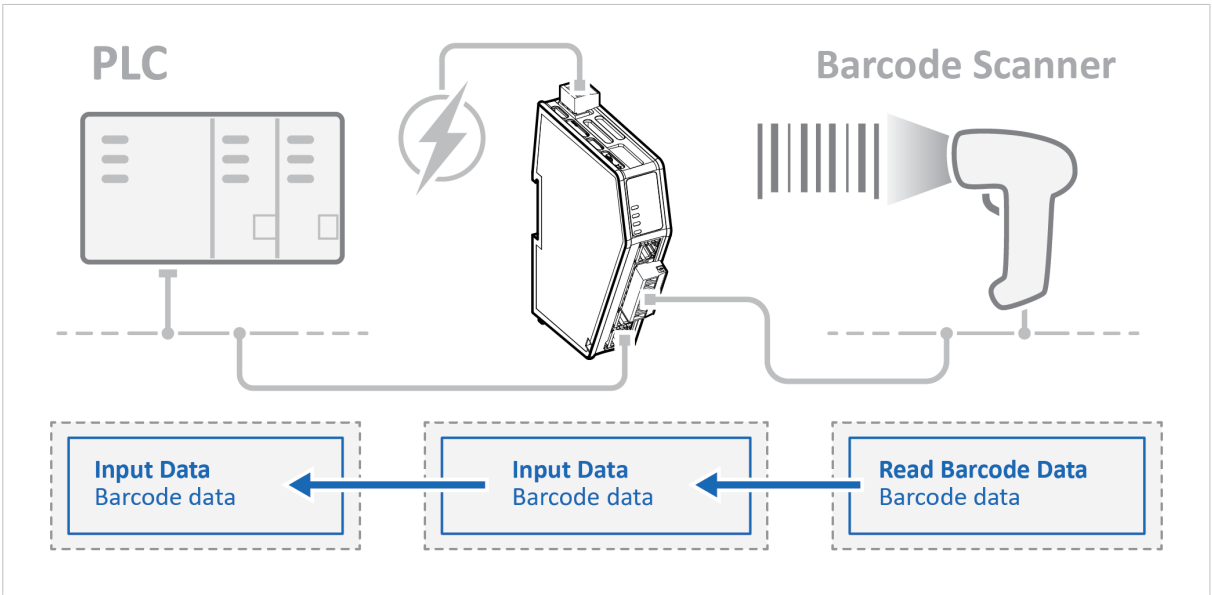
Table 12. Status word and actual speed from My Drive to PROFIBUS

Address	Drive Parameter
0-1	Control Word
2-3	Speed

10.3. Barcode Scanner - Custom Produce/Consume Use Case

10.3.1. About the Use Case

The purpose of this use case is to explain how to use the **Custom Produce/Consume** serial protocol.



In this use case we use the Communicator to enable data exchange between an Barcode Scanner and a PLC.

The use case describes how to map the communication in the Communicator.

The Barcode Scanner is connected to the serial subnetwork via a custom RS-232 protocol.

The PLC is connected to an PROFIBUS network (high level network).

We use the Custom Produce/Consume serial protocol and create a customized transaction template.

Figure 94. Barcode Scanner - Custom Produce/Consume Use Case

10.3.2. Before You Begin

- Connect the Communicator configuration port to your computer.
- Power on the Communicator.
- Ensure that your computer can find the Communicator IP address.
- Enter the Communicator built-in web interface of the.

For more information refer to Communicator Configuration (page 39).

10.3.3. Choose Serial Protocol Type

The purpose of this use case is to explain how to use the **Custom Produce/Consume** serial protocol.

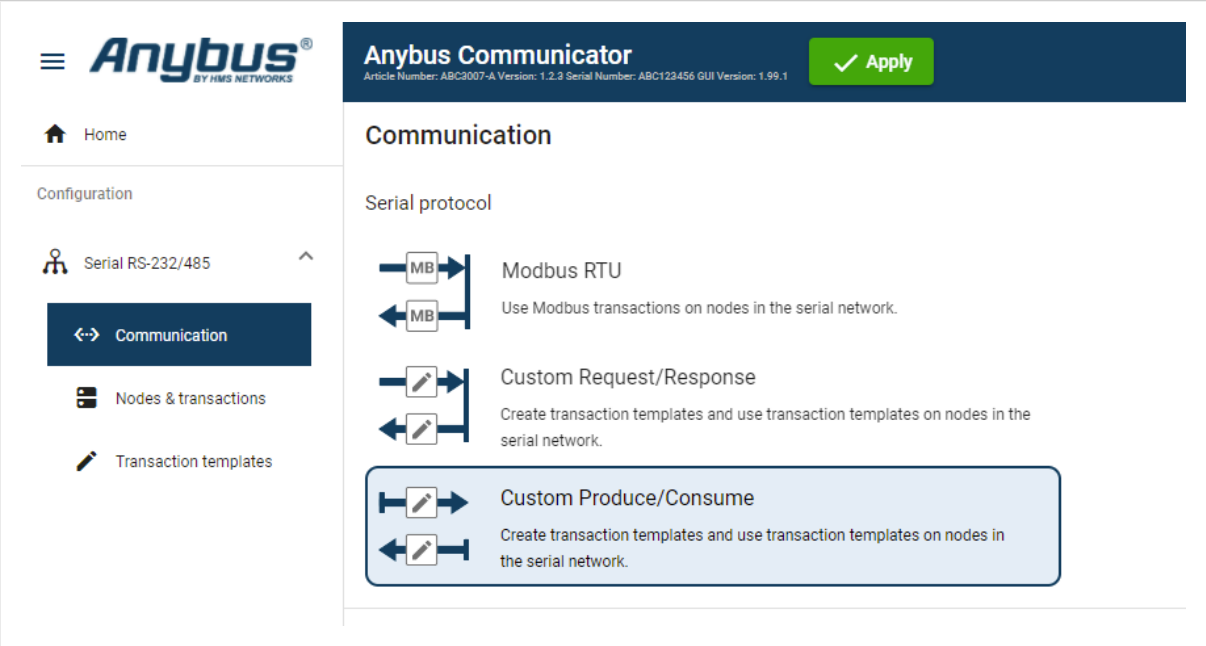


Figure 95. Communication page, **Custom Produce/Consume**

- On the **Serial RS232/485** page, select **Custom Produce/Consume**.

10.3.4. Setup Serial Communication

Set up the communication between the Communicator and Barcode Scanner.

In the **Serial RS232/485** page, configure the **Communication** settings.

Figure 96. Serial RS232/485, Basic settings

Table 13. Used the following settings:

Frame objects	Value
Physical standard	RS-232
Baud rate	9600 baud
Data bits	8 bits
Parity	None
Stop bits	1 stop bit

10.3.5. Create Transaction Templates

Create Read Barcode Data Parameter

Before You Begin

The Communicator reads values delivered from to the Barcode Scanner node on to the PLC.

The Barcode Scanner sends data whenever it is available, without any request or handshake from the Communicator.

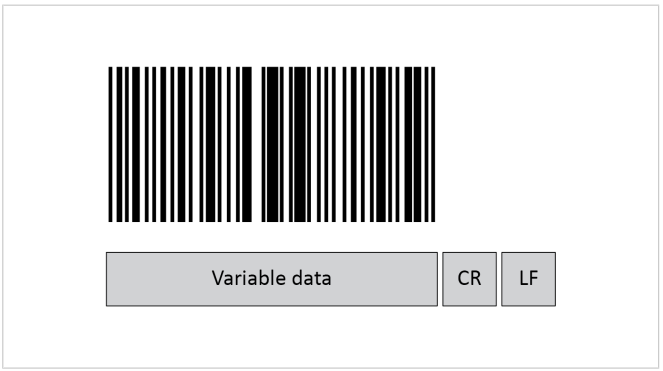


Figure 97. Barcode Variable data, CR and LF

In this example we have added three frame fields for the barcode data transaction:

- One Variable data frame for the length of the barcode.
  We use a fixed value.
  The maximum payload length is 31 (ASCII).
- The Barcode Scanner is configured to append:
  - One Carriage Return character (CR) to the barcode.
    So we create one Constant frame with the Value 13 (ASCII).
  - One Line Feed character (LF) to the barcode.
    So we create one Constant frame with the Value 10 (ASCII).

