

Anybus[®] Communicator[™] - Modbus TCP to Modbus RTU/Serial USER MANUAL

SCM-1202-162
Version 1.3
Publication date 2026-05-27



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 © 2026 HMS Networks

Contact Information

Postal address:

Box 4126

300 04 Halmstad, Sweden

E-Mail: info@hms-networks.com

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. Cybersecurity	4
3.1. General Cybersecurity	4
3.2. Security Advisories	4
3.3. How to Report a Vulnerability	4
3.4. Product Cybersecurity Context	5
3.4.1. Security Defense in Depth Strategy	5
3.4.2. Purdue Model	6
3.5. Supported Transport Layer Security (TLS) 1.2 Cipher Suites	7
4. Preparation	8
4.1. Support and Resources	8
4.2. Cabling	8
4.3. Mechanical Tools and Equipment	8
4.4. System Requirements	8
4.4.1. Supported Operating Systems	8
4.4.2. Supported Web Browsers	8
4.5. HMS Software Applications	9
4.6. Third-Party Software Applications	9
4.7. Software License Information	9
5. About Anybus Communicator	10
5.1. Serial Protocol Communication	10
5.1.1. Serial Protocol Types	10
5.1.2. Serial Protocol Building Blocks	11
5.2. How the Communication Works	14
5.3. How the Data Exchange Works	16
5.4. Data Integrity	16
6. Installation	17
6.1. External Parts	17
6.2. Connector Port Guide	18
6.3. DIN Rail Mounting	19
6.4. Connect to Modbus RTU Serial Network via RS-232/485	20
6.5. Connect to Modbus TCP Network	22
6.6. Connect to Power	24
6.7. Security Switch	26
6.8. Lock the Cables	28
6.9. DIN Rail Demount	29
7. Accessing the Communicator	30
7.1. Connect to Configure the Communicator	30
7.2. Access the Built-In Web Interface from HMS IPconfig	31
7.3. Access the Built-In Web Interface from a Web Browser	32

7.4. Communicator Built-In Web Interface Overview	34
8. Role Management	35
8.1. Roles Overview	35
8.2. Roles Configuration Access	36
8.3. Set Up Roles	38
8.4. Set or Change Password	40
8.5. Logout From the Communicator Web Interface	40
9. Configuration Quick Guide	41
9.1. Prepare Configuration	41
9.2. Security Setup	44
9.3. Setup New Configuration	45
9.4. PLC Configuration	47
9.5. Verify Operation	48
10. Communicator Configuration	50
10.1. General Subnetwork Settings	51
10.1.1. Communication Serial Protocol	51
10.1.2. Communication Basic Settings	52
10.1.3. Communication Advanced Settings	54
10.2. About Transaction Templates	61
10.2.1. Transaction Template Example	61
10.2.2. Transaction Template Types	63
10.2.3. Frame Field Types	65
10.3. Build Transaction Template	67
10.3.1. Add Transaction Template	67
10.3.2. Add Frame Fields	71
10.3.3. Configure Frame Field Settings	73
10.3.4. Data Delimiter and Subnet Delimiter Options	76
10.3.5. Store Transaction Templates	79
10.4. Nodes and Transactions	80
10.4.1. Node and Broadcast Node	80
10.4.2. Add Node	81
10.4.3. Import Node Settings From Other Communicator Unit	82
10.4.4. Import Transaction Template From Other Communicator Unit	86
10.4.5. Node Settings	89
10.4.6. Add Transactions	90
10.4.7. Transaction Settings	93
10.4.8. Activate/Deactivate Transaction	95
10.4.9. Duplicate Transaction	95
10.4.10. Delete Transaction	96
10.5. Modbus TCP Settings	97
10.5.1. To Use Automatic I/O Sizes	97
10.5.2. To Configure I/O Sizes Manually	97
10.5.3. IP Settings	98
10.5.4. Timeout Time Settings	99
10.6. Modbus TCP Advanced Settings	100
10.6.1. Legacy Mode for Anybus Address Mode Modbus TCP	100
10.7. Security Settings	100
10.7.1. Local Time and Date	100
10.7.2. Network Time Protocol (NTP) Synchronization	101
10.7.3. Installing a Web Server Certificate	102
10.7.4. View Current Certificate	104
10.7.5. Update or Delete a Certificate	105

10.7.6. Secure Connection With HTTPS Mode	107
10.7.7. None Secure HTTP Mode	109
10.7.8. Device IP Address Discovery via HMS Software Applications	110
10.8. I/O Configuration	111
10.8.1. Optimize the I/O Configuration	112
10.8.2. Map Area Transactions Order	113
10.8.3. Map Area	114
10.8.4. Trigger Byte	115
10.8.5. Endian Swap	115
10.8.6. Offline Option	118
10.8.7. Live List	119
10.8.8. Data Exchange Control	120
10.9. Configuration Notes	121
10.9.1. Add Configuration Note	121
10.9.2. View and Edit Configuration Notes	123
10.10. Apply Configuration	124
10.11. To Use an Existing Configuration	125
10.12. To Use a Communicator Classic Configuration	126
11. PLC Configuration	128
11.1. PLC Device Security	128
11.2. Export I/O Configuration	128
11.3. Modbus Addressing and Register Mapping	129
11.3.1. The Difference Between Address and Register Start	129
11.3.2. To Read and Write Data	130
11.3.3. Default Address Mode I/O Data Mapping	131
11.3.4. Legacy Address Mode I/O Data Mapping	132
12. Verify Operation	133
12.1. Communicator Status Monitor	133
12.2. Communicator LED Indicators	135
12.3. Ethernet RJ45 LED Indicators	137
13. Use Cases	138
13.1. Temperature Regulator - Modbus RTU Use Case	138
13.1.1. About the Use Case	138
13.1.2. Before You Begin	138
13.1.3. Choose Serial Protocol Type	139
13.1.4. Setup Serial Communication	139
13.1.5. Setup the Node	140
13.1.6. Setup the Transactions	141
13.1.7. Check the I/O Configuration	143
13.2. AC Motor Drive - Custom Request/Response Use Case	144
13.2.1. About the Use Case	144
13.2.2. Before You Begin	144
13.2.3. Choose Serial Protocol Type	145
13.2.4. Setup Serial Communication	145
13.2.5. Create Transaction Templates	146
13.2.6. Setup Node and Transactions	150
13.2.7. Check the I/O Configuration	151
13.3. Barcode Scanner - Custom Produce/Consume Use Case	152
13.3.1. About the Use Case	152
13.3.2. Before You Begin	152
13.3.3. Choose Serial Protocol Type	153
13.3.4. Setup Serial Communication	153

13.3.5. Create Transaction Templates	154
13.3.6. Setup Node and Transactions	156
13.3.7. Check the I/O Configuration	157
14. Maintenance	158
14.1. Action on Fatal Error	158
14.2. Configuration Port IP Settings	159
14.3. Dashboard	160
14.4. Configuration File Handling	163
14.4.1. Export Configuration	163
14.4.2. Import Configuration	164
14.5. Clear and Revert Configuration	165
14.6. Firmware Management	166
14.6.1. View the Firmware Version	166
14.6.2. Firmware and Configuration Compatibility	166
14.6.3. Firmware File Validation	166
14.6.4. Update Firmware	167
14.7. Change Language	168
15. Troubleshooting	169
15.1. Diagnostics	169
15.1.1. Serial RS-232/485 Data Monitor	169
15.1.2. I/O Data	170
15.1.3. Event Log	171
15.1.4. LED Status	172
15.2. Reset to Factory Settings	173
15.3. Firmware Upgrade Error Management	176
15.4. Support	179
15.4.1. Support Package	179
16. Technical Data	180
16.1. Technical Specifications	180
17. End Product Life Cycle	181
17.1. Secure Data Disposal	181
18. Reference	182
18.1. About Input Registers and Holding Registers	182
18.2. Modbus Data Model	182
18.3. Modbus Function Codes	183
18.4. Modbus Exception Codes	183
18.5. ASCII Table	184
18.6. RS232/RS485 Electrical Connection	185
18.6.1. RS485 Typical Connection	185
18.6.2. RS232 Typical Connection	185

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.hms-networks.com/technical-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.hms-networks.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. Cybersecurity

3.1. General Cybersecurity

**IMPORTANT**

It is important to maintain the cybersecurity of the Communicator.

Before connecting the Communicator to a PLC, ensure the PLC is configured and installed in accordance with the PLC supplier hardening guidelines.

**IMPORTANT**

To physically secure networks and equipment and to prevent unauthorized access, it is recommended to install the equipment in a locked environment.

**IMPORTANT**

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

**IMPORTANT**

To avoid exposure of sensitive data, always perform a factory reset before decommissioning the equipment.

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

See [Reset to Factory Settings \(page 173\)](#).

3.2. Security Advisories

For cybersecurity reasons, stay informed about new vulnerabilities and follow the recommended actions.

HMS Networks Security Advisories includes information about our product vulnerabilities and available solutions.

You find our Safety Advisories at www.hms-networks.com/cybersecurity/security-advisories.

3.3. How to Report a Vulnerability

HMS Networks place the utmost importance on the security of our products and systems, however, despite all the measures we take, it cannot be excluded that vulnerabilities persist.

To report a potential vulnerability in an HMS product or service, please visit www.hms-networks.com/cybersecurity/report-a-vulnerability and follow the instructions.

3.4. Product Cybersecurity Context

3.4.1. Security Defense in Depth Strategy

The defense in depth strategy of the Communicator includes the following security measures:

- Secure Boot: Security standard used to ensure that the Communicator boots using only software that is trusted by HMS Networks.
- Signed firmware: HMS Networks delivers digitally signed firmware. Before the firmware file is imported into the Communicator, the firmware upgrade function performs a validation of the file, to ensure that is authentic.
- Security switch: Used to lock unauthorized access to the Communicator built-in web interface.
- The Communicator is intended to be installed in a Process Control Network (PCN) environment. See Level 1 in the [Purdue Model \(page 6\)](#).
- To physically secure networks and equipment and to prevent unauthorized access, the Communicator is intended to be installed in a locked environment.

3.4.2. Purdue Model

The Communicator is intended to be part of the process control network in Level 1 (E), to enable communication between PLCs or between a PLC and peripheral devices.

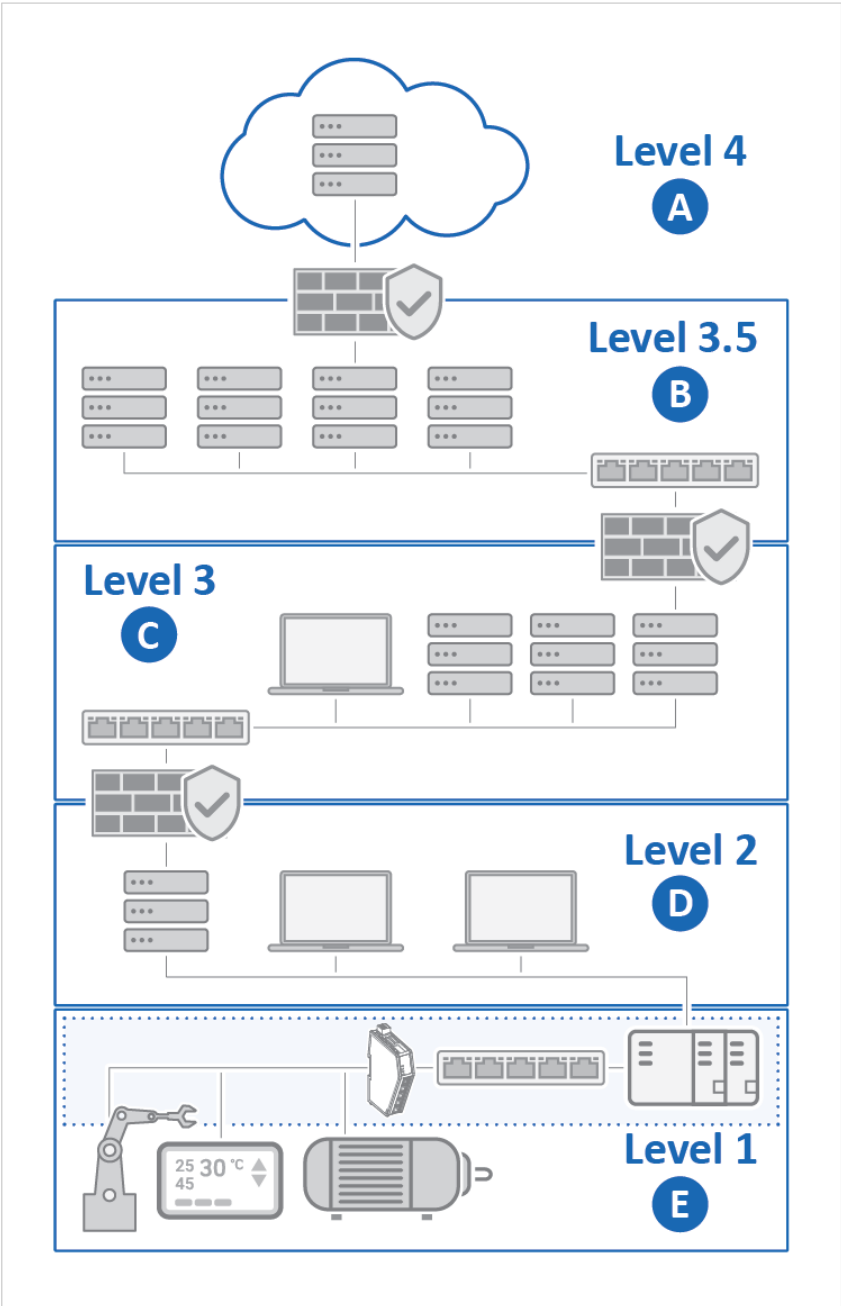


Figure 1. Purdue model, product security context

IT Network	OT Network
A. Level 4: Enterprise Network Example: Cloud solution, Business LAN (VPN)	C. Level 3: Advanced Control Network (ACN) Example: SCADA systems, Business control
B. Level 3.5: Perimeter Network Example: Demilitarized Zone (DMZ)	D. Level 2: Supervisory Control Example: Operator panels, Operator stations, Engineering stations
	E. Level 1: Process Control Network (PCN) Environment where the Communicator is installed Example: Factory floor, Industrial product line

3.5. Supported Transport Layer Security (TLS) 1.2 Cipher Suites

The Communicator supports encrypted communication using TLS 1.2.

The following industry-recommended TLS 1.2 cipher suites are supported:

- ECDHE-ECDSA-WITH-AES128-GCM-SHA256
- ECDHE-ECDSA-WITH-AES128-CBC-SHA256
- ECDHE-RSA-WITH-AES128-GCM-SHA256
- ECDHE-RSA-WITH-AES128-CBC-SHA256
- RSA-WITH-AES128-GCM-SHA256
- RSA-WITH-AES128-CBC-SHA256
- RSA-WITH-AES256-CBC-SHA256

4. Preparation

4.1. Support and Resources

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

4.2. Cabling

Have the following cables available:

- Power cable.
- Ethernet cable for configuration.
- 7-pin screw terminal block connector is included with the product.
- Ethernet cable for connecting to network.

4.3. Mechanical Tools and Equipment

Have the following tool(s) 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.

4.4. System Requirements

4.4.1. 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
Windows 11 64-bit	Windows 11 64-bit

4.4.2. Supported Web Browsers

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

- Google Chrome
- Microsoft Edge
- Mozilla Firefox

4.5. HMS Software Applications

Download the software installation files and user documentation from www.hms-networks.com/technical-support.

HMS IPconfig

Use the software application HMS 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

HMS IPconfig is only available for Windows.

4.6. Third-Party Software Applications

Microsoft Excel

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 map** file and the **Event log** file.

4.7. Software License Information

For license agreements regarding the third-party software used in the Communicator, refer to the *LICENSE.txt* file(s) included in the Communicator firmware update package zip file.

To download the Communicator firmware update package zip file, please visit www.hms-networks.com/technical-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.

5. About Anybus Communicator

5.1. Serial Protocol Communication

5.1.1. 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.

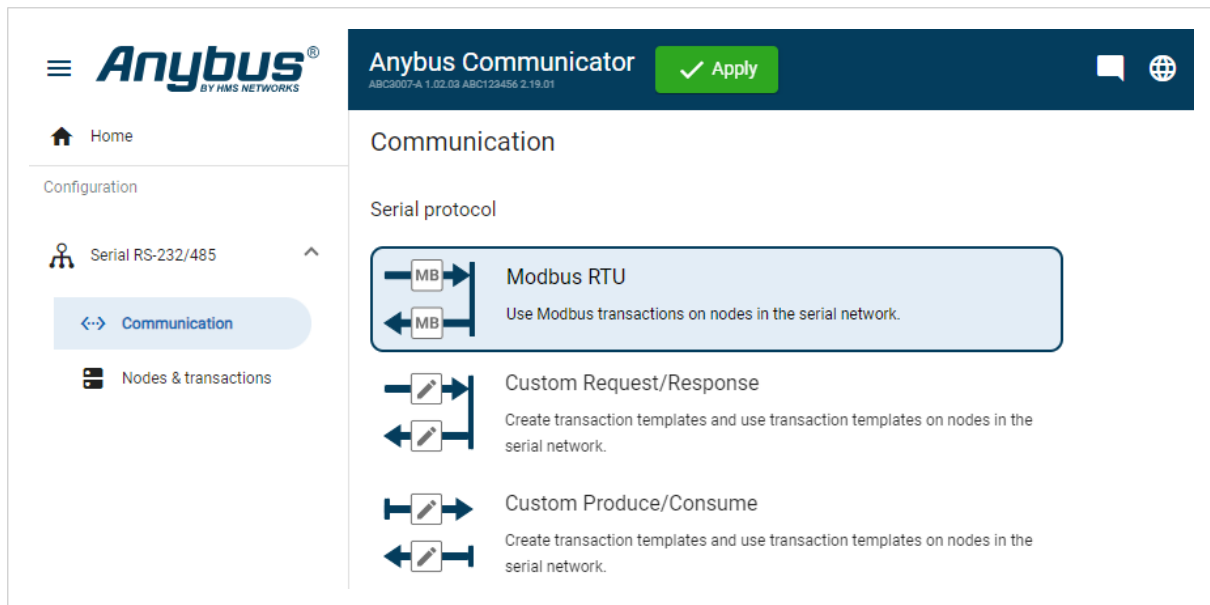


Figure 2. Communication, Serial protocol

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.

5.1.2. Serial Protocol Building Blocks

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

Node

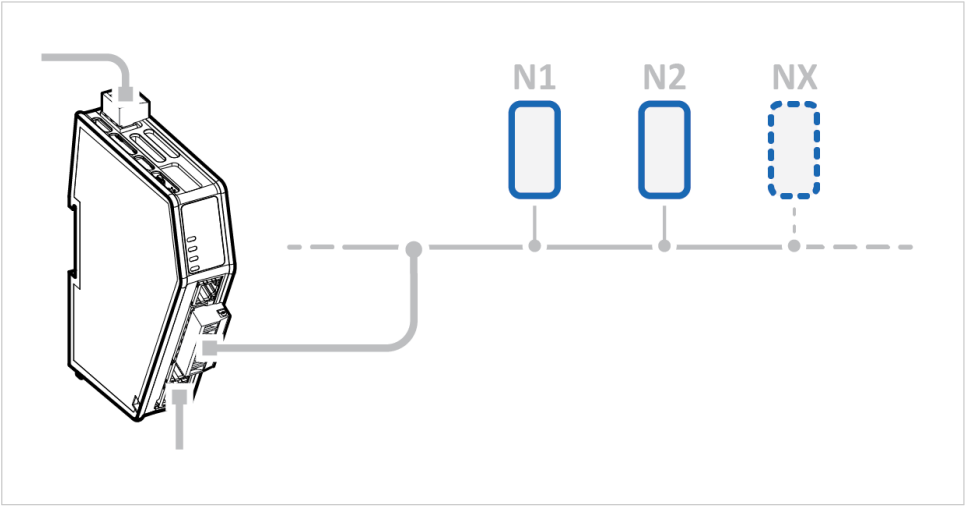


Figure 3. Nodes on a subnetwork

A node represents a single device on the subnetwork.

Each node can be associated with a number of transactions.

Nodes and Transactions

Anybus Communicator

Article Number: ABC2807-A Version: 1.2.0 Serial Number: ABC123456 GUI Version: 1.2.1

✓ Apply

Nodes

Custom request/response

+ Add node

1 My Drive

Node settings

Node address

1

Name

My Drive

Timeout time

1000 ms

Reconnection ti...

10000 ms

Retries

0

Transactions

+ Add

▼

Duplicate

Delete

	Active	Transaction name	Transaction template name	Size to EtherNet/IP™ (bytes)	Size from EtherNet/IP™ (bytes)
☒	☒	Control Word	Write Parameter (0x02)	0	2
☐	☒	Speed	Write Parameter (0x02)	0	2
☐	☒	Status Word	Read Parameter (0x01)	2	0
☐	☒	Actual Speed	Read Parameter (0x01)	2	0

Figure 4. 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.

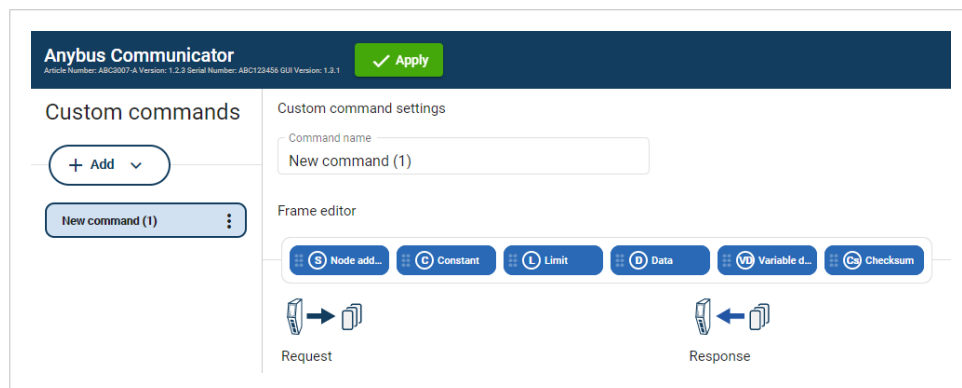


Figure 5. Frame editor

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.

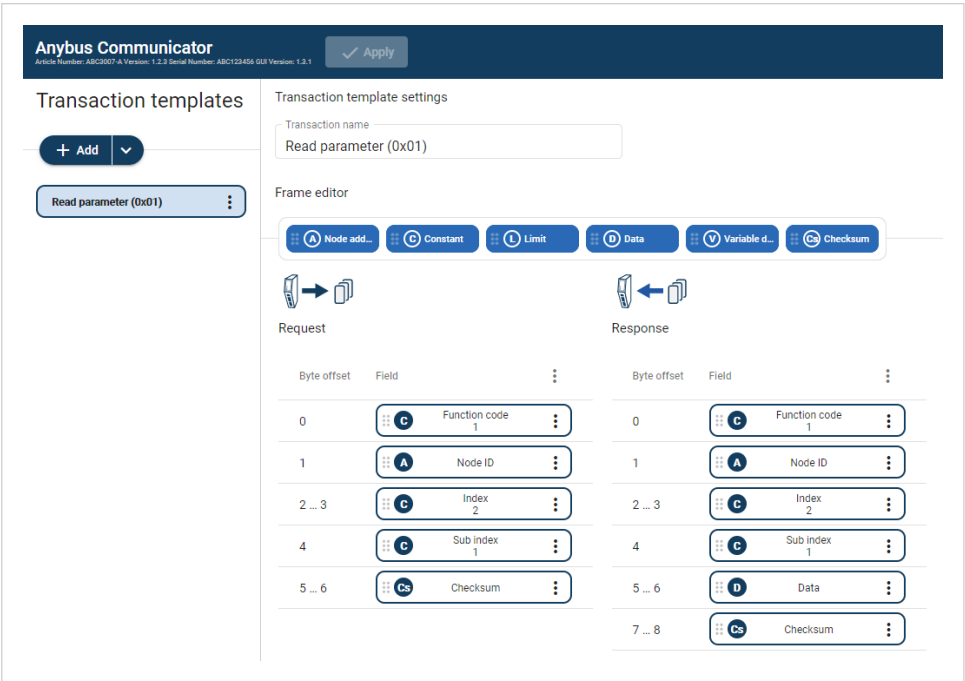


Figure 6. Frame editor, Request and Response

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 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.

5.2. 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.

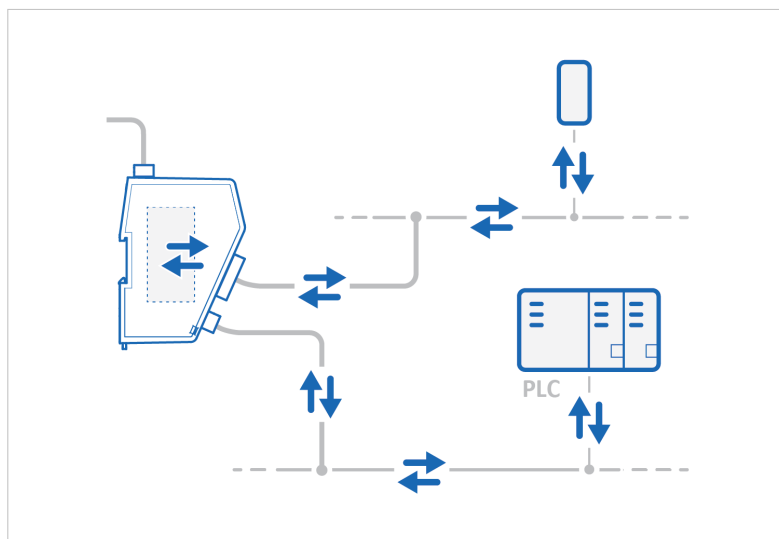


Figure 7. Process data traffic overview

For example:

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

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

Request Process Data

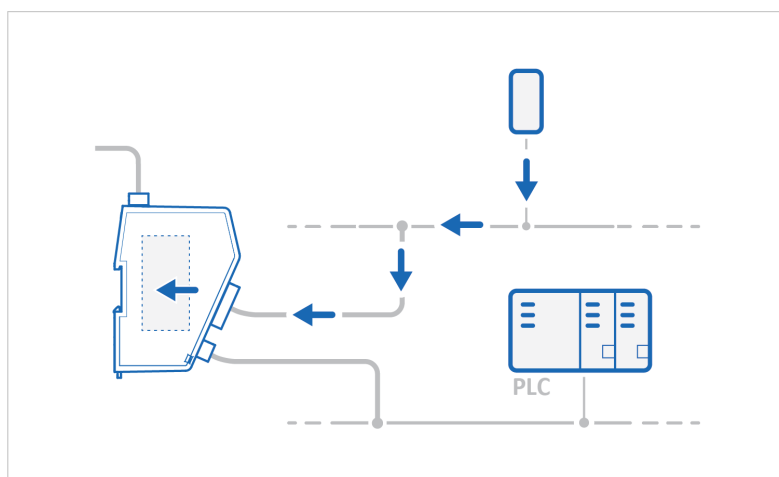


Figure 8. Process data traffic from nodes to client

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

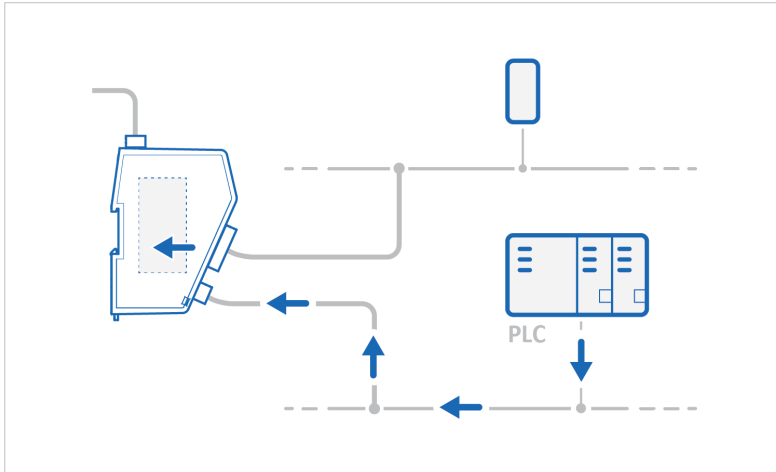


Figure 9. Process data traffic from client to nodes

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.

5.3. How the Data Exchange Works

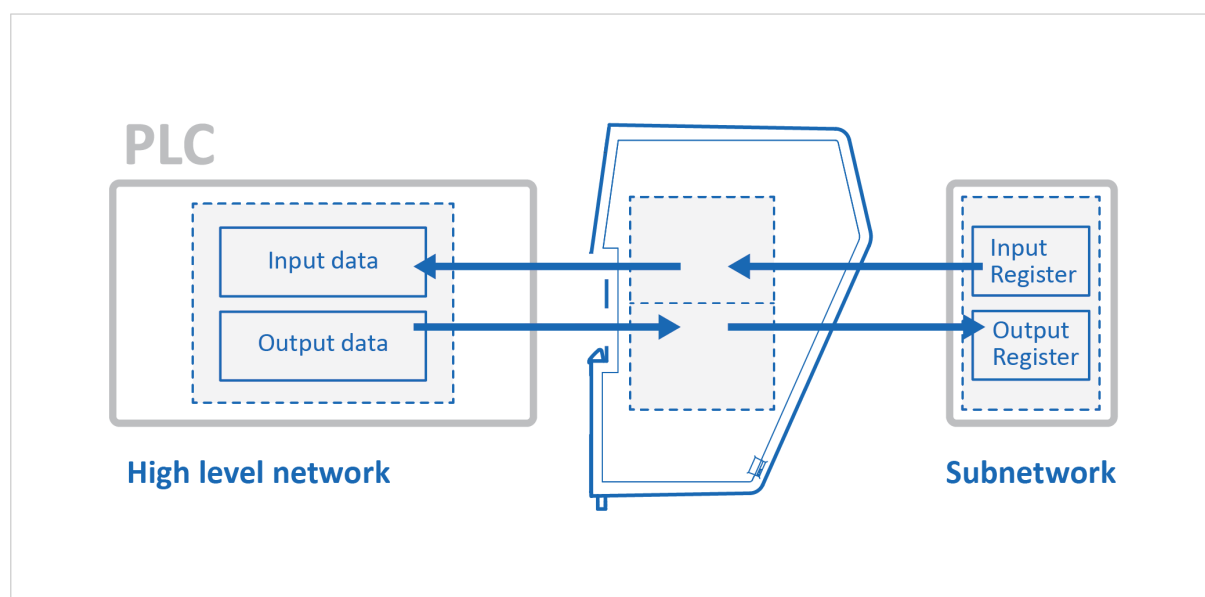


Figure 10. The Communicator internal memory areas

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.

Input Data

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

Output Data

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

5.4. Data Integrity

A snapshot of the process data buffer between the 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.

6. Installation

6.1. External Parts

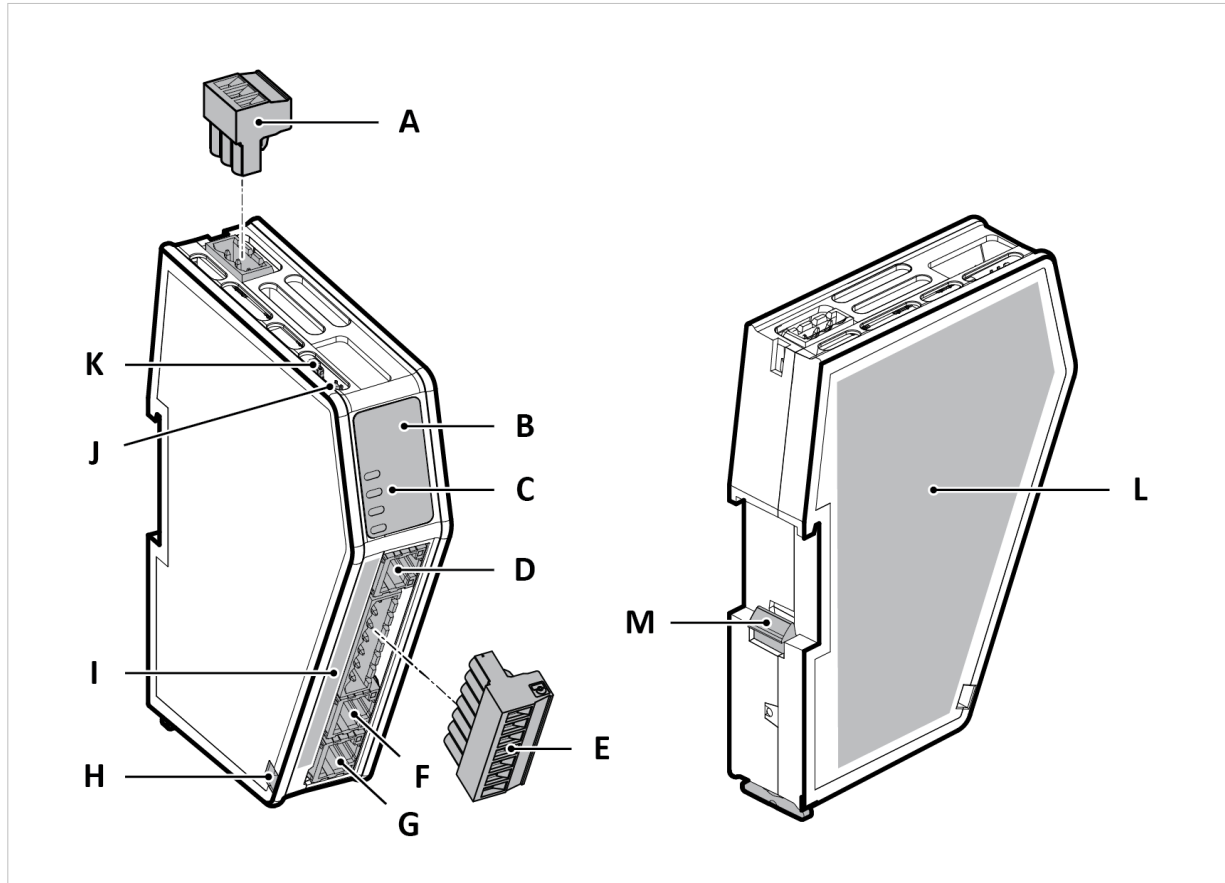


Figure 11. Communicator external parts

- | | | |
|-------------------------------|---|--|
| A. Power connector | E. Modbus RTU Serial (X3) RS-232/485 via 7-pin screw terminal block | J. Security switch |
| B. Label with LED designation | F. Modbus TCP (X2.1) connector | K. Factory reset button |
| C. Status LEDs | G. Modbus TCP (X2.2) connector | L. Laser engraved label with product information |
| D. Configuration port | H. Cable tie mount | M. DIN rail locking mechanism |
| | I. Laser engraved connectors designation | |

6.2. Connector Port Guide

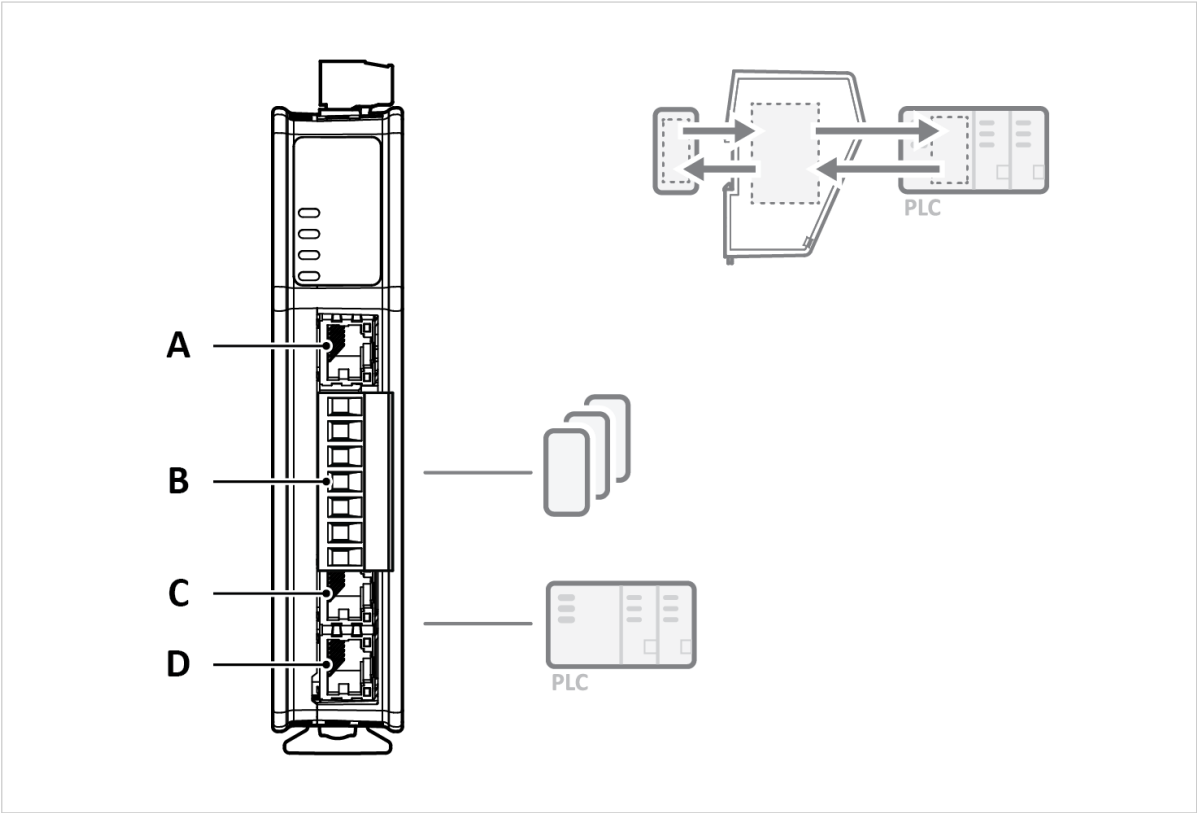


Figure 12. Communicator connector ports

Position	Port Number	Connector	Port Usage
A	X1	Ethernet RJ45	Configuration port
B	X3	7-pin screw terminal block	RS-232/485 Serial network
C	X2.1	Ethernet RJ45	Modbus TCP network
D	X2.2	Ethernet RJ45	Modbus TCP network

See Also

[Connect To Modbus RTU Serial Network Via RS-232/485 \(Page 20\)](#)

[Connect To Modbus TCP Network \(Page 22\)](#)

6.3. 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.

**IMPORTANT**

To physically secure networks and equipment and to prevent unauthorized access, it is recommended to install the equipment in a locked environment.

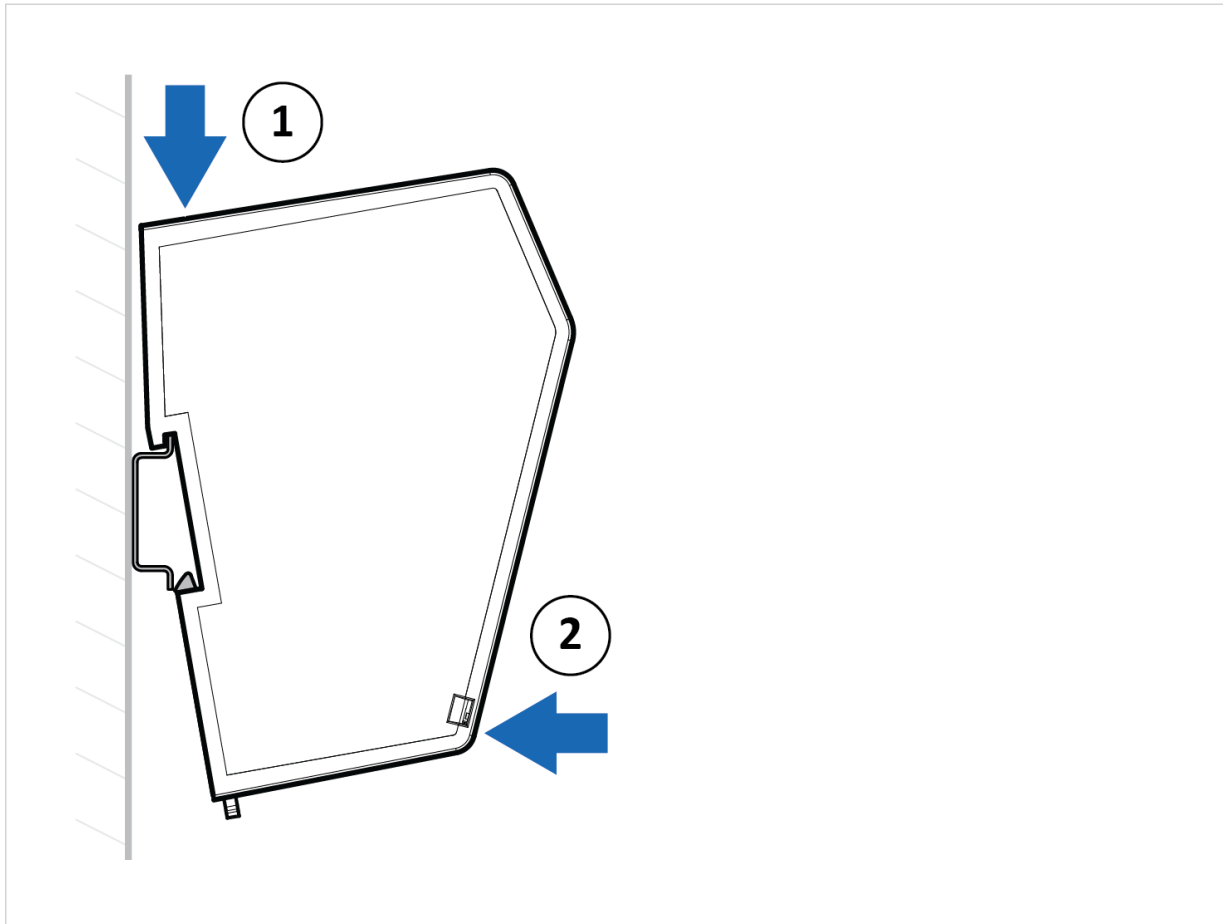


Figure 13. Attach the Communicator on the DIN rail

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.

6.4. Connect to Modbus RTU Serial Network via RS-232/485

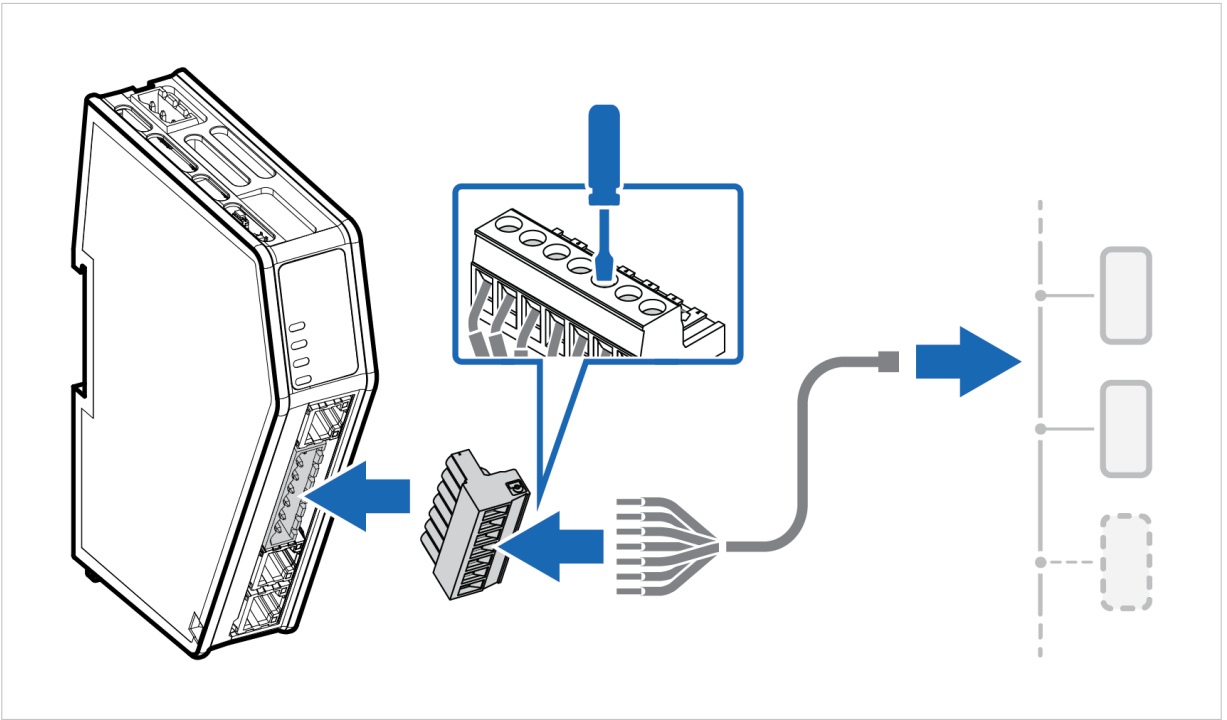
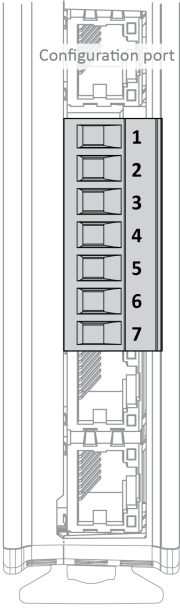


Figure 14. Connect to Modbus RTU network

Modbus RTU Connector Pinout

**NOTE**

Use minimum 90 °C copper (Cu) wire only.

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

Procedure

1. Insert the cable wires into the 7-pin connector and tighten the wire clamp screws.
2. Connect the 7-pin connector to the Communicator.
3. Connect the Communicator to your Modbus RTU network.

See Also

[Connector Port Guide \(Page 18\)](#)

[Connect To Modbus TCP Network \(Page 22\)](#)

[Connect To Power \(Page 24\)](#)

[Communicator Status Monitor \(Page 133\)](#)

[Communicator LED Indicators \(Page 135\)](#)

[LED Status \(Page 172\)](#)

[Firmware Upgrade Error Management \(Page 176\)](#)

6.5. Connect to Modbus TCP Network

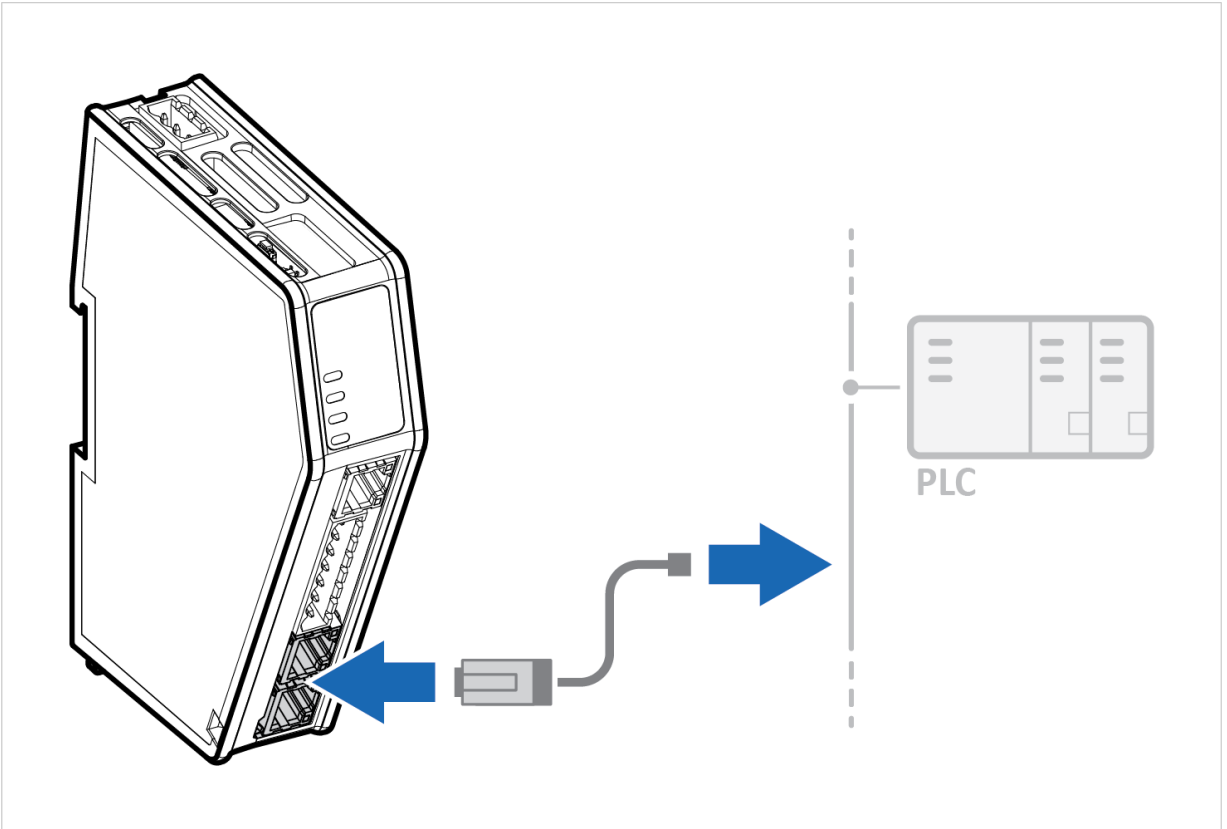
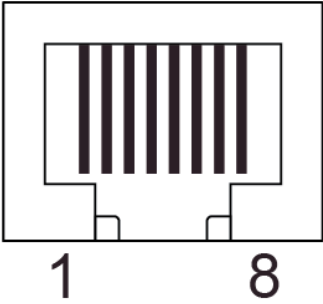


Figure 15. Connect to Modbus TCP network

Ethernet RJ45 Connector Pinout

Ethernet RJ45 Connector	Pin	Description
	1	TD+
	2	TD-
	3	RD+
	4	Not used
	5	Not used
	6	RD-
	7	Not used
	8	Not used

Procedure

Connect the Communicator to your Modbus TCP network.

See Also

- [Connector Port Guide \(Page 18\)](#)
- [Connect To Modbus RTU Serial Network Via RS-232/485 \(Page 20\)](#)
- [Connect To Power \(Page 24\)](#)

[Communicator Status Monitor \(Page 133\)](#)
[Communicator LED Indicators \(Page 135\)](#)
[LED Status \(Page 172\)](#)
[Firmware Upgrade Error Management \(Page 176\)](#)

6.6. 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.

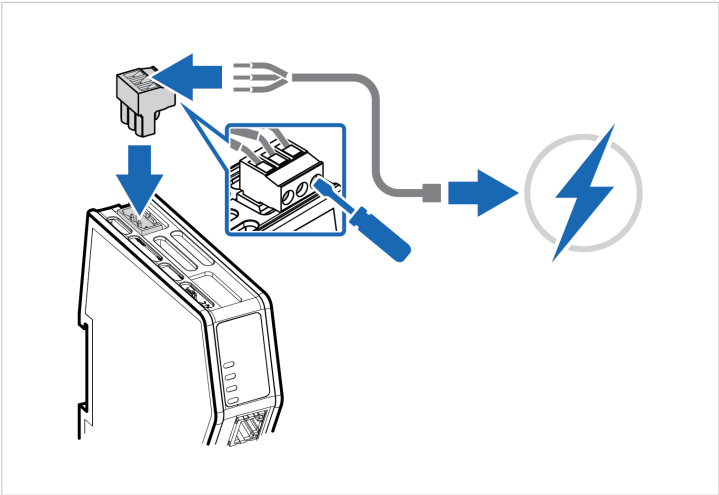
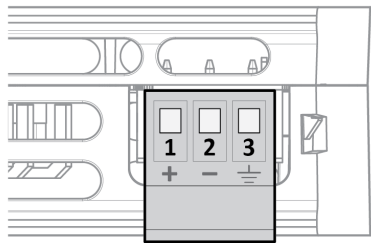


Figure 16. Connect to power

Power Connector Pinout

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

Procedure

1. Insert the cable wires to the terminal block and tighten the wire clamp screws.
2. Connect the terminal block to the Communicator.
3. Connect the Communicator to a power supply.
4. Turn on the power supply.

See Also

[Connect To Modbus RTU Serial Network Via RS-232/485 \(Page 20\)](#)

[Connect To Modbus TCP Network \(Page 22\)](#)

[Communicator Status Monitor \(Page 133\)](#)

[Communicator LED Indicators \(Page 135\)](#)

[LED Status \(Page 172\)](#)

[Firmware Upgrade Error Management \(Page 176\)](#)

6.7. Security Switch



IMPORTANT

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 cannot be accessed, and the Communicator cannot 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

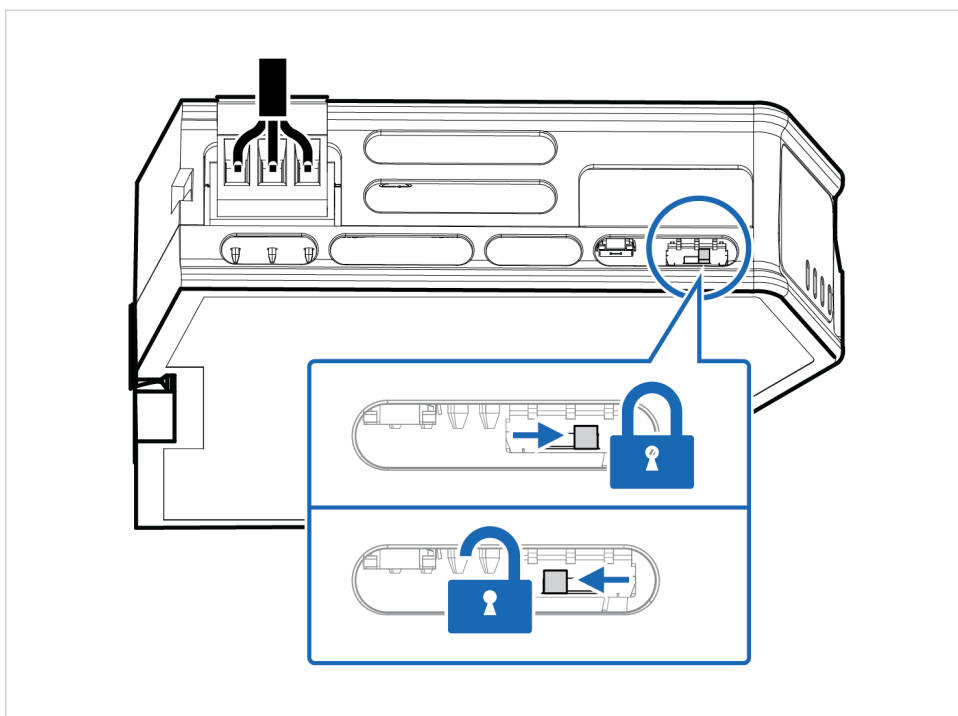


Figure 17. Security switch in locked and unlocked position

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

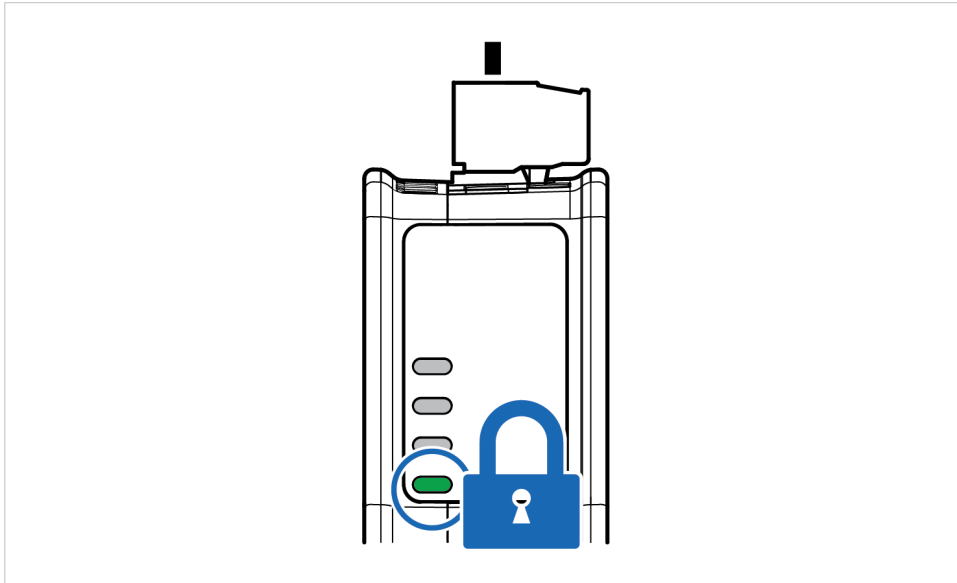


Figure 18. Security switch locked status LED

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.

6.8. Lock the Cables

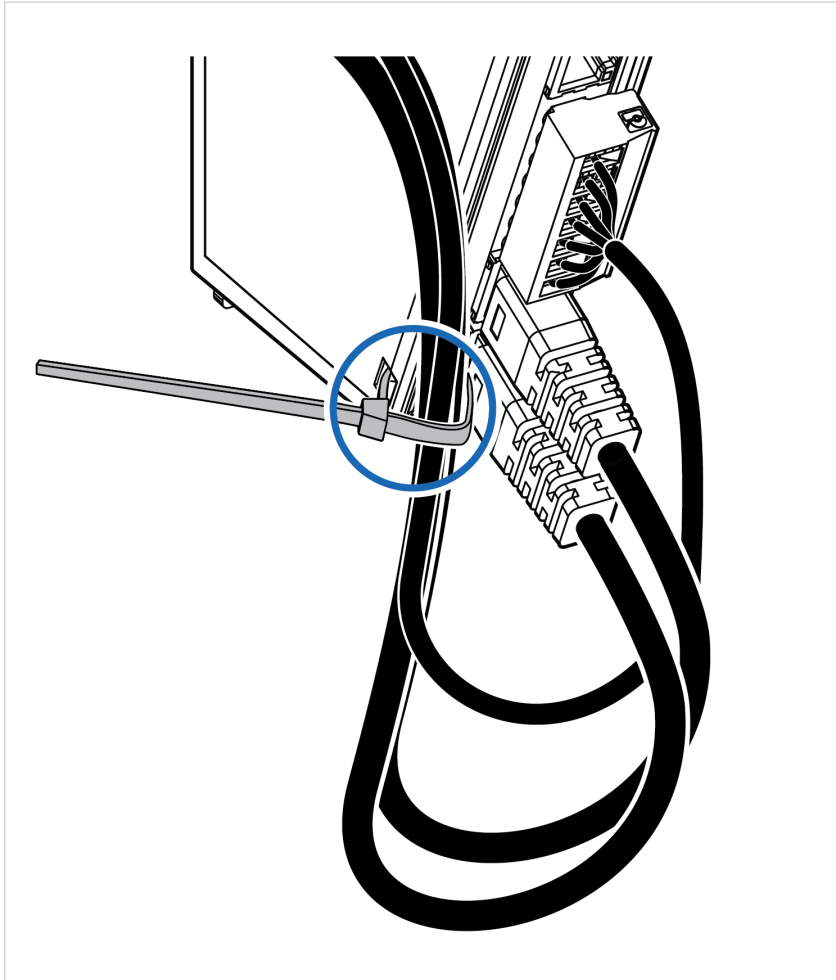


Figure 19. Lock the cables

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

6.9. 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:

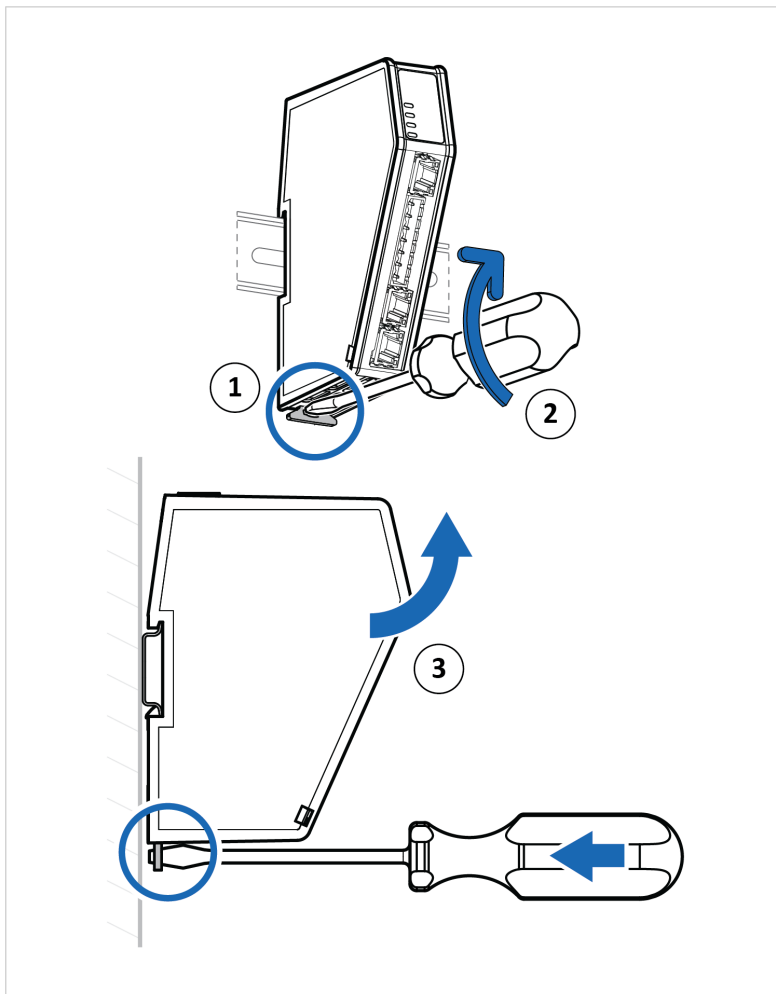


Figure 20. Unlock and unhook the Communicator

1. Insert the screwdriver into the Communicator DIN rail locking mechanism.
2. To unlock the Communicator DIN rail locking mechanism, turn the screwdriver clockwise.
3. Hold the screwdriver in the DIN rail locking mechanism while you unhook the Communicator from the DIN rail.

7. Accessing the Communicator

7.1. Connect to Configure the Communicator

Procedure

1. Connect the Communicator to your:
 - Serial RS-232/485 network — (1) (2) and (3)
 - Modbus TCP network — (4)

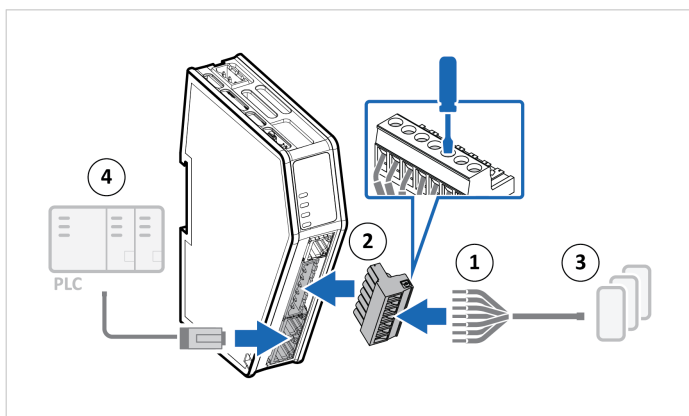


Figure 21. Connect to networks

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

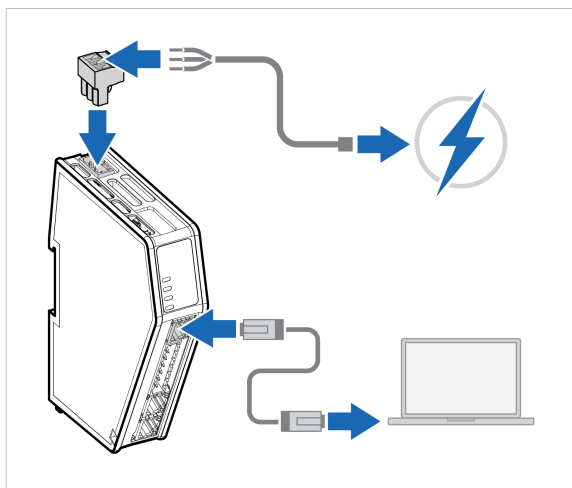


Figure 22. Connect to PC and power

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

Before You Begin

Security Settings



NOTE

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



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 3250 UDP is open in your PC Windows Firewall.

IP Address



NOTE

The Communicator default IP address is 192.168.0.10.



TIP

When you have accessed the Communicator built-in web interface, you can change the IP settings for the Communicator configuration port on the **System > Configuration port** page.

Download HMS IPconfig Installation Files

Download the software application HMS IPconfig installation files and user documentation from www.hms-networks.com/technical-support.

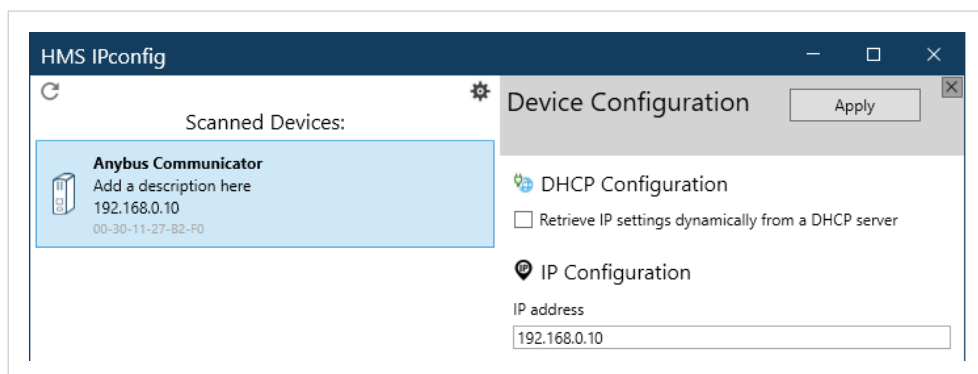
Procedure

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

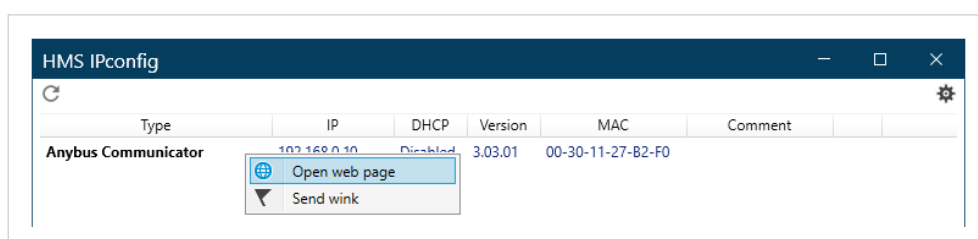


- HMS IPconfig automatically starts scanning for compatible and active HMS devices.
 - Found HMS devices are added to the device list.
3. To open the settings pane, click on the Communicator in the device list.

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

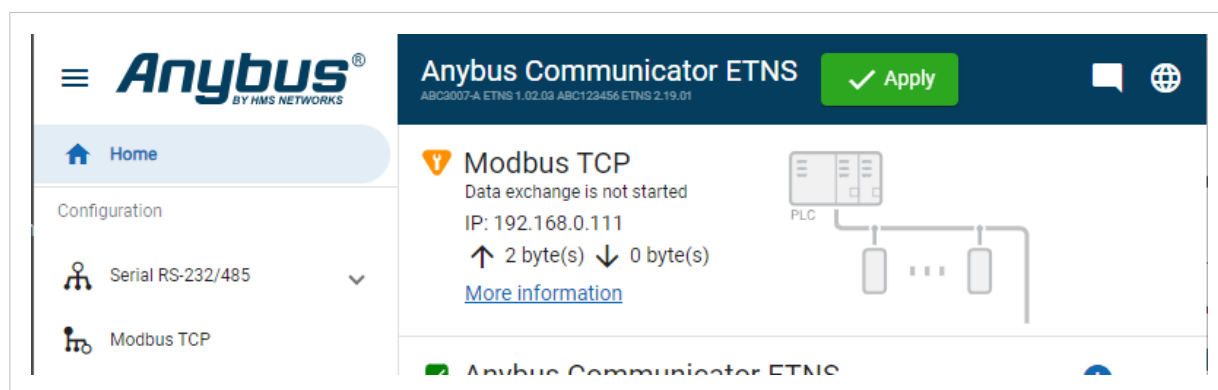


5. To open the **Open web page** built-in web interface, click Communicator.



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

Security Settings



NOTE

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

**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.

IP Address**NOTE**

The Communicator configuration port default IP address is 192.168.0.10.

**NOTE**

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

**TIP**

When you have accessed the Communicator built-in web interface, you can change the IP settings for the Communicator configuration port on the **System > Configuration port** page.

Procedure

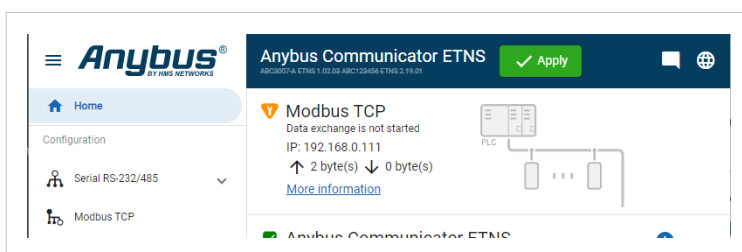
1. On the PC accessing the Communicator built-in web interface, set a static IP address within the same IP address range as the Communicator IP address.



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

Menu item	Description
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.
Modbus TCP	High Level Network with Client. Configure I/O Size and IP settings.
I/O configuration	View the added commands mapped to the Communicator internal memory area.
System	Define how the device should behave if a serious error occurs. Configure the Communicator configuration port IP settings.
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.

8. Role Management

8.1. Roles Overview

The Communicator provides enhanced security through user roles with different permissions levels.

Role	Default	Permission	Password Management
Administrator	Enabled	Full access to all features and settings. Can manage other roles.	Require password to login. Default password is printed on the product housing.
Operator	Disabled	Read-only access.	Blank password by default. Administrator can enable password protection. The password can be set and changed by the Operator.
User	Disabled	Configure IO-device/adaptor/ server/slave network, maintenance and diagnostics.	Administrator can enable role. Default password is printed on the product housing.

8.2. Roles Configuration Access

Administrator Role - Full Access

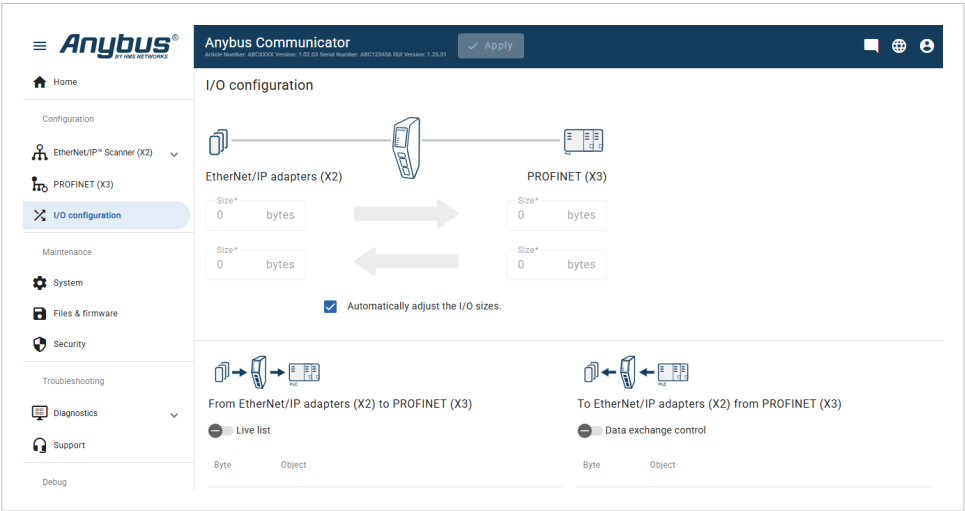


Figure 24. Example: Editable I/O configuration settings

The Administrator role has full access to all configuration settings in the Communicator web interface.

All settings are available for viewing and editing without restrictions.

Operator Role - Limited Access

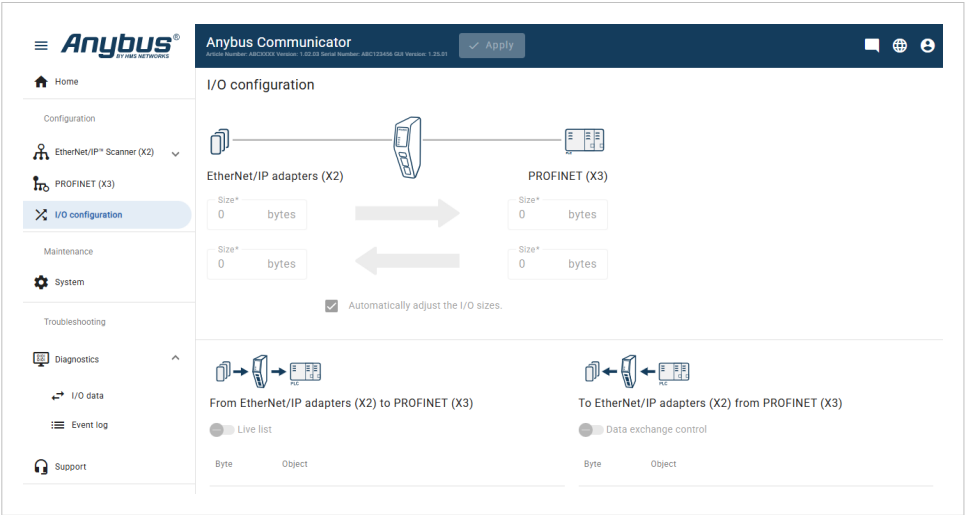
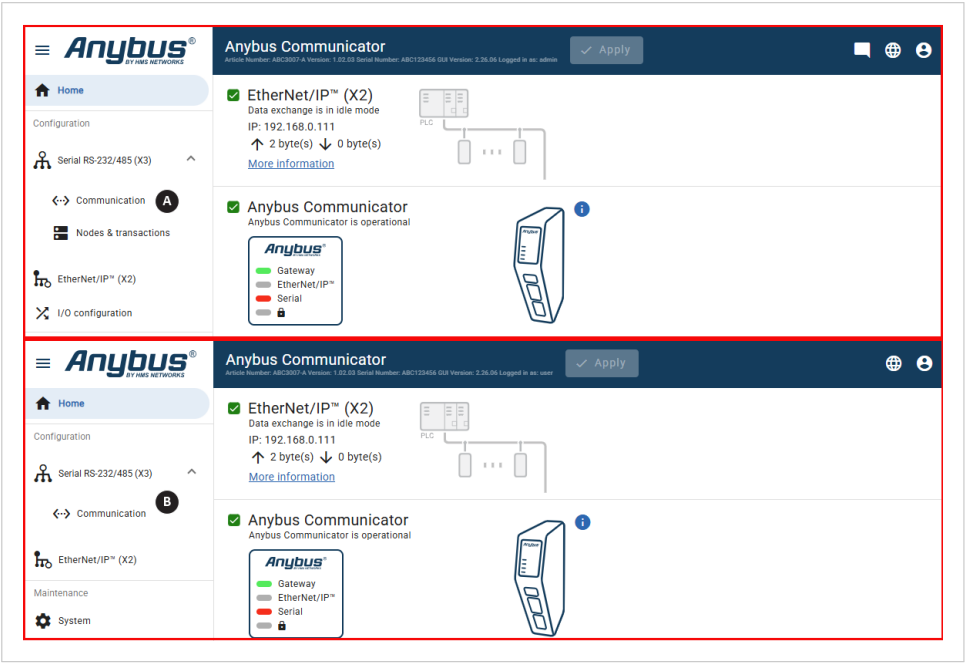


Figure 25. Example: Locked and grayed-out I/O configuration settings

The Operator role has read-only access to the configuration settings in the Communicator web interface.

Restricted settings are locked and grayed out.

User Role - Configuration Access



- A. Administrator
- B. User

Figure 26. Example: Menu visibility by user role

The User role provides limited access to the configuration options in the Communicator web interface.

Controller network configuration settings are not available for the User role.

Other restricted settings are locked and grayed out.

8.3. Set Up Roles

By default, **Roles** are disabled, and the user is automatically logged in to the Communicator web interface. No user identity or credentials are required.

Before You Begin



NOTE

The Communicator includes default credentials: username *admin* (lowercase) and password printed on the product housing.



NOTE

If no password is set for a User account, the default password printed on the product housing is used. Always set a password during account activation.

Procedure



IMPORTANT

Enable HTTPS before activating roles to ensure passwords are encrypted.

Activate Roles:

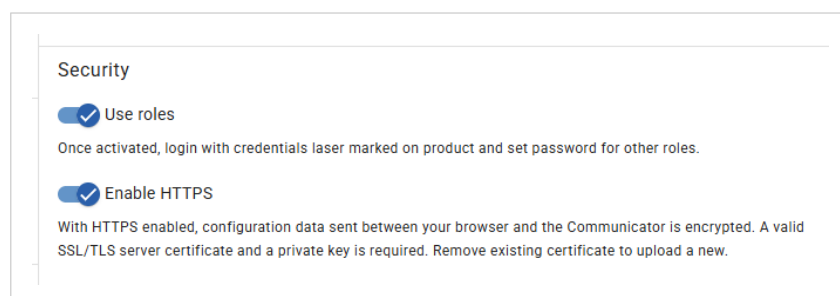


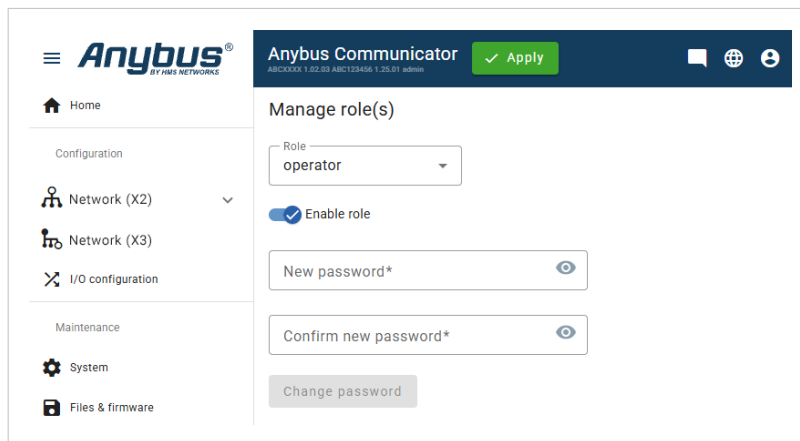
Figure 27. System page with Use role(s) activated

1. Open the **System** page.
2. In the **Security** section, click the **Use roles** toggle to activate roles.
3. Click **Apply** in the web interface header.

Manage Roles and Passwords:

1. When roles are enabled for the first time, log in using the default credentials printed on the product housing.
If an administrator password has already been configured, use that password instead.

- From the **Role** drop-down menu, select the role to configure: **Administrator**, **Operator**, or **User**.



The screenshot shows the 'Anybus Communicator' web interface. The top header is dark blue with the 'Anybus' logo on the left and a status bar on the right showing 'Anybus Communicator' and a green 'Apply' button. Below the header is a sidebar with navigation links: Home, Configuration, Network (X2), Network (X3), I/O configuration, Maintenance, System, and Files & firmware. The main content area is titled 'Manage role(s)' and contains a 'Role' dropdown menu set to 'operator', an 'Enable role' toggle switch, and two password input fields labeled 'New password*' and 'Confirm new password*'. A 'Change password' button is at the bottom.

Figure 28. Manage role(s) page

- Enter the password in the **New password** field.
 - Re-enter the password in the **Confirm password** field.
 - Click **Change password** to confirm.
- The settings are applied. If the password is changed for the currently logged-in role, you are automatically logged out.

Result

Roles are enabled, and the factory default mode that does not require user credentials is turned off.

You will now be prompted to enter a username and password to access the Communicator web interface.

8.4. Set or Change Password

Procedure

Change the Communicator web interface login password:

1. In the web interface header, click  **Account** > **Manage role(s)**.

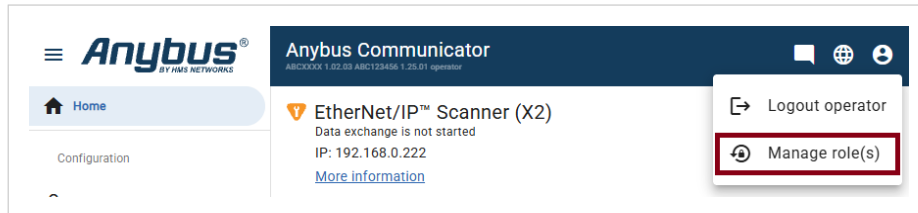


Figure 29. Account menu, **Manage role(s)**

2. From the **Role** drop-down menu, select the role for password update.

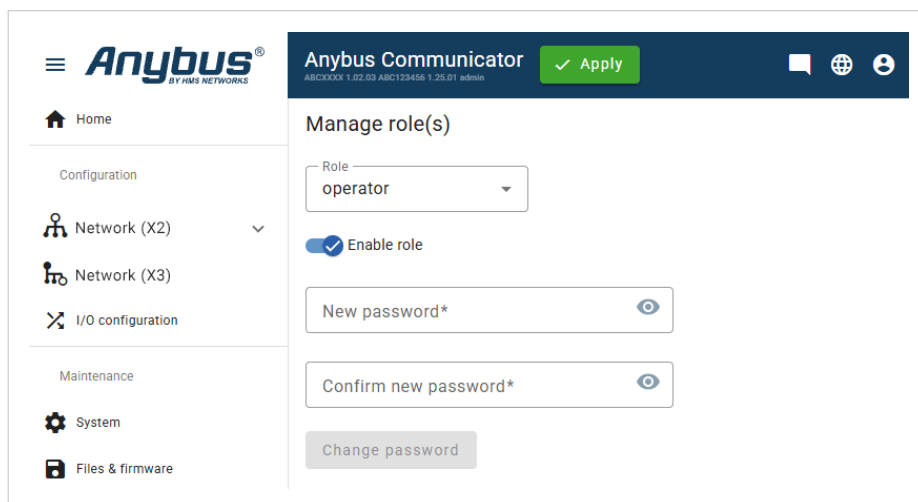


Figure 30. Manage roles(s) page

3. Enter the new password in the **New password** field.
4. Re-enter the new password in the **Confirm password** field.
5. Click **Change password** to confirm.
The password is applied and you are automatically logged out.

8.5. Logout From the Communicator Web Interface

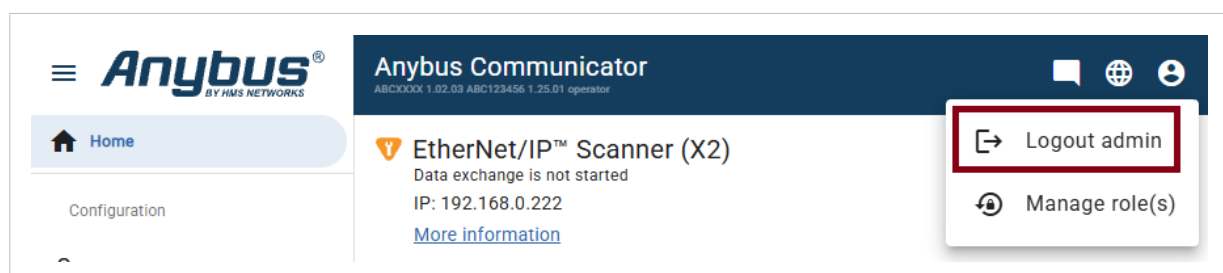


Figure 31. Account menu, Logout

Click  **Account** in the web interface header, then select **Logout** from the menu.

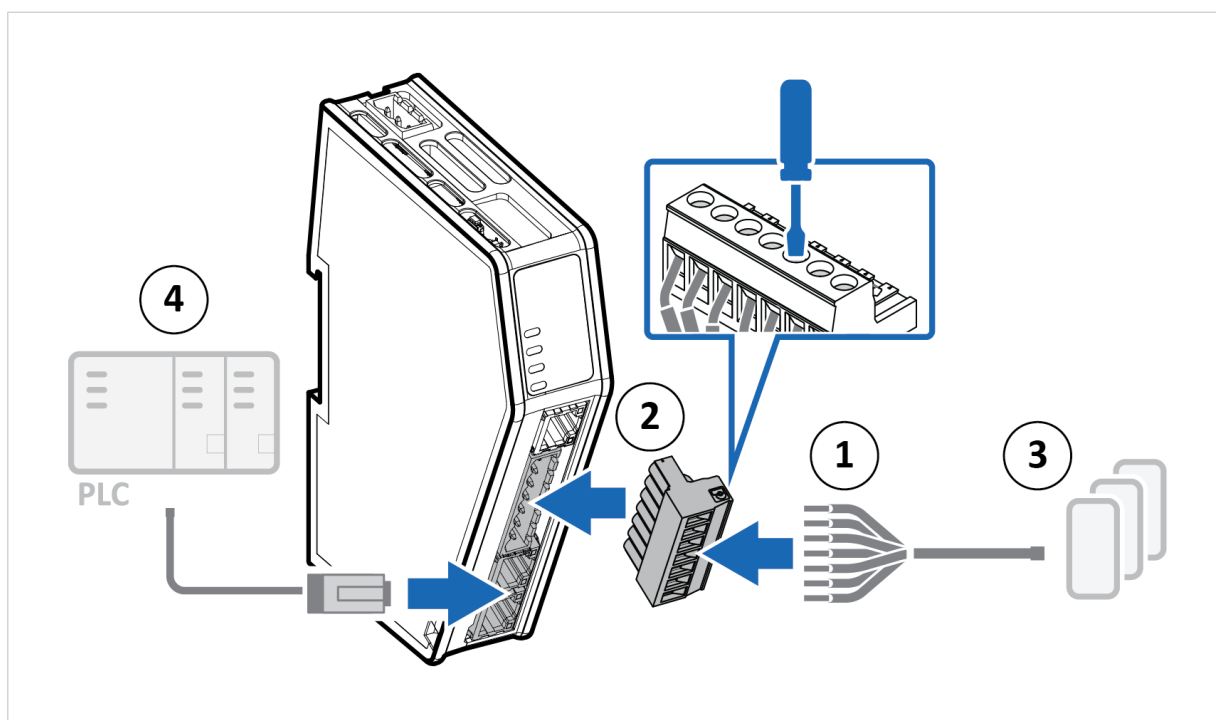
9. Configuration Quick Guide

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

For detailed information, please refer to [Communicator Configuration \(page 50\)](#).

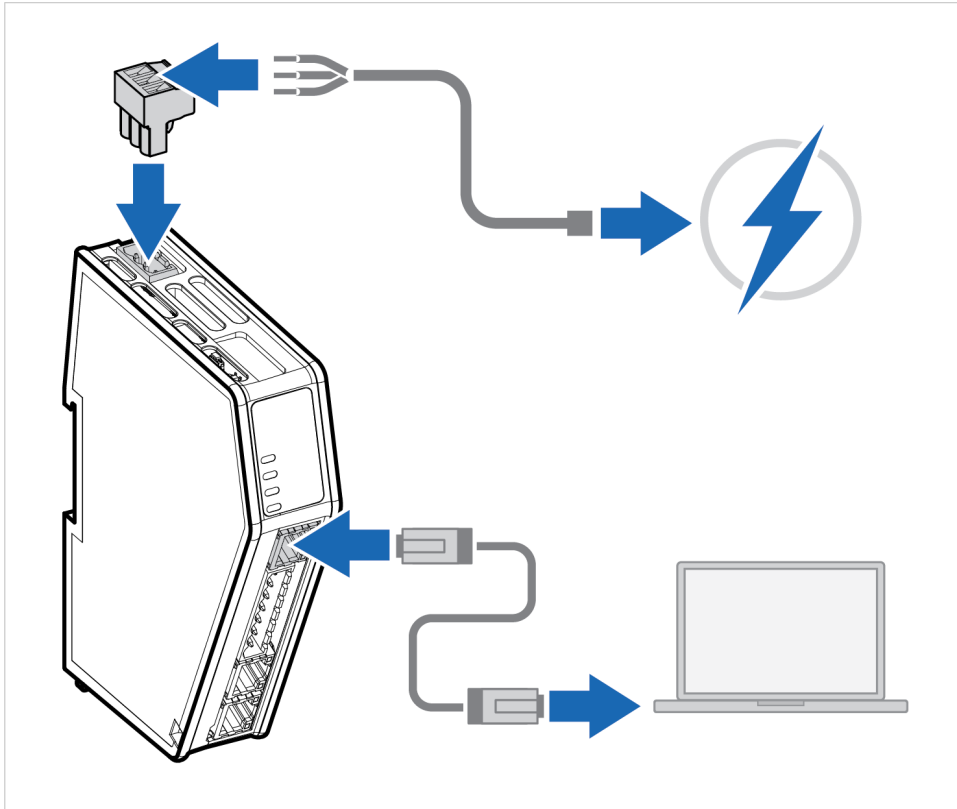
9.1. Prepare Configuration

Connect to Networks



1. Insert the cable wires into the 7-pin connector and tighten the wire clamp screws (1). See [Modbus RTU Connector Pinout \(page 20\)](#).
2. Connect the 7-pin connector to the Communicator (2).
3. Connect the Communicator to the RS-232/485 Serial network (3).
4. Connect the Communicator to the Modbus TCP network (4).

Connect to PC and Power



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

Access the Built-In Web Interface

1. Find the Communicator on your PC

**NOTE**

The Communicator default IP address is 192.168.0.10.

Option 1 | Static IP Address



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.