Procedure

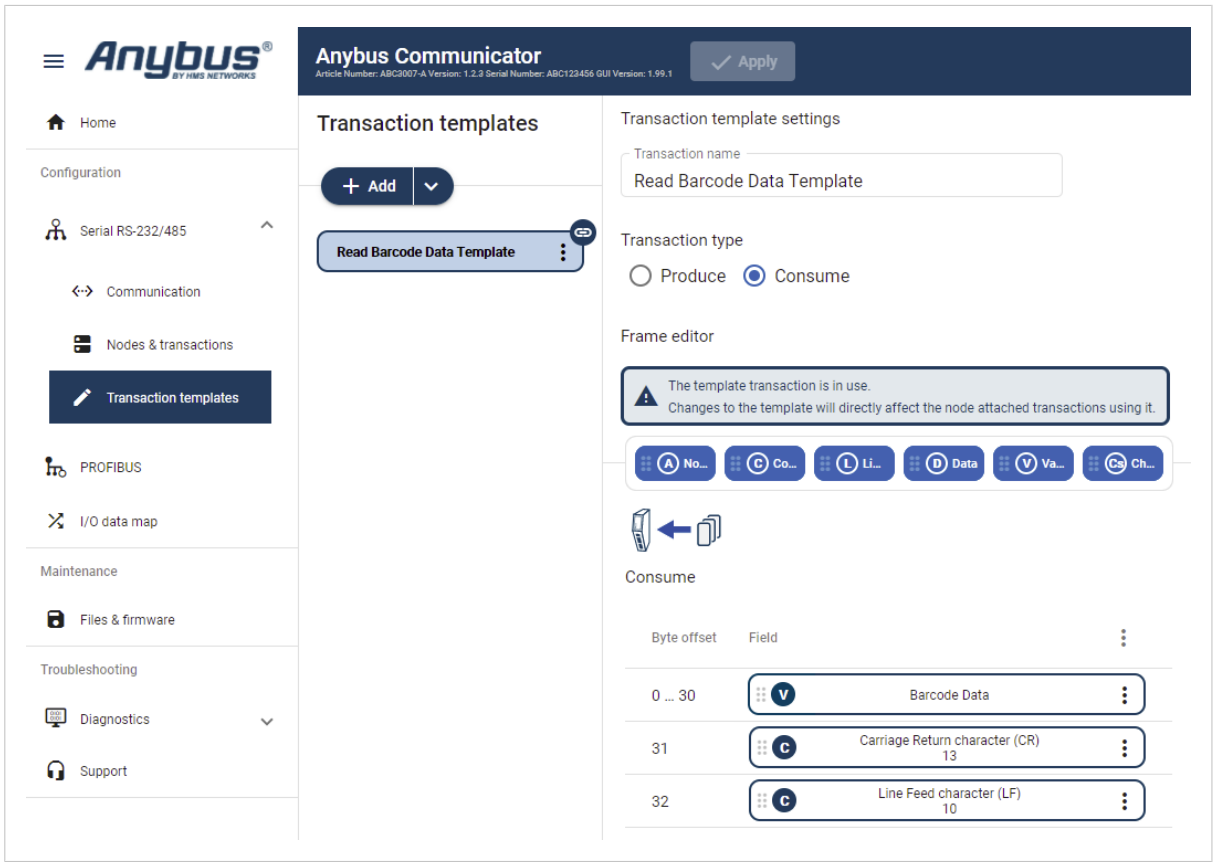


Figure 98. Read Barcode Data Parameter

1. Add an **Empty consume template** and select it.
2. Name the template **Read Barcode Data**.
3. In the **Frame editor**, add four frame field with the following settings:

Table 14. Consume frame fields

Frame fields	Name	Type	Value	Fixed field	Maximum payload length	Process data delimiter	Subnet delimiter	End pattern
Variable data	Barcode Data	N/A	N/A	Yes, set here	31 bytes	None	None	0
Constant	Carriage Return character (CR)	Byte (1 byte)	13	Yes, set here	N/A	N/A	N/A	N/A
Constant	Line Feed character (LF)	Byte (1 byte)	10	Yes, set here	N/A	N/A	N/A	N/A

10.3.6. Setup Node and Transactions

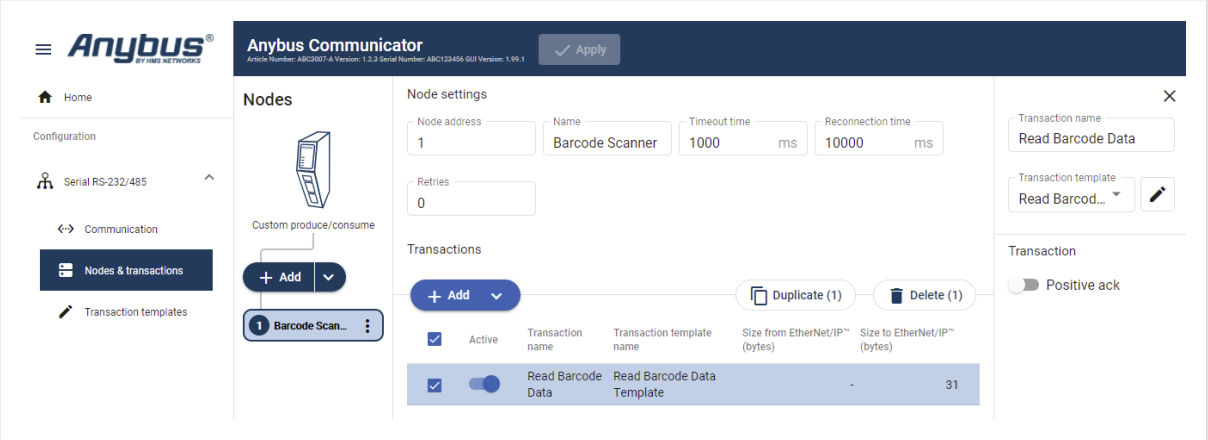


Figure 99. Node settings

1. Add a node and select it.
2. In **Node settings** configure the node with the following settings:

Node settings	Value
Node address	The Barcode Scanner is set up as a node with Node address 1.
Name	Barcode Scanner
Timeout time	1000 ms (default)
Reconnecting time	1000 ms (default)
Retries	0 (default)

3. Add one transactions to the Barcode Scanner node and configure it with the following settings:

Table 15. The Barcode Scanner contains the following parameters

Transaction name	Transaction template
Read Barcode Data	Read Barcode Data Template

10.3.7. Check the I/O Data Map

The transactions from the Barcode Scanner is mapped as follows in the **I/O data map** page.

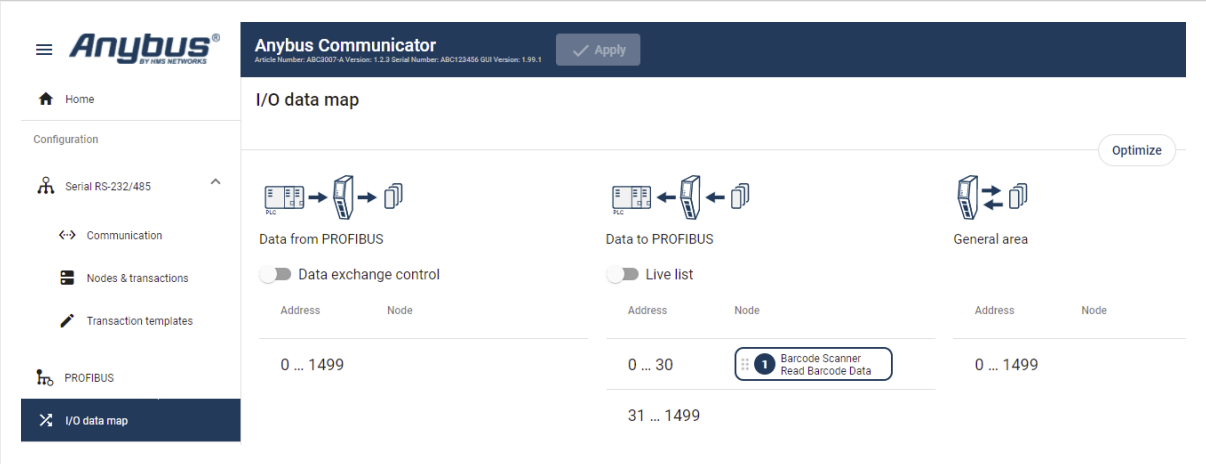


Figure 100. I/O data map

Table 16. Status word and actual speed from My Drive to PROFIBUS

Address	Barcode Scanner Parameter
0-31	The variable data, 31 bytes, are forwarded from the Barcode Scanner to the PLC.