Option 2 | Configuration Port IP Address

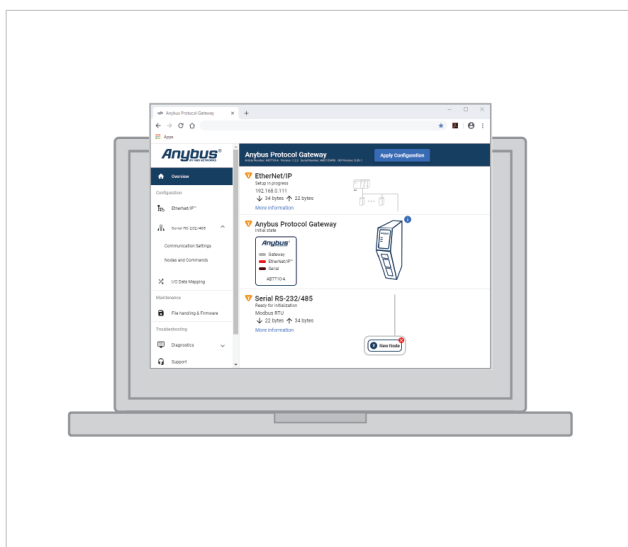


Change the IP address on the Communicator configuration port to one within the same IP address range as your PC.

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.hms-networks.com/technical-support.

2. 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.

9.2. Security Setup

On the System page in the **Security** panel:

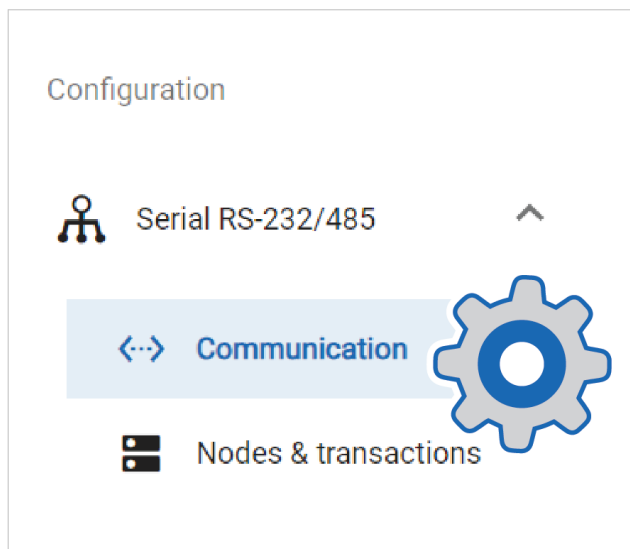
Before you begin configuring the Communicator, review these key **Security** settings:

- User roles: Enable login and assign access levels.
- Secure Connection (HTTPS): Turn on encrypted access and upload the private key and certificate.
- Time settings: Set the correct local time and enable time synchronization to ensure a valid certificate and accurate event timing.

9.3. Setup New Configuration

Follow these steps to setup a new Communicator configuration.

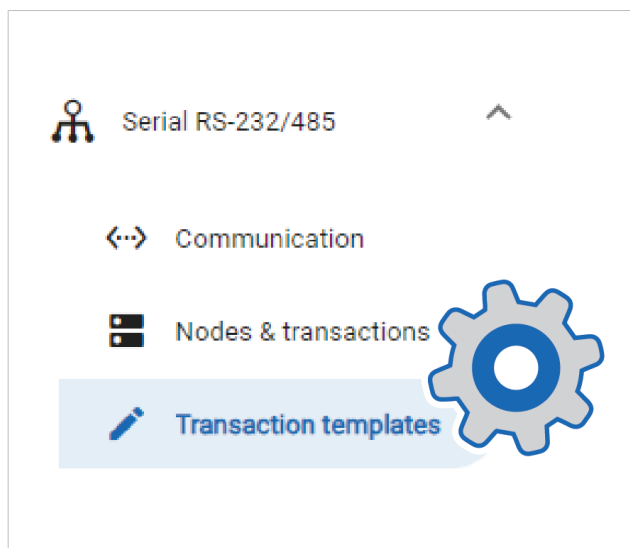
1. Subnetwork configuration



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 \(page 10\)](#).
- b. Configure the basic settings Physical standard:, Baud rate, Data bits, Parity and Stop bits.

2. Create Transaction Templates

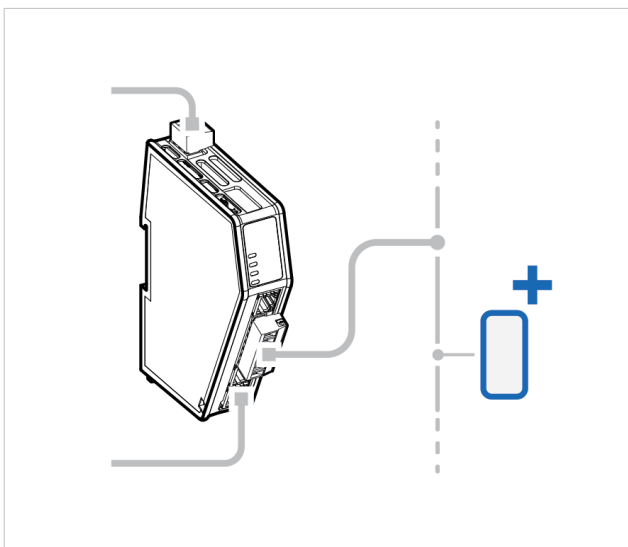


This step applies when the serial protocol **Custom request/response** is selected.

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.

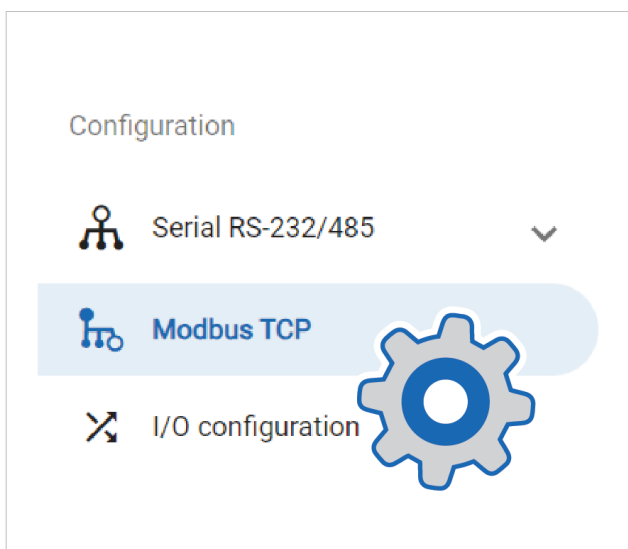
3. Add Nodes and Transactions



On the **Nodes & transactions** 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.

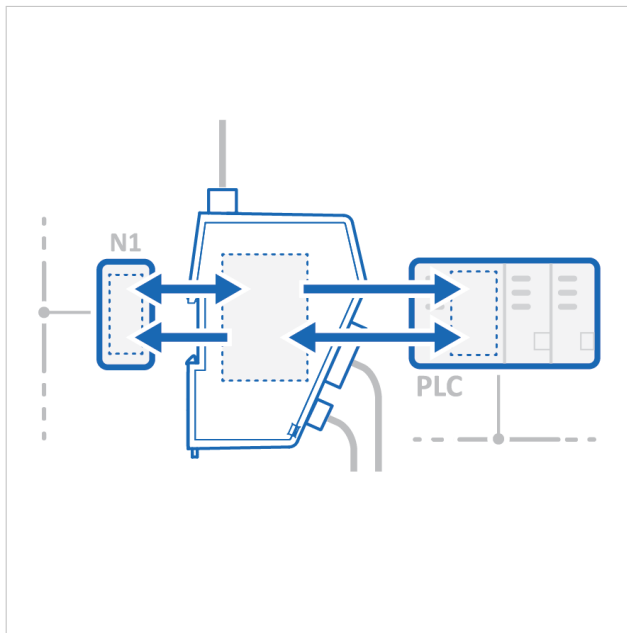
4. High level network configuration



On the **Modbus TCP** page:

- a. Use Automatic I/O sizes provided by the subnetwork or choose to set them manually.
- b. Enable DHCP server or choose to set a specific IP address.

5. I/O Data Mapping



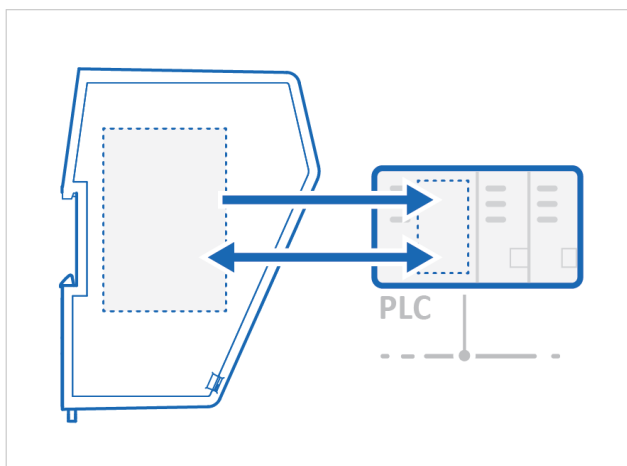
The commands you added to the nodes are automatically mapped to the Communicator internal memory area.

View the added nodes and commands on the **I/O configuration** page.

9.4. PLC Configuration

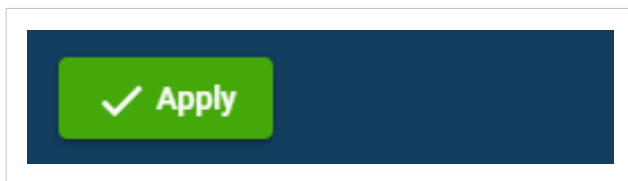
In the PLC Program

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



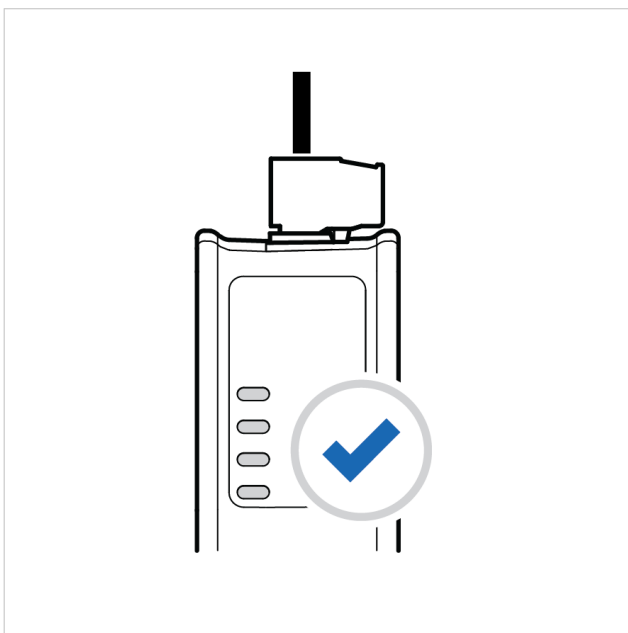
9.5. Verify Operation

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.
- **I/O data** page to monitor how the data flow between the **Serial RS-232/485** side and the **Modbus TCP** side, including any configured endian conversions.
- **Event log** page to detect failures and unexpected behavior over time.

10. 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 \(page 41\)](#).

10.1. General Subnetwork Settings

10.1.1. 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:

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

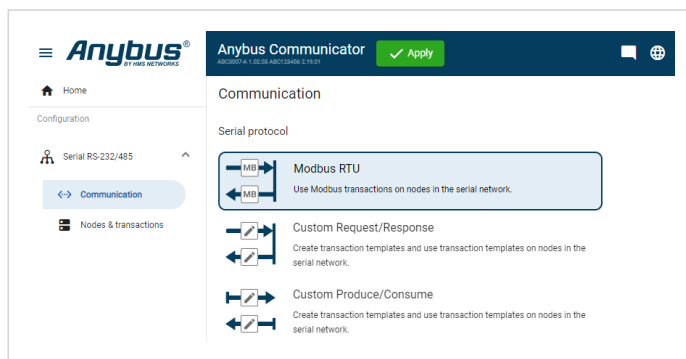


Figure 32. Communication, Serial protocol

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

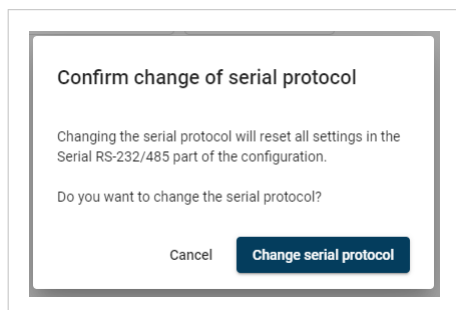


Figure 33. Confirm change of serial protocol

Apply Configuration

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

10.1.2. Communication Basic Settings

Anybus Communicator

Article Number: AB7710-A Version: 1.2.3 Serial Number: ABC123456 GUI Version: 0.44.1

✓ Apply

Communication

Basic settings

Physical standard

RS232

Baud rate

19200 baud

Data bits

8 data bits

Parity

None

Stop bits

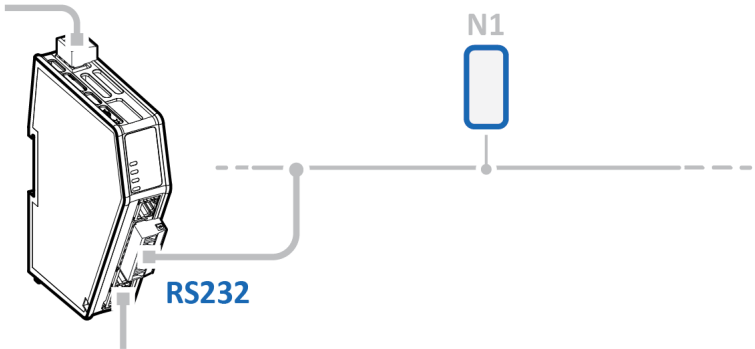
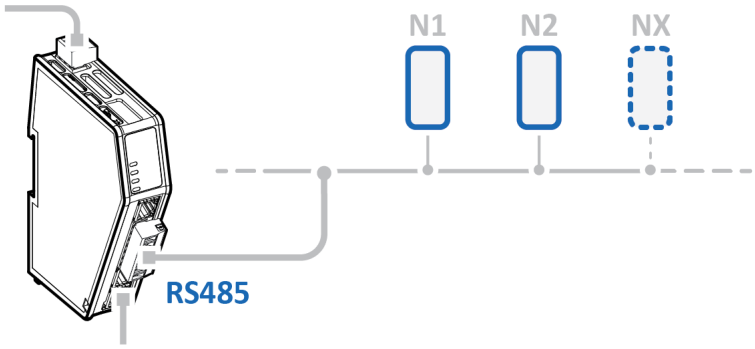
1 stop bit

Figure 34. 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.

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

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.

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

10.1.3. Communication Advanced Settings

Pre Extension and Post Extension for RS-485 Communication

Option for Basic settings Physical standard RS-485.

About Pre Extension and Post Extension

Pre Extension

A time extension is added before the actual data transfer begins.

For example, this allows the signal to stabilize before data is sent, reducing the risk of errors and ensuring that all devices on the network are synchronized and ready to receive data.

Post Extension

A time extension is added after the data transfer is complete.

For example, this ensures that the transmission line is clear of any residual signals before the next data packet is transmitted. By keeping the line free before another device starts transmitting, data collisions are also prevented.

Procedure

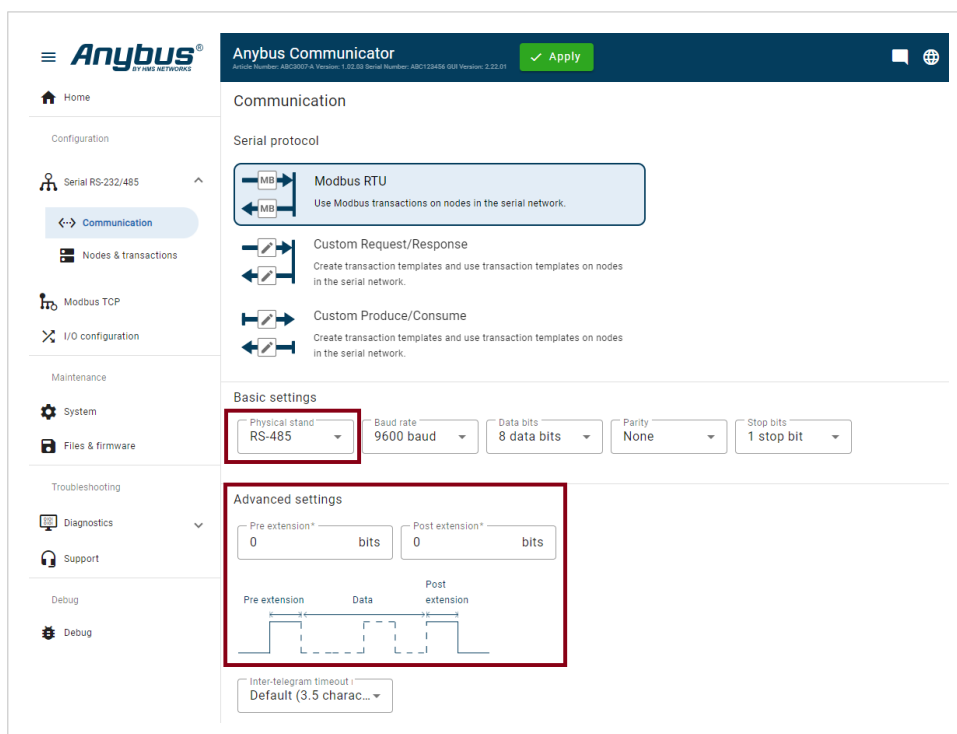


Figure 35. Communication page, Physical standard RS-485 Advanced settings



NOTE

Pre extension and post extension do not overlap with or occur during inter-telegram timeout.

Ensure that the **Physical standard RS-485** is selected.

By default, **Pre extension** and **Post extension** is set to 0 bits.

Enter **Pre extension** and **Post extension** values between 0 and 1000 bits.

Inter-Telegram Timeout Mode Settings

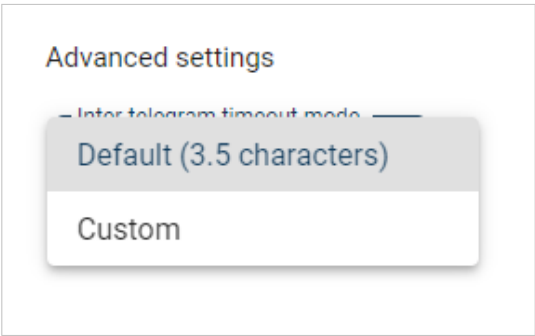


Figure 36. Advansed settings, Default (3.5 characters)

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.

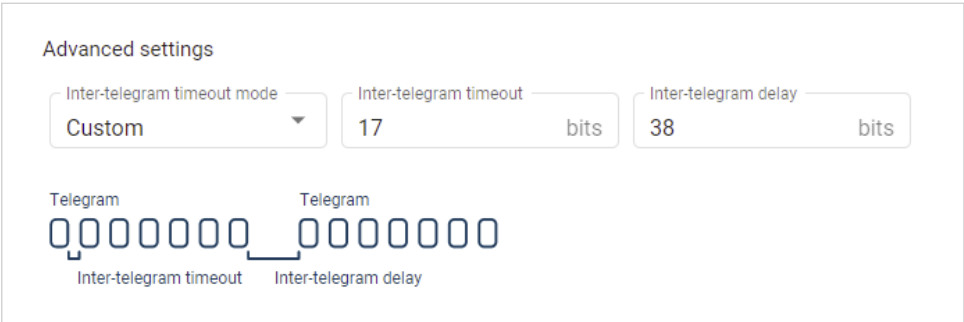


Figure 37. Custom settings, Inter-telegram timeout and Inter-telegram delay

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.

About Inter-Telegram Start and End Character

This topic describes scenarios for using Start character and End character.

Example 2. Both Start Character and End Character Disabled

Default setting, no **Start character** or **End character** is used.

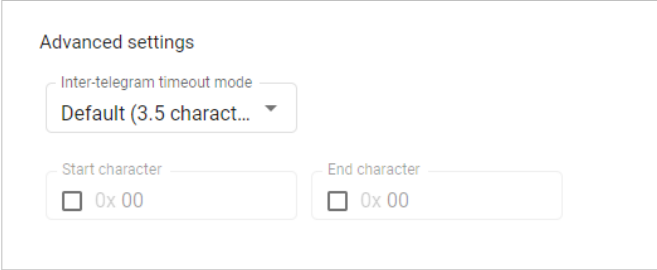


Figure 38. Start character and End character are Disabled

The size of the telegram is defined between two inter-telegram delays.

Standard **Inter-telegram delay**: 3.5 characters

Inter-telegram delay is set in the Advanced settings, see [Inter-Telegram Timeout Mode Settings \(page 55\)](#).

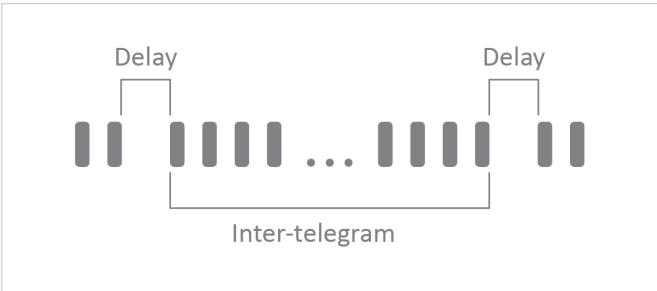


Figure 39. Data stream without Start and End characters

Example 3. Start Character Enabled and End Character Disabled

Start character is used.

Advanced settings

Inter-telegram timeout mode

Default (3.5 charact...

Start character *

☒ 0x 01

End character

☐ 0x 00

Figure 40. Example: Start character Enabled, Hex value 0x01

First, a start character is identified in the data stream, then the inter-telegram follows. The inter-telegram ends when an inter-telegram delay is identified in the data stream.

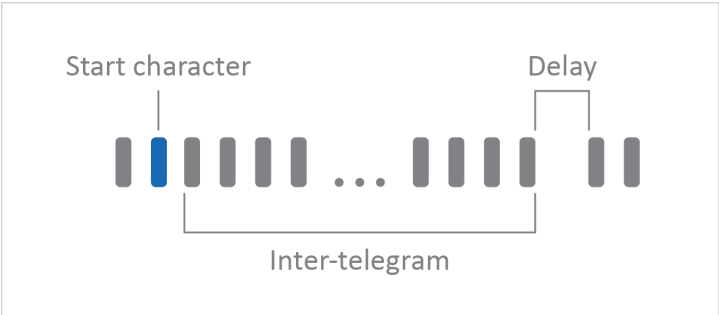


Figure 41. Data stream with Start character

Example 4. Start Character Disabled and End Character Enabled
End character is used.

Advanced settings

Inter-telegram timeout mode

Default (3.5 charact...

Start character

☐ 0x 00

End character *

☒ 0x 04

Figure 42. Example: End character Enabled, Hex value 0x04

First, an inter-telegram delay is identified in the data stream, then the inter-telegram follows. The inter-telegram ends when an end character is identified in the data stream.

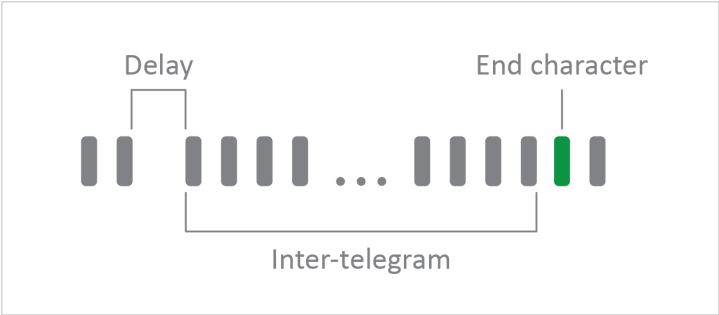


Figure 43. Data stream with End character

Example 5. Bothe Start Character and End Character Enabled

Both **Start character** and **End character** is used.

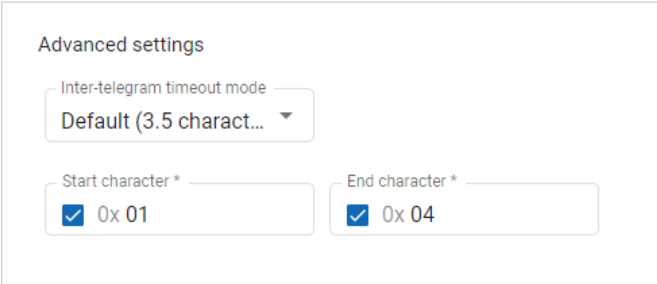


Figure 44. Example: Start character Enabled, Hex value 0x01 and End character Enabled, Hex value 0x04

First, a start character is identified in the data stream, then the inter-telegram follows. The inter-telegram ends when an end character is identified in the data stream.

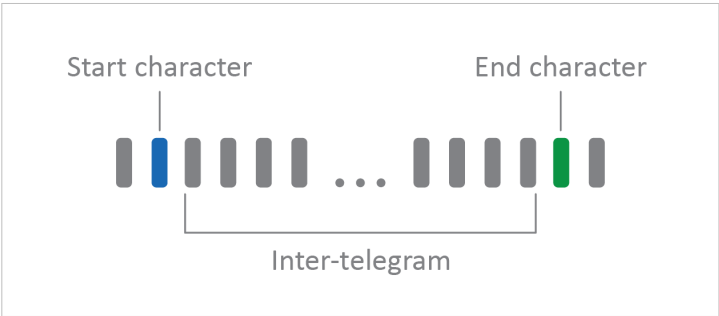


Figure 45. Data stream with Start character and End character

Inter-Telegram Start and End Character

Start character and **End character** is only valid for the **Custom Produce/Consume Serial protocol** mode; **Produce** or **Consume** transactions, see [Communication Serial Protocol \(page 51\)](#).



NOTE

Inter-telegram timeout mode Custom settings for **Inter-character timeout** and **Inter-telegram delay** still apply if **Start character** and/or **End character** is enabled.



NOTE

If a inter-telegram exceeds the maximum allowed transaction frame size of 1500 bytes, the telegram is invalid and discarded.

Start and End character Use Case Example

In serial channel data stream, there is a **Consume** transaction for which you want to obtain the content between two defined characters as an Inter-telegram.

Procedure

Advanced settings

Inter-telegram timeout mode
Default (3.5 charact... ▼

Start character
☐ 0x 00

End character
☐ 0x 00

Figure 46. Start and End character settings

1. Navigate to the **Serial RS-232/485 Communication** page.
2. Select the **Custom Produce/Consume Serial protocol** mode.
3. To enable **Start character** and/or **End character**, select the checkbox(es).
4. Enter the decried character hexadecimal value(s) for the Inter-telegram.

Result

For **Produce** transactions, a start and/or end character is added to the inter-telegram when it is sent.

10.2. About Transaction Templates

This section applies when the Custom Request/Response or Custom Produce/Consume serial protocol is applied, refer to [Communication Serial Protocol \(page 51\)](#).

10.2.1. Transaction Template Example

Custom Request/Response

Anybus Communicator
Article Number: ABC3007-A Version: 1.2.3 Serial Number: ABC123456 GUI Version: 1.2.1

✓ Apply

Transaction templates

+ Add

▼

Read parameter (0x01) ⋮

Transaction template settings

Transaction name
Read parameter (0x01)

Frame editor

⋮ A Node add...

⋮ C Constant

⋮ L Limit

⋮ D Data

⋮ V Variable d...

⋮ Cs Checksum

➡

Request

Byte offset	Field	⋮
0	⋮ C Function code 1 ⋮	
1	⋮ A Node ID ⋮	
2 ... 3	⋮ C Index 2 ⋮	
4	⋮ C Sub index 1 ⋮	
5 ... 6	⋮ Cs Checksum ⋮	

⬅

Response

Byte offset	Field	⋮
0	⋮ C Function code 1 ⋮	
1	⋮ A Node ID ⋮	
2 ... 3	⋮ C Index 2 ⋮	
4	⋮ C Sub index 1 ⋮	
5 ... 6	⋮ D Data ⋮	
7 ... 8	⋮ Cs Checksum ⋮	

Figure 47. 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

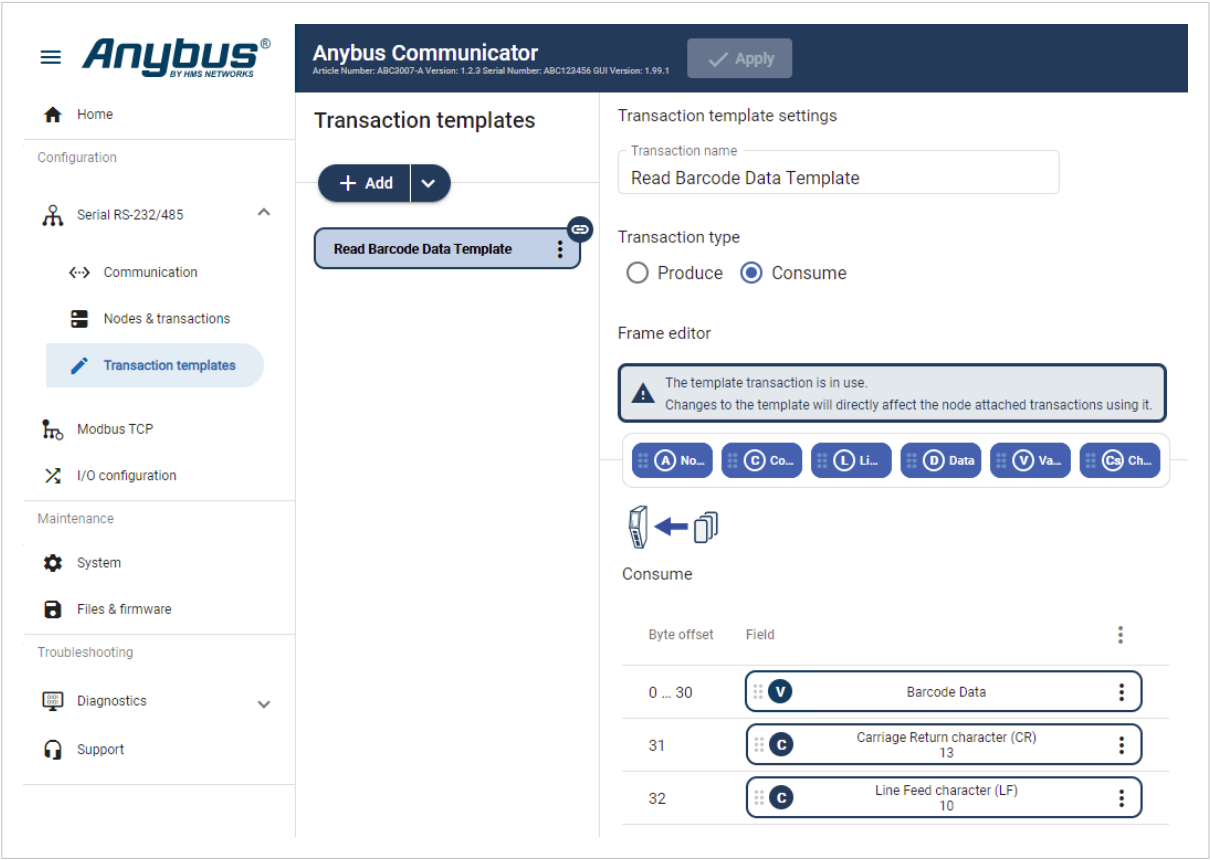


Figure 48. Produce transaction template example

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.

10.2.2. 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.

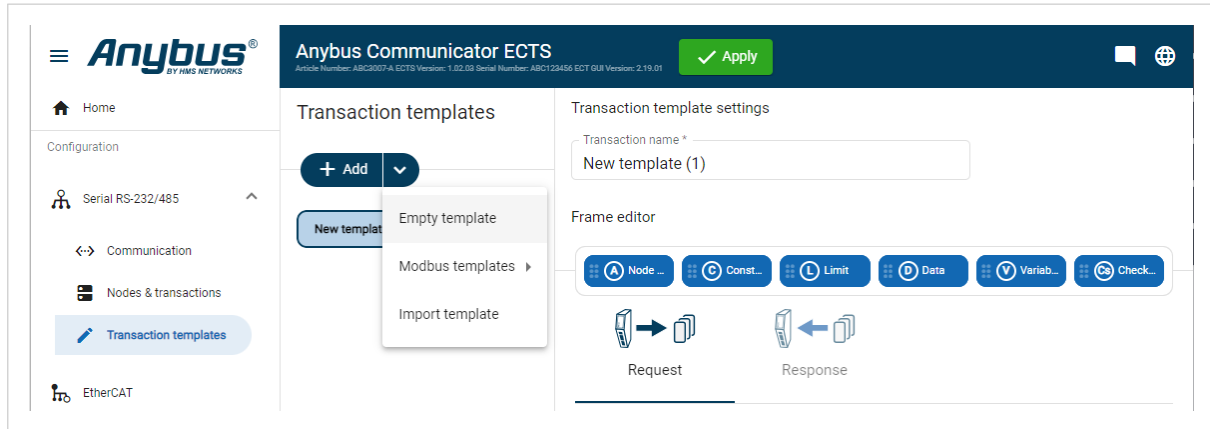


Figure 49. Transaction template, Empty template

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

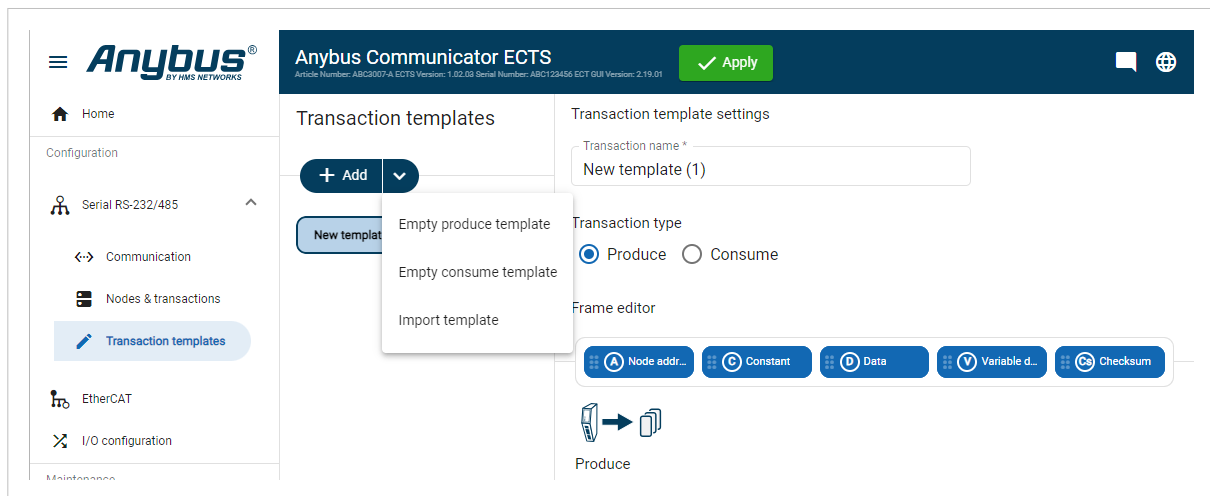


Figure 50. Transaction template, Empty Produce and Consume templates

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.

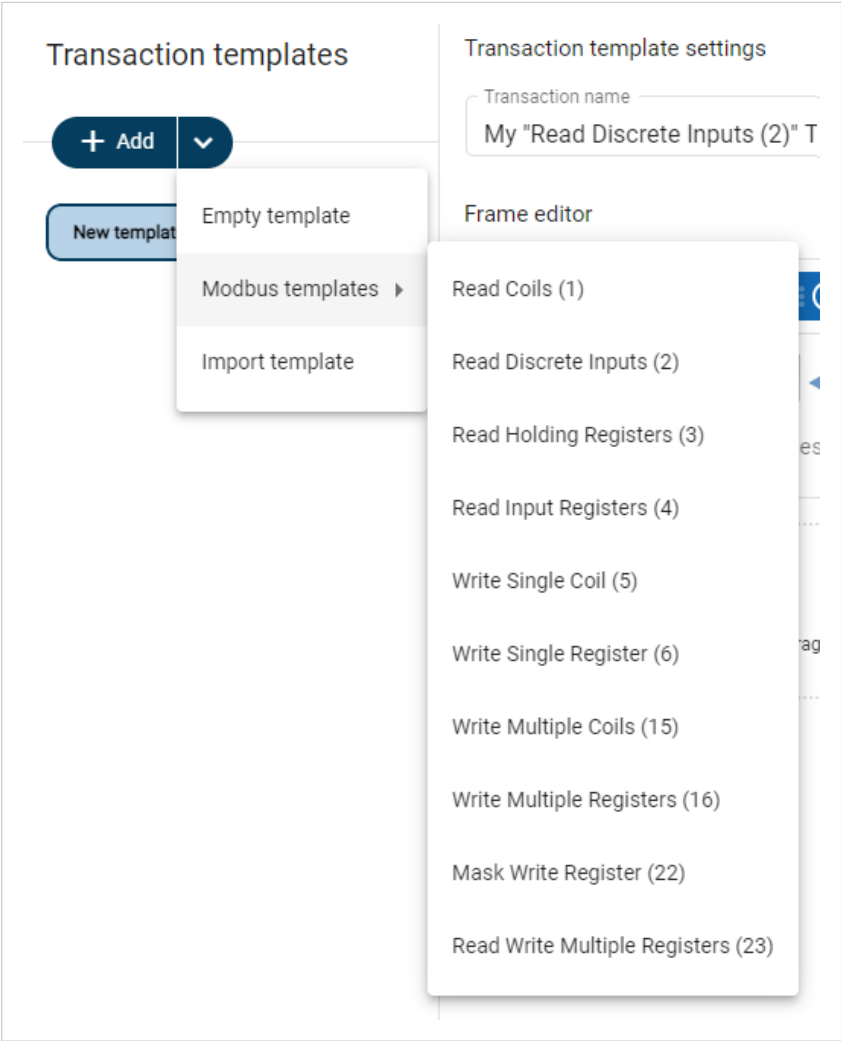
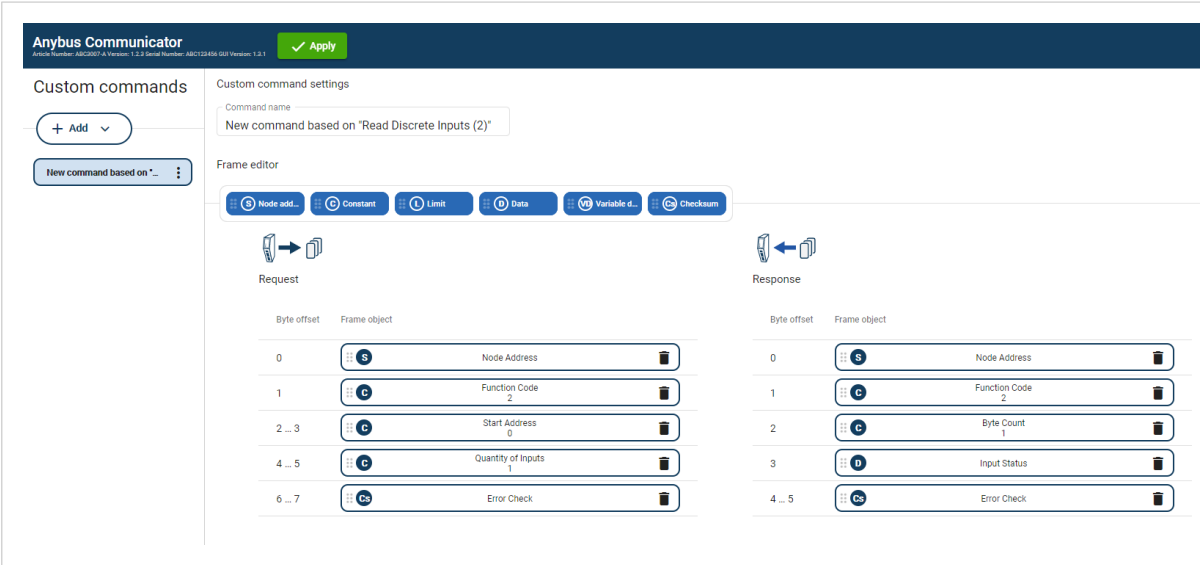


Figure 51. Transaction template, Modbus template

Example 6. New transaction template based on the Modbus template Read Discrete Inputs



10.2.3. 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.

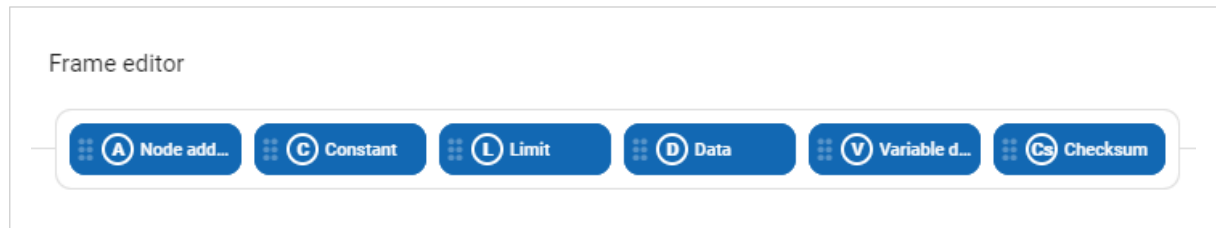


Figure 52. The following frame objects are available

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.

10.3. Build Transaction Template

Before You Begin

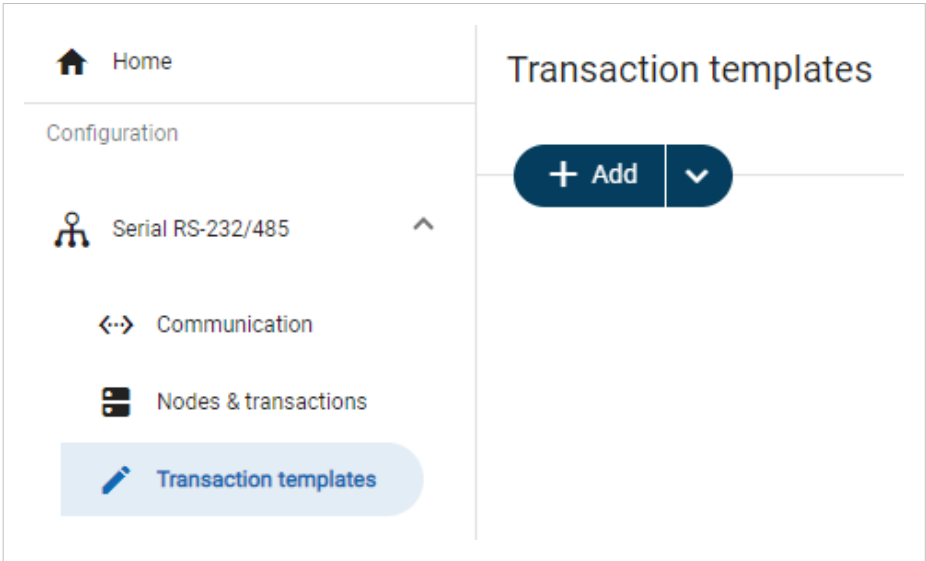
Ensure that you have applied the Custom Request/Response or Custom Produce/Consume serial protocol, refer to [Communication Serial Protocol \(page 51\)](#).

10.3.1. Add Transaction Template

Procedure

Add a transaction template:

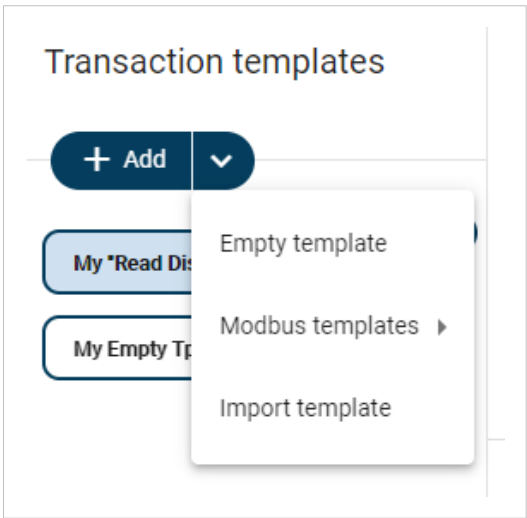
1. In the web-interface left sidebar menu, click **Transaction templates**.



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

Anybus Communicator

Article Number: ABC3007-A Version: 1.2.2 Serial Number: ABC123456 GUI Version: 1.2.1

✓ Apply

Transaction templates

+ Add

▼

New template (1)

⋮

Transaction template settings

Transaction name

New template (1)

Frame editor

A Node add...

C Constant

L Limit

D Data

V Variable ...

CS Checksum

Request

Drop fields here

Drag a field from the toolbar above.