11. Maintenance

11.1. Configuration File Handling

11.1.1. Export Configuration

You can export the current configuration, in order to import and use the same settings to configure additional Communicator.

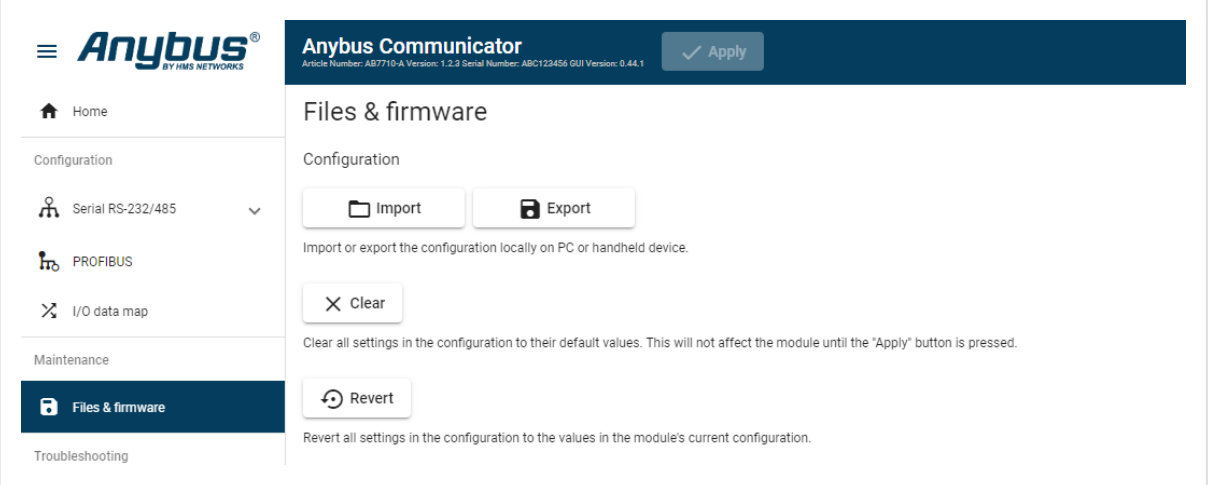


Figure 101. Files & firmware page

To export a configuration file:

- In **Files & firmware**, click **Export**.
  The configuration settings are stored in a .conf file and downloaded to your PC.

11.1.2. Import Configuration

To easily configure multiple Communicator with the same settings, you can import a configuration file.

Before You Begin

> **NOTE**
> Importing a configuration replaces the current applied configuration.

The supported file format is .conf.

Procedure

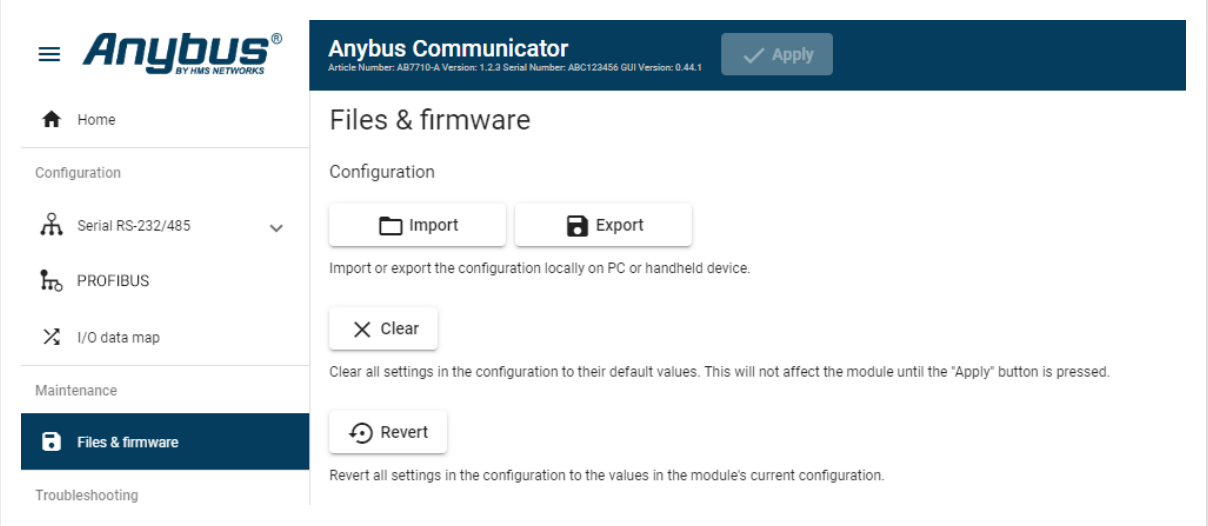


Figure 102. Files & firmware page

Import configuration file:

1. On the **Files & firmware** page, click **Import**.
2. In the Import configuration window, click **Select file (.conf)**.
3. In the Open dialog box, browse to and select the configuration file and click **Open**.
4. In the Import configuration window, click **Import**.
5. In the Communicator address settings window:
   - To import IP settings from the selected configuration file, click **Imported settings**.
     All configuration settings are imported.
   - To continue using the current IP settings, click **Configured settings**.
     All configuration settings except the IP settings are imported.
6. The configuration file is parsed.
   - If the configuration is compatible, the settings are imported.
   - If any compatibility mismatches occurs, a message about the mismatch appears.
7. To apply the settings, click **Apply** in the web-interface header, and follow the instructions.

11.2. Clear and Revert Configuration

You can restore all settings in a configuration to the default settings.

Procedure

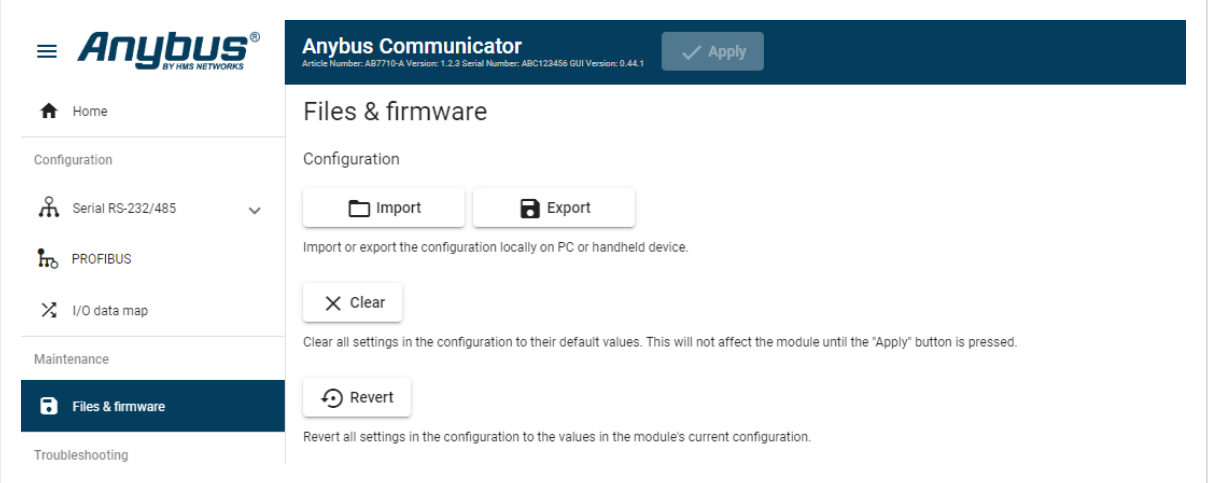


Figure 103. Files & firmware page

When you want to clear a configuration and return to the default settings.

To clear the configuration:

1. On the **Files & firmware** page, click **Clear**.
2. In the Confirm clear window, click **Clear**.
3. To apply the change, click **Apply** in the web-interface header, and follow the instructions.