Response

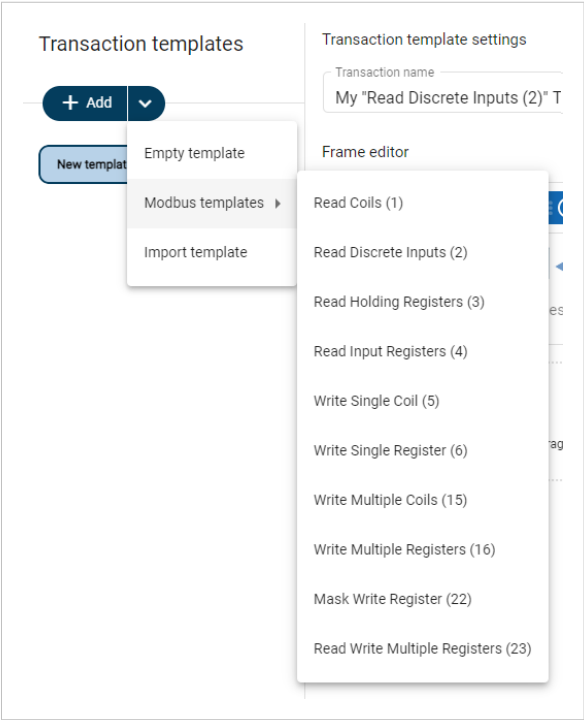
Drop fields here

Drag a field from the toolbar above.

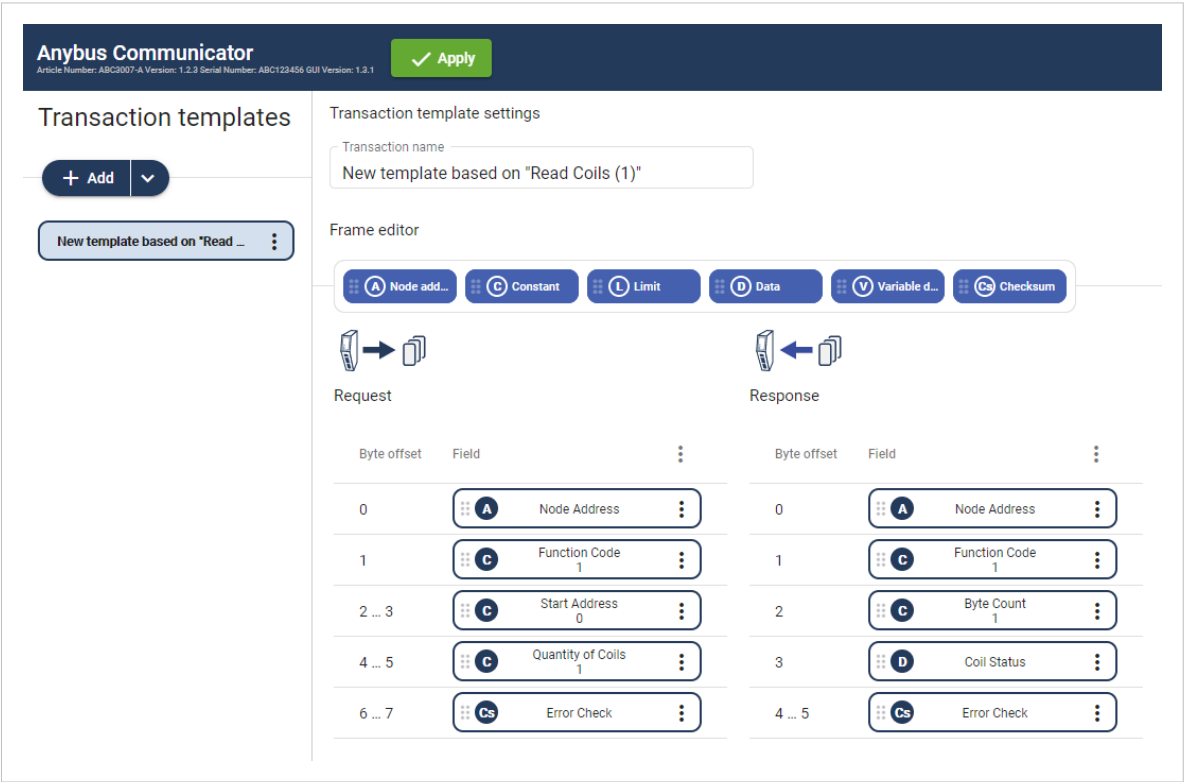
Page 68 of 185

SCM-1202-162 Version 1.3

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

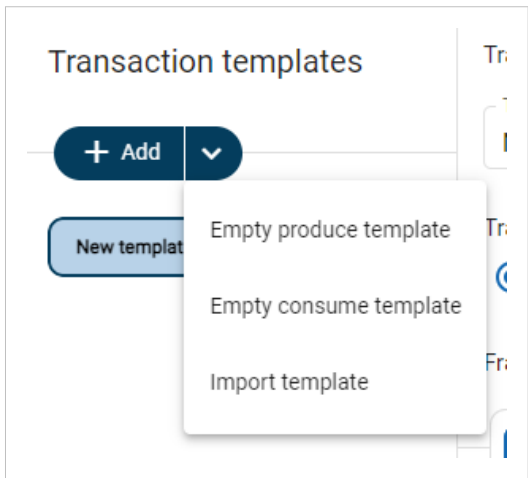


Example 8. A new request/response template based on “Read Coils (1)” is added to the transaction template list

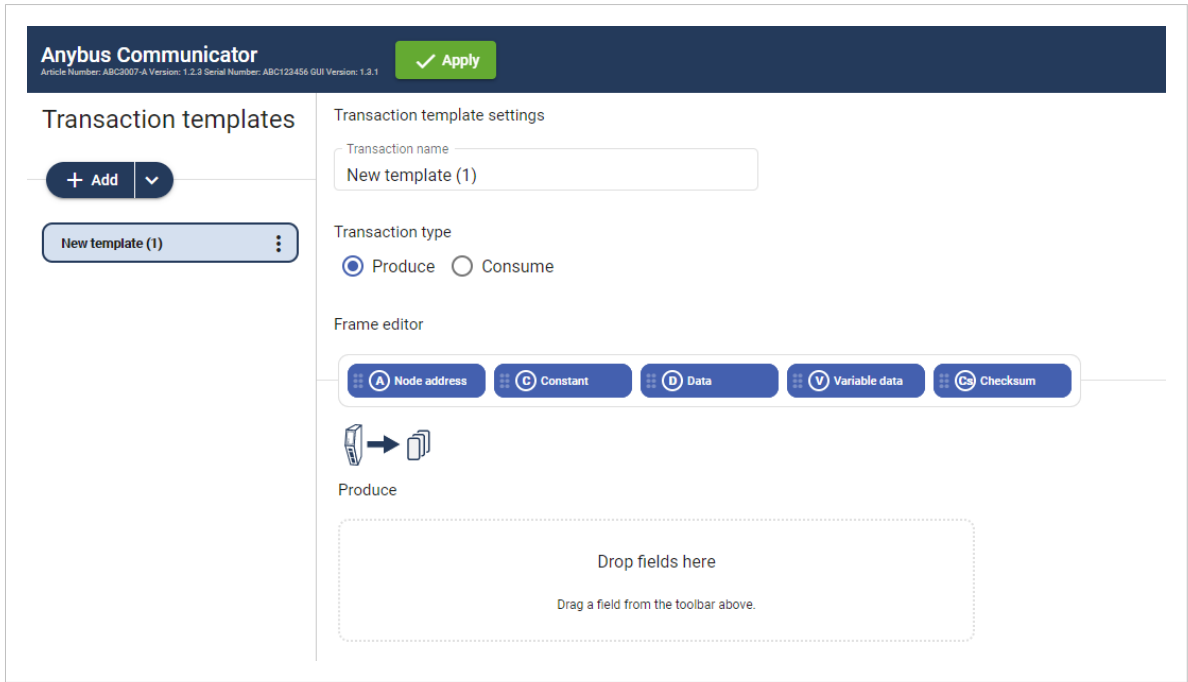


Option for the Custom Produce/Consume Protocol

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



Example 9. 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 \(page 71\)](#).

10.3.2. Add Frame Fields

Procedure

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

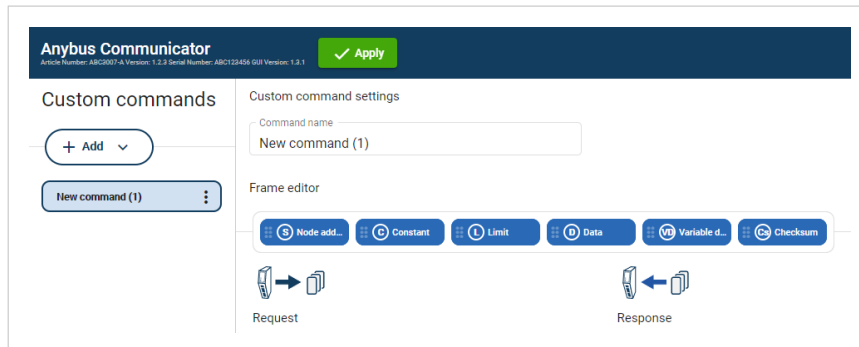


Figure 53. Select transaction template

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.

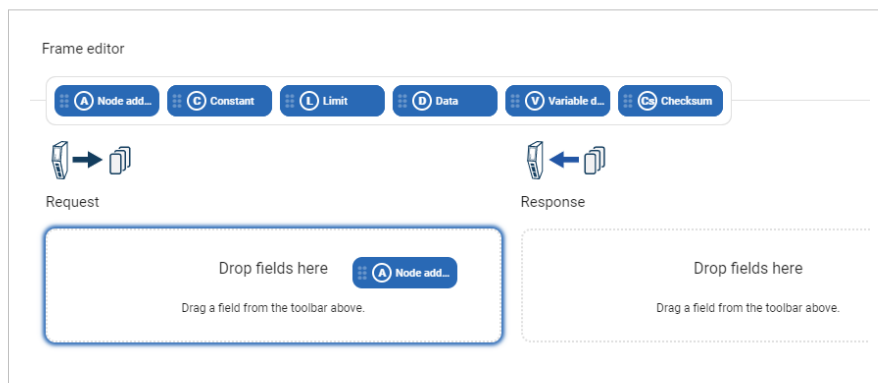


Figure 54. Add frame fields

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

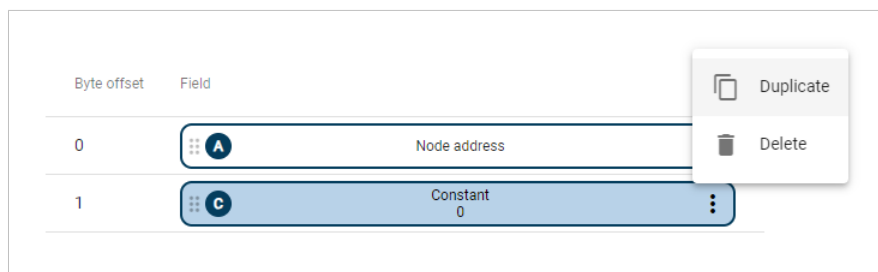


Figure 55. Duplicate frame fields

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

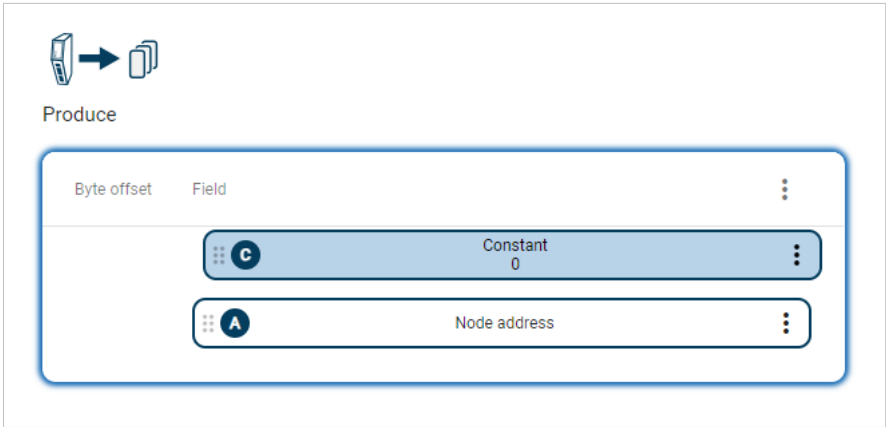


Figure 56. Change frame fields order

- 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.

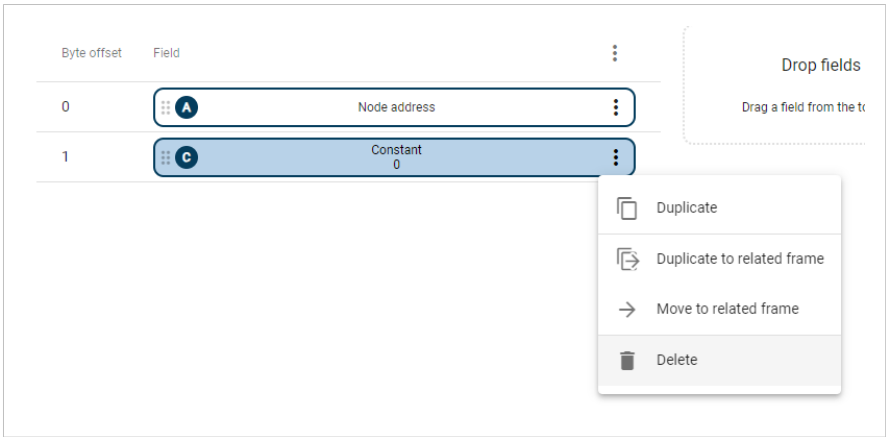


Figure 57. Delete frame field

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 Field Settings \(page 73\)](#).

10.3.3. Configure Frame Field Settings

Procedure

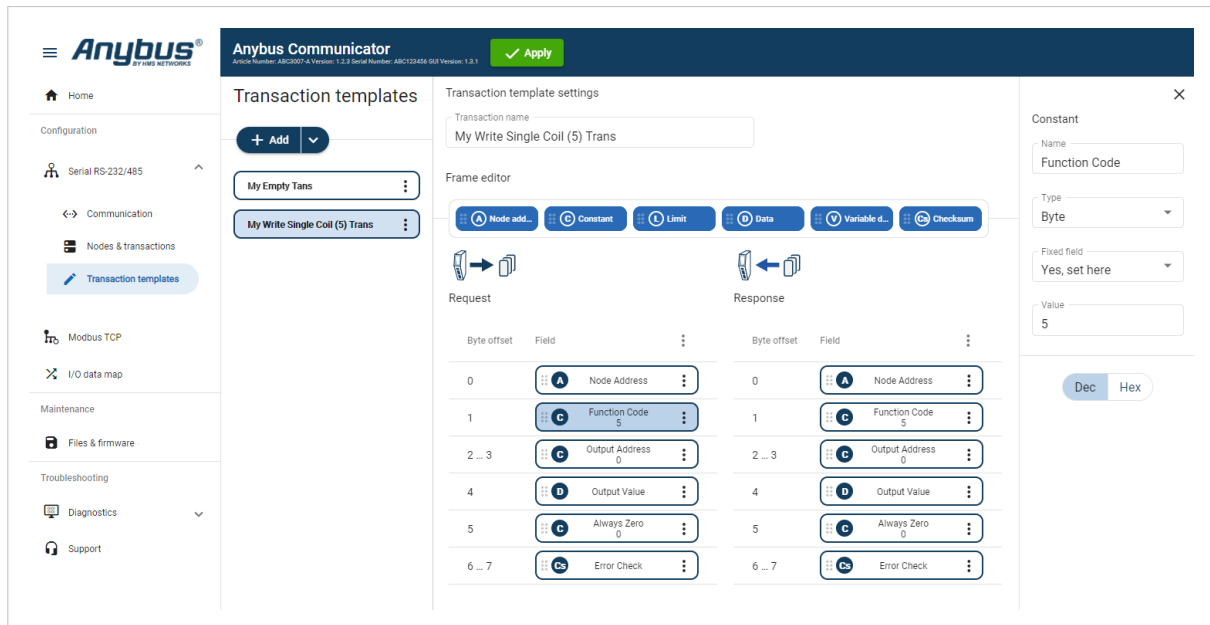


Figure 58. Frame Fields Settings

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.
- **Endianness:** 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).
- **Endianness:** 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 \(page 76\)](#).
- **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 \(page 76\)](#).
- **End pattern:** When a **Data delimiter** or **Subnet delimiter** is set to End pattern, specify the value defining the end of the payload.
- **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.
- **Destination Representation:**
Each byte represents two ASCII characters.
Allowed characters: Digits 0-9 and Letters a-f
The first two characters represent the 4 most significant bits of the byte. The second character represents the 4 least significant bits.
Specify how the destination checksum is represented:
 - Binary (Default): Data is transmitted and received as-is, no pre- or post-processing is performed.
 - ASCII (Lower case): Received telegrams are case insensitive and sent telegrams are lowercase.
 - ASCII (Upper case): Received telegrams are case insensitive and sent telegrams are uppercase.

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 built-in web interface header and follow the instructions.

10.3.4. 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.

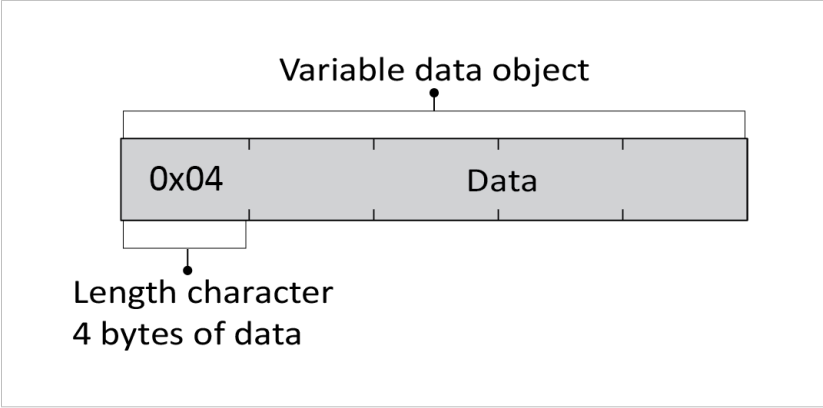
Transaction Template Variable Data Settings

Figure 59. Transaction template variable data settings

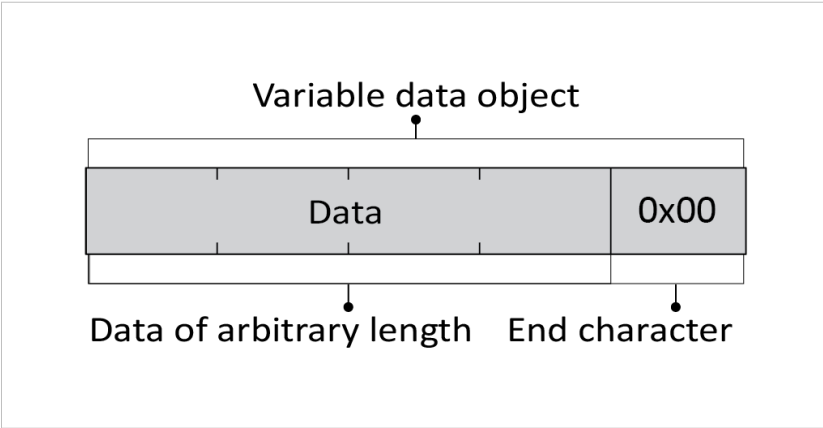
1. Select a desired **Variable data** object.
2. In the **Data delimiter** and/or **Subnet delimiter** drop down menu, select one of the following options.

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.

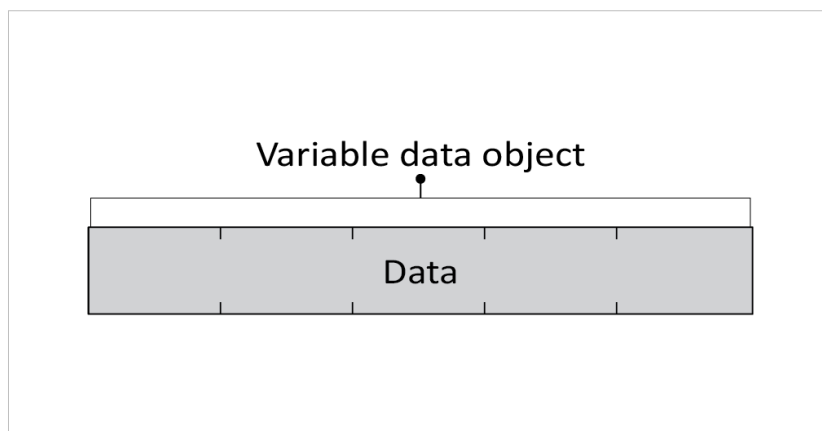


- End pattern
The package consists of a data section followed by an end character to indicate where the data section ends. End pattern is used to define whether the delimiter is an end character or an end pattern, which depends on whether the message is forwarded from the subnetwork or sent as process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



- None (Default)

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.

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.

10.3.5. Store Transaction Templates

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

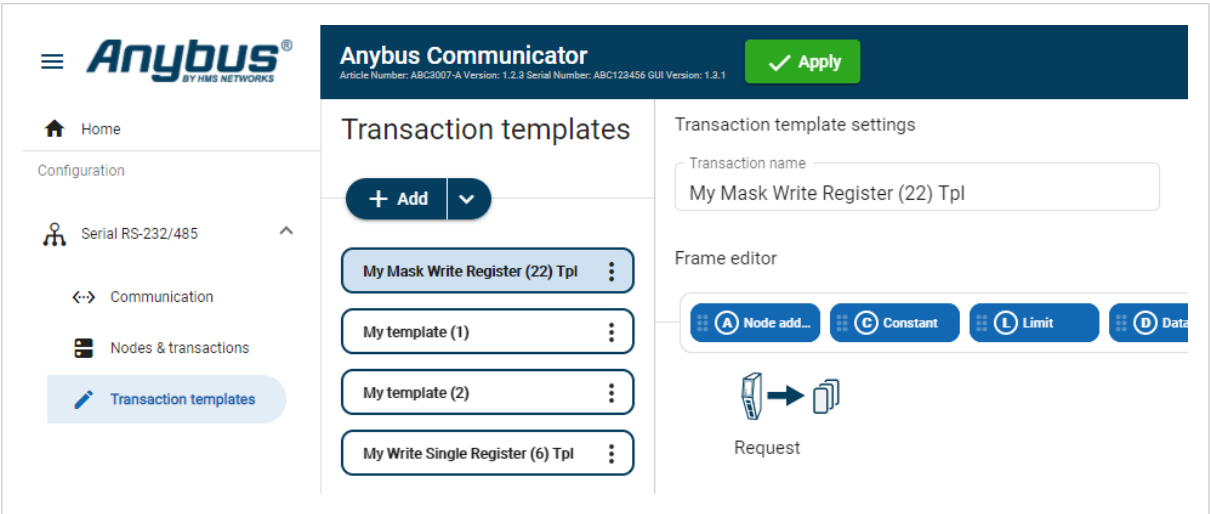


Figure 60. Transaction templates page

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

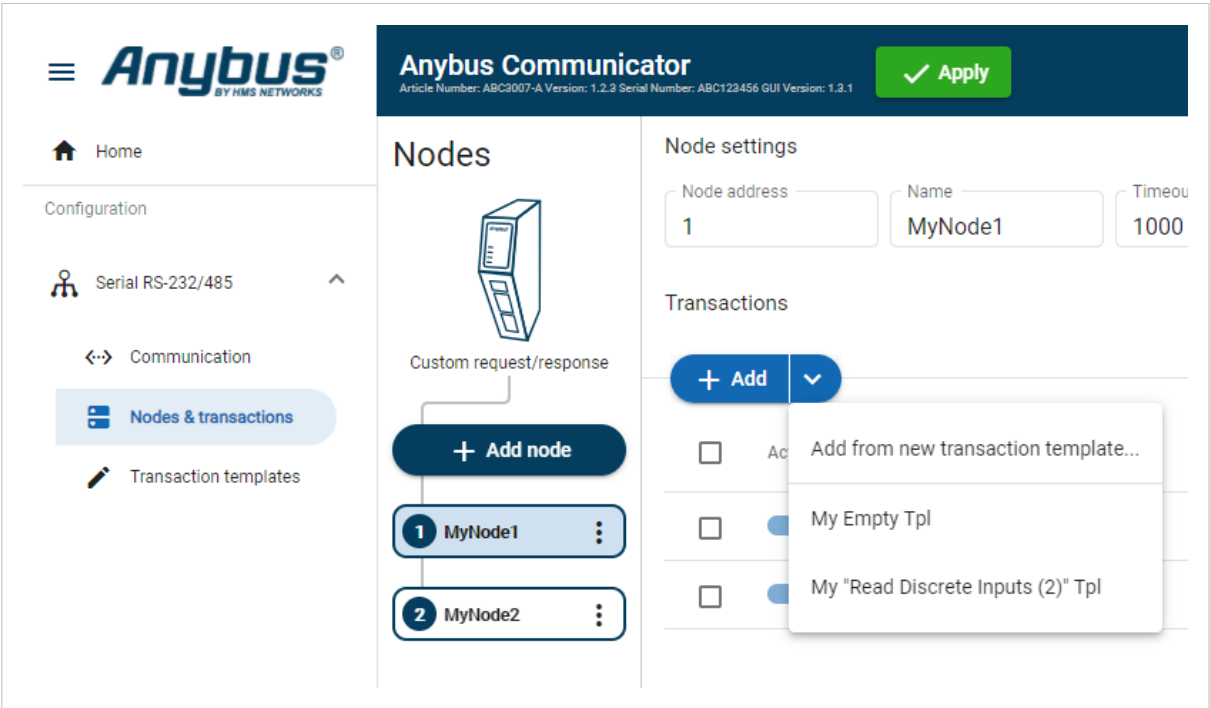


Figure 61. Nodes & transaction page

For information on how to add the transaction templates to the nodes, refer to [Transaction Settings \(page 93\)](#).

10.4. 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.

10.4.1. Node and Broadcast Node

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

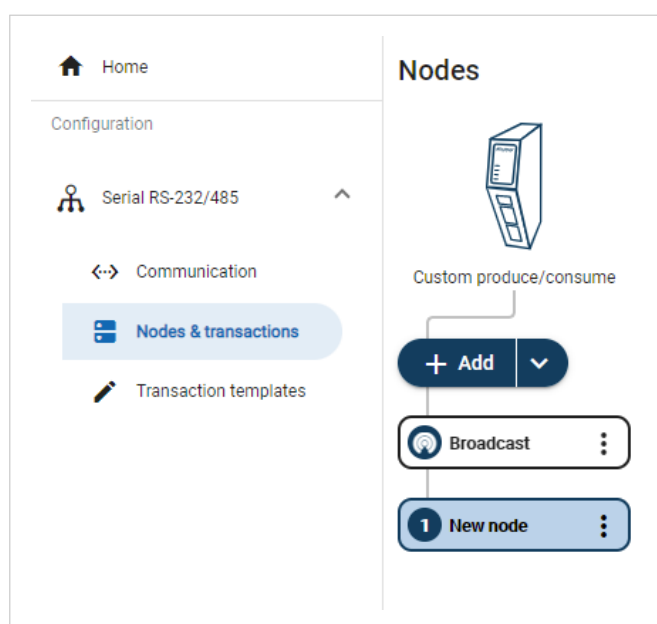


Figure 62. Add Node or Broadcast Node

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 \(page 61\)](#).

10.4.2. 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. In the **Add** split button drop-down menu, select **Add broadcast node** or **Add node**.
To **Import node**, see [Import Node Settings From Other Communicator Unit \(page 82\)](#).

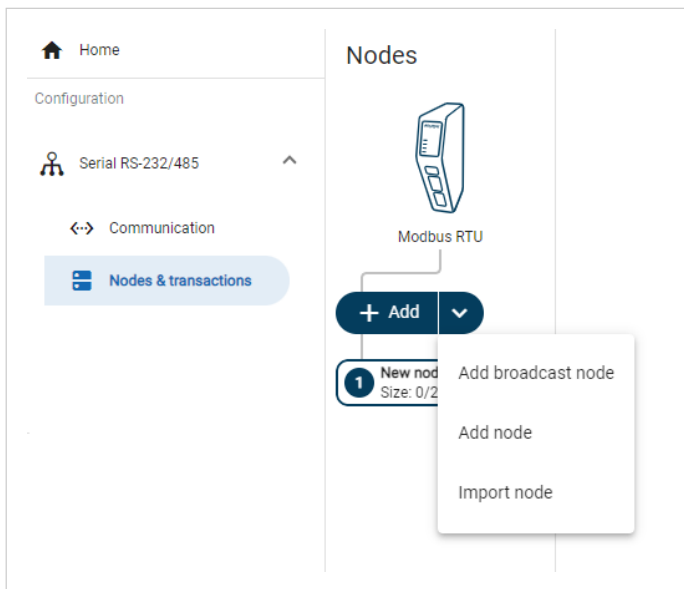


Figure 63. Add node

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

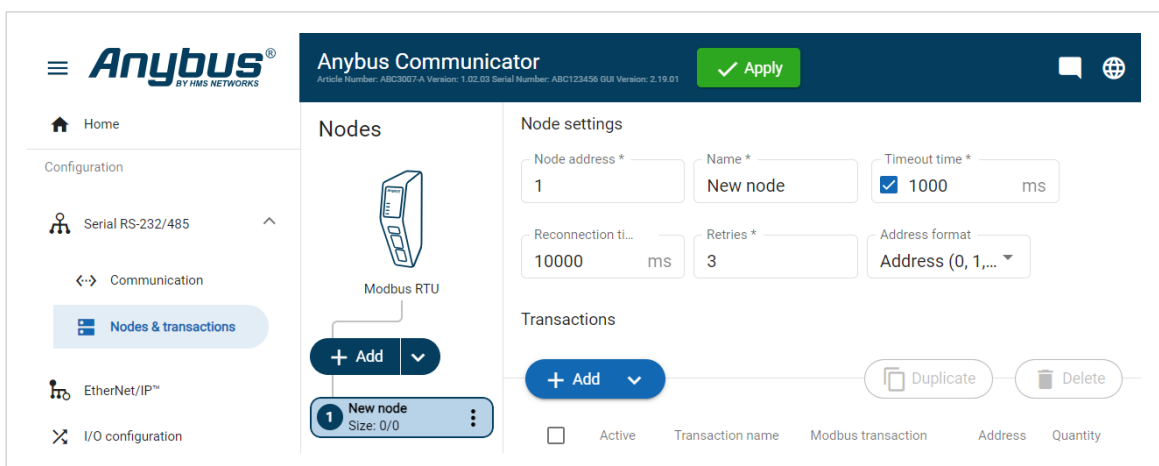


Figure 64. A new node is added to the node list

To Do Next

Configure the Node Settings, [Node Settings \(page 89\)](#).

10.4.3. Import Node Settings From Other Communicator Unit

When you have configured a node and want to use the same node for additional Communicator units, do the following.

Procedure

To Export the Node

In the built-in web-interface of the Communicator with the node you want to export:

1. In the web-interface left sidebar menu, click **Nodes & transactions**.
2. On the node to be exported, click the three dots icon and then click **Export node**.
A *.node* file including the node settings and the related transaction templates is downloaded to your computer.

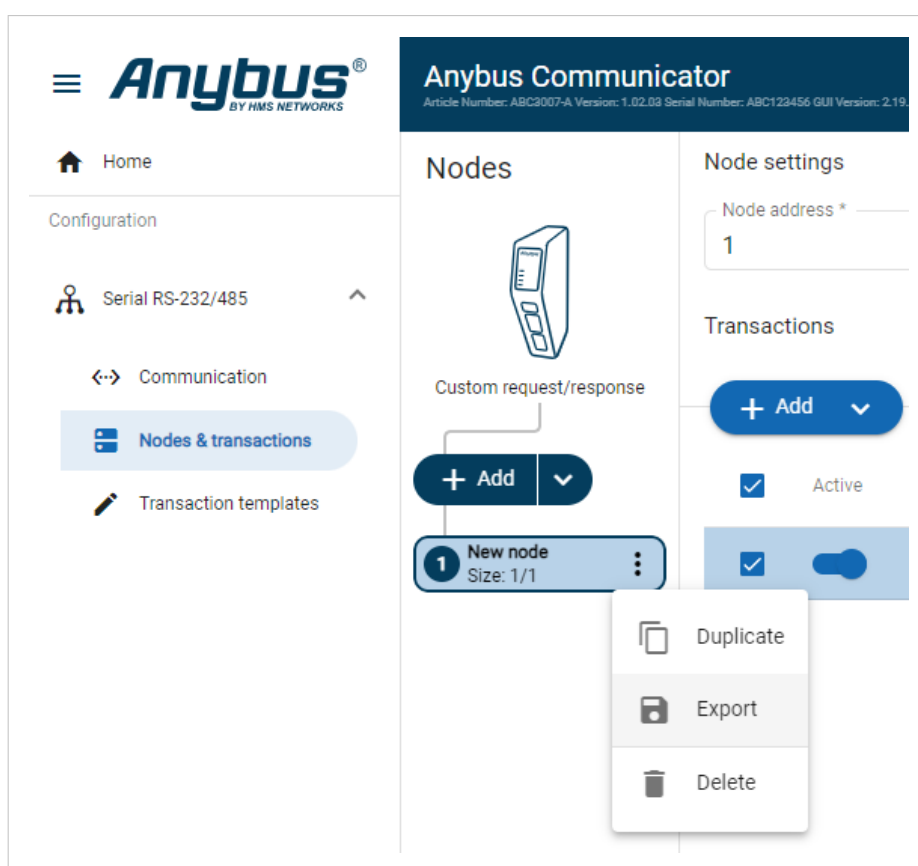


Figure 65. Export node

To Import the Node

In the built-in web-interface of the Communicator to which the node is to be imported:

3. In the web-interface left sidebar menu, click **Nodes & transactions**.

4. In the **Add** split button drop-down menu, select **Import node**.

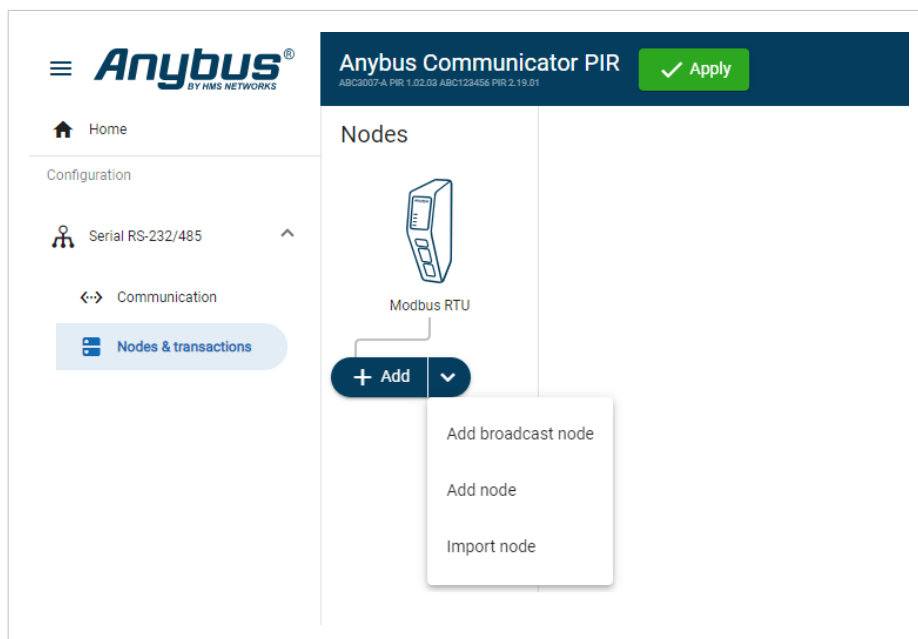


Figure 66. Import node

5. In the **Open** dialog box, browse to and select the *.node* file and click **Open**.
6. To import the *.node* file, click **Import**.

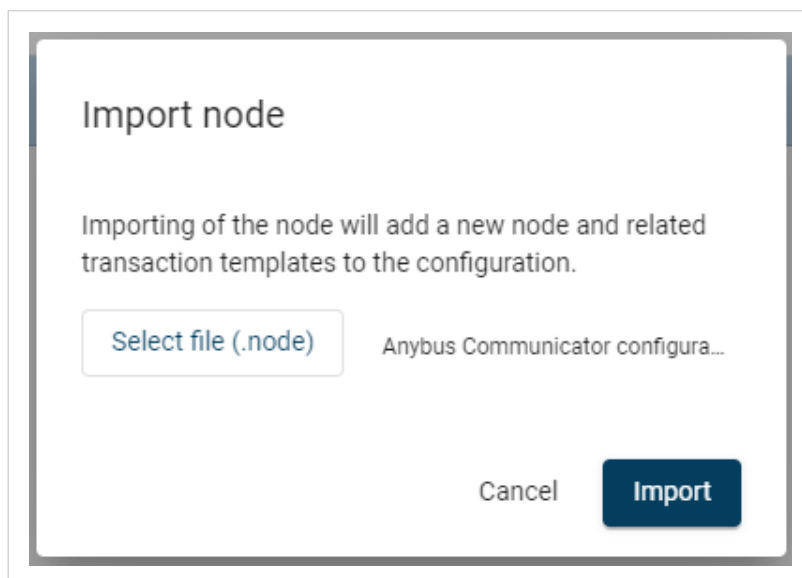


Figure 67. Example, selected .node file

Result

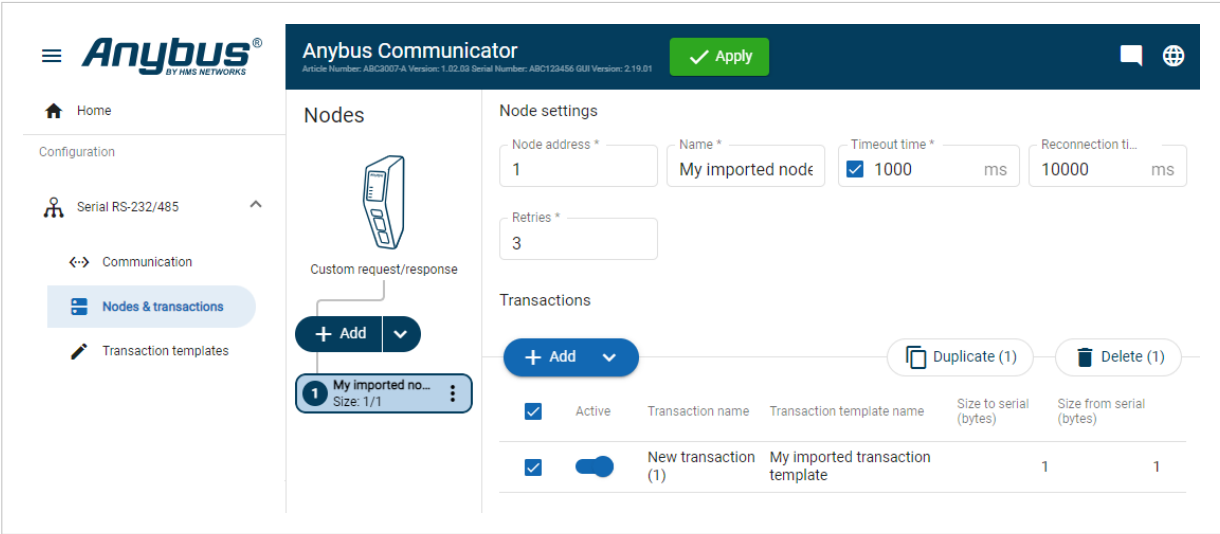


Figure 68. Example, imported node

The imported node is added to the **Nodes** list.

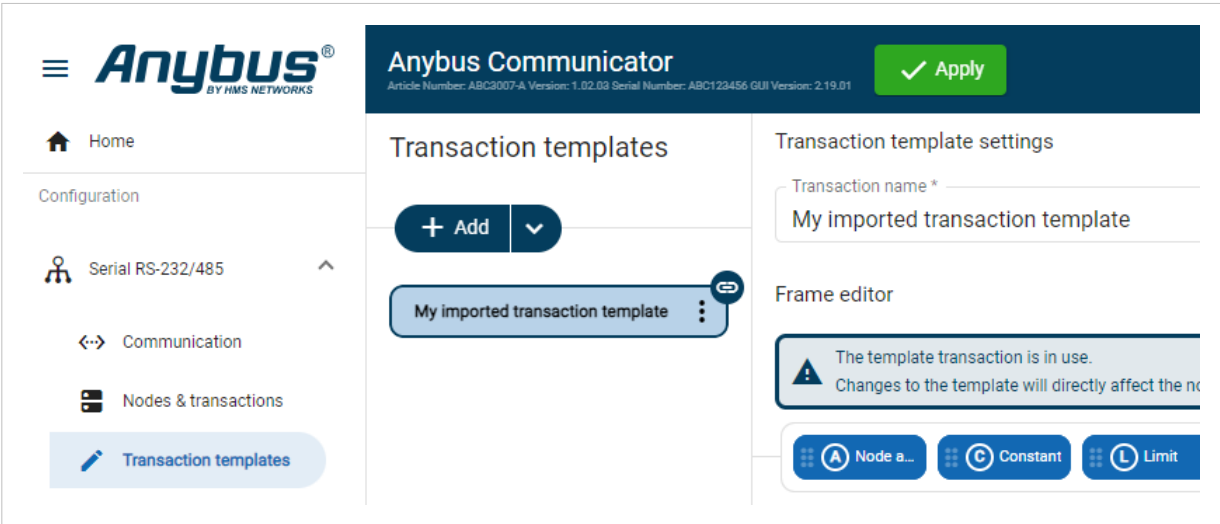


Figure 69. Example, imported transaction template

Any included transaction templates are added to the **Transaction templates** list on the **Transaction templates** page.

To Do Next



IMPORTANT

To identify the nodes and avoid node address conflicts on the subnetwork, each node (device) must have a unique node address number.

In the **Node settings** for the imported node, ensure that the **Node address** number is unique.

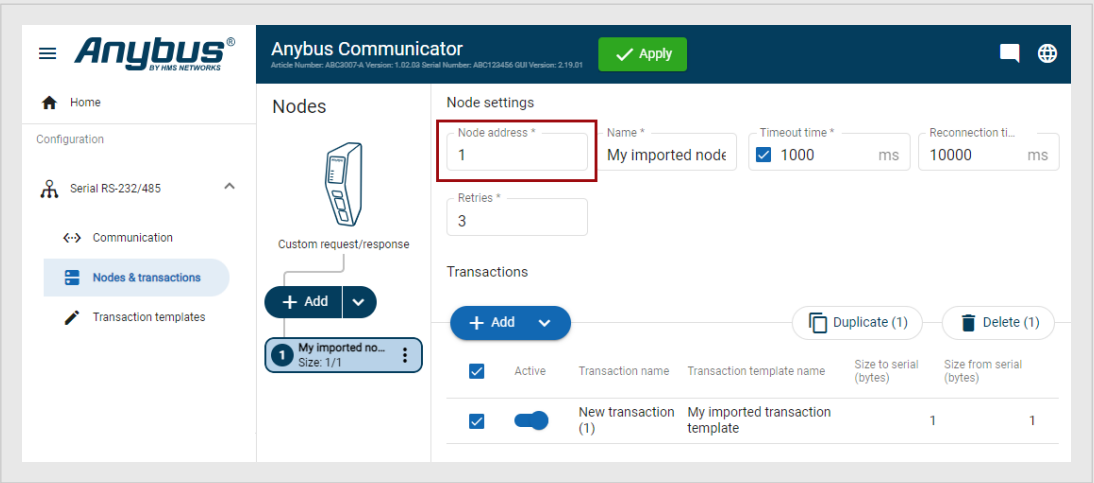


Figure 70. Node settings, Node address number

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

10.4.4. Import Transaction Template From Other Communicator Unit

When you have configured a node and want to use the same node for additional Communicator units, do the following.

Procedure

To Export the Transaction Template

In the built-in web-interface of the Communicator with the transaction template you want to export:

1. In the web-interface left sidebar menu, click **Transaction templates**.
2. On the transaction template to be exported, click the three dots icon and then click **Export node**.
A *.trans* file including the transaction template settings and frame fields is downloaded to your computer.

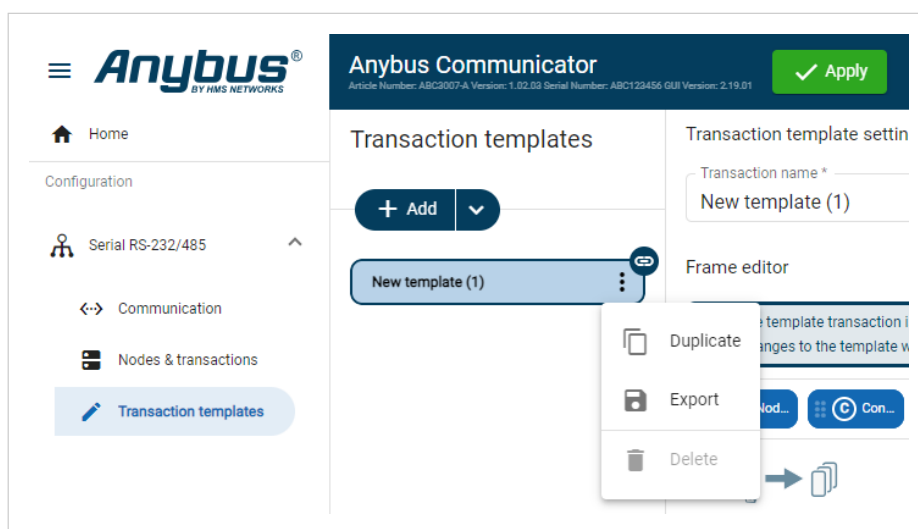


Figure 71. Export template

To Import the Transaction Template

In the built-in web-interface of the Communicator to which the transaction template is to be imported:

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

4. In the **Add** split button drop-down menu, select **Import template**.

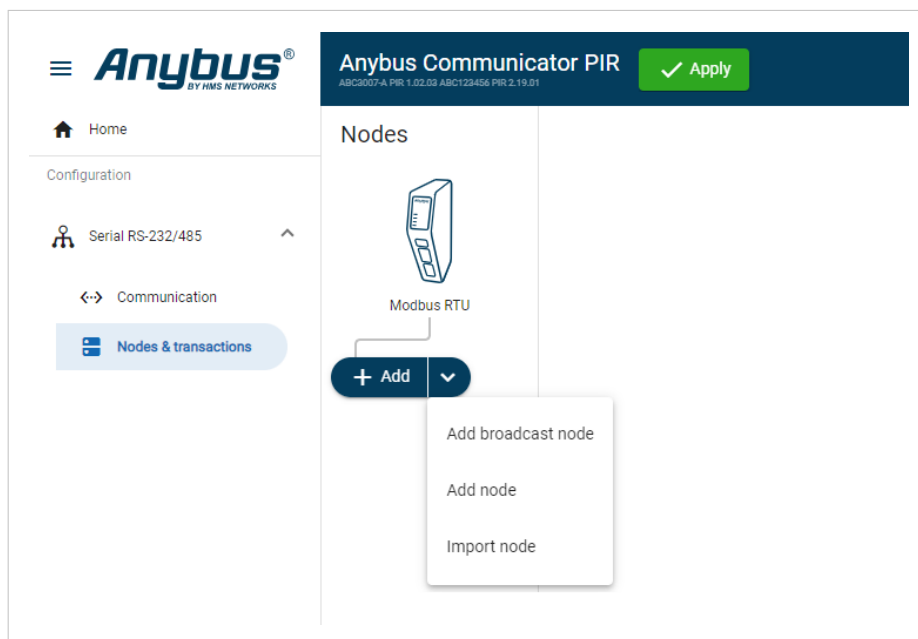


Figure 72. Import template

5. In the **Open** dialog box, browse to and select the `.trans` file and click **Open**.
6. To import the `.trans` file, click **Import**.

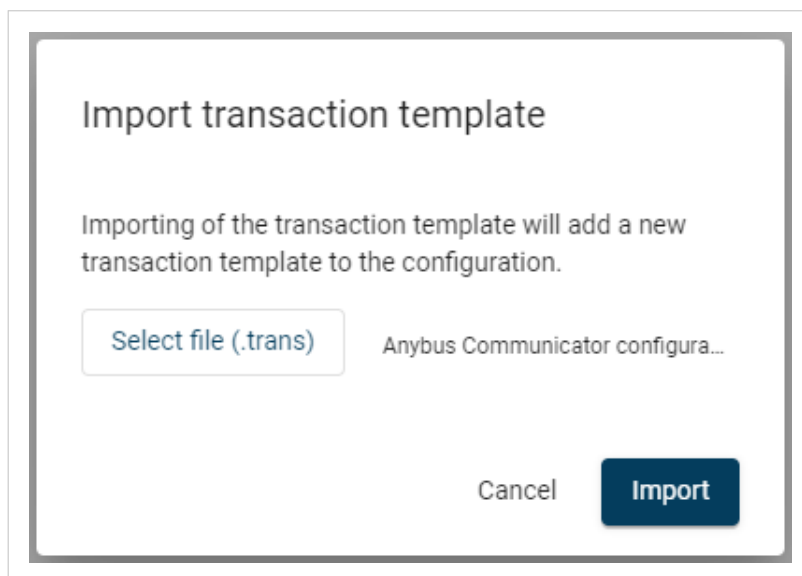


Figure 73. Example, selected .trans file

Result

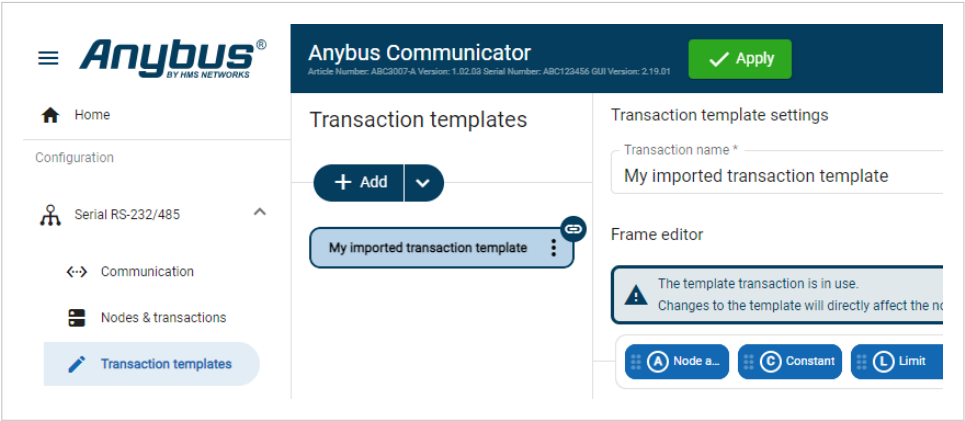


Figure 74. Example, imported transaction template

The imported transaction template is added to the **Transaction templates** list.

10.4.5. 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

The screenshot shows the 'Anybus Communicator' web interface. The top header includes the product name and version information, along with an 'Apply' button. The main area is divided into two sections: 'Nodes' on the left and 'Node settings' on the right. The 'Nodes' section shows a list of nodes, with Node 1 selected. The 'Node settings' section contains fields for 'Node address' (set to 1), 'Name' (set to 'New node'), 'Timeout time' (set to 1000 ms), 'Reconnection time' (set to 10000 ms), and 'Retries' (set to 3). Below these fields are buttons for '+ Add', 'Duplicate', and 'Delete'. At the bottom, there is a table with columns for 'Active', 'Transaction name', 'Transaction template name', 'Size to serial (bytes)', and 'Size from serial (bytes)'.

Figure 75. 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 Enabled by default 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 timeout setting. If enabled, 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. If disabled, the Communicator immediately sends the message again, when no response is received from the node
Reconnection time	Min 10 ms Max 60 0000 ms Default 10 000 ms	Specify for how long the Communicator should wait before attempting to reconnect, if the node is disconnected. Reconnect time (10 ms) is not applicable for the broadcast node, that hold transactions destined to all nodes.
Retries	0 to 10 Default 3	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 RTU 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 \(page 90\)](#).

10.4.6. 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.

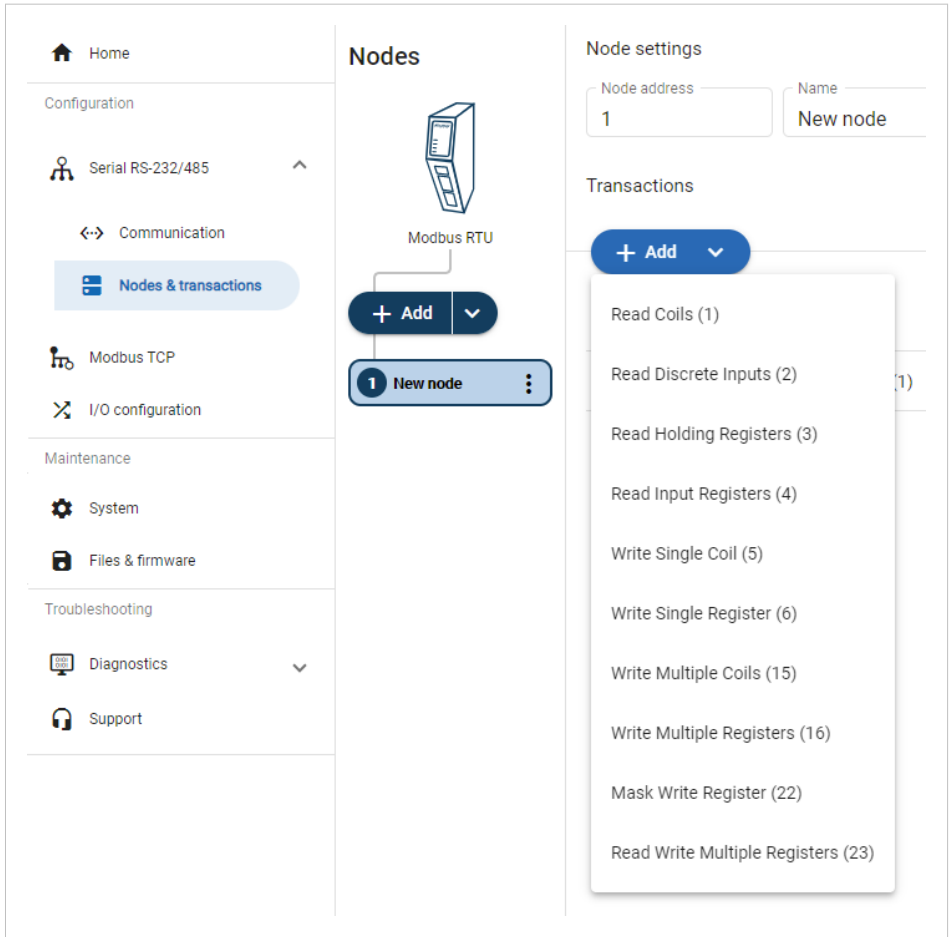


Figure 76. Add Modbus RTU transactions

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

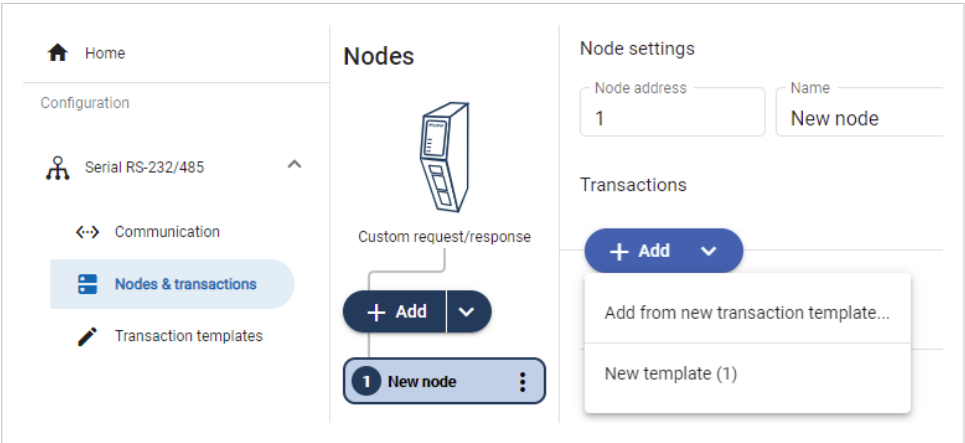


Figure 77. Add new empty template

- 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 \(page 67\)](#).

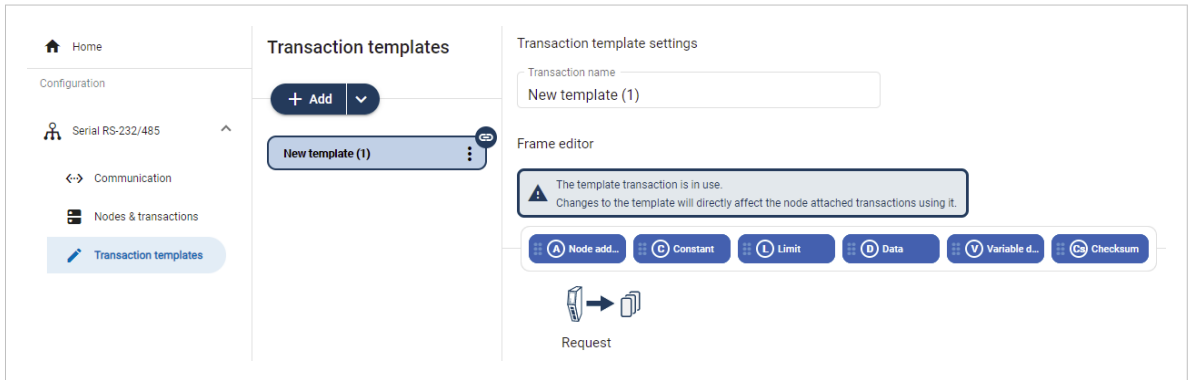


Figure 78. Add new empty template

- 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.

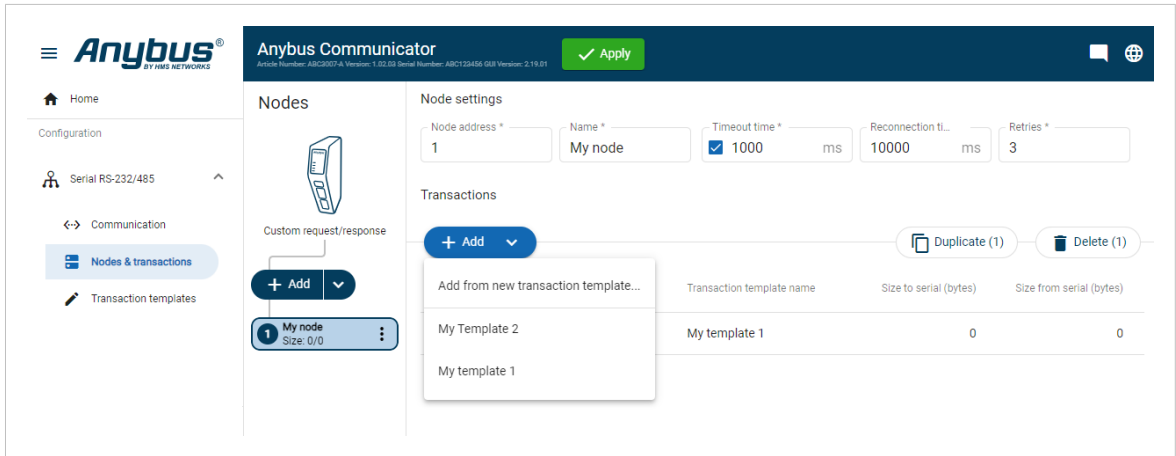


Figure 79. Add new transaction

To Do Next

Configure the Transactions settings, [Transaction Settings \(page 93\)](#).

10.4.7. 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 Function Codes \(page 183\)](#).

Procedure

The screenshot shows the 'Anybus Communicator' interface for the 'Modbus RTU' protocol. The 'Nodes' sidebar on the left shows a 'New node' button and a list with one node. The main area is divided into 'Node settings' and 'Transactions'. The 'Node settings' section includes fields for 'Node address *' (1), 'Name *' (New node), 'Timeout time *' (1000 ms), 'Reconnection ti...' (10000 ms), and 'Retries *' (3). The 'Address format' is set to 'Address (0, 1,...)'. The 'Transactions' section has a table with one transaction: 'New transaction (1)' with 'Read Holding Registers (3)' as the 'Modbus transaction', address '0', and quantity '1'. The right sidebar shows the 'Command name' (New command), 'Modbus command' (Read Holding Regist...), 'Address' (0), 'Quantity' (1), 'Update mode' (Cyclically), and 'Update time' (1000 ms).

Figure 80. Modbus RTU serial protocol

The screenshot shows the 'Anybus Communicator' interface for the 'Custom request/response' protocol. The 'Nodes' sidebar on the left shows a 'My first Node' button and a list with one node. The main area is divided into 'Node settings' and 'Commands'. The 'Node settings' section includes fields for 'Node address' (1), 'Name' (My first Node), 'Timeout time' (1000 ms), 'Reconnection ti...' (10000 ms), and 'Retries' (0). The 'Commands' section has a table with one command: 'New command (1)' with 'My first custom command' as the 'Custom command name', 'Size to EtherNet/IP™ (bytes)' (1), and 'Size from EtherNet/IP™ (bytes)' (0). The right sidebar shows the 'Command name' (New command (1)), 'Custom command' (My first custom com...), 'Command' (My first custom command), 'Update mode' (Cyclically), and 'Update time' (1000 ms).

Figure 81. 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	Byte counter: 0 to 255 End pattern: 0 to 1499 None (Default): 0 to 1500	The maximum payload length of the variable data field.
Update mode	Cyclically On data change Single shot Change of state on trigger ABC Classic on data change	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 configuration , the node will be marked with a flash icon. In the I/O configuration you can also configure the area map and the trigger byte address. See Trigger Byte (page 115) . ABC Classic on data change: This setting makes the Communicator compatible with the Communicator Classic On data change setting. If the initial value sent from the PLC at startup is 0, no transaction is triggered on the serial subnetwork.
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 configuration 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 configuration is incremented each time this transaction fails.

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

10.4.8. Activate/Deactivate Transaction

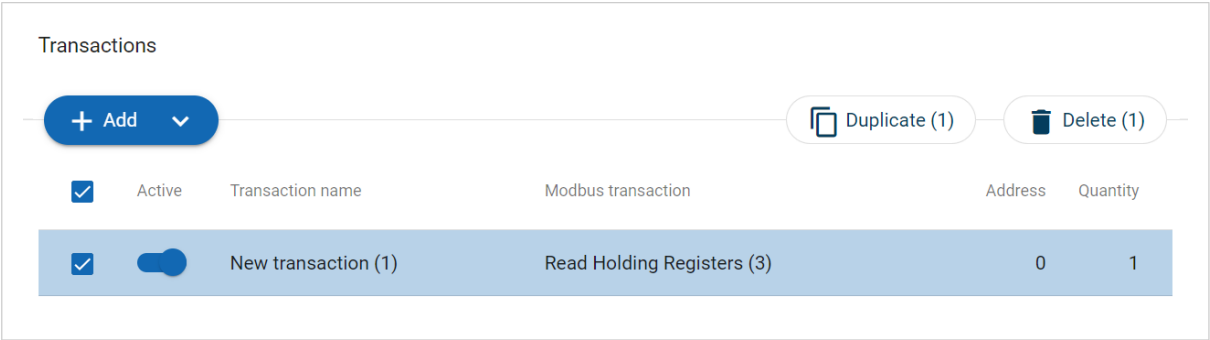


Figure 82. Activate/Deactivate Transaction

The transaction default status is **Active**.

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

10.4.9. Duplicate Transaction

When you duplicate a transaction, all settings are preserved.

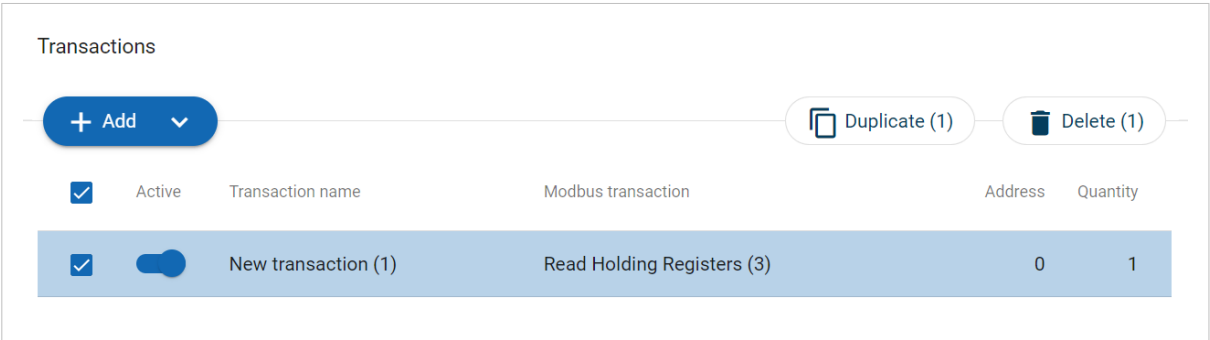


Figure 83. Duplicate transaction

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.

10.4.10. Delete Transaction

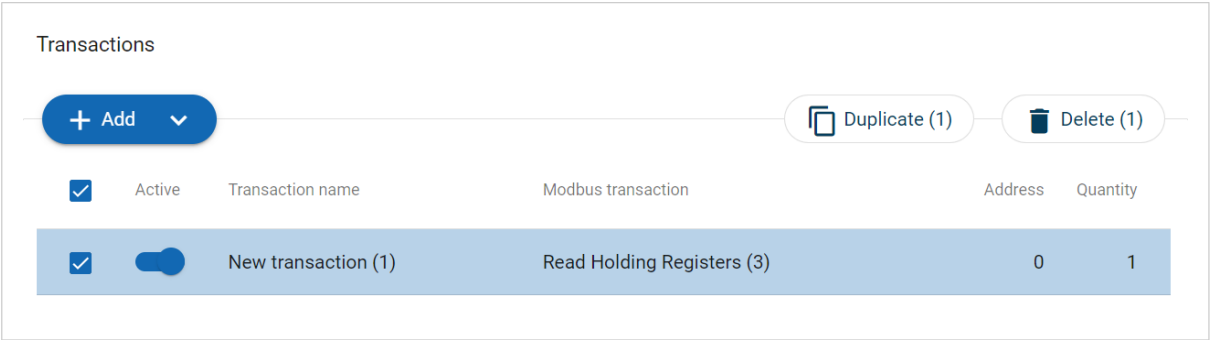


Figure 84. Delete transaction

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

10.5. Modbus TCP Settings

Configure the Modbus TCP network settings.

10.5.1. To Use Automatic I/O Sizes

Anybus Communicator
Article Number: AB7710-A Version: 1.2.3 Serial Number: ABC123456 GUI Version: 0.44.1

✓ Apply

Modbus TCP

I/O sizes

☒ Use automatic I/O sizes

When "Use automatic I/O sizes" is checked the size of the I/O data to and from the OT network will be set to the same size as provided by the serial subnetwork.

Data size to Modbus-TCP
2

Data size from Modbus-TCP
0

Figure 85. Modbus TCP, I/O sizes

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

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

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

10.5.2. To Configure I/O Sizes Manually



NOTE
The maximum data size in each direction is N/A bytes.

Anybus Communicator
Article Number: AB7710-A Version: 1.2.3 Serial Number: ABC123456 GUI Version: 0.44.1

✓ Apply

Modbus TCP

I/O sizes

☐ Use automatic I/O sizes

When "Use automatic I/O sizes" is checked the size of the I/O data to and from the OT network will be set to the same size as provided by the serial subnetwork.

Data size to Modbus TCP
2

Data size from Modbus TCP
0

Figure 86. Modbus TCP, I/O sizes

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

10.5.3. IP Settings

Anybus Communicator
Article Number: ABC2007-A EIP Version: 1.2.3 Serial Number: ABC123456 EIP GUI Version: 1.2.2 ✓ Apply

IP Settings

☒ DHCP enabled

IP address: 192.168.0.222 Subnet mask: 255.255.255.0 Gateway address: 192.168.0.1

Primary DNS: 0.0.0.0 Secondary DNS: 0.0.0.0

Hostname

Figure 87. IP Settings

By default, DHCP is Enabled.

Default IP Settings

The Communicator comes with the following factory default IP settings:

Setting	Default value	Description
DHCP	Enabled	DHCP (Dynamic Host Configuration Protocol) automatically gives devices on a network the IP settings they need to connect and communicate.
IP address	There is no default IP address.	The EtherNet/IP network IP address in IPv4 dot-decimal notation.
Subnet mask	255.255.255.0	The EtherNet/IP network Subnet mask in IPv4 dot-decimal notation.
Gateway address	There is no default Gateway address.	The EtherNet/IP network Gateway address in IPv4 dot-decimal notation. If there is no gateway available, set the Gateway address to: 0.0.0.0
Primary DNS server	There is no default Primary DNS server.	The EtherNet/IP network Primary DNS in IPv4 dot-decimal notation.
Secondary DNS server	There is no default Secondary DNS server.	The EtherNet/IP network Secondary DNS in IPv4 dot-decimal notation.

Using DHCP Server for IP Configuration

Procedure

With DHCP enabled, the IP settings are provided by the high level network DHCP server.

Ensure the **DHCP enabled** checkbox is selected.

To Configure IP Settings Manually

Procedure

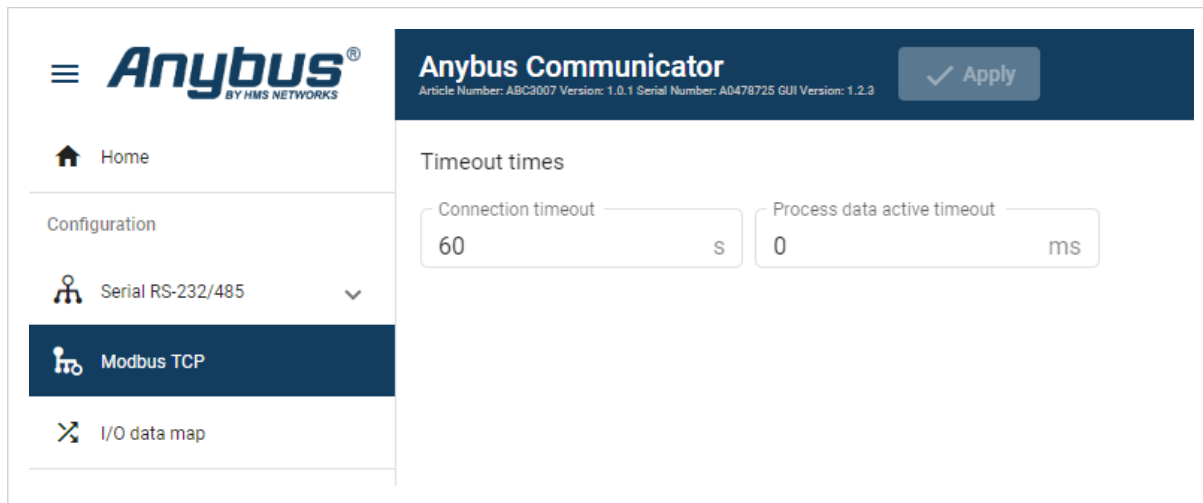
1. To disable DHCP, deselect the **DHCP enabled** checkbox.
2. Then, enter the desired IP settings values.

Hostname

You can label the Communicator.

- The maximum allowed length of the Hostname is 64 characters.
- No symbol characters, punctuation characters, or whitespace are permitted.
- Write the Hostname as one single word.

10.5.4. Timeout Time Settings



The screenshot shows the 'Anybus Communicator' web interface. The top header includes the 'Anybus' logo and version information: 'Article Number: ABC9007 Version: 1.0.1 Serial Number: A0478725 GUI Version: 1.2.3'. A navigation sidebar on the left contains links for 'Home', 'Configuration', 'Serial RS-232/485', 'Modbus TCP' (which is highlighted), and 'I/O data map'. The main content area is titled 'Anybus Communicator' and features an 'Apply' button. Below this, the 'Timeout times' section contains two input fields: 'Connection timeout' set to '60' with a unit of 's', and 'Process data active timeout' set to '0' with a unit of 'ms'.

Figure 88. Modbus TCP page, Timeout times

Connection Timeout

Specify how long a Modbus TCP connection may be idle before it is closed by the Communicator.

The default value is 60 seconds.

To disable **Connection timeout**, set the value to 0.

Process Data Active Timeout

Specify the maximum allowed time between two incoming messages in steps of 10 ms.

If this time is exceeded, the Serial RS-232/485 network is considered to be offline.

The data sent to the Modbus TCP network is frozen.

A value of 0 disables this feature, i.e. the Serial RS-232/485 network can never go offline.

The default value is 0 ms.

10.6. Modbus TCP Advanced Settings

10.6.1. Legacy Mode for Anybus Address Mode | Modbus TCP

Advanced settings option for Modbus TCP.

About Legacy Mode

Use Legacy Mode when you want to make a new Communicator compatible with the Communicator Classic Anybus Address Mode.

See also [Legacy Address Mode I/O Data Mapping \(page 132\)](#).

To Use Legacy Mode

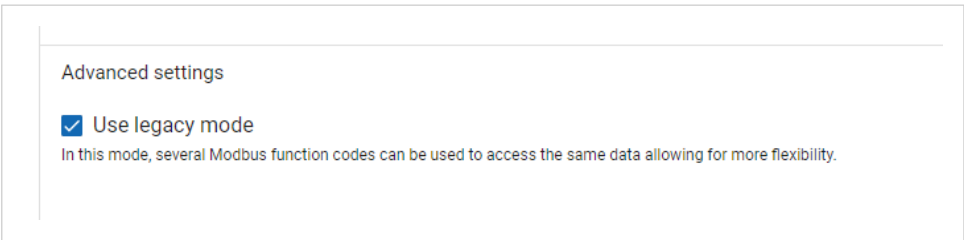


Figure 89. Advanced settings > Use legacy mode

To enable Legacy mode, select the **Use legacy mode** checkbox.

See Also

[Legacy Address Mode I/O Data Mapping \(Page 132\)](#)

10.7. Security Settings

10.7.1. Local Time and Date

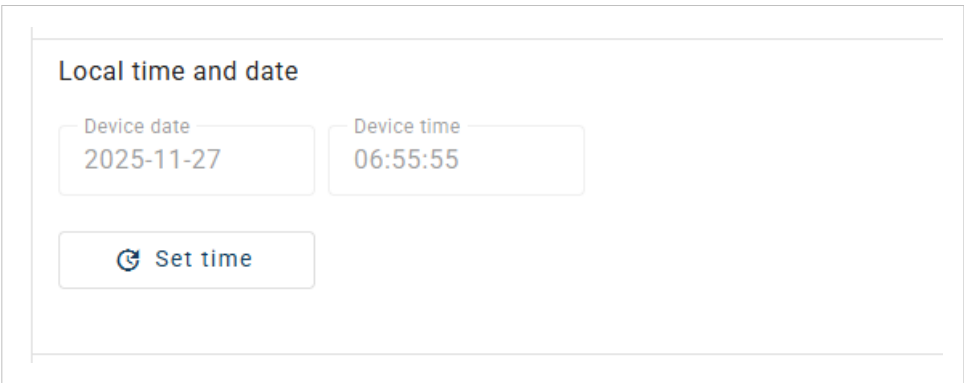


Figure 90. System page, Local time and date

You can synchronize the time in the Communicator with the current browser time and date.

Make sure the system clock on the computer connected to the Communicator is accurate before setting the time.

On the **System** page, in the **Local time and date** panel, click **Set time** to synchronize the Communicator with the time and date from the browser on the connected computer.

10.7.2. Network Time Protocol (NTP) Synchronization

The Communicator can use the **Network Time Protocol (NTP)** to synchronize its clock with network time servers.

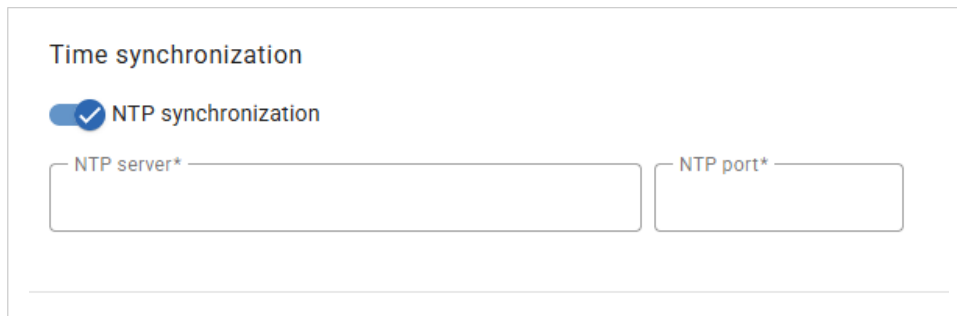


Figure 91. System page, Time synchronization

By default, **NTP synchronization** is disabled.

To use **NTP synchronization**:

1. On the **System** page, click the **NTP synchronization** toggle.
2. In the **NTP server*** field, enter the server name or IP number of the NTP server.
3. In the **NTP port*** field, enter the port number used by the NTP server for communication.

10.7.3. Installing a Web Server Certificate

Install a web server certificate in the Communicator.

Before You Begin



IMPORTANT

A factory reset of the Communicator will erase all stored web server certificates.



NOTE

The web server certificate and private Key must be provided as two separate files.



NOTE

The web server certificate must be a Base64 encoded DER certificate in PEM format.



NOTE

If the web server certificate is used for HTTPS, the subject name (CN) must match the device address (IP or DNS name).

Procedure

To install web server certificate:

Security

☒ Use roles

Once activated, login with credentials laser marked on product and set password for other roles.

☐ HTTPS disabled

With HTTPS enabled, configuration data sent between your browser and the Communicator is encrypted. A valid SSL/TLS server certificate and a private key is required.
Remove existing certificate to upload a new.
NOTICE: The device must be rebooted after toggling HTTPS before the change becomes active.

Private key

Certificate

Select a new certificate file and upload it to the Communicator. You must first upload the private key before your certificate.

Device discovery

Read-only

Controls if the device IP address can be discovered by stand alone HMS tools including IP Config.

Local time and date

Device date

2026-01-27

Device time

09:14:17

Set time

Figure 92. System page, Security settings

1. Open the **System** page.

2. Upload the private key:
 - a. Click **Private key**,
 - b. In the **Upload private key** window, click **Select private key file (.pem)**.
 - c. In the **Open** dialog box, browse to and select the private key file and click **Open**.

**NOTE**

Ensure the .pem file corresponds to the SSL/TLS certificate.

- d. Click **Upload key**.
The message 'Private key uploaded successfully' appears when the upload is complete.
3. Upload the web certificate:
 - a. Click **Certificate**.
 - b. In the **Upload certificate** window, click **Select certificate file (.crt)**.
 - c. In the **Open** dialog box, browse to and select the web server certificate file and click **Open**
 - d. Click **Upload certificate**.
The message 'A new certificate has been uploaded.' appears when the upload is complete.
 - e. The web server certificate appears in the **Security** section.
4. Enable **Secure Connection (HTTPS)**:
Click the slide toggle HTTPS.
Secure connection encrypts data sent between your browser and the server.
See also [Secure Connection With HTTPS Mode \(page 107\)](#).

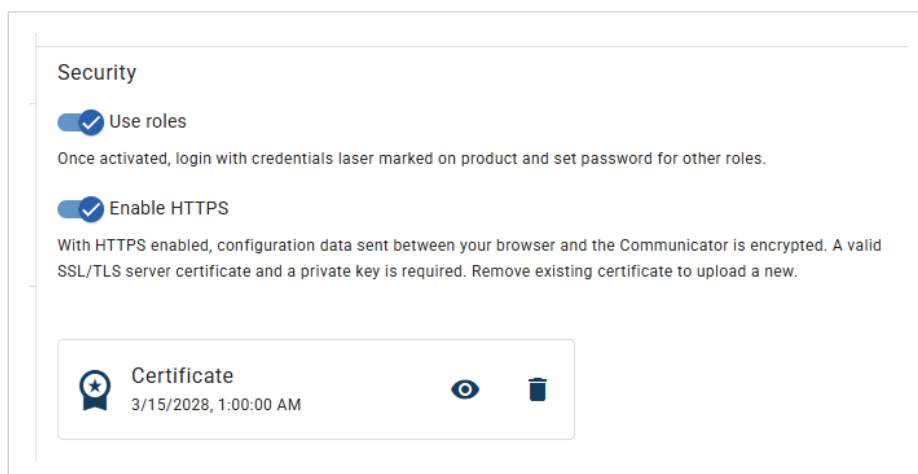


Figure 93. HTTPS enabled

5. Click **Apply** in the web interface header.
6. For the setting to take effect, restart the Communicator by disconnecting and reconnecting its power supply.

10.7.4. View Current Certificate

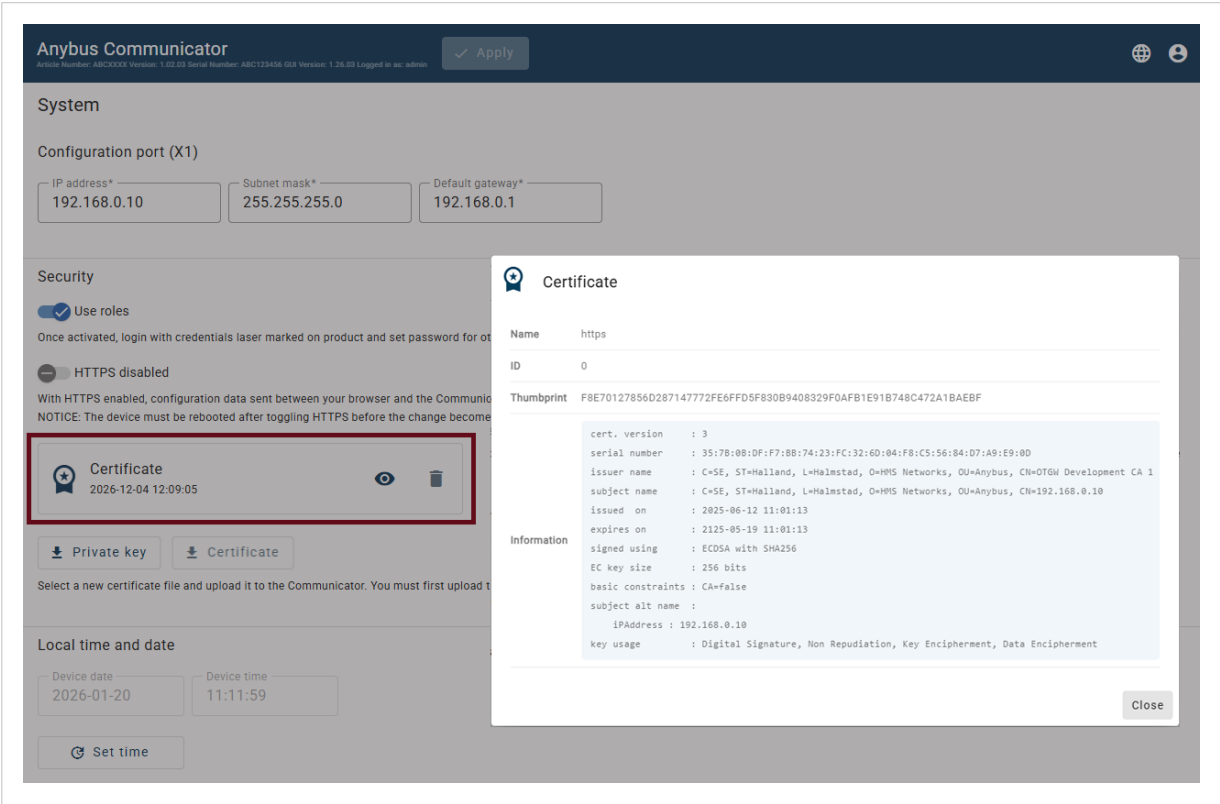


Figure 94. System page, View current certificate

To view the currently uploaded web server certificate, click the **View** (eye) icon.

The **Certificate** window opens.

10.7.5. Update or Delete a Certificate

Update an Existing Certificate

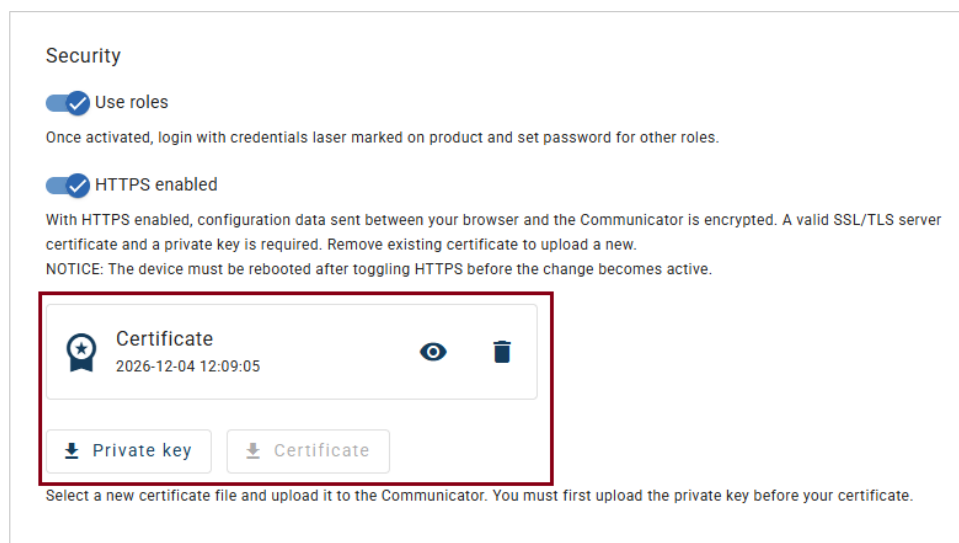


Figure 95. System page showing installed certificate

To update the existing HTTPS setup without removing it, you can upload a new certificate.

If the updated certificate uses a new key pair, first upload the matching private key.

Follow the steps in [Installing a Web Server Certificate \(page 102\)](#).

The uploaded file replaces the existing certificate — and private key, if included.

Delete a Certificate

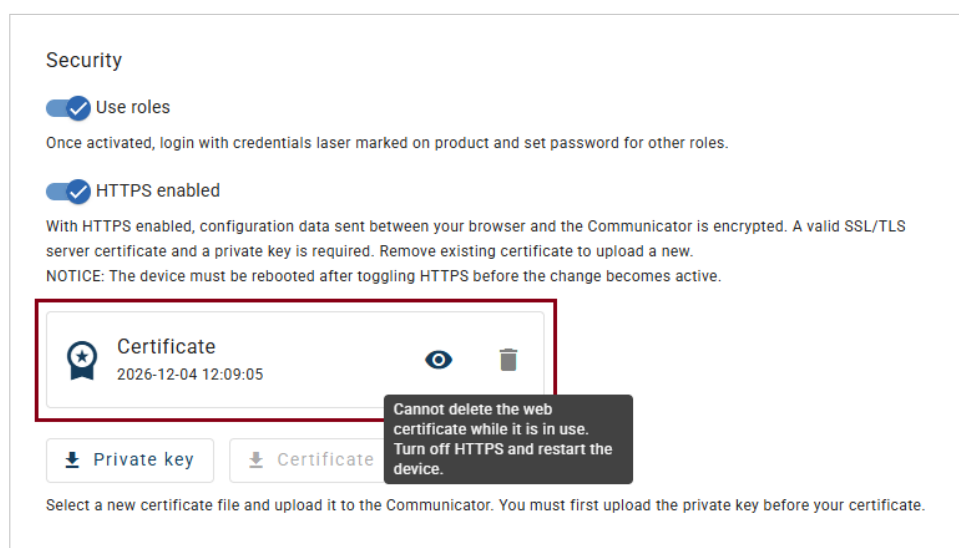


Figure 96. System page showing a locked certificate

When a certificate has been uploaded and HTTPS mode is enabled, the certificate becomes locked and cannot be deleted. In this state the trash-can delete button is disabled and grayed out.

To unlock the certificate, you must first disable HTTPS mode and then reboot the Communicator.

After the reboot, the certificate unlocks, and you can delete it.

See Also

[Secure Connection With HTTPS Mode \(Page 107\)](#)

[None Secure HTTP Mode \(Page 109\)](#)

10.7.6. Secure Connection With HTTPS Mode

About the Secure Connection (HTTPS) Mode

Secure connection (HTTPS) encrypts data sent between your browser and the server to keep the connection secure.



IMPORTANT

TCP Port 443 is used for HTTPS over Transport Layer Security (TLS) in HTTPS mode.

When using HTTPS mode, ensure TCP Port 443 is available,



IMPORTANT

For the HTTPS mode to work and enable secure access to the Communicator built-in web interface, a valid SSL/TLS server certificate is required.

Install the SSL/TLS certificate before enabling the HTTPS mode. See [Installing a Web Server Certificate \(page 102\)](#).



IMPORTANT

When HTTPS mode is enabled but no valid SSL/TLS server certificate is available, users can still access a limited HTTP version of the built-in web interface to update or install a SSL/TLS certificate. See [Installing a Web Server Certificate \(page 102\)](#).



IMPORTANT

A factory reset of the Communicator restores it to the default HTTP mode.

Before You Begin

Ensure Web Server Certificate is Installed

Ensure there is a valid web server certificate installed, see [Installing a Web Server Certificate \(page 102\)](#).