When you want to remove any configuration made in a current session and re-load the configuration from the gateway.

To Revert the configuration:

1. On the Files & firmware page, click **Revert**.
2. In the Confirm revert window, click **Revert**.
3. To apply the change, click **Apply** in the web-interface header, and follow the instructions.

11.3. Firmware Management

11.3.1. View the Firmware Version

On the **Support** page, you can view the current applied firmware version.

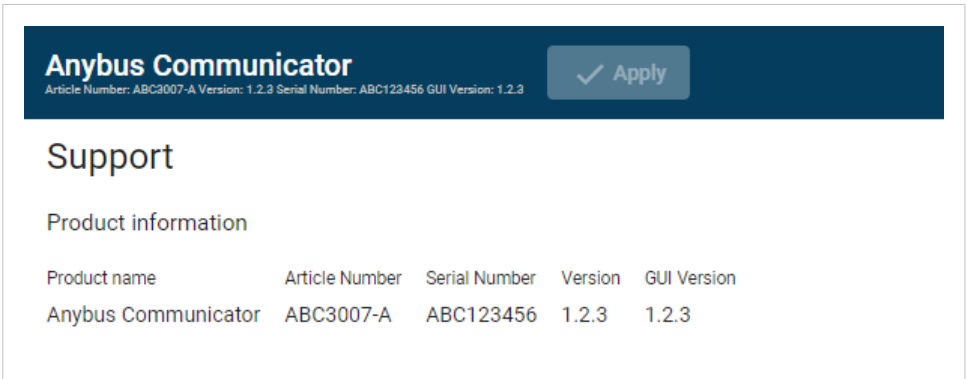


Figure 104. Support page, Product information example

11.3.2. Firmware and Configuration Compatibility

Compatibility after firmware upgrade

Current configuration is still compatible after upgrading the firmware.

Compatibility after firmware downgrade

IMPORTANT

Compatibility after a firmware downgrade can not be guaranteed.

The current configuration may use features not available in the older firmware version.

11.3.3. Firmware File Validation

Before the firmware file is imported into the system, the firmware upgrade function perform a validation of the file, to ensure that:

- the firmware is compatible with the Communicator hardware
- the firmware is suited for the product
- the officially HMS software signatures are valid
- that the firmware file is not corrupt or damaged

If the firmware file does not pass the validation, the firmware file is rejected and an error message appear.

11.3.4. Update Firmware

Before You Begin

IMPORTANT
To eliminate the risk of interference with plant operation, firmware update is only available when the Communicator is disconnected from the OT networks.

Ensure that the Communicator is disconnected from the OT networks.

Procedure

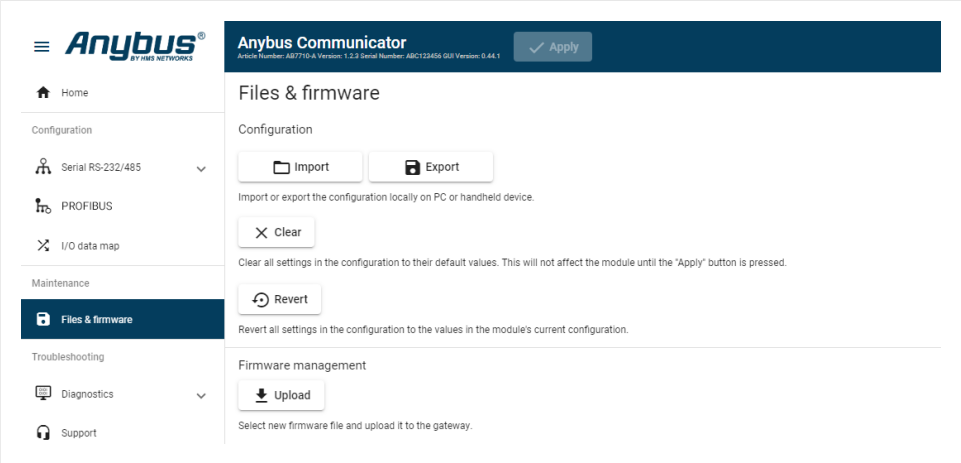


Figure 105. Files & firmware page

To update the firmware:

1. On the **Files & firmware** page, click **Upload**.
2. In the Upload Firmware window, click **Select firmware (.hiff)**.
3. In the Open dialog box, browse to and select the firmware file and click **Open**.
4. To start the firmware upgrade, click **Update firmware**.
   The firmware file is validated and transferred.

Result

- If the firmware file pass the validation: The firmware is upgraded and then the Communicator automatically reboots, for the upgrade to take effect.
- If the firmware file is rejected: An error message appear.

12. Troubleshooting

12.1. Diagnostics

12.1.1. Serial RS-232/485 Data Monitor

On the Serial RS-232/485 page you can monitor how the data flow between the nodes and the gateway changes over time.

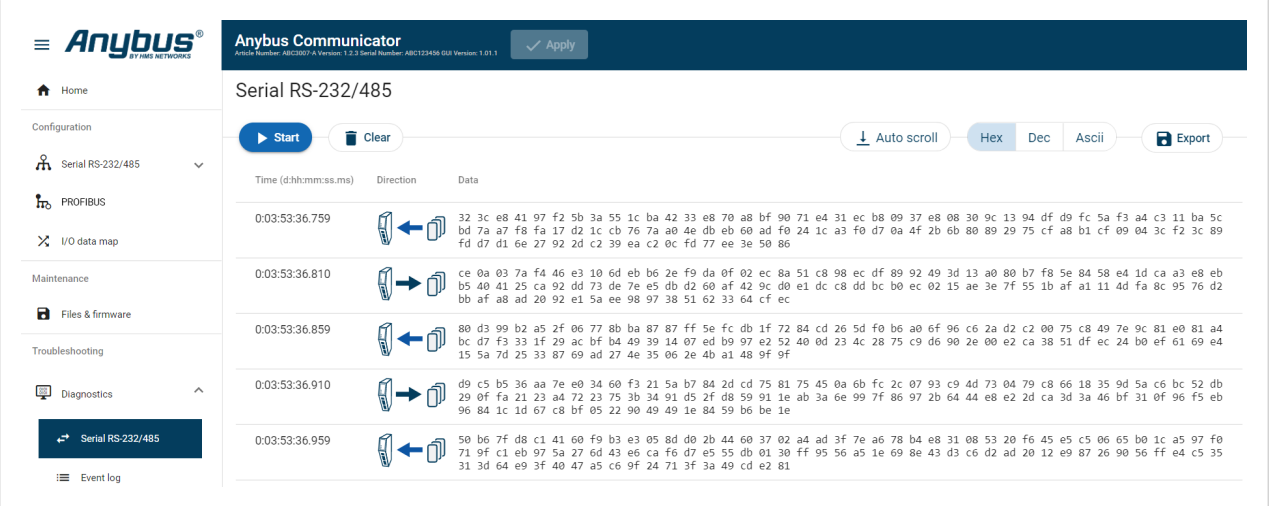


Figure 106. Serial RS-232/485 page

The table can contain at most 10000 messages. When the limit is reached, the oldest messages are discarded when new messages are added.

Choose how data is displayed

- To choose if the data should be displayed in Hexadecimal, Decimal or ASCII, click **Hex**, **Dec** or **Ascii**.

Start and Stop Data flow

- To start the data flow, click **Start**.
- To end the data flow, click **Stop**.

Export data flow

- To export the data flow, click **Export**.
  An Excel file with the data flow is downloaded to your PC.