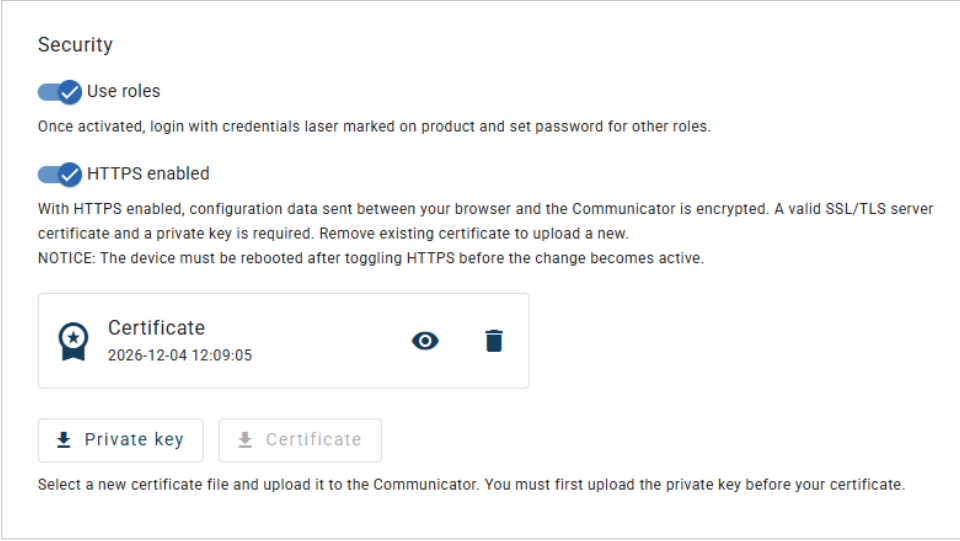
Ensure Accurate Time Settings

Incorrect time can cause SSL/TLS certificates to fail. The Communicator might interpret the certificate as expired or not yet valid, which is a security risk.

Ensure **Local time and date** are set and **Time synchronization** is enabled, see [Local Time and Date \(page 100\)](#) and [Network Time Protocol \(NTP\) Synchronization \(page 101\)](#).

Procedure

To enable HTTPS mode:



Security

☒ Use roles

Once activated, login with credentials laser marked on product and set password for other roles.

☒ HTTPS enabled

With HTTPS enabled, configuration data sent between your browser and the Communicator is encrypted. A valid SSL/TLS server certificate and a private key is required. Remove existing certificate to upload a new.

NOTICE: The device must be rebooted after toggling HTTPS before the change becomes active.

 **Certificate**
2026-12-04 12:09:05





 Private key

 Certificate

Select a new certificate file and upload it to the Communicator. You must first upload the private key before your certificate.

Figure 97. System page

1. Open the **System** page.
2. In the **Security** section, click the **HTTPS disabled** slide toggle.
3. Click **Apply** in the web interface header.
4. For the setting to take effect, restart the Communicator by disconnecting and reconnecting its power supply.

See Also

[Update Or Delete A Certificate \(Page 105\)](#)

[None Secure HTTP Mode \(Page 109\)](#)

10.7.7. None Secure HTTP Mode

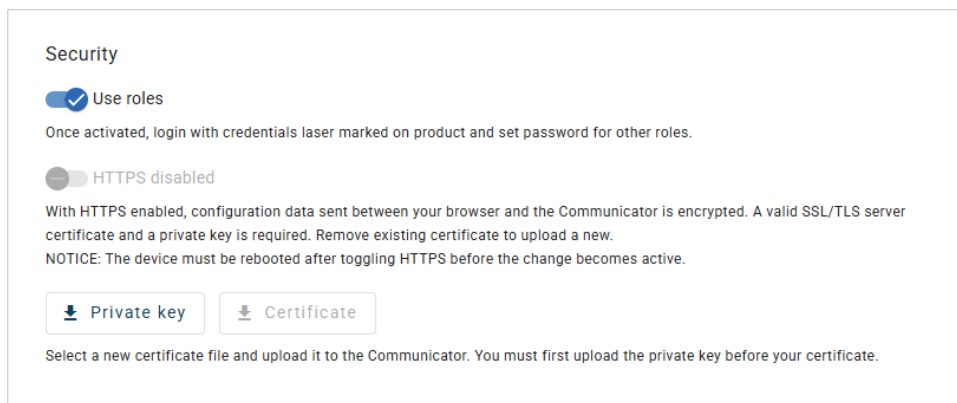


Figure 98. System page, Security settings, and Secure Connection (HTTPS) disabled

About the Hypertext Transfer Protocol (HTTP) Mode



IMPORTANT

By default, **Secure Connection (HTTPS)** is disabled and Communicator uses the HTTP mode.

To protect unauthorized access to the Communicator built-in web interface, ensure the security switch is locked.

To lock the security switch, see [Security Switch \(page 26\)](#).



IMPORTANT

TCP Port 80 is used for HTTP mode and for the limited version of the built-in web interface used when no certificate is available in HTTPS mode. Ensure that TCP Port 80 is open on the Communicator and network.

When using HTTP mode, ensure that TCP Port 443 is closed to prevent HTTPS access.

HTTP Mode Overview

- Data sent between the Communicator built-in web interface and the networks is transmitted in unencrypted plain text.
- Anyone on the same network could potentially intercept or manipulate the data.
- Recommended only in internal environments, such as during initial setup, or in isolated or trusted networks.

See Also

[Update Or Delete A Certificate \(Page 105\)](#)

[Secure Connection With HTTPS Mode \(Page 107\)](#)

10.7.8. Device IP Address Discovery via HMS Software Applications

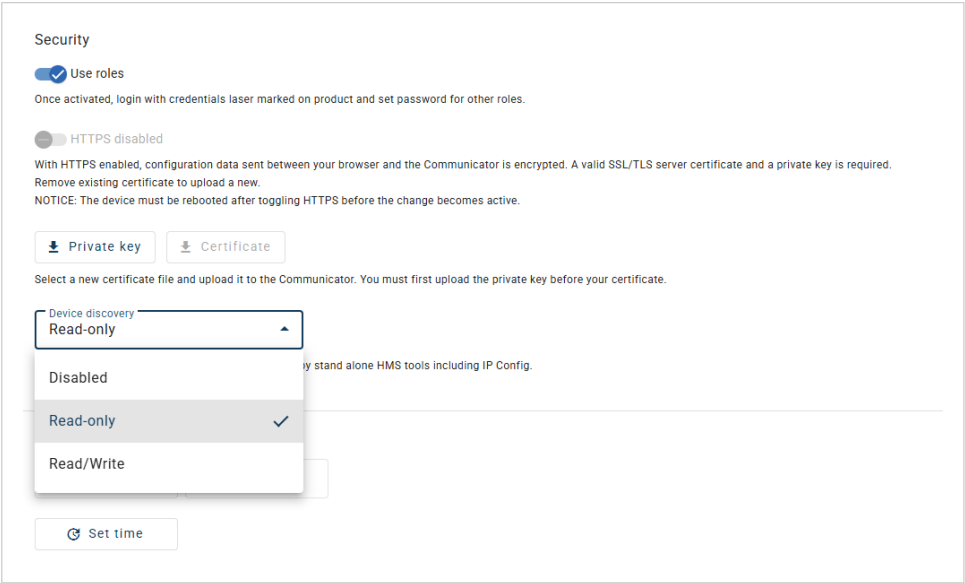


Figure 99. System page, Security settings, and Device discovery menu options

You can control whether the Communicator IP address can be discovered by stand-alone HMS software applications, including HMS IPconfig.

By default, **Read-only** mode is enabled.

The following **Device discovery** modes are available:

Mode	Description
Disabled	The Communicator does not expose its IP address to HMS software applications.
Read-only	The Communicator IP address is visible to HMS software applications and can be read, but not changed.
Read/Write	The Communicator IP address is visible to HMS software applications and can be read and changed.

10.8. I/O Configuration

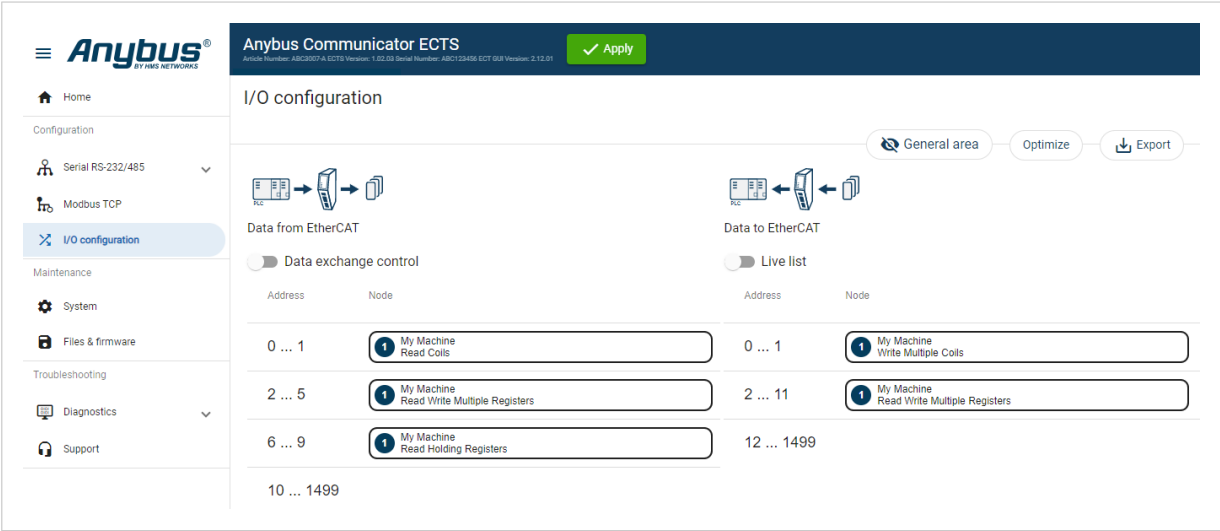


Figure 100. I/O configuration page

On the **I/O configuration** 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 \(page 97\)](#).

There are three areas: **Data from Modbus TCP**, **Data to Modbus TCP** and **General area**. See [Map Area \(page 114\)](#).

10.8.1. Optimize the I/O Configuration

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 configuration, the current I/O configuration will be overwritten.

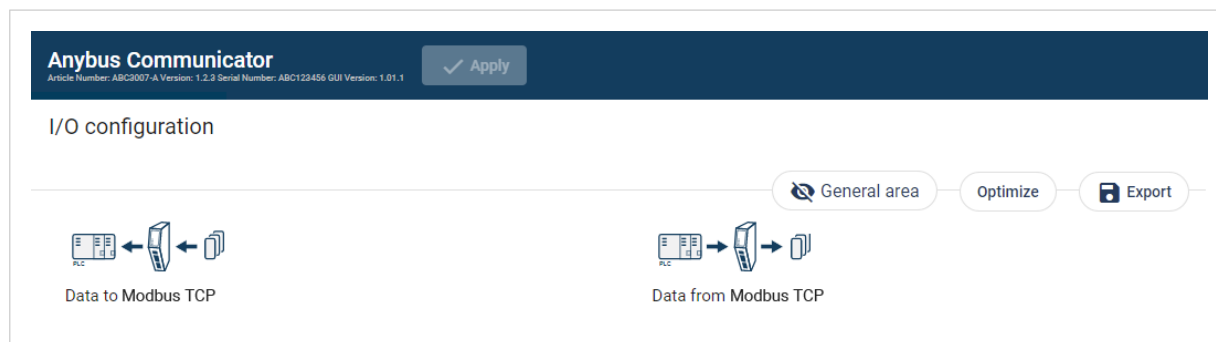


Figure 101. I/O configuration page, Optimize

To optimize the map:

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

10.8.2. 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.

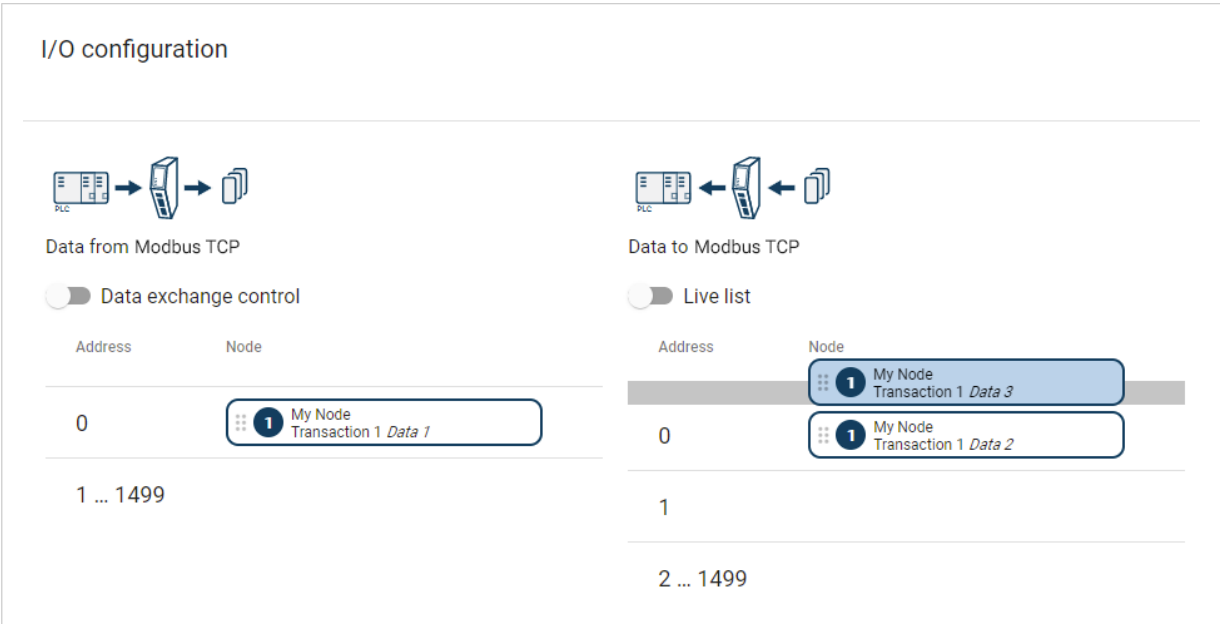


Figure 102. I/O configuration page, change the order of transactions

Transactions can not share the same I/O area.

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

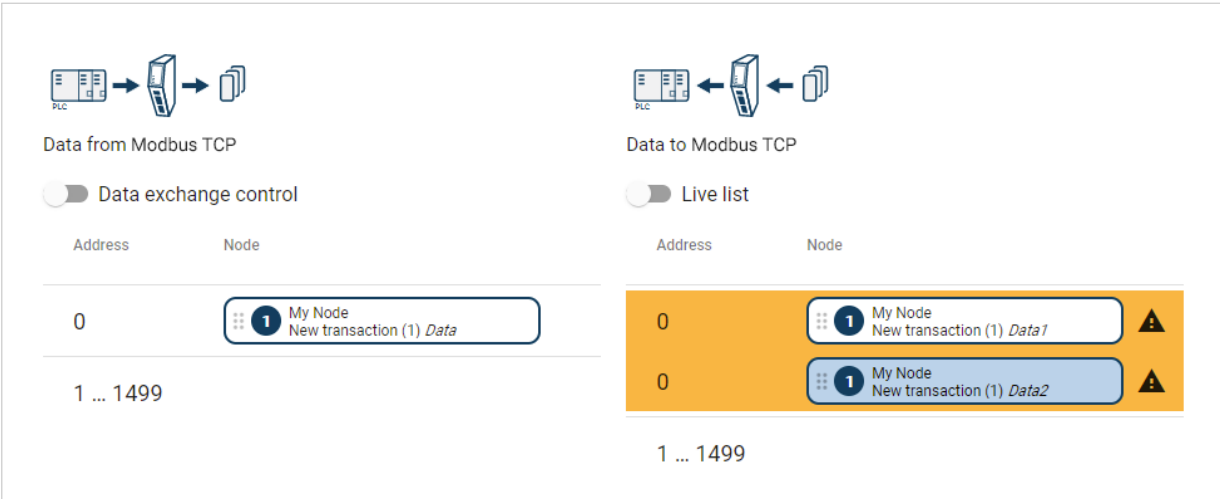


Figure 103. Highlighted I/O area

10.8.3. Map Area

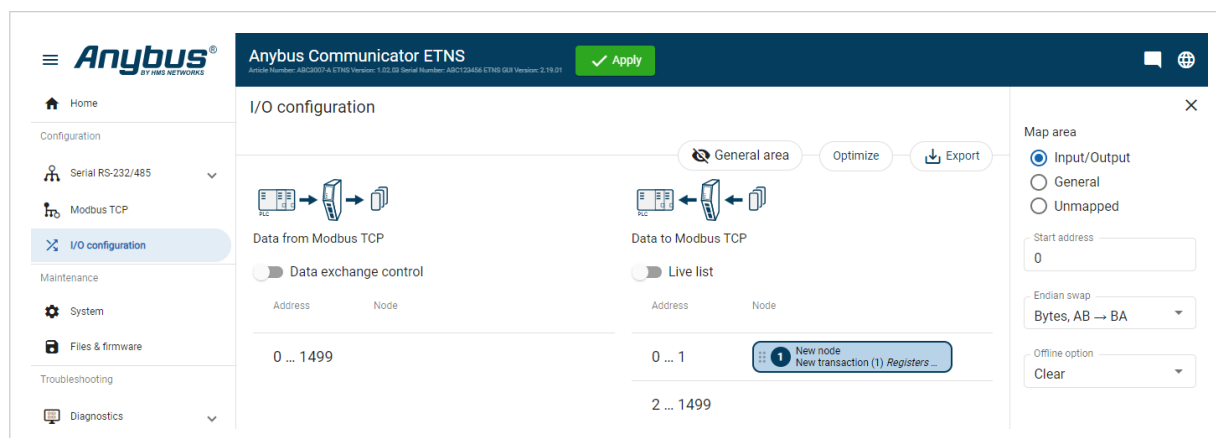


Figure 104. I/O configuration page, Map area options

Map Area Options

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

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

- **Input/Output:** The transaction data is sent/received 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.

10.8.4. 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 Modbus TCP** area or the **General** area.

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

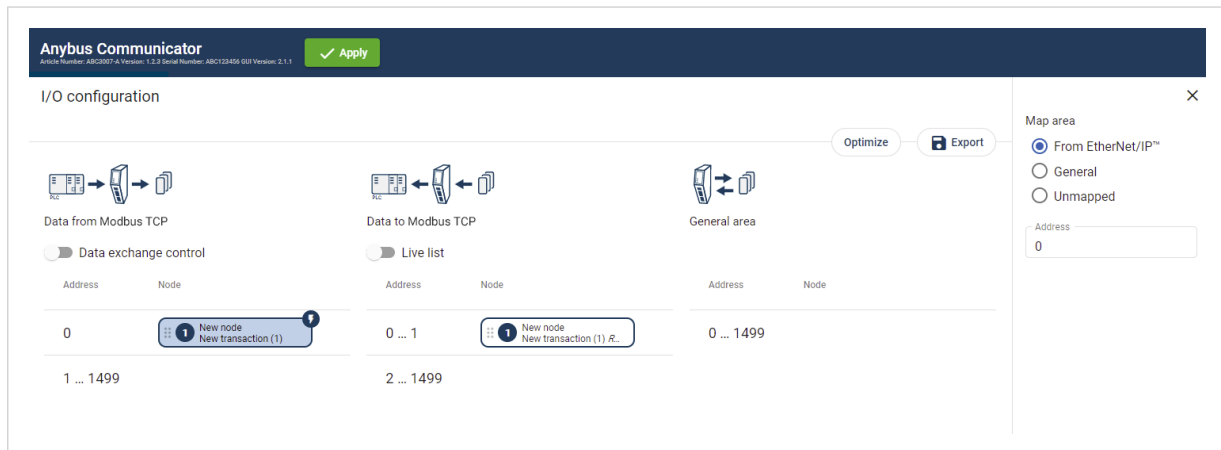


Figure 105. I/O configuration page

How to Enable Trigger Byte on a Node Procedure

1. Navigate to the **Nodes & transactions** page.
2. Select the desired node and transaction.
3. In the transaction sidebar **Update mode** menu, select **Change of state on trigger**.
4. Navigate to the **I/O configuration** 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 Modbus TCP:** The trigger byte is stored in the I/O configuration **Data from Modbus TCP** area.
- **General:** The trigger byte is stored in the I/O configuration **General area**.
- **Unmapped:** The transaction data is not used.

Address

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

10.8.5. Endian Swap

About Endianness

Big-endian (BE)

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

Little-endian (LE)

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

Convert Between Big-Endian and Little-Endian

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

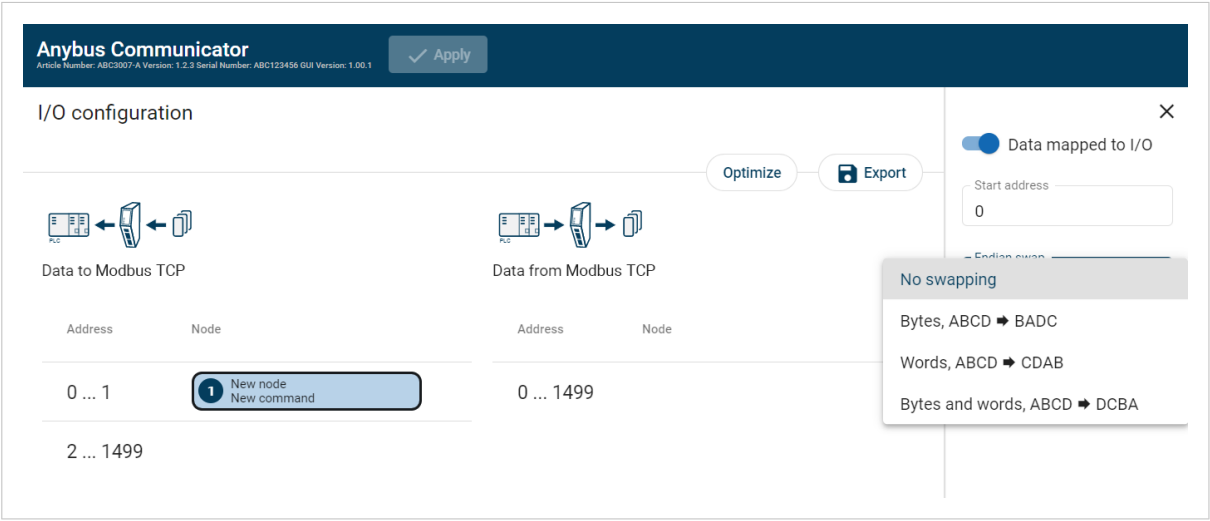


Figure 106. I/O configuration page, Endian swap

To reverse the byte order:

- 1. In the web-interface left sidebar menu, click **I/O configuration**.
- 2. In the data map, select the transaction for which you want to do swap the byte order.
- 3. Select the endian swap type from the **Endian swap** drop-down menu.

Setting	Description
No swapping	Default setting No swapping is performed on the data.
Bytes	Swap 2 bytes A B C D becomes B A D C
Words	Swap 4 bytes A B C D becomes C D A B
Bytes and words	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.

10.8.6. Offline Option

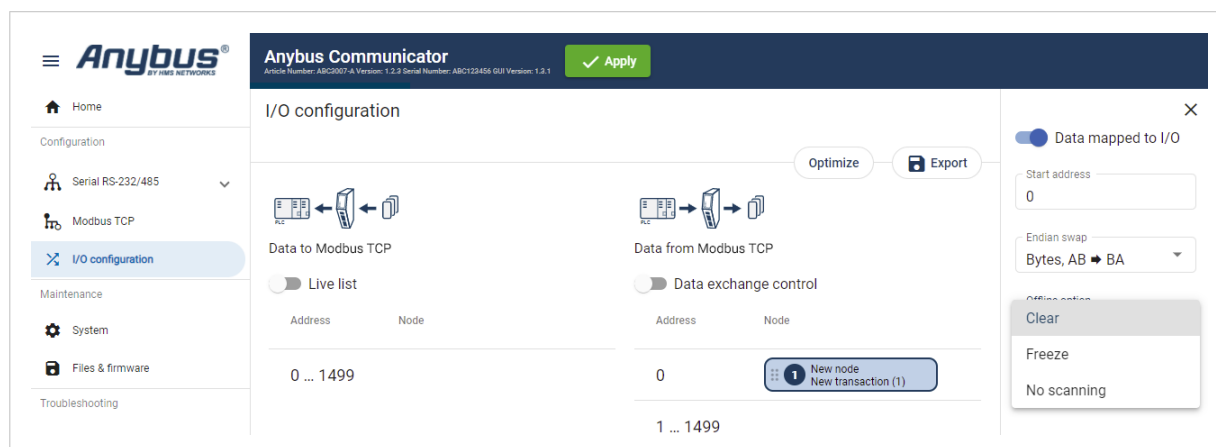


Figure 107. I/O configuration page, Offline options

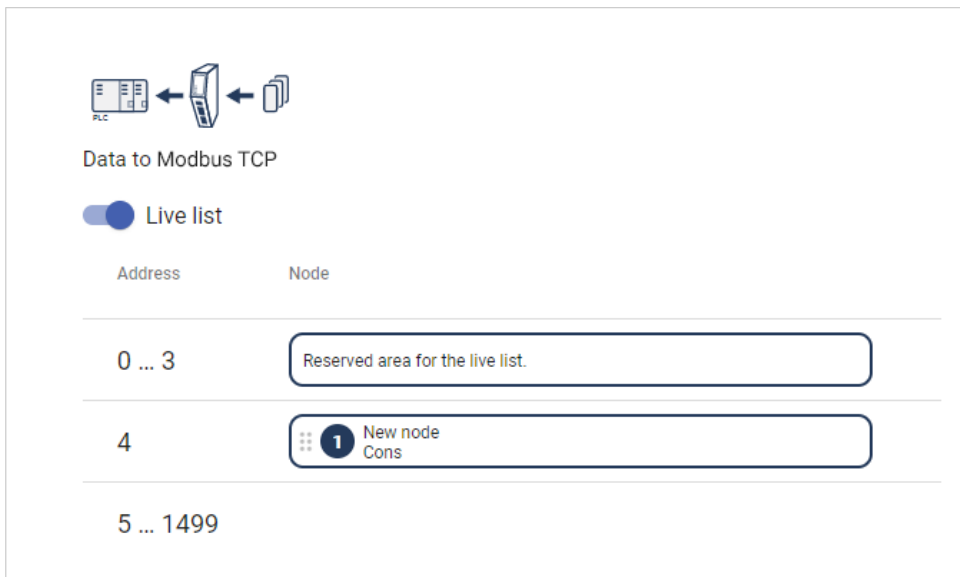
Offline mode is used to define what data to send if the network connection or connection with a specific node is lost.

You must specify the offline mode to use for each transaction in the I/O configuration.

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.

10.8.7. Live List



PLC ← Communicator ← Modbus TCP

Data to Modbus TCP

☒ Live list

Address	Node
0 ... 3	Reserved area for the live list.
4	1 New node Cons
5 ... 1499	

Figure 108. I/O configuration page, Live list enabled

By default, **Live list** is disabled.

About the Live List

- When **Live list** is enabled, the first four bytes of process data on the Modbus TCP network contain the live list.
- Each bit in the **Live list** can hold the status for one node.
- The **Live list** holds 32 bits, a total of 32 nodes connected to the Communicator.
- The bit is 0 when the bit does not correspond to a configured node.
For example, this occurs when the number of configured nodes is less than 32.
- Each bit is 1 when the corresponding nodes is online.

10.8.8. Data Exchange Control

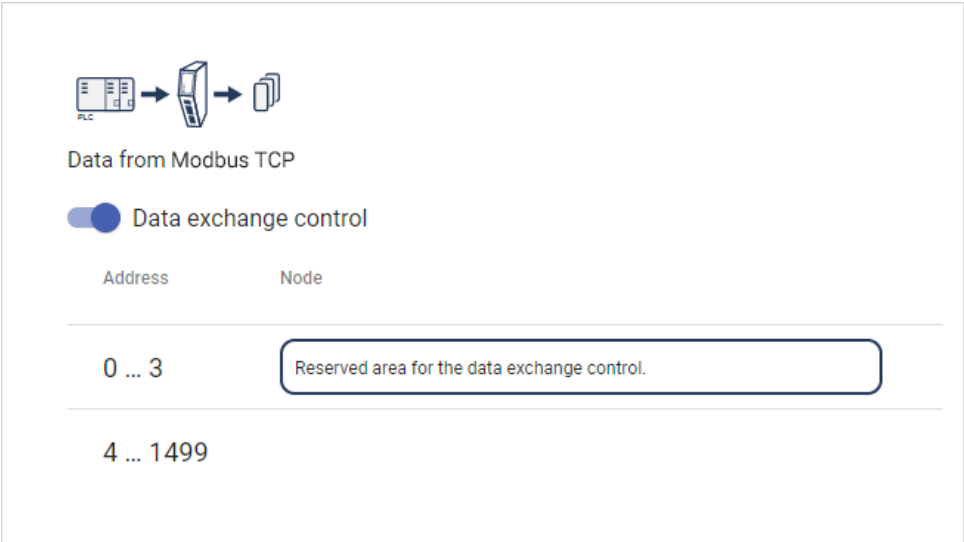


Figure 109. I/O configuration, **Data exchange control** enabled

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

When **Data exchange control** is enabled, the first four bytes of process data on the Modbus TCP 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 node 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 data exchange is enabled for the corresponding node, the control bit is 1.

10.9. Configuration Notes

You can add notes to describe the Communicator configuration.

10.9.1. Add Configuration Note

Procedure

1. To open the **Configuration Notes** window, click on the **comments** icon .



Figure 110. Configuration note, comment icon

2. To add a new configuration note, click **Add**.

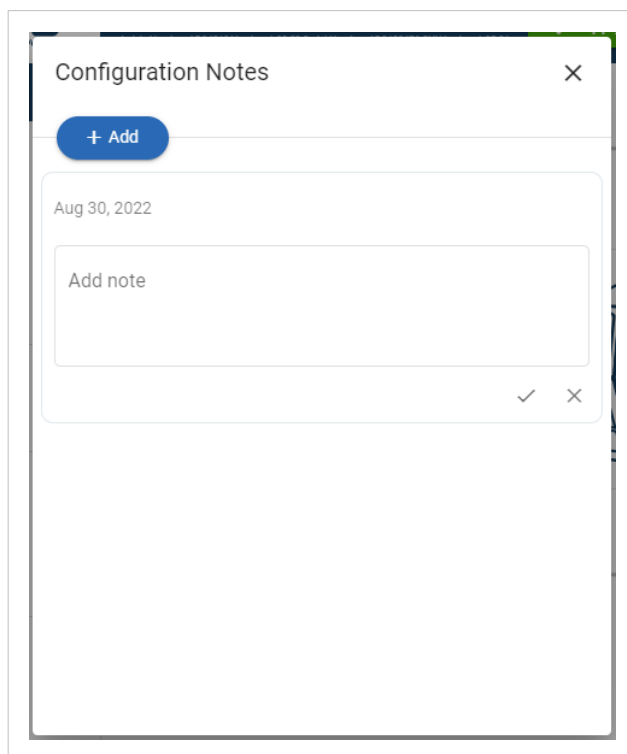


Figure 111. Add new configuration note

3. Write your configuration note and click **accept** ✓.

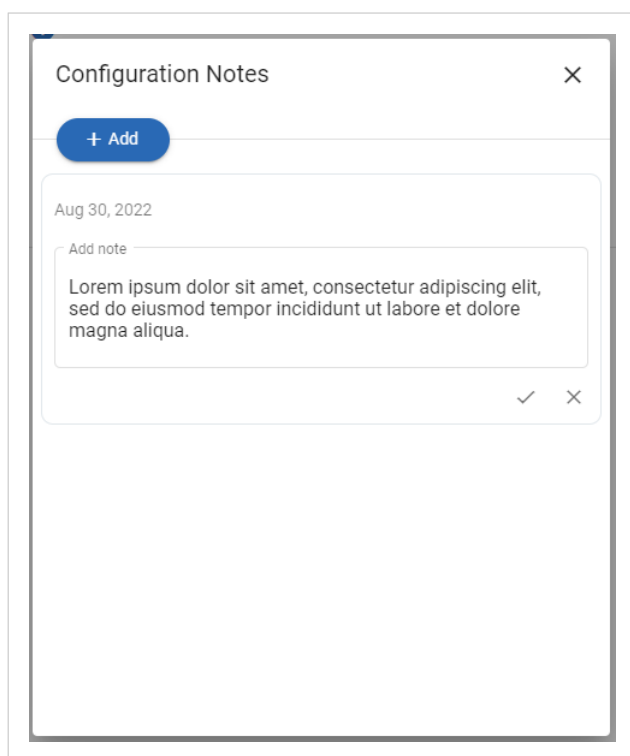


Figure 112. Write a configuration note

The configuration note is added to the list.

4. To close the window, click **close** ✕.
5. To save the configuration note, click **Apply** in the web-interface header, and follow the instructions.

10.9.2. View and Edit Configuration Notes

To view and/or edit a note, click on the **comments** icon .

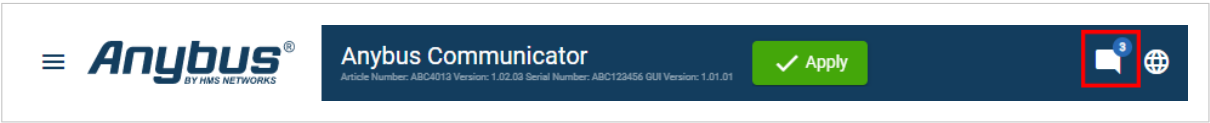


Figure 113. Example: The comment icon indicates that there are three added notes

The configuration notes are listed in the **Configuration Note** window.

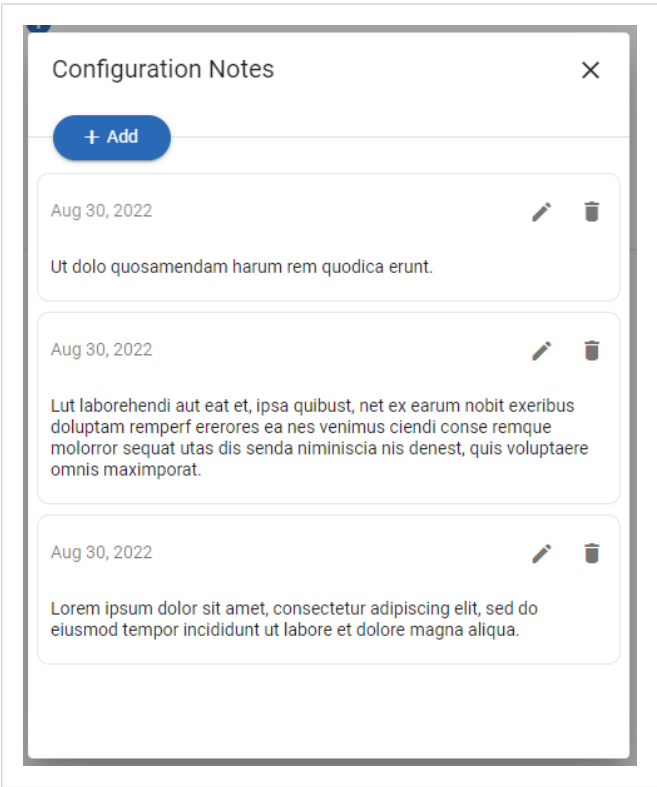


Figure 114. Example: The Configuration Notes window with added notes

10.10. Apply Configuration

Before You Begin

**NOTE**

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

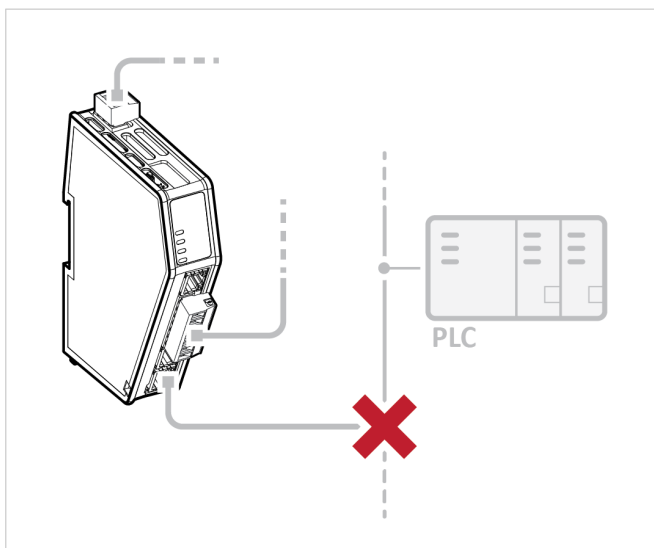


Figure 115. Disconnect the Communicator from the Modbus TCP network

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

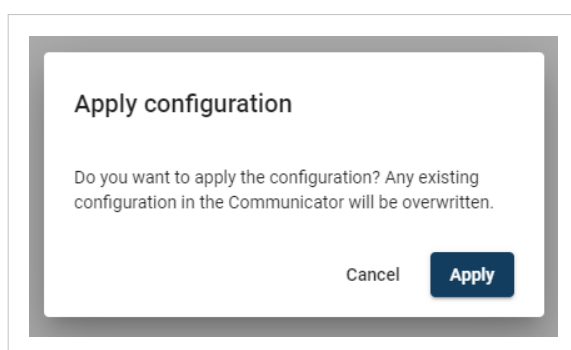
Procedure

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.



10.11. 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

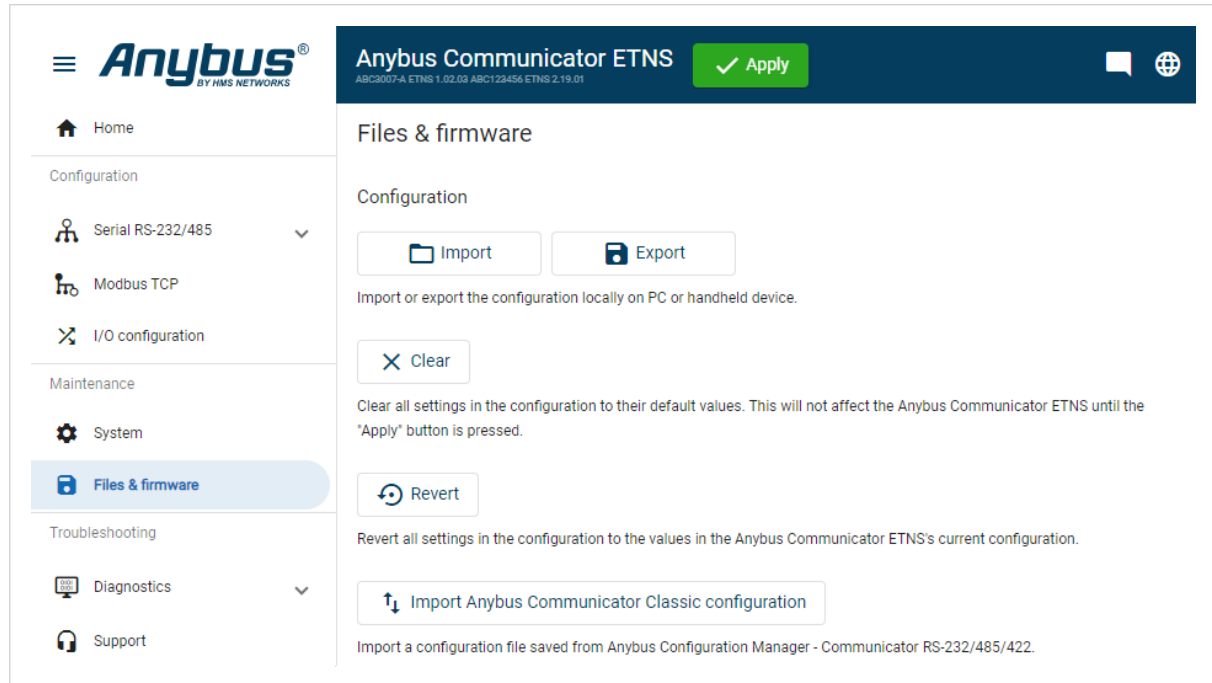


Figure 116. Files & firmware page

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 a configuration 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.

10.12. To Use a 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 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 \(page 51\)](#).
- 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

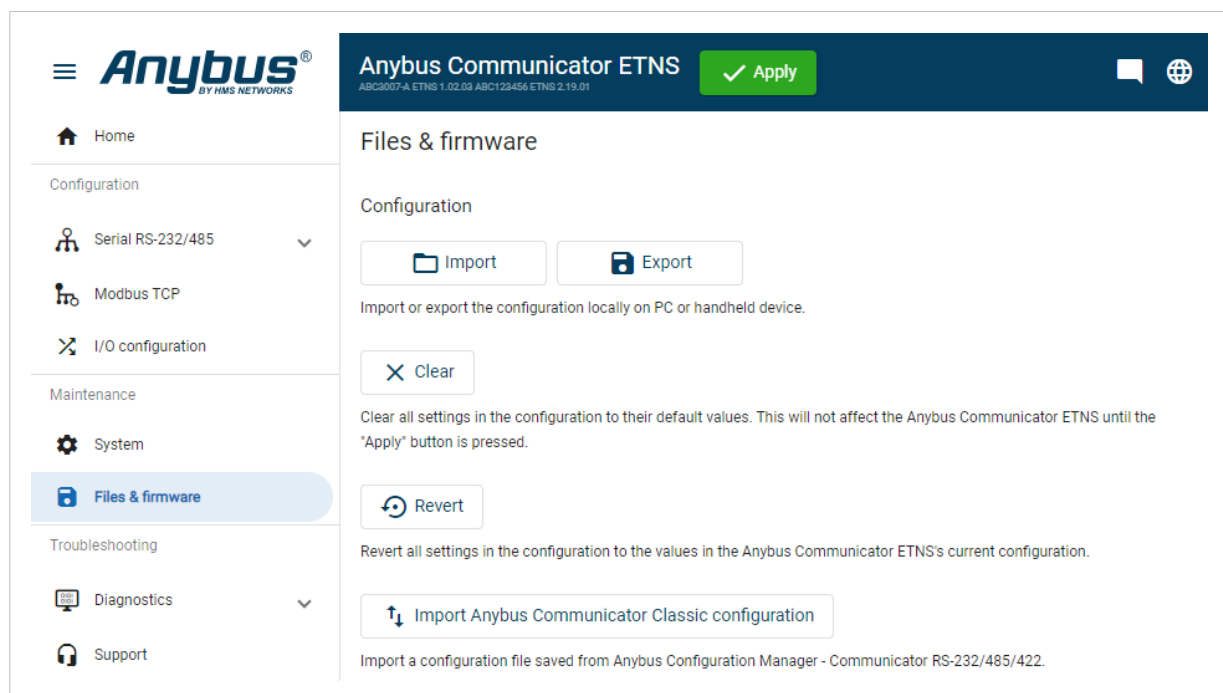


Figure 117. Files & firmware page

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**.

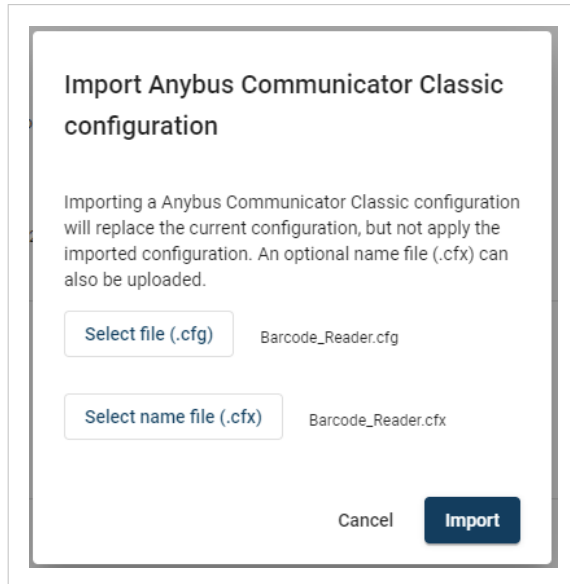
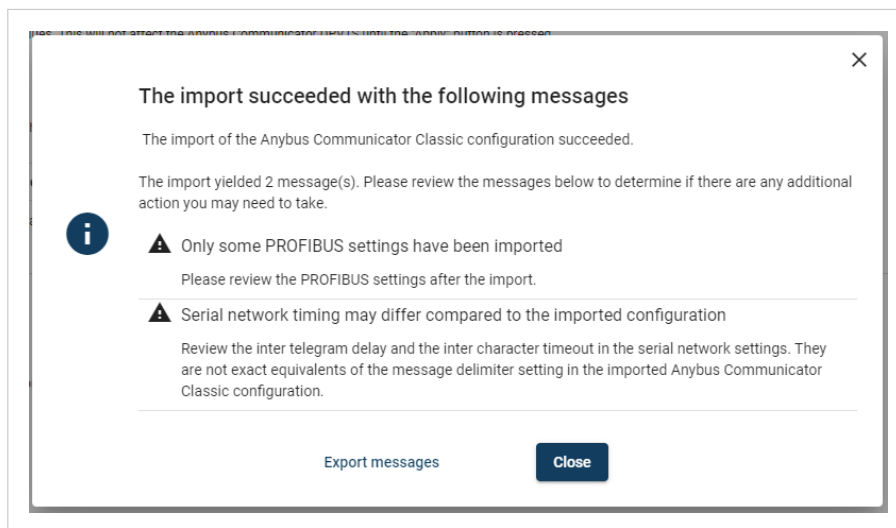


Figure 118. Example, selected .cfg and .cfx files

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**.