12.1.2. Event Log

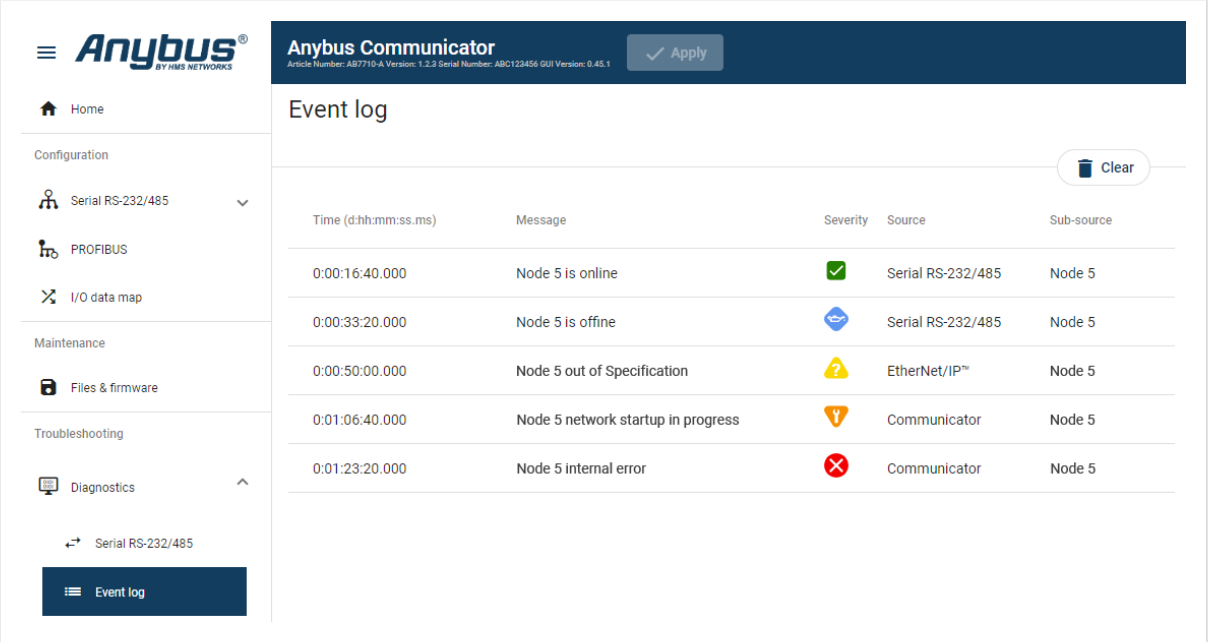


Figure 107. Event log page

How To Analyze the Information

The log follows the FIFO principle, first in and first out. The oldest (first) value is processed first.

Time (d:hh:mm:ss.ms)	The date and time when the event occurred.	
Message	A brief description of the event.	
Severity	The severity of the event occurred. For description of the symbols, see Communicator Status Monitor (page 94).	
Source	0	Communicator
	1	High level network, PROFIBUS
	2	Subnetwork, Serial RS-232/485
Sub-source	The nodes connected to the subnetwork and the PLC connected to the high level network. If there is a problem with a node the node name is displayed in the Sub-source column. Example 10. Sub-source number If the node name is 5, number 5 is displayed in the Sub-source column.	

To clear the current log, click **Clear log**.

12.1.3. LED Status

On the Home page, you can remotely monitor the Communicator LED status.

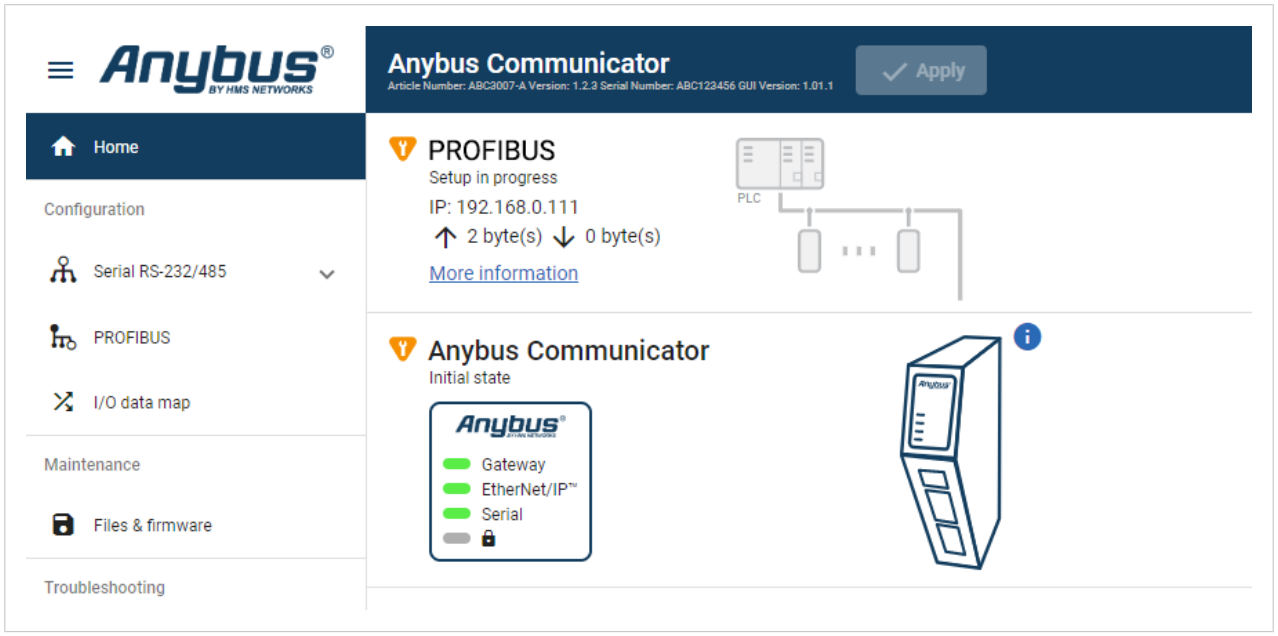


Figure 108. Home page

For information about the LED indication, see Communicator LED Indicators (page 96).

12.2. Reset to Factory Settings

Before You Begin

Factory reset will reset any on site made configuration changes and set the Communicator to the same state as leaving HMS production.

If the Firmware has been updated, factory reset will revert the Communicator configuration to initial state after the update.

Procedure

To reset the Communicator:

1. Disconnect the Communicator from power.

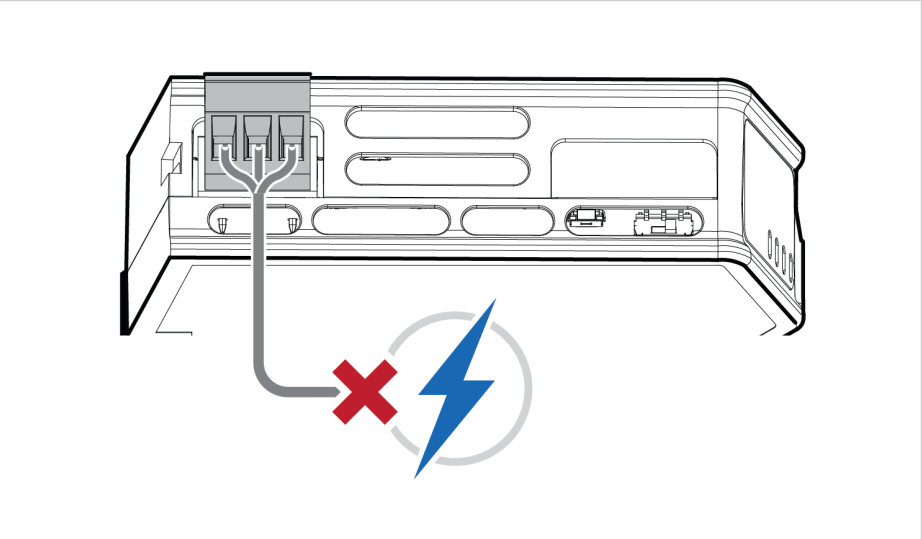


Figure 109. Disconnect power

2. Use a pointed object, such as a ballpoint pen to press and hold the **Reset** button.

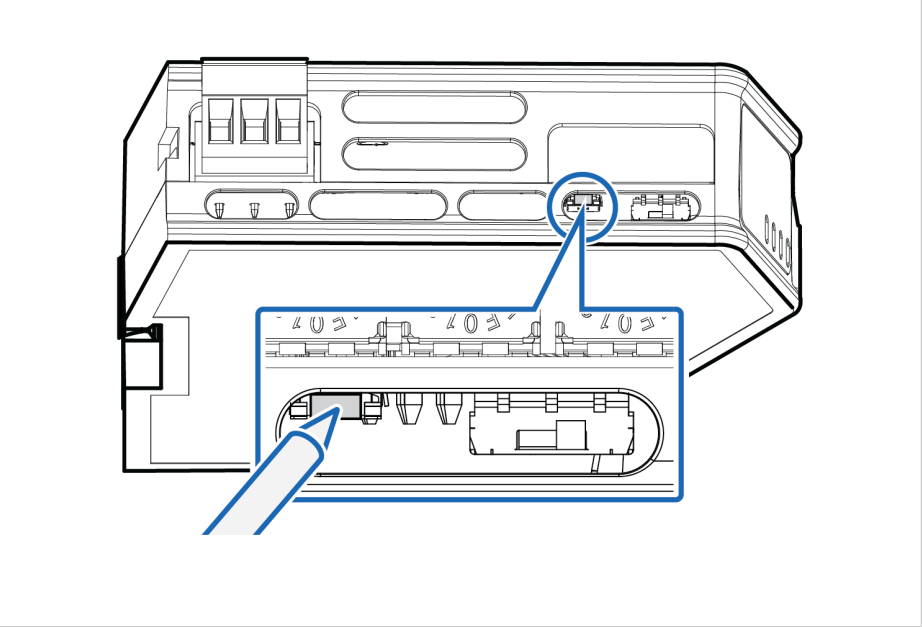


Figure 110. Press and hold **Reset** button

3. While holding the **reset** button, reconnect the Communicator to power.

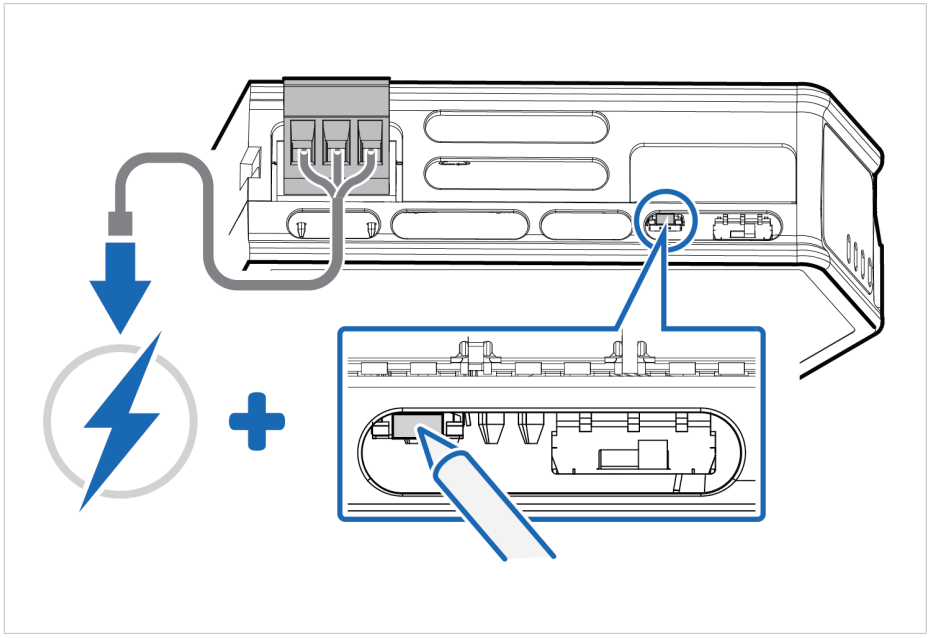


Figure 111. Hold **Reset** button and reconnect power

4. Release the **reset** button.
   The Communicator enters exception state.

5. Reboot the Communicator.

Result

- When the Communicator has successfully rebooted, the Communicator configuration is reset to the factory default configuration or the current configuration after firmware upgrade.

To Do Next

To ensure that the Communicator built-in web-interface is synchronized.

1. Open the the Communicator built-in web interface.
2. Navigate to the **Files & firmware** page and click **Revert**.

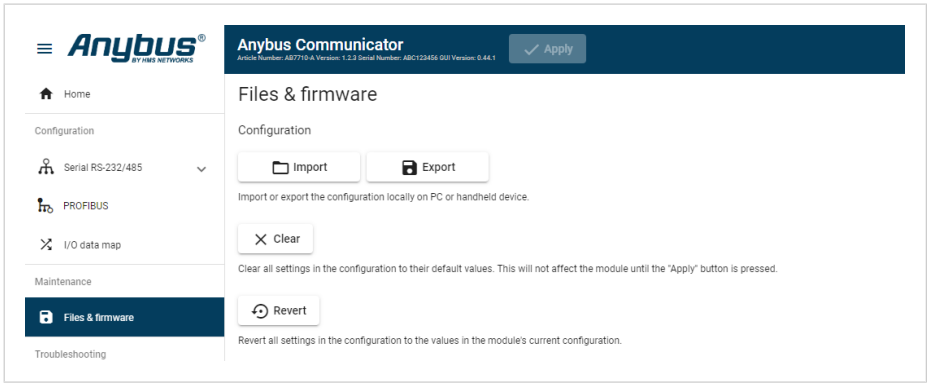


Figure 112. Files & firmware, Revert

12.3. Firmware Upgrade Error Management

If the firmware update process is interrupted or if the power is lost during the update process, the Communicator goes into fallback mode.

The last working firmware is still available on the flash, but it is not active.

Procedure

To complete the interrupted firmware update:

1. Disconnect the Communicator from power.

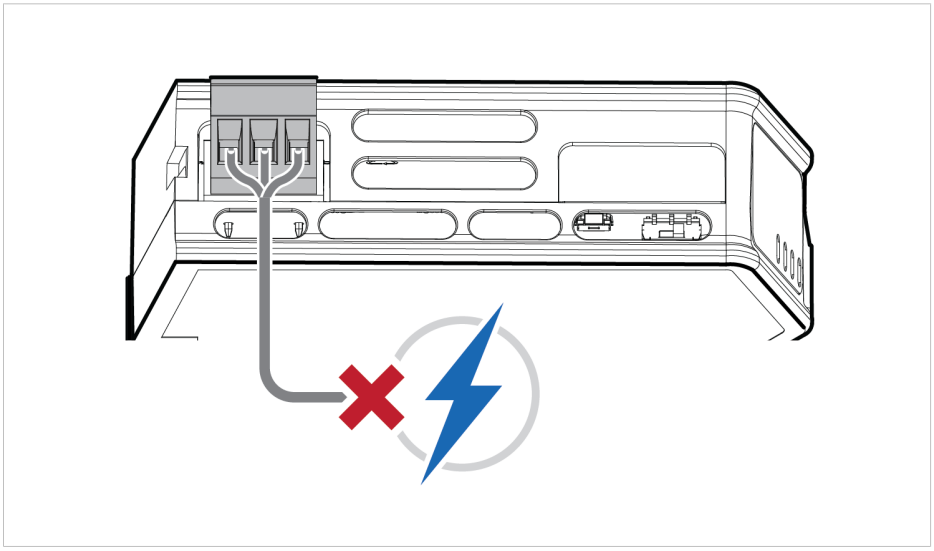


Figure 113. Disconnect power

2. Reconnect the Communicator to power.

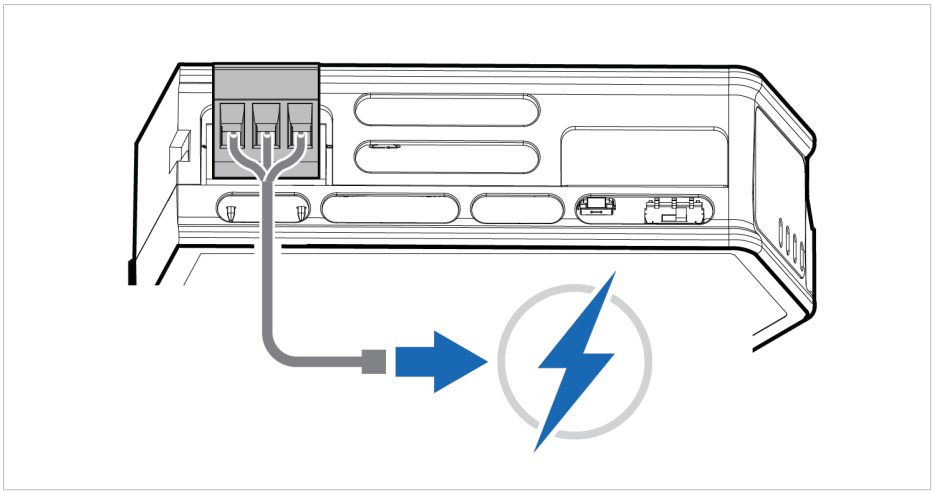


Figure 114. Reconnect power

3. Leave the Communicator for 10 minutes.
   The Gateway status led indicator flashes red and green until the firmware upgrade is completed.