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

Figure 119. Example, list with messages about the import

11. PLC Configuration

11.1. PLC Device Security



IMPORTANT

It is important to maintain the cybersecurity of the Communicator.

Before connecting the Communicator to a PLC, ensure the PLC is configured and installed in accordance with the PLC supplier hardening guidelines.

11.2. Export I/O Configuration

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.

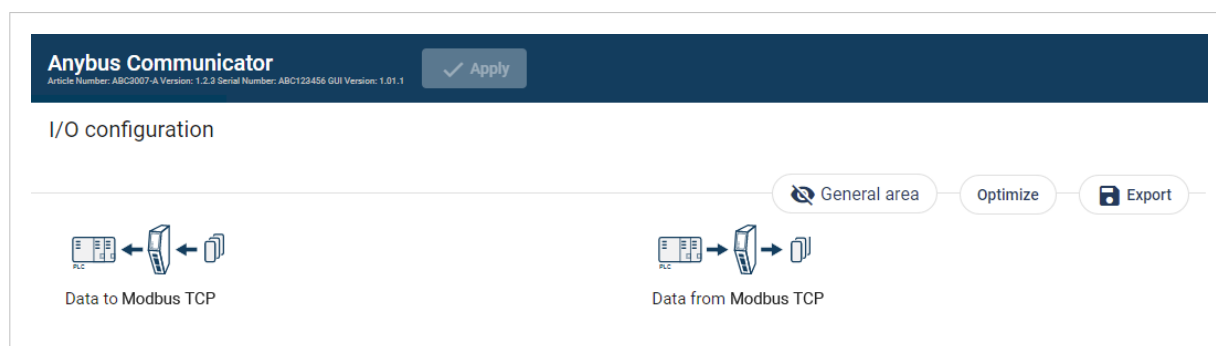


Figure 120. I/O configuration page

On the **I/O configuration** 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.

11.3. Modbus Addressing and Register Mapping

For information about the Modbus Transactions, see [Modbus Data Model](#) and [Modbus Function Codes \(page 183\)](#).

11.3.1. The Difference Between Address and Register Start

To read or write data from a Modbus device you can use either Address or Register.

For information on whether to use Register or Address, see [Nodes and Transactions \(page 80\)](#) and the user documentation for the client device and the node(s) that will be exchanging data.



TIP
If Register is used, add +1 to the Address value.

Example 10. Modbus Function Code 0x04 Read Input Registers

Table 1. Modbus client values

Address	Register	Modicon Register
0	1	30001
1	2	30002
2	3	30003

Example 11. Modbus Function Code 0x03 Read Holding Registers

Table 2. Modbus client values

Address	Register	Modicon Register
0	1	40001
1	2	40002
2	3	40003

11.3.2. To Read and Write Data

To Read Data

To read data, you can use different Modbus Functions.

Discrete Inputs

- Modbus Function 0x02 Read Discrete Input Status

Discrete Input start at address 0.

Input Registers

- Modbus Function 0x04 Read Input Register
- Modbus Function 0x03 Read Holding Registers

The Input Registers starts at address 0 which corresponds to Register 1.

To use Read Holding Registers the address is offset to address 2048 which corresponds to Register 2049.

The first Input Register (30001)

The first Input Register (30001) can be read using Modbus Function code 0x04 Read Input Registers at Address 0 or Register 1.

You can also use Read Holding Registers (Modbus Function code 0x03) to read the same data (at Address 42048 = Register 42049) or use Read Discrete Inputs 0x02 at address 0..15.

To Write Data

To write data, you can use different Modbus Functions.

Coils

- Modbus Function 0x05 Force Single Coil
- Modbus Function 0x0F Force Multiple Coils

Coils starts at address 0.

Holding Registers

- Modbus Function 0x06 Write Single Register
- Modbus Function 0x10 Write Multiple Registers

The Holding Registers starts at address 0 which corresponds to Register 1.

Holding Register (40001)

The first Holding Register (40001) can be written to using Modbus Function code 0x06 Write Single Register at Address 0 or Register 1.

To read back what has been written, you can use Read Holding Registers 0x03 at address 0..749 = Register 1..750 (40001..40750) or Read Coils 0x01 at address 0..11999.

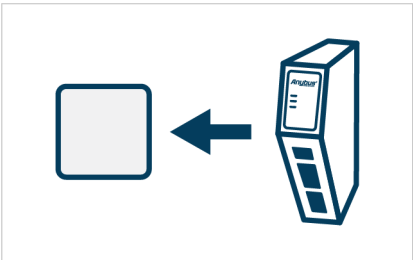
Idle Mode

Transaction	Holding Register	Description
Enter/Exit Idle Mode	0x1004	Used by the Modbus TCP client to indicate an idle/offline mode on the network.

11.3.3. Default Address Mode I/O Data Mapping

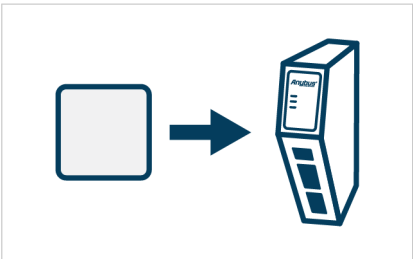
The following Modbus function codes are compatible with the default addressing mode.

Input Data to Client/Server - Data to Modbus



Process I/O data offset		Discrete input address		Input register address		Holding register address	
HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC
0x0000 - 0x0001	0 - 1	0x0000 - 0x000F	0 - 15	0x0000	0	0x0800	2048
0x0002 - 0x0003	2 - 3	0x0010 - 0x001F	16 - 31	0x0001	1	0x0801	2049
...	...	...	...	...	...	...	...
0x05D8 - 0x05D9	1496 - 1497	0x2EC0 - 0x2ECF	11968 - 11983	0x02EC	748	0x0AEC	2796
0x05DA - 0x05DB	1498 - 1499	0x2ED0 - 0x2EDF	11983 - 11999	0x02ED	749	0x0AED	2797

Output Data From Client/Server - Data From Modbus TCP

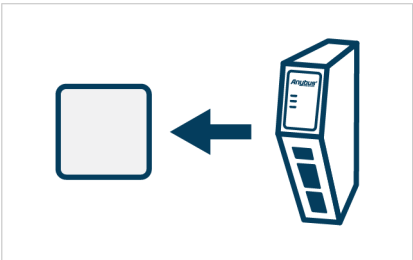


Process I/O data offset		Coil address		Holding register address	
HEX	DEC	HEX	DEC	HEX	DEC
0x0000 - 0x0001	0 - 1	0x0000 - 0x000F	0 - 15	0x0000	0
0x0002 - 0x0003	2 - 3	0x0010 - 0x001F	16 - 31	0x0001	1
...	...	...	...	...	...
0x05D8 - 0x05D9	1496 - 1497	0x2EC0 - 0x2ECF	11968 - 11983	0x02EC	748
0x05DA - 0x05DB	1498 - 1499	0x2ED0 - 0x2EDF	11983 - 11999	0x02ED	749

11.3.4. Legacy Address Mode I/O Data Mapping

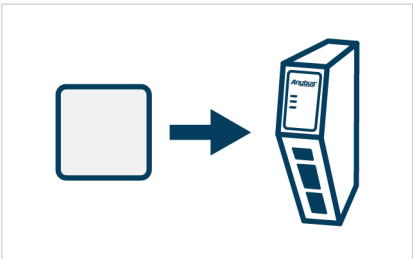
The following Modbus function codes are compatible with the Modbus **Use legacy mode** setting.

Input Data to Client/Server - Data to Modbus



Process I/O data offset		Discrete input address		Coil address		Input register address		Holding register address	
HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC
0x0000 - 0x0001	0 - 1	0x0000 - 0x000F	0 - 15	0x0000 - 0x000F	0 - 15	0x0000	0	0x0000	0
0x0002 - 0x0003	2 - 3	0x0010 - 0x001F	16 - 31	0x0010 - 0x001F	16 - 31	0x0001	1	0x0001	1
...	...	...	...	...	...	...	...	...	...
0x01FC - 0x01FD	508 - 509	0x0FE0 - 0x0FEF	4064 - 4079	0x0FE0 - 0x0FEF	4064 - 4079	0x00FE	254	0x00FE	254
0x01FE - 0x01FF	510 - 511	0x0FF0 - 0x0FFF	4080 - 4095	0x0FF0 - 0x0FFF	4080 - 4095	0x00FF	255	0x00FF	255

Output Data From Client/Server - Data From Modbus



Process data offset		Discrete input address		Coil address		Input register address		Holding register address	
HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC
0x0000 - 0x0001	0 - 1	0x4000 - 0x400F	16384 - 16399	0x4000 - 0x400F	16384 - 16399	0x0400	1024	0x0400	1024
0x0002 - 0x0003	2 - 3	0x4010 - 0x401F	16400 - 16415	0x4010 - 0x401F	16400 - 16415	0x0401	1025	0x0401	1025
...	...	...	...	...	...	...	...	...	...
0x01FC - 0x01FD	508 - 509	0x4FE0 - 0x4FEF	20448 - 20463	0x4FE0 - 0x4FEF	20448 - 20463	0x04FE	1278	0x04FE	1278
0x01FE - 0x01FF	510 - 511	0x4FF0 - 0x4FFF	20464 - 20479	0x4FF0 - 0x4FFF	20464 - 20479	0x04FF	1279	0x04FF	1279

See Also

[Legacy Mode For Anybus Address Mode | Modbus TCP \(Page 100\)](#)

12. Verify Operation

Before You Begin

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

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

12.1. Communicator Status Monitor

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

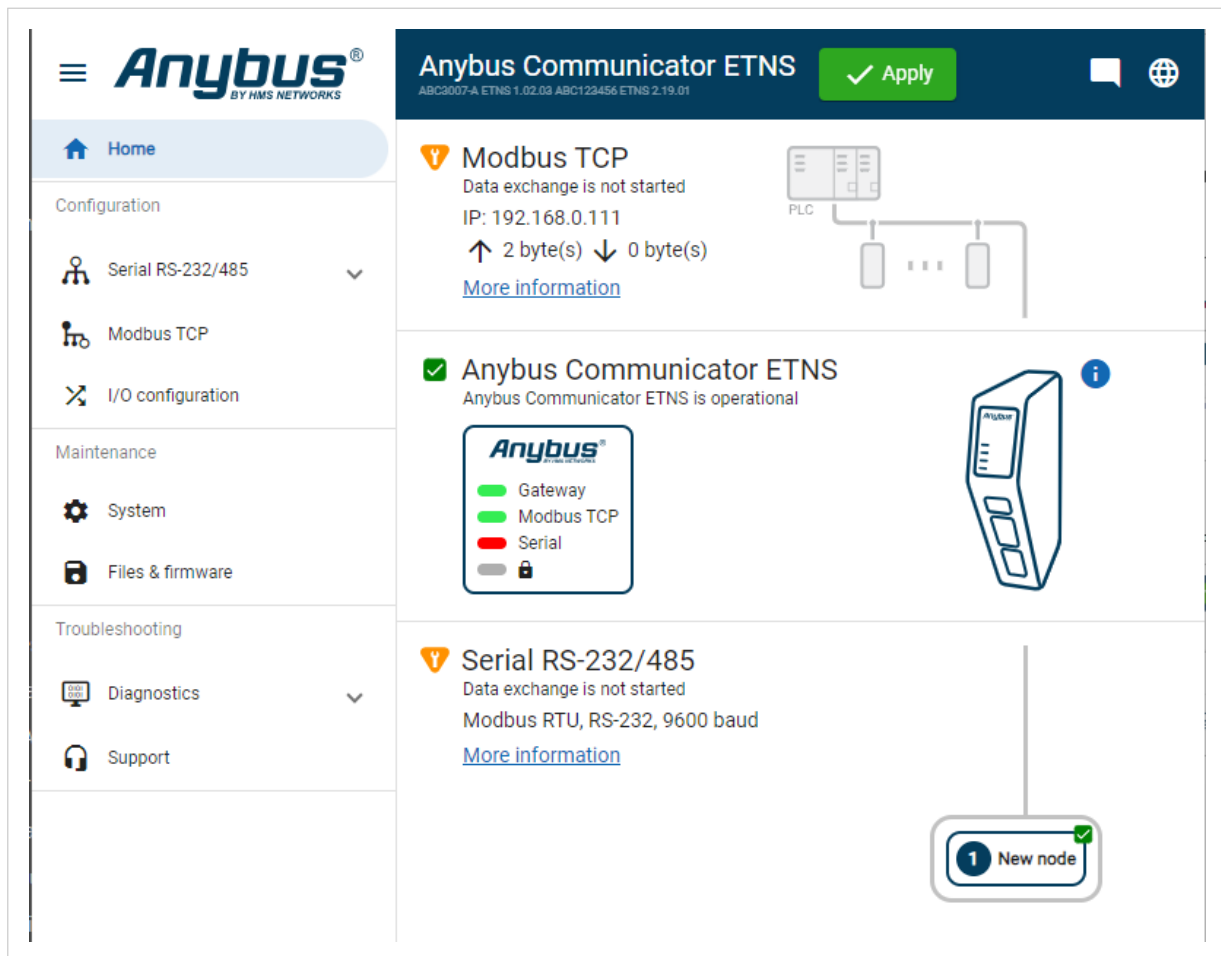


Figure 121. Home page

Gateway Status

Overview the Communicator LED indications remotely.

Node Status

Overview the status for each node added to the subnetwork.

Network Status and Settings

Overview communication status and the current networks settings.

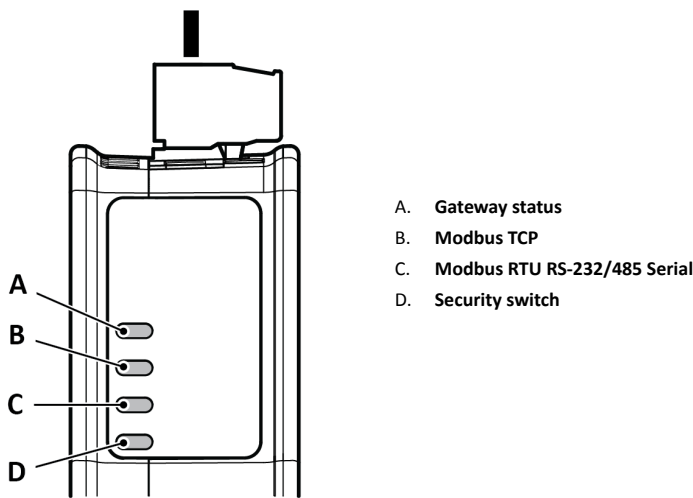
Status Symbols

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.

See Also

- [Connect To Modbus RTU Serial Network Via RS-232/485 \(Page 20\)](#)
- [Connect To Modbus TCP Network \(Page 22\)](#)
- [Connect To Power \(Page 24\)](#)
- [Communicator LED Indicators \(Page 135\)](#)
- [LED Status \(Page 172\)](#)
- [Firmware Upgrade Error Management \(Page 176\)](#)

12.2. Communicator LED Indicators



Gateway Status

Operation status	Description
Off	No power.
Green, flashing	Startup phase.
Green, solid	Operational.
Red, flashing	Invalid configuration.
Green/Red, flashing	Power up self-test. Firmware update. Firmware recovery.

Modbus RTU Serial

Operation status	Description
Off	No power. Subnetwork not running. Node is switched off via a control word.
Green, flashing	Running, one or more nodes are offline.
Green, solid	Running.
Red, flashing	All nodes are offline.

Modbus TCP

Operation status	Description
Off	No power or no IP address.
Green, flashing	Modbus TCP online, no messages received.
Green, solid	Modbus TCP online, at least one message received.
Red, solid	IP address conflict detected.
Red, flashing	Connection timeout.

Security Switch

Operation status	Description
Off	No power. Security switch is unlocked. Exception.
Green, solid	Security switch is locked.

Fatal Error and Exception Error

Fatal error: A fatal error causes the Communicator firmware application to crash in an uncontrolled manner.

Exception error: An exception error causes the Communicator to enter a controlled error state. The Communicator firmware application is still running.

LED	Fatal error	Exception error
A	Red, solid	Red, solid
B	Red, solid	Off
C	Red, solid	Off
D	Off	Off

See Also

[Connect To Modbus RTU Serial Network Via RS-232/485 \(Page 20\)](#)

[Connect To Modbus TCP Network \(Page 22\)](#)

[Connect To Power \(Page 24\)](#)

[Communicator Status Monitor \(Page 133\)](#)

[LED Status \(Page 172\)](#)

[Firmware Upgrade Error Management \(Page 176\)](#)

12.3. Ethernet RJ45 LED Indicators

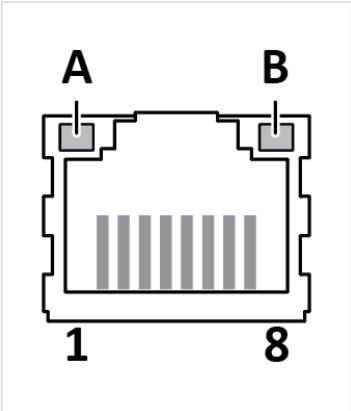


Figure 122. LED A. Activity LED B. Not used

LED A	Function
Off	No link (or no power)
Green	Link (100 Mbit/s) established
Green, flashing	Activity (100 Mbit/s)
Yellow	Link (10 Mbit/s) established
Yellow, flashing	Activity (10 Mbit/s)

LED B	Function
Off	Not used

13. Use Cases

13.1. Temperature Regulator - Modbus RTU Use Case

13.1.1. About the Use Case

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

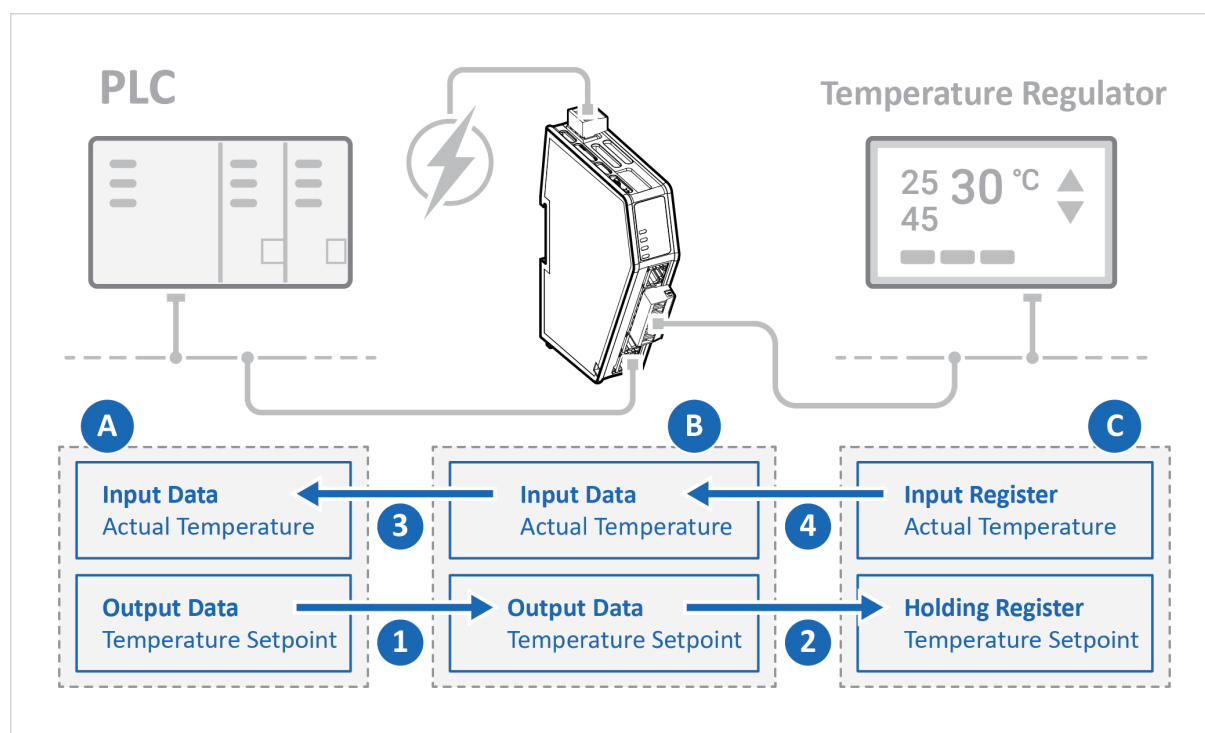


Figure 123. Temperature Regulator - Modbus RTU Use Case

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 Modbus TCP network (high level network).

13.1.2. 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 \(page 50\)](#).

13.1.3. Choose Serial Protocol Type

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

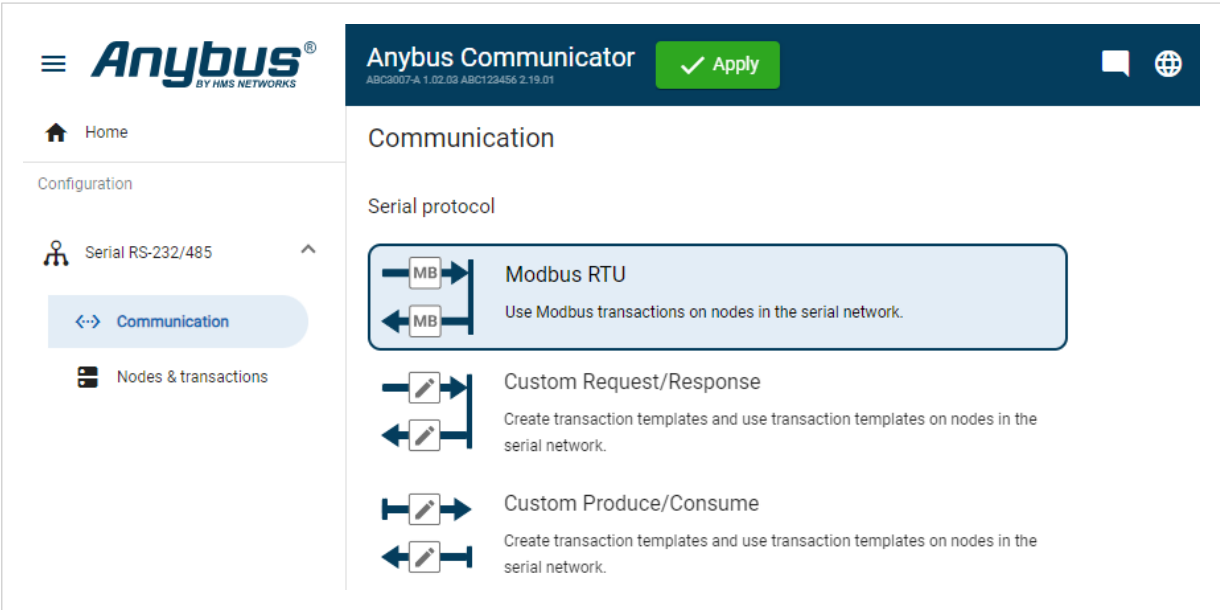


Figure 124. Communication page, **Modbus RTU**

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

13.1.4. Setup Serial Communication

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

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

Basic settings

Physical standard
RS232

Baud rate
19200 baud

Data bits
8 data bits

Parity
None

Stop bits
2 stop bit

Figure 125. Serial RS232/485, Basic settings

Table 3. 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

13.1.5. Setup the Node

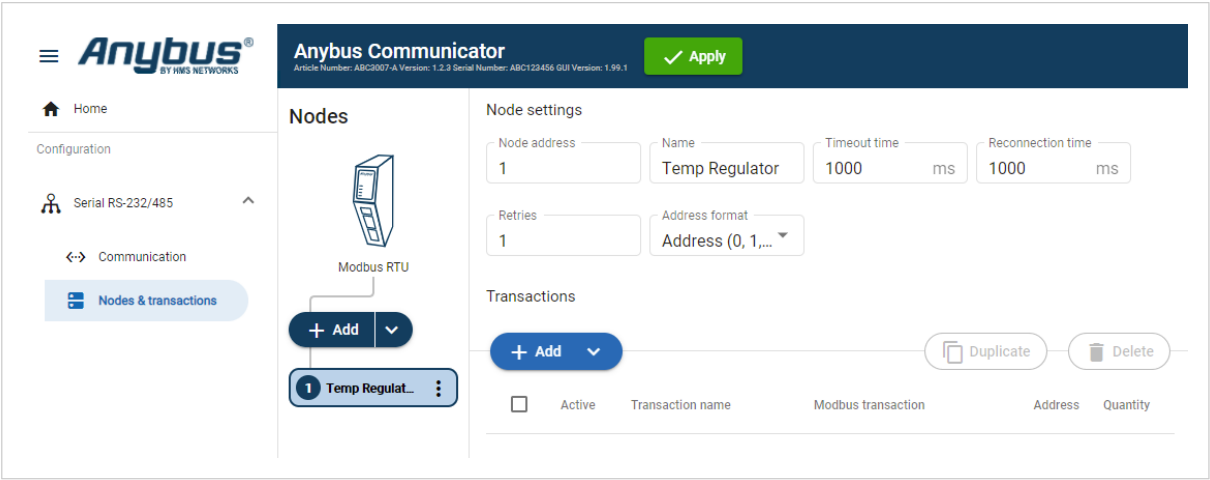


Figure 126. Add the Temperature Regulator node

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

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

13.1.6. 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 127. 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 4. 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

The screenshot shows the Anybus Communicator GUI. The 'Nodes' section on the left lists the 'Temp Regulator' node. The 'Transactions' section in the center shows a table with two transactions: 'Temp Setpoint' and 'Actual Temp'. The 'Actual Temp' transaction is selected, and its settings are shown in a sidebar on the right. The settings include: Transaction name: Actual Temp, Modbus transaction: Read Holding Registers (3), Address: 0, Quantity: 1, Update mode: Cyclically, Update time: 1000 ms, and Positive/Negative ack options.

Figure 128. 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 5. Actual temperature transaction settings:

Setting	Value	Description
Transaction 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.

13.1.7. Check the I/O Configuration

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

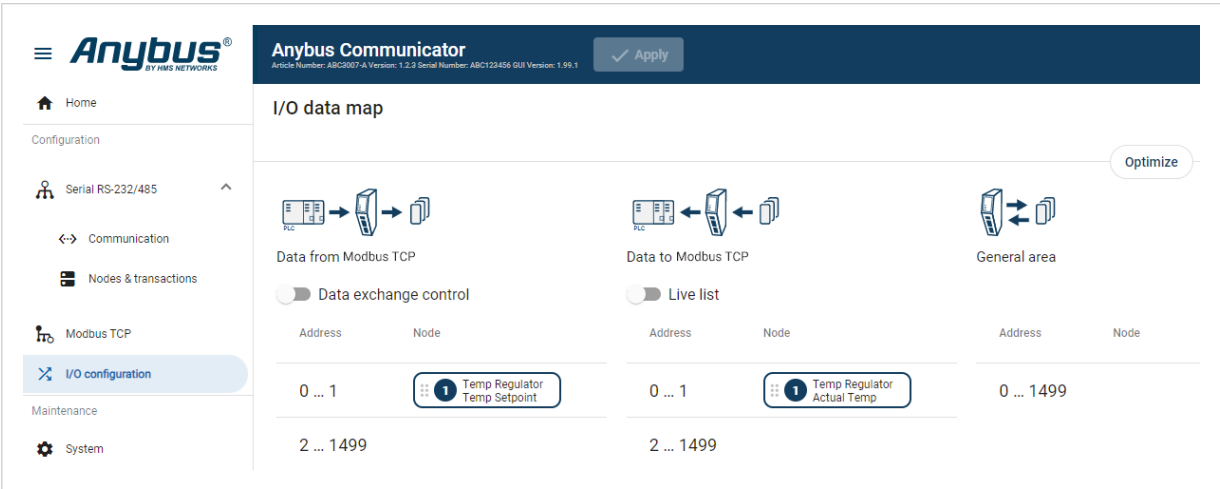


Figure 129. I/O configuration page

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

Address	Data from Modbus TCP
0-1	Actual temperature from the Temperature Regulator to Modbus TCP.

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

13.2.1. About the Use Case

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

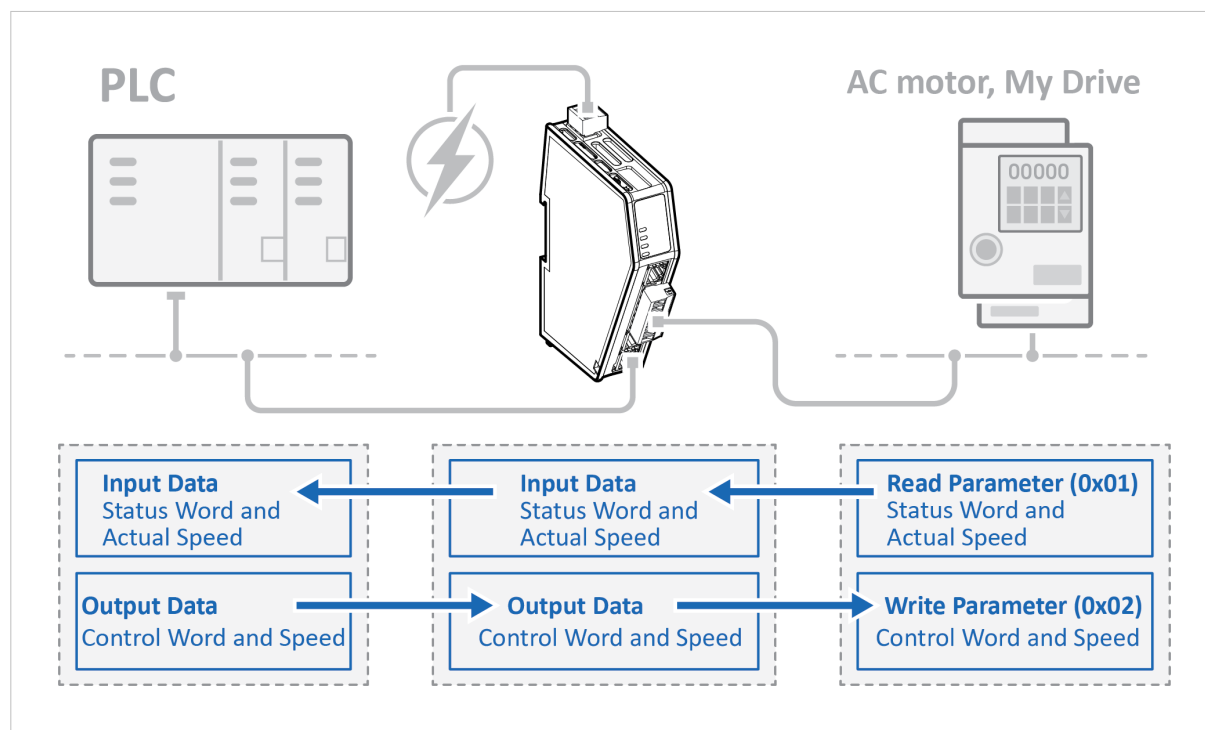


Figure 130. AC Motor Drive - Custom Request/Response Use Case

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 Modbus TCP 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 Modbus TCP network.
- Control word and speed from the Modbus TCP network to My Drive.

13.2.2. 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 \(page 50\)](#).

13.2.3. Choose Serial Protocol Type

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

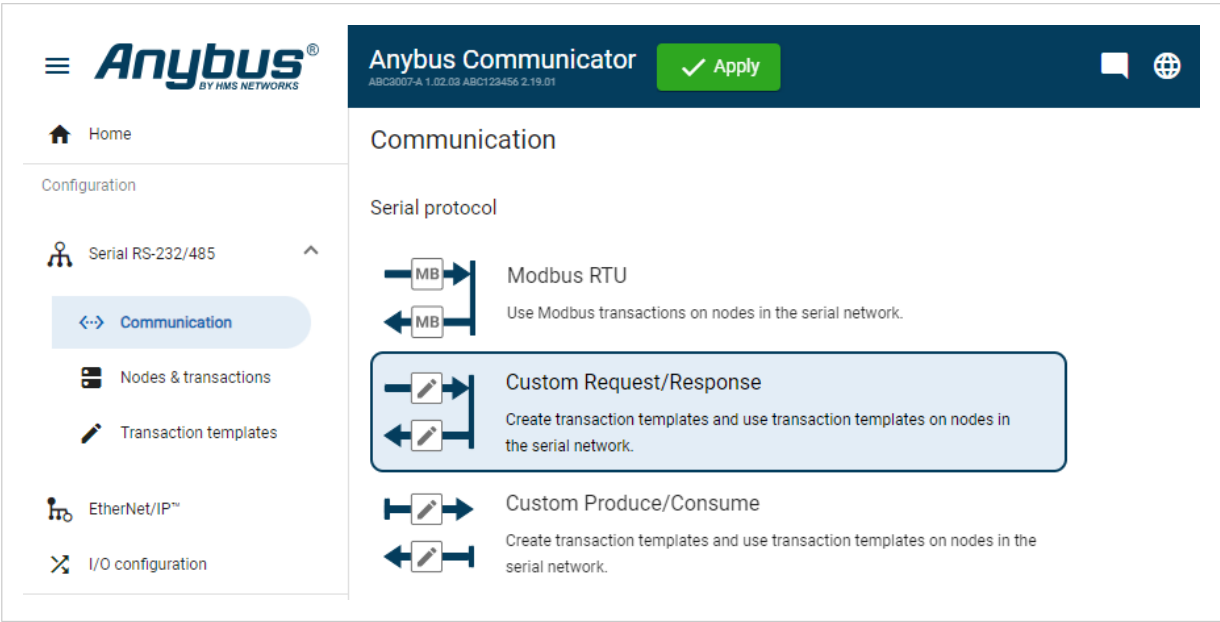


Figure 131. Communication page, Custom Request/Response

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

13.2.4. Setup Serial Communication

Set up the communication between the Communicator and My Drive.

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

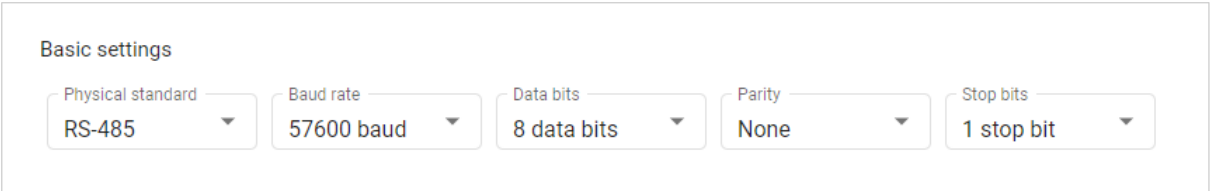


Figure 132. Serial RS232/485, Basic settings

Table 6. 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

13.2.5. 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 Modbus TCP to My Drive and status word and actual speed from the drive to Modbus TCP.

Create Read Parameter (0x01)

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

Anybus Communicator
Article Number: ABC2007-A Version: 1.2.2 Serial Number: ABC123456 GUI Version: 1.2.1

Transaction templates

+ Add ▼

Read parameter (0x01) ⋮

Transaction template settings

Transaction name
Read parameter (0x01)

Frame editor

Node add... Constant Limit Data Variable d... Checksum

Request

Byte offset	Field
0	Function code 1
1	Node ID
2 ... 3	Index 2
4	Sub index 1
5 ... 6	Checksum

Response

Byte offset	Field
0	Function code 1
1	Node ID
2 ... 3	Index 2
4	Sub index 1
5 ... 6	Data
7 ... 8	Checksum

Figure 133. 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 7. Request frame objects

Frame objects	Name	Bytes/Length	Type/Checksum type	Endianness	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 8. Response frame objects

Frame object	Name	Bytes/Length	Type/Checksum type	Endianness	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.

Anybus Communicator
Article Number: ABC3007-A Version: 1.2.3 Serial Number: ABC123456 GUI Version: 1.2.1 ✓ Apply

Transaction templates

+ Add ▼

Read parameter (0x01) ⋮

Write Parameter (0x02) ⋮

Transaction template settings

Transaction name
Write Parameter (0x02)

Frame editor

[A] Node add... [C] Constant [L] Limit [D] Data [V] Variable d... [Cs] Checksum

Request

Byte offset	Field
0	[C] Function code 1 ⋮
1	[A] Node ID ⋮
2 ... 3	[C] Index 1 ⋮
4	[C] Sub index 1 ⋮
5 ... 6	[D] Data ⋮
7 ... 8	[Cs] Checksum ⋮

Response

Byte offset	Field
0	[C] Function code 0 ⋮
1	[A] Node ID ⋮
2 ... 3	[C] Index 2 ⋮
4	[C] Sub index 1 ⋮
5 ... 6	[Cs] Checksum ⋮

Figure 134. 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 9. Request frame objects

Frame object	Name	Bytes/Length	Type/Checksum type	Endianness	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 10. Response frame objects

Frame objects	Name	Bytes	Type/Checksum type	Endianness	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

13.2.6. Setup Node and Transactions

The screenshot shows the 'Anybus Communicator' interface. On the left, a 'Nodes' sidebar lists '1 My Drive'. The main area is divided into 'Node settings' and 'Transactions'. The 'Node settings' section includes fields for Node address (1), Name (My Drive), Timeout time (1000 ms), Reconnection tL (10000 ms), and Retries (0). The 'Transactions' section shows a table with four transactions: 'Control Word' (Write Parameter (0x02), Index 0, Sub index 2), 'Speed' (Write Parameter (0x02), Index 0, Sub index 2), 'Status Word' (Read Parameter (0x01), Index 2, Sub index 0), and 'Actual Speed' (Read Parameter (0x01), Index 2, Sub index 0). On the right, a 'Request' and 'Response' section shows the configuration for the selected transaction, with Index 1 and Sub index 1.

Figure 135. 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 11. 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

13.2.7. Check the I/O Configuration

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



Figure 136. I/O configuration page

Table 12. Control word and speed from Modbus TCP to My Drive

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

Table 13. Status word and actual speed from My Drive to Modbus TCP

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

13.3. Barcode Scanner - Custom Produce/Consume Use Case

13.3.1. About the Use Case

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

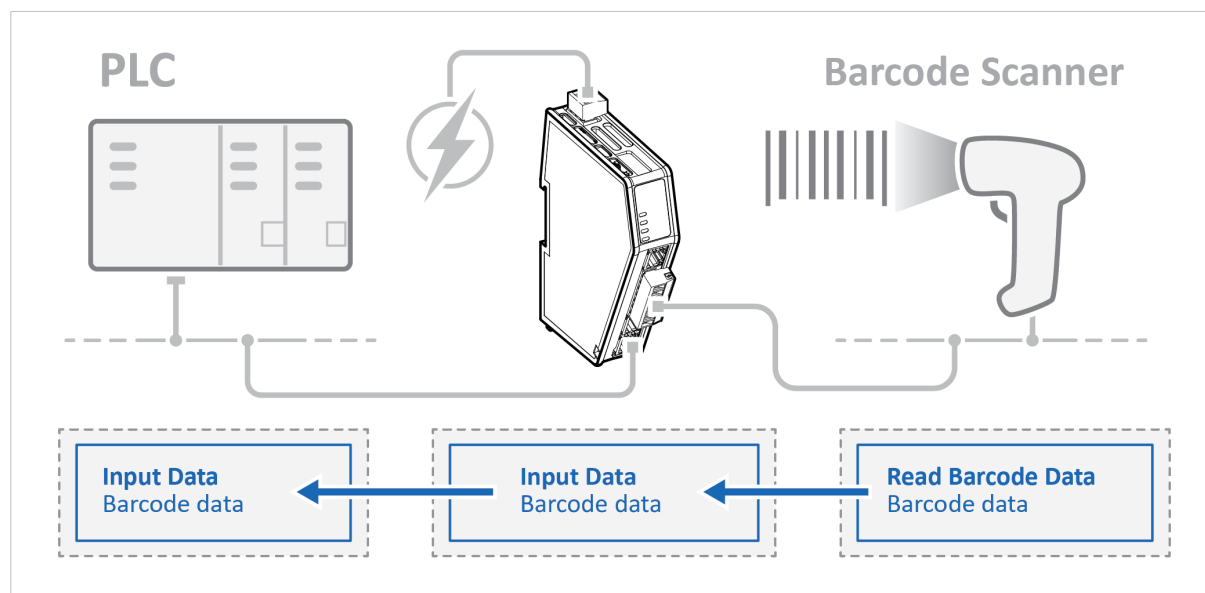


Figure 137. Barcode Scanner - Custom Produce/Consume Use Case

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 Modbus TCP network (high level network).

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

13.3.2. 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 \(page 50\)](#).

13.3.3. Choose Serial Protocol Type

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

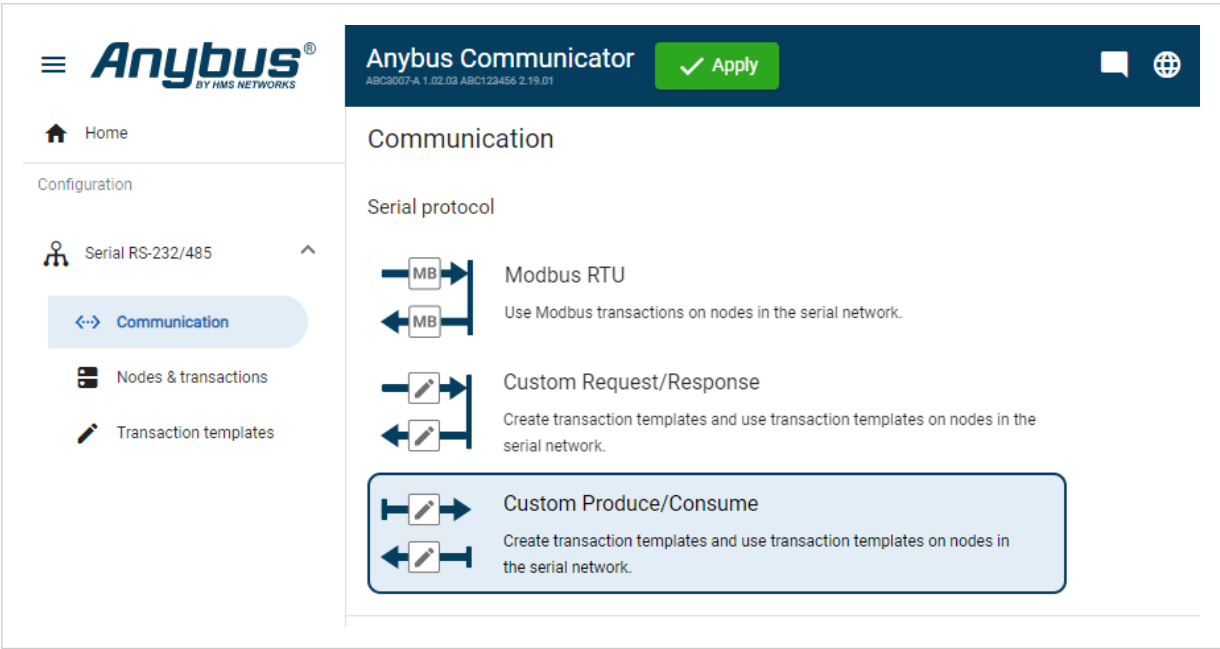


Figure 138. Communication page, Custom Produce/Consume

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

13.3.4. Setup Serial Communication

Set up the communication between the Communicator and Barcode Scanner.

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

Basic settings

Physical standard
RS-232

Baud rate
9600 baud

Data bits
8 data bits

Parity
None

Stop bits
1 stop bit

Figure 139. Serial RS232/485, Basic settings

Table 14. 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

13.3.5. 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.