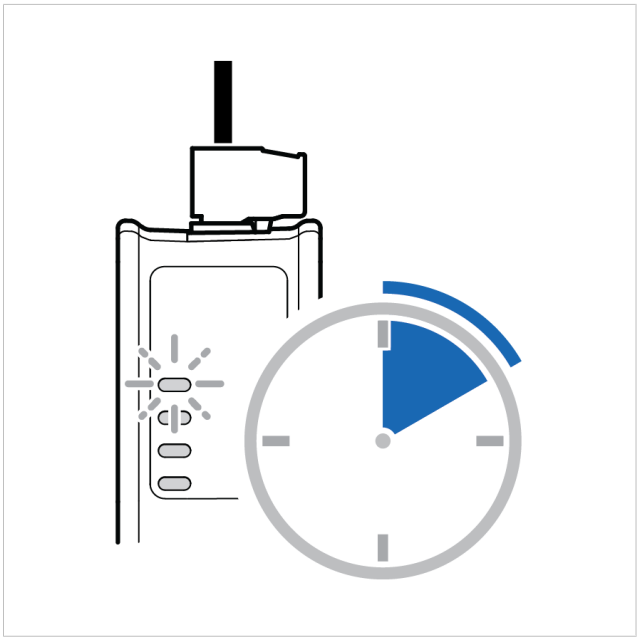


Figure 115. Firmware upgrade LED indication

Result

The Communicator recover and return to normal operation.

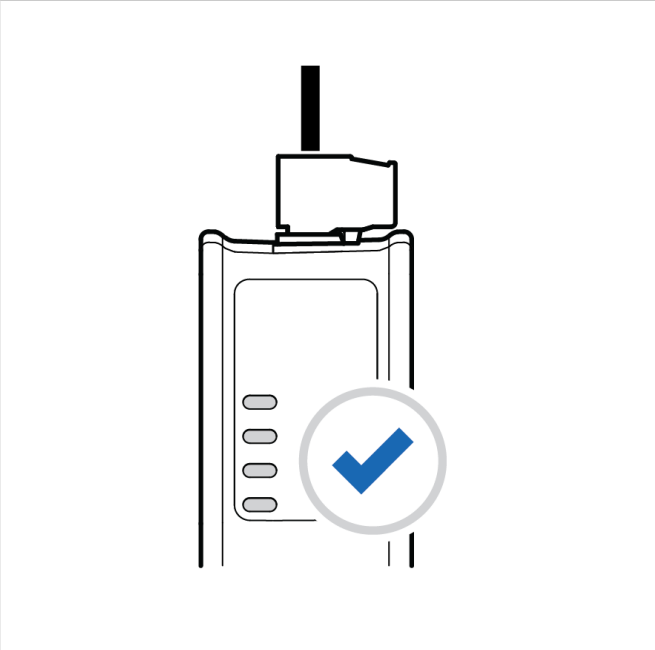


Figure 116. Recover and return to normal operation

To Do Next

To check LED status, refer to Communicator LED Indicators (page 96).

12.4. Support

12.4.1. Support Package

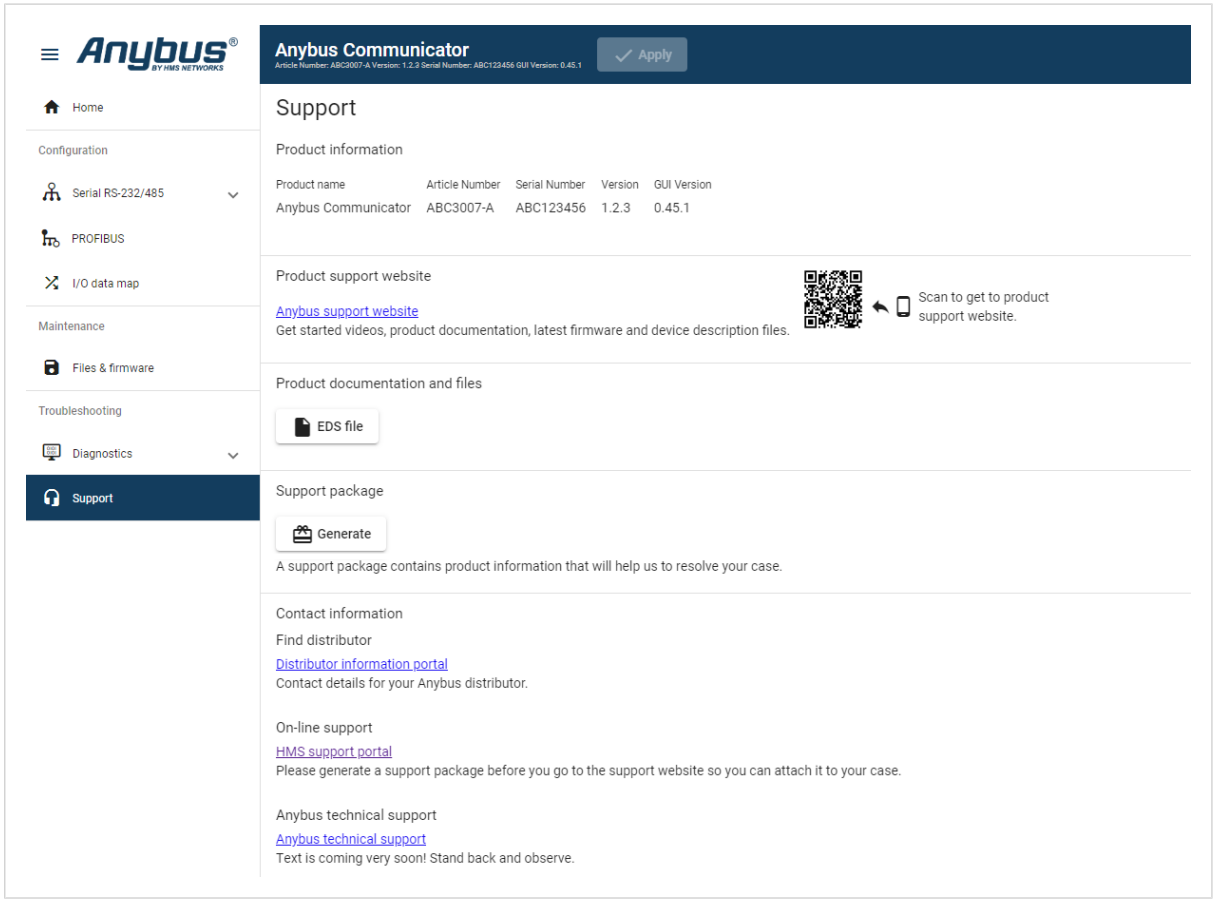


Figure 117. Support page

Before you create a ticket for technical support, generate a support package.

The support package contains information about what has occurred and will help the Anybus technical support team resolve the support case as quickly and efficiently as possible.

Support Package Content

The information in the support package are available to open and read, the files are not locked or encrypted.

Generate Support Package

- On the **Support** page, click **Generate**.
  A zip file with the support files is downloaded to your PC.

Create a Support Ticket

1. On the Support page, click **Anybus support website**.
   You are redirected to the Anybus support website.

2. On the Anybus support website, create a support ticket and upload the support package.

13. Technical Data

For complete technical specifications and regulatory compliance information, please visit www.anybus.com/support.

13.1. Technical Specifications

Article identification	ABC3000-A
Communication connector	PROFIBUS DSUB connector
Configuration connector	RJ45
Serial connector	7-pin screw connector
Power connector	3-pin screw connector
Power supply	12-30 VDC, Reverse voltage protection and short circuit protection
Power consumption	Typical: 160 mA @ 24 V Max: 400 mA @ 12 V
Storage temperature	-40 to +85 °C
Operating temperature	-25 to +70 °C
Humidity	EN 600068-2-78: Damp heat, +40°C, 93% humidity for 4 days EN 60068-2-30: Damp heat, +25°C – +55°C, 95% RH, 2 cycles
Vibration	See datasheet
Housing material	Plastic, See datasheet for details
Protection class	IP20
Product weight	150 g
Dimensions	27 x 144 x 98 mm (H x W x D) with connectors included
Mounting	DIN-rail

For additional technical data and information related to the installation and use of this product, please visit www.anybus.com/support.

14. Reference Guides

14.1. About Input Registers and Holding Registers

Modbus data is most often read and written as registers which are 16-bit pieces of data.

Holding registers and Input registers are both 16-bit registers.

Input registers

Input registers can only be read.

Holding registers

Holding registers can be read or written.

These registers can be used for a variety of things such as inputs, outputs, configuration data, or other requirement for holding data.

14.2. Modbus Data Model

Discretes Input	Single bit	Read-Only	Data can be provided by the I/O system.
Coils	Single bit	Read-Write	Data can be alterable by the application program.
Input Registers	16-bit word	Read-Only	Data can be provided by the I/O system
Holding Registers	16-bit word	Read-Write	Data can be alterable by the application program.

Reference: MODBUS Application Protocol Specification V1.1b3, April 26 2012

For more information refer to the Modbus organisation website.

14.3. Modbus Transactions

Nr	Transaction	Function Code	Description
1	Read Coils	0x01	Read from 1 to 2000 contiguous status of coils in a remote device.
2	Read Discrete Inputs	0x02	Read from 1 to 2000 contiguous status of discrete inputs in a remote device.
3	Read Holding Registers	0x03	Read the contents of a contiguous block of holding registers in a remote device.
4	Read Input Registers	0x04	Read from 1 to 125 contiguous input registers in a remote device.
5	Write Single Coil	0x05	Write a single output to ON or OFF in a remote device.
6	Write Single Register	0x06	Write a single holding register in a remote device.
15	Write Multiple Coils	0x0F	In a sequence of coils, force each coil to either ON or OFF in a remote device.
16	Write Multiple Registers	0x10	Write a block of contiguous registers in a remote device.
22	Mask Write Register	0x16	In a single transaction, modify the contents of a specified holding register using a combination of an AND mask, an OR mask, and the register's current contents. Can be used to set or clear individual bits in the register.
23	Read/Write Multiple Registers	0x17	Performs a combination of one read operation and one write operation. The write operation is performed before the read.

Reference: MODBUS Application Protocol Specification V1.1b3, April 26 2012

For more information refer to the Modbus organisation website.

14.4. Modus Exception Codes

Exception Code	Name	Description
01	Illegal Function	The server does not recognize or permit the function code.
02	Illegal Data Address	The data address (register, discrete input or coil number) is not an permitted address for the server. If multiple registers were requested, at least one was not permitted.

Reference: MODBUS Application Protocol Specification V1.1b3, April 26 2012

For more information refer to the Modbus organisation website.

14.5. ASCII Table

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL 0	SOH 1	STX 2	ETX 3	EOT 4	ENQ 5	ACK 6	BEL 7	BS 8	HT 9	LF 10	VT 11	FF 12	CR 13	SO 14	SI 15
1x	DLE 16	DC1 17	DC2 18	DC3 19	DC4 20	NAK 21	SYN 22	ETB 23	CAN 24	EM 25	SUB 26	ESC 27	FS 28	GS 29	RS 30	US 31
2x	(sp) 32	! 33	" 34	# 35	$ 36	% 37	& 38	' 39	(40	) 41	* 42	+ 43	, 44	- 45	. 46	/ 47
3x	0 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55	8 56	9 57	: 58	; 59	< 60	= 61	> 62	? 63
4x	@ 64	A 65	B 66	C 67	D 68	E 69	F 70	G 71	H 72	I 73	J 74	K 75	L 76	M 77	N 78	O 79
5x	P 80	Q 81	R 82	S 83	T 84	U 85	V 86	W 87	X 88	Y 89	Z 90	[91	\ 92	] 93	^ 94	_ 95
6x	` 96	a 97	b 98	c 99	d 100	e 101	f 102	g 103	h 104	i 105	j 106	k 107	l 108	m 109	n 110	o 111
7x	p 112	q 113	r 114	s 115	t 116	u 117	v 118	w 119	x 120	y 121	z 122	{ 123	| 124	} 125	~ 126	DEL 127

14.6. RS485/RS232 Electrical Connection

14.6.1. RS485 Typical Connection

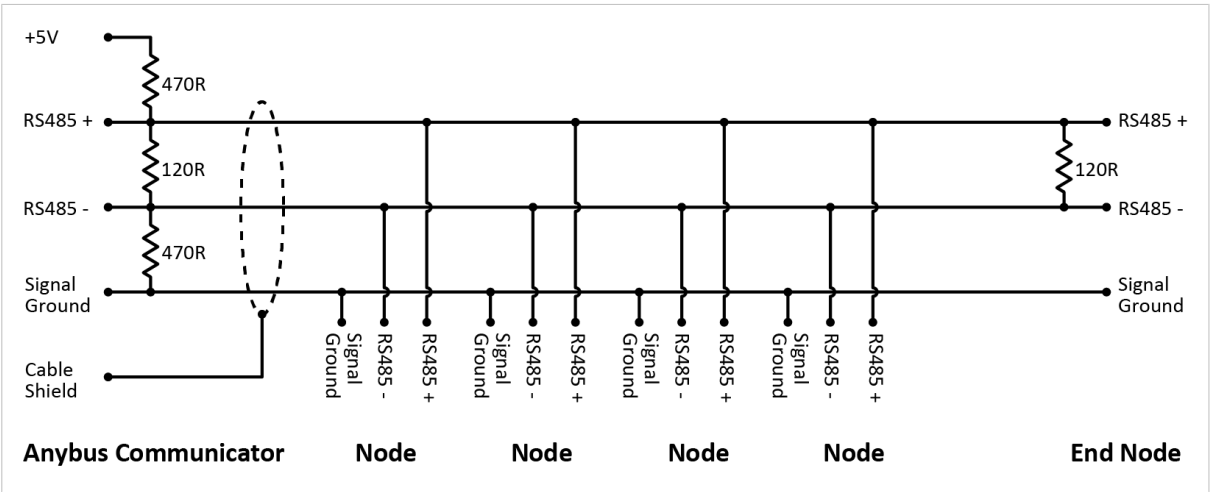


Figure 118. RS485

14.6.2. RS232 Typical Connection

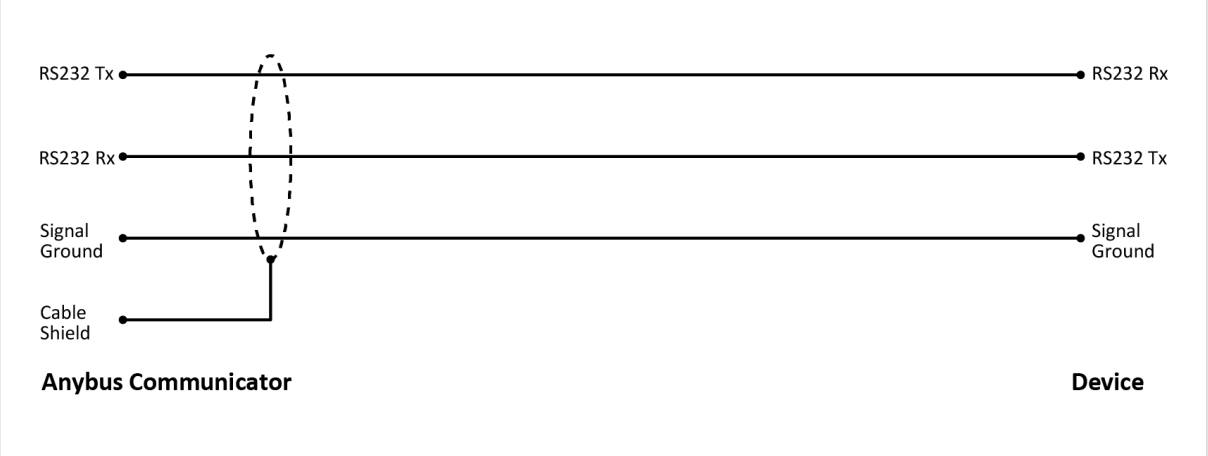


Figure 119. RS232

This page is intentionally left blank.