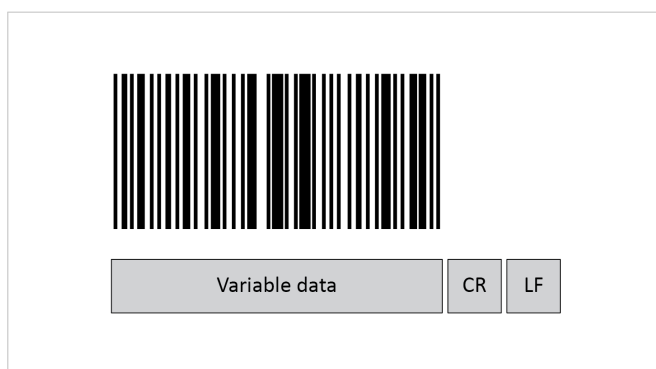


Figure 140. Barcode Variable data, CR and LF

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

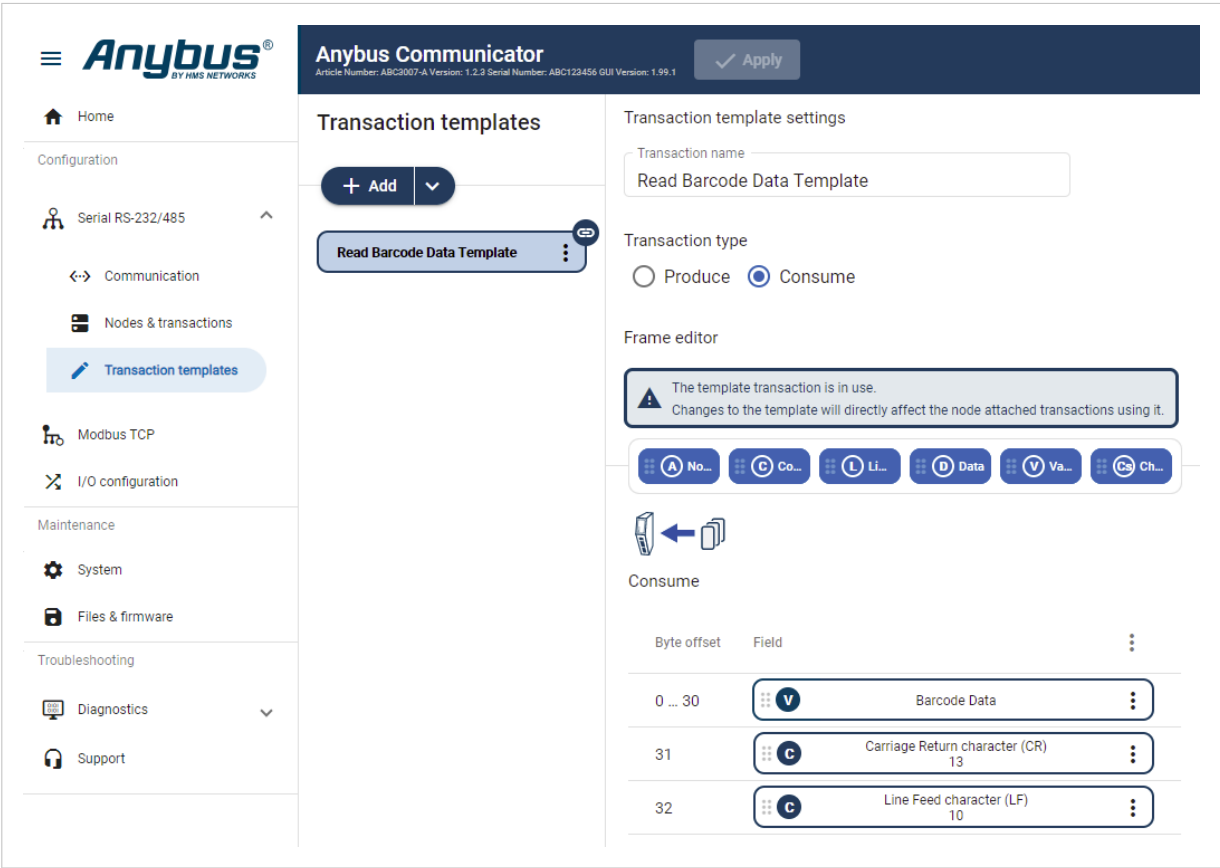


Figure 141. Read Barcode Data Parameter

- 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 15. 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

13.3.6. Setup Node and Transactions

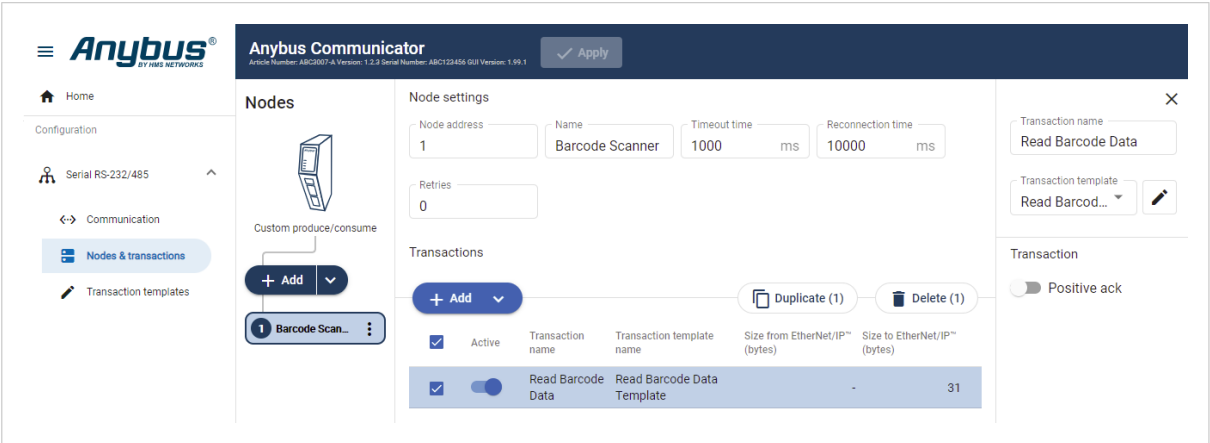


Figure 142. Node settings

- 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 16. The Barcode Scanner contains the following parameters

Transaction name	Transaction template
Read Barcode Data	Read Barcode Data Template

13.3.7. Check the I/O Configuration

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

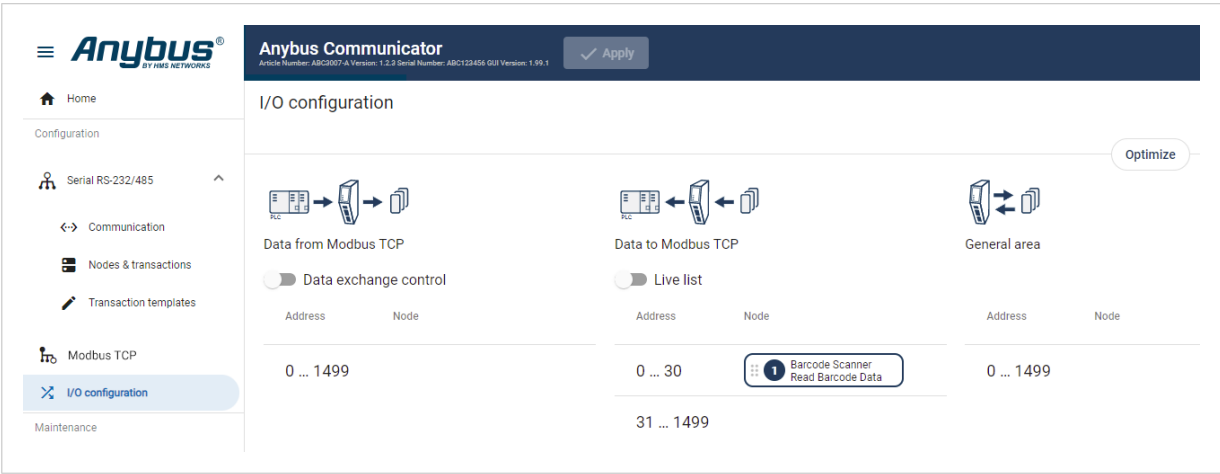


Figure 143. I/O configuration page

Table 17. Status word and actual speed from My Drive to Modbus TCP

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

14. Maintenance

14.1. Action on Fatal Error

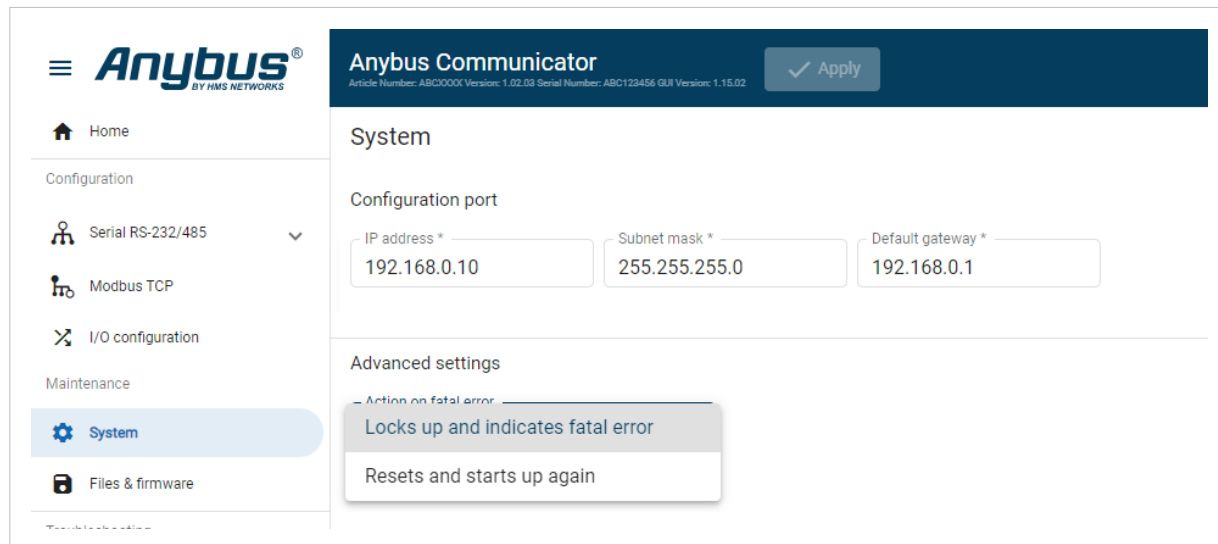


Figure 144. System page, Action on fatal error menu

A fatal error causes the Communicator firmware application to crash in an uncontrolled manner.

You can configure how the Communicator should behave if a fatal error occurs.

In the **Action on fatal error** menu, select one of the following settings:

- **Locks up and indicates fatal error:** Default setting, the Communicator locks up and the LED indicators indicate a fatal error.
- **Resets and starts up again:** The Communicator is rebooted to reset the system and return to normal operation.

14.2. Configuration Port IP Settings

On the **System** page you can change the IP address of the Communicator configuration port.

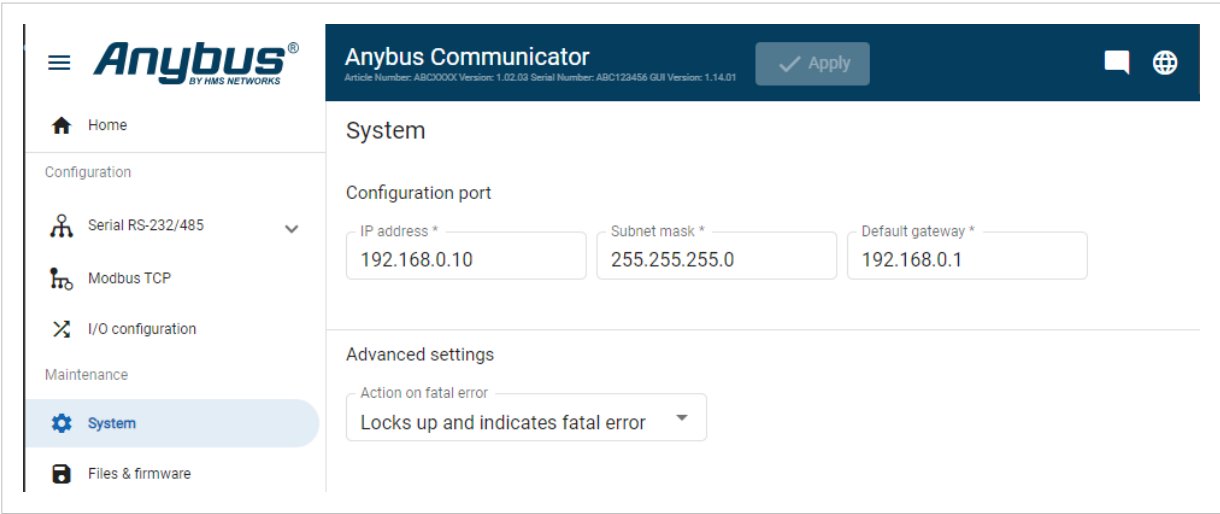


Figure 145. System page, Configuration port settings

Default Configuration Port IP settings

Setting	Default value
IP address	192.168.0.10
Subnet mask	255.255.255.0
Gateway	There is no default Gateway address.

14.3. Dashboard

The **Dashboard** allows you to create a customized view of the live data that the Communicator exchanges between device(s) on the connected networks.

Using a dashboard helps reduce complexity by presenting key data and actions in one place.

Enable the Dashboard

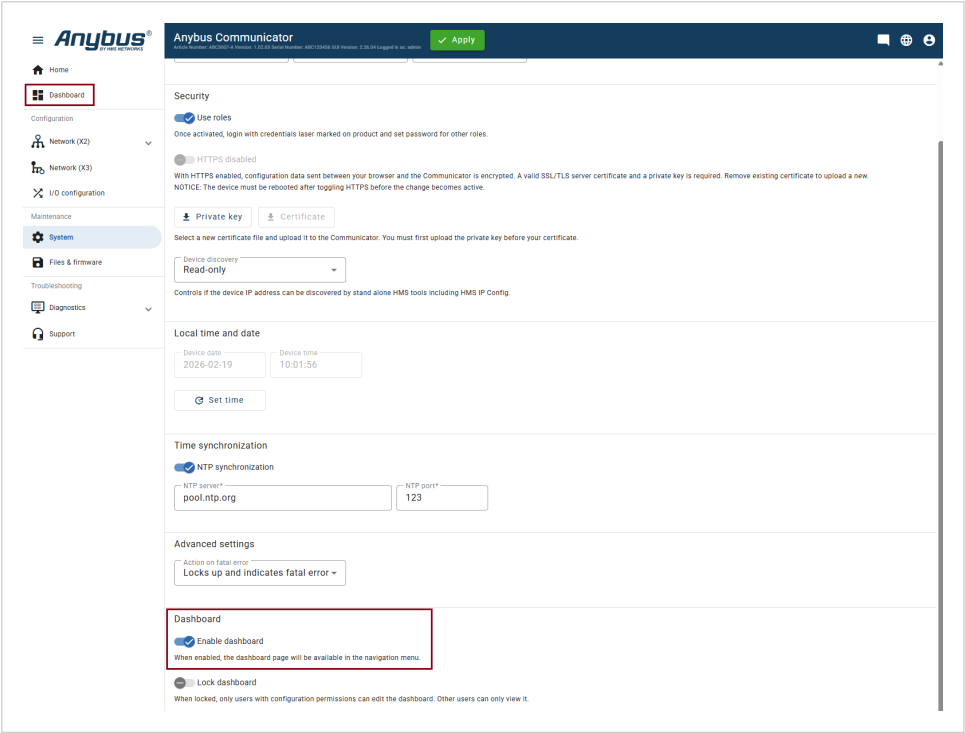


Figure 146. System page with Dashboard enabled and visible in the main menu

Procedure

1. On the **System** page, go to the **Dashboard** panel.
2. Click the **Enable dashboard** slide toggle.
3. When enabled, the **Dashboard** appears in the main menu.

To Setup the Dashboard

Before You Begin

You can add labels, data values, and images to create a simple visualization of the data exchange.

This helps you monitor values in real time and confirm that communication between networks is working as expected.

Dashboard Elements and Settings

Label

Use labels to add text descriptions on the dashboard, such as section names, group elements, or context for images or data.

Setting	Description
Label title	Enter the text you want to display in the label.

Setting	Description
Font	Select font type, size, and color.
Background	Choose a background color and opacity.
Border	Choose to show or hide the border. Adjust corner radius and border width if needed.

Image

Use images to create a simple graphical layout, for example to represent pumps, motors, or sensors.

Images can enhance clarity or support an HMI-style overview.

Data Value

Displays the current live value from a selected data point.

Setting		Description
Data	Byte address	Enter the starting address in the I/O data.
	Data type	Select UInt8 (1 byte), Int8 (1 byte), UInt16 (2 bytes), Int16 (2 bytes), UInt32 (4 bytes), Int32 (4 bytes), or Float32 (4 bytes)
	Direction	Choose whether the data direction is to or from the Communicator.
	Network	Select the network the data comes from, X2 or X3.
	Display format	Select Hex, Dec, or ASCII.
Value transformation	Formula	Apply a mathematical formula to transform the raw value. Use x to represent the raw value. Supports basic math operators (+, -, *, /) and parentheses. Example: $x * 0.1 + 10$
Label font	Label, Font, Font size, and Font color	Add a label to describe the value. Select the font type, size, and color.
Value font	Font, Font size, and Font color	Select the font type, size and color for the displayed value.
Background	Background color and opacity	Choose background color and opacity.
Border	Corner radius and Border width	Choose to show or hide border. Adjust corner radius and border width if needed.

Procedure

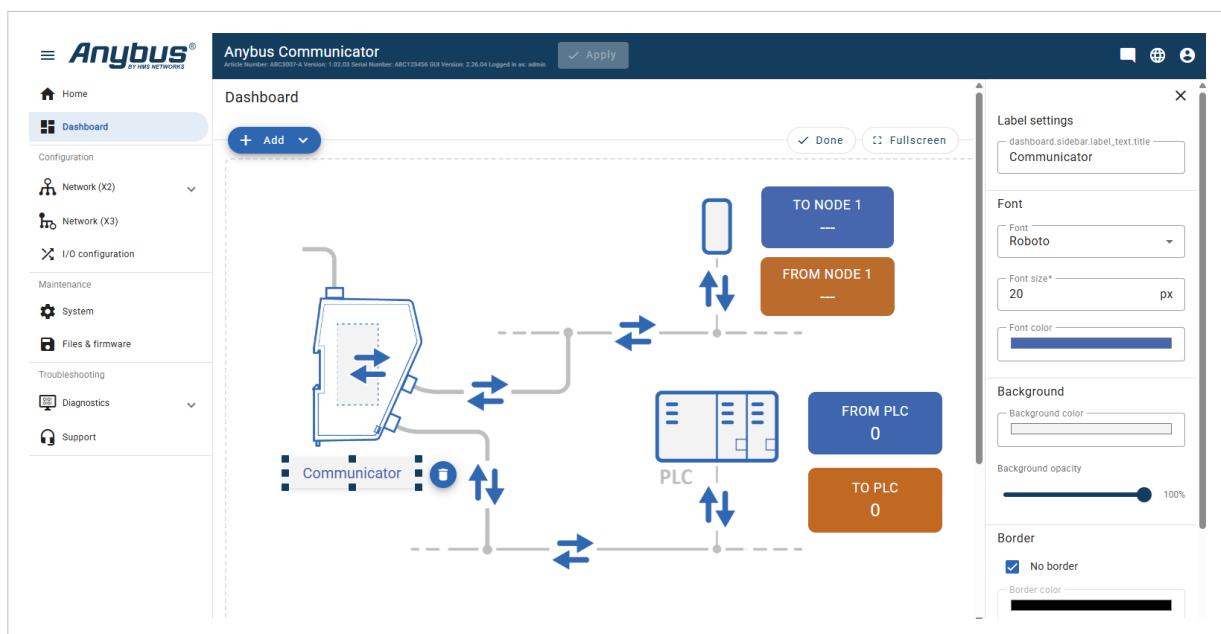


Figure 147. Dashboard example with added element

1. Navigate to the **Dashboard** page.
2. To enable edit mode, click **Edit**
3. To add an element, click **Add**, then select **Label**, **Image**, or **Data Value**.
The element is placed on the dashboard.
For **Labels** and **Data Values**, the settings panel is opened.
4. Configure the settings. See [Dashboard Elements and Settings \(page 160\)](#).
5. Drag the element to the preferred location if needed.
6. Resize the element if necessary.
7. To close edit mode, click **Done**.

Lock Dashboard

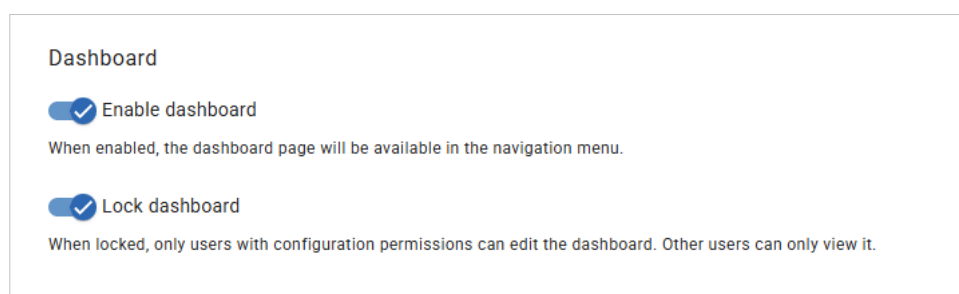


Figure 148. Dashboard enabled and locked

On the **System** page, you can lock the dashboard to prevent unauthorized changes.

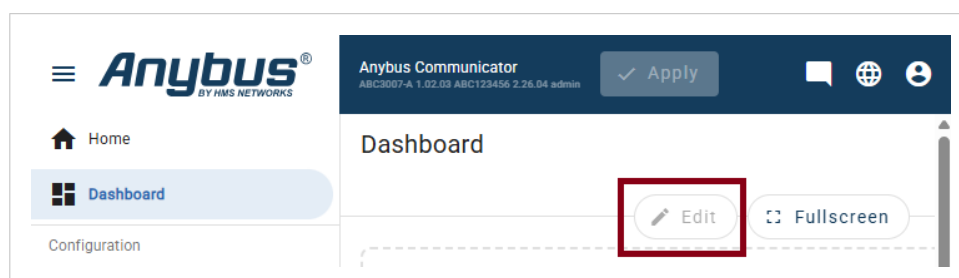


Figure 149. Dashboard locked

When locked:

- Only users with Administrator authority can edit the dashboard.
- All other users can view the dashboard but cannot modify it.

14.4. Configuration File Handling

14.4.1. Export Configuration

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

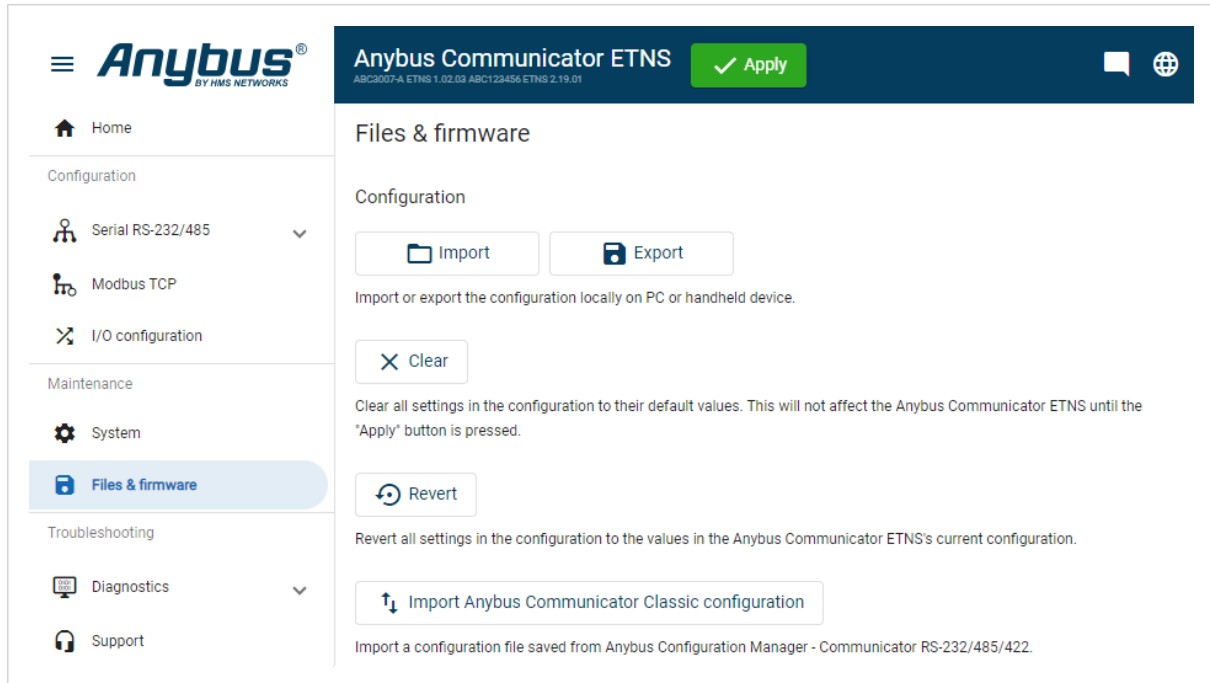


Figure 150. Files & firmware page

To export a configuration file:

In **Files & firmware**, click **Export**.

The configuration settings are stored in a .conf file and downloaded to your PC.

14.4.2. 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

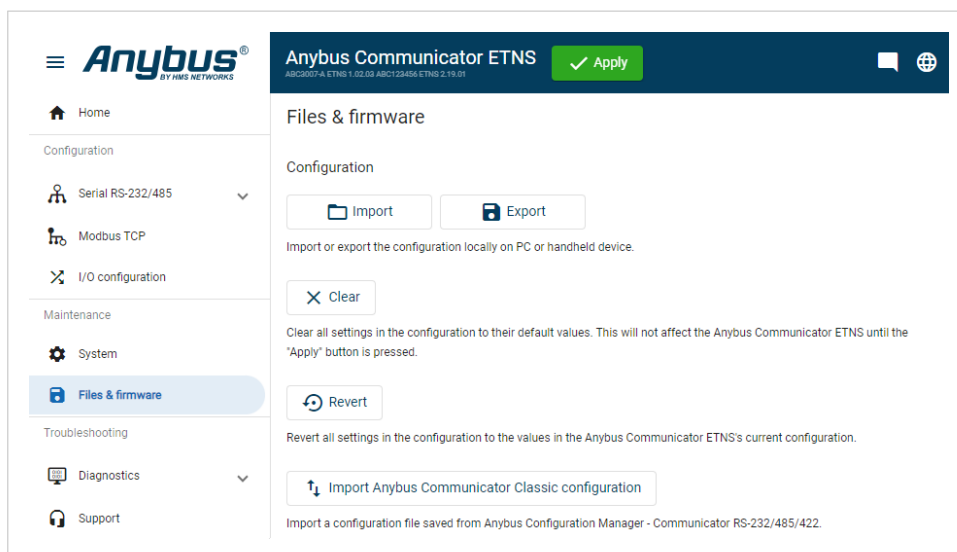


Figure 151. Files & firmware page

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 occur, a message about the mismatch appears.
7. To apply the settings, click **Apply** in the web-interface header, and follow the instructions.

14.5. Clear and Revert Configuration

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

Procedure

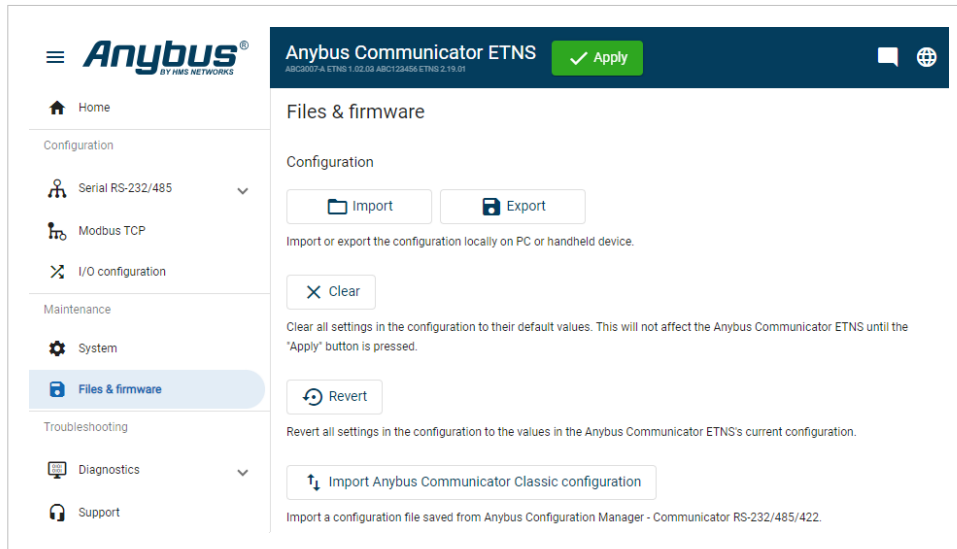


Figure 152. Files & firmware page

To Clear the Configuration

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

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.

To Revert the Configuration

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

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.

14.6. Firmware Management

14.6.1. View the Firmware Version

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

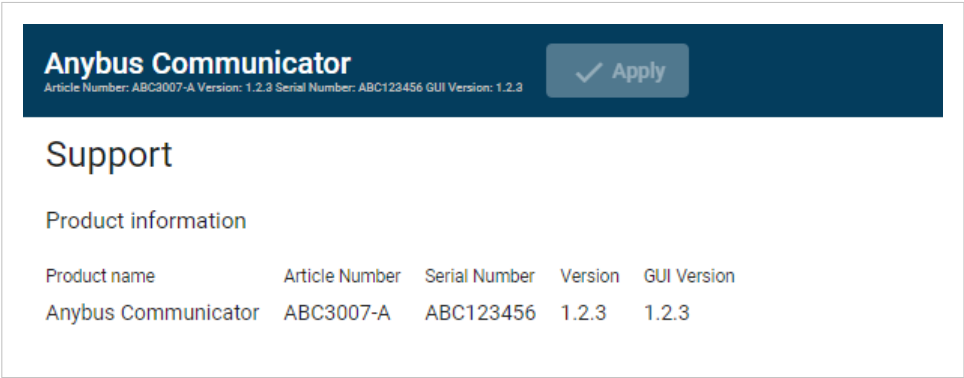


Figure 153. Support page, Product information example

14.6.2. 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 cannot be guaranteed.

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

14.6.3. Firmware File Validation

Before the firmware file is imported into the system, the firmware upgrade function performs 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.

14.6.4. 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 to disconnect the Communicator from the OT networks.

Procedure

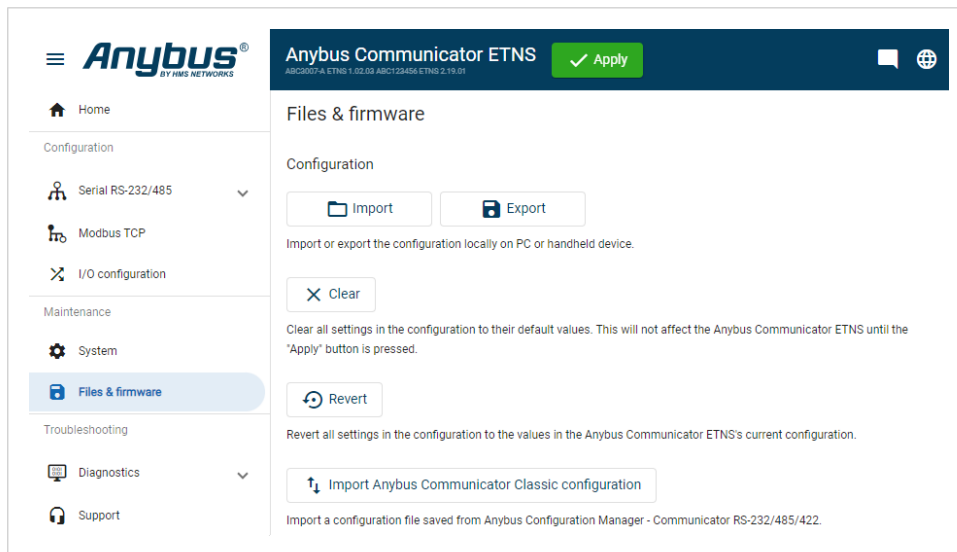


Figure 154. Files & firmware page

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 passes 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 appears.

14.7. Change Language

Default language is **English**.

To change the language of the Communicator built-in web interface:

1. In the Communicator built-in web-interface header, click the **Language** icon .

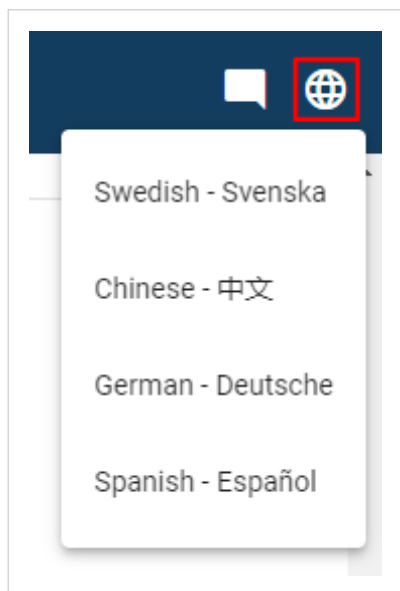


Figure 155. Language menu

2. Select a new language from the list.

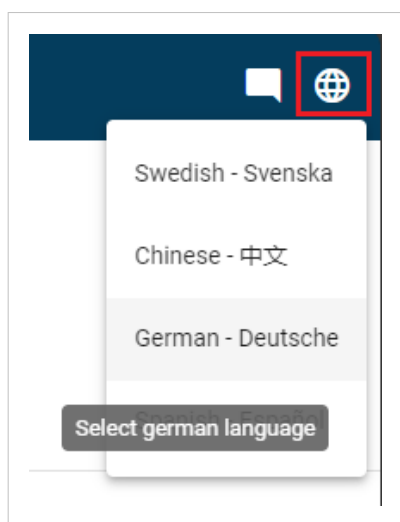


Figure 156. Example: Change language to German

The language change takes effect immediately.

15. Troubleshooting

15.1. Diagnostics

15.1.1. 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.

The screenshot shows the 'Anybus Communicator' interface for the 'Serial RS-232/485' section. The left sidebar contains navigation links: Home, Configuration (Serial RS-232/485, Modbus TCP, I/O configuration), Maintenance (System, Files & firmware), and Troubleshooting (Diagnostics, I/O data). The main area displays a table of data flow messages. The table has columns for Time (d:h:mm:ss.ms), Direction, and Data. The data is displayed in Hexadecimal format. The table shows several messages with timestamps and hex data. The interface also includes buttons for Start, Clear, Auto scroll, Hex, Dec, Ascii, and Export.

Time (d:h:mm:ss.ms)	Direction	Data
0:03:53:36.759	←	32 3c e8 41 97 f2 5b 3a 55 1c ba 42 33 e8 70 a8 bf 90 71 e4 31 ec b8 09 37 e8 08 30 9c 13 94 df d9 fc 5a f3 a4 c3 11 ba 5c bd 7a a7 f8 fa 17 d2 1c cb 76 7a a0 4e db eb 60 ad f0 24 1c a3 f0 d7 0a 4f 2b eb 80 89 29 75 cf a8 b1 cf 09 04 3c f2 3c 89 fd d7 d1 6e 27 92 2d c2 39 ea c2 0c fd 77 ee 3e 50 86
0:03:53:36.810	→	ce 0a 03 7a f4 46 e3 10 6d eb b6 2e f9 da 0f 02 ec 8a 51 c8 98 ec df 89 92 49 3d 13 a0 80 b7 f8 5e 84 58 e4 1d ca a3 e8 eb b5 40 41 25 ca 92 dd 73 de 7e e5 db d2 60 af 42 9c d0 e1 dc c8 dd bc b0 ec 02 15 ae 3e 7f 55 1b af a1 11 4d fa 8c 95 76 d2 bb af a8 ad 20 92 e1 5a ee 98 97 38 51 62 33 64 cf ec
0:03:53:36.859	←	80 d3 99 b2 a5 2f 06 77 8b ba 87 87 ff 5e fc db 1f 72 84 cd 26 5d f0 b6 a0 6f 96 c6 2a d2 c2 00 75 c8 49 7e 9c 81 e0 81 a4 bc d7 f3 33 1f 29 ac bf b4 49 39 14 07 ed b9 97 e2 52 40 0d 23 4c 28 75 c9 d6 90 2e 00 e2 ca 38 51 df ec 24 b0 ef 61 69 e4 15 5a 7d 25 33 87 69 ad 27 4e 35 06 2e 4b a1 48 9f 9f
0:03:53:36.910	→	d9 c5 b5 36 aa 7e e0 34 60 f3 21 5a b7 84 2d cd 75 81 75 45 0a 6b fc 2c 07 93 c9 4d 73 04 79 c8 66 18 35 9d 5a c6 bc 52 db 29 0f fa 21 23 a4 72 73 75 3b 34 91 d5 2f d8 59 91 1e ab 3a 6e 99 7f 86 97 2b 64 44 e8 e2 2d ca 3d 3a 46 bf 31 0f 96 f5 eb 96 84 1c 1d 67 c8 bf 05 22 90 49 49 1e 84 59 b6 be 1e
0:03:53:36.959	←	50 b6 7f d8 c1 41 60 f9 b3 e3 05 8d d0 2b 44 60 37 02 a4 ad 3f 7e a6 78 b4 e8 31 08 53 20 f6 45 e5 c5 06 65 b0 1c a5 97 f0 71 9f c1 eb 97 5a 27 6d 43 eb ca f6 07 e5 55 db 01 30 ff 95 56 a5 1e 69 8e 43 d3 c6 d2 ad 20 12 e9 87 26 90 56 ff e4 c5 35 31 3d 64 e9 3f 40 47 a5 c6 9f 24 71 3f 3a 49 cd e2 81

Figure 157. 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.

15.1.2. I/O Data

On the **Diagnostics**, **I/O data** page you can monitor how the data flow between the **Serial RS-232/485** side and the **Modbus TCP** side, including any configured endian conversions.

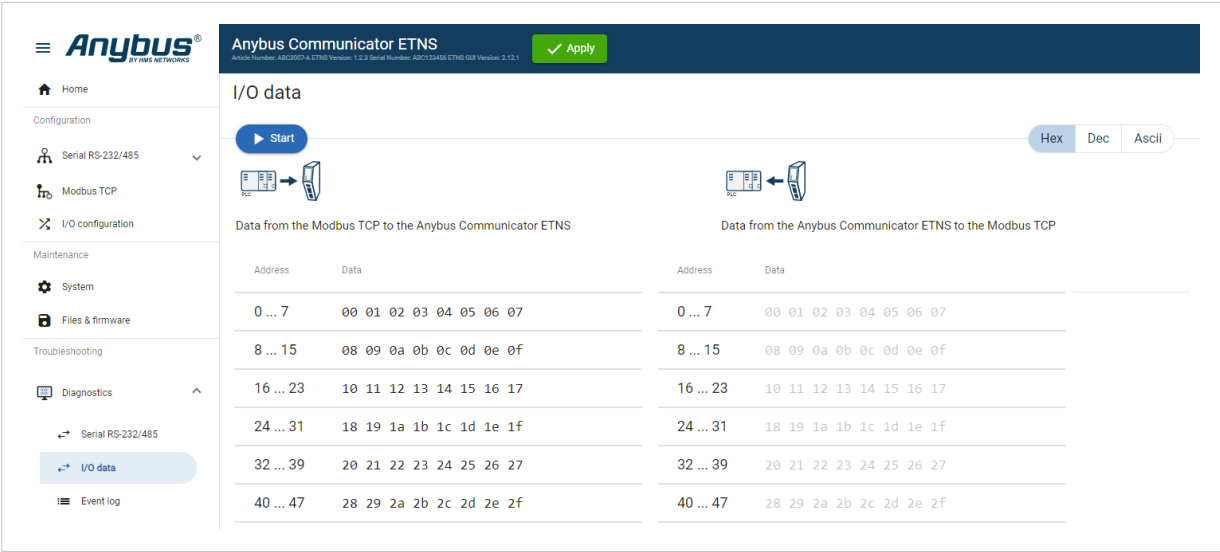


Figure 158. I/O data

I/O data is updated twice every second.

Select 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**.

15.1.3. Event Log

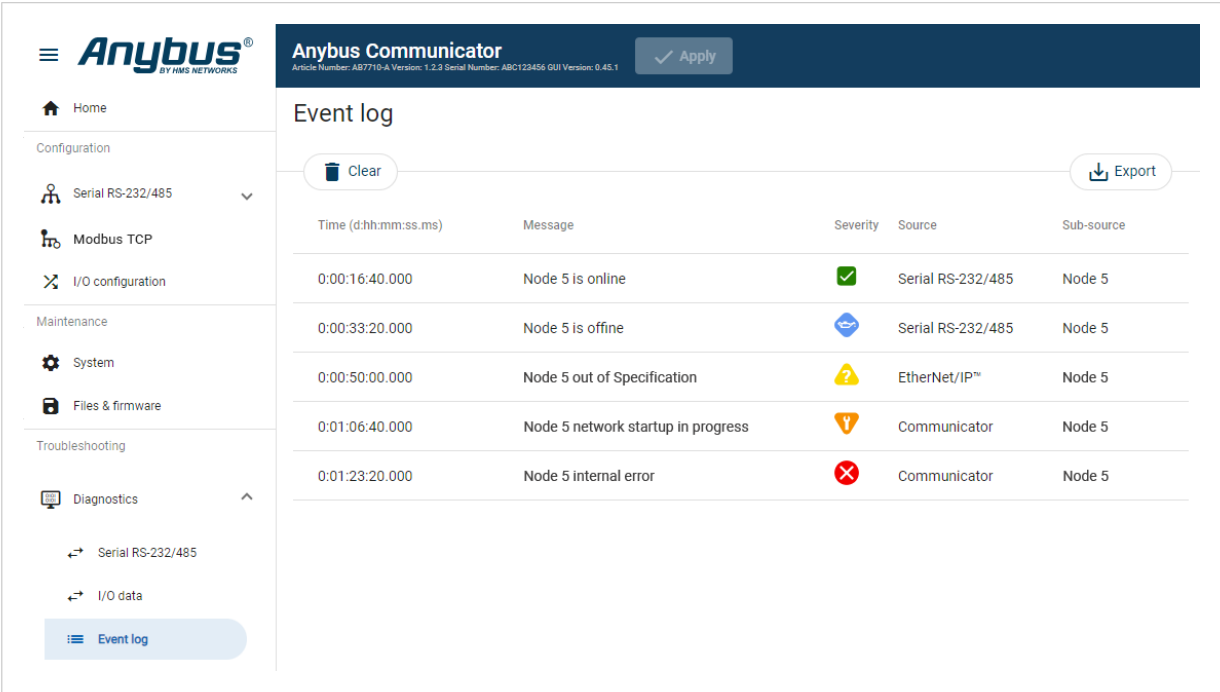


Figure 159. Event log page example

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 duration from when the Communicator Classic was powered on until the event occurred.
Message	A brief description of the event.
Severity	The severity of the event occurred.
Source	Communicator
	Serial RS-232/485
	Modbus TCP
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 12. 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**.

15.1.4. LED Status

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



Figure 160. Home page

See Also

[Connect To Modbus RTU Serial Network Via RS-232/485 \(Page 20\)](#)

[Connect To Modbus TCP Network \(Page 22\)](#)

[Connect To Power \(Page 24\)](#)

[Communicator Status Monitor \(Page 133\)](#)

[Communicator LED Indicators \(Page 135\)](#)

[Firmware Upgrade Error Management \(Page 176\)](#)

15.2. 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.

When 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.

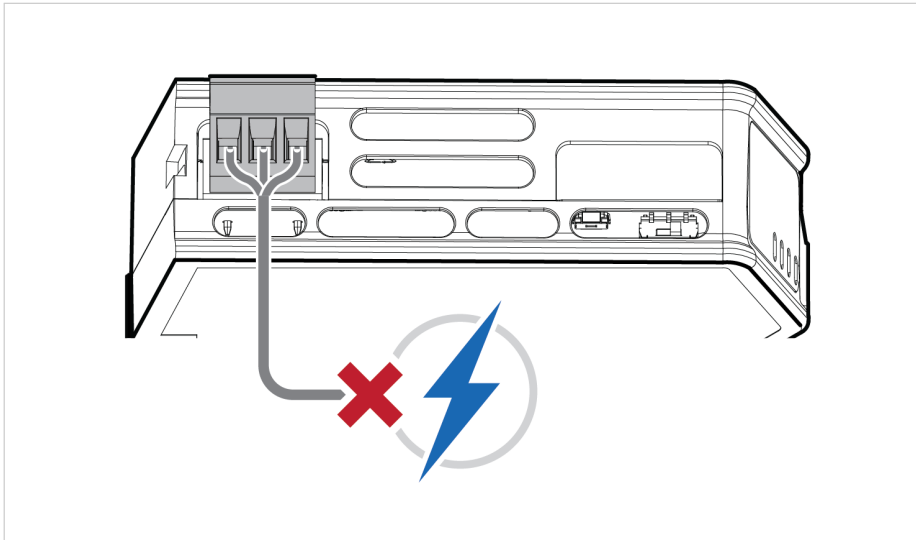


Figure 161. Disconnect power

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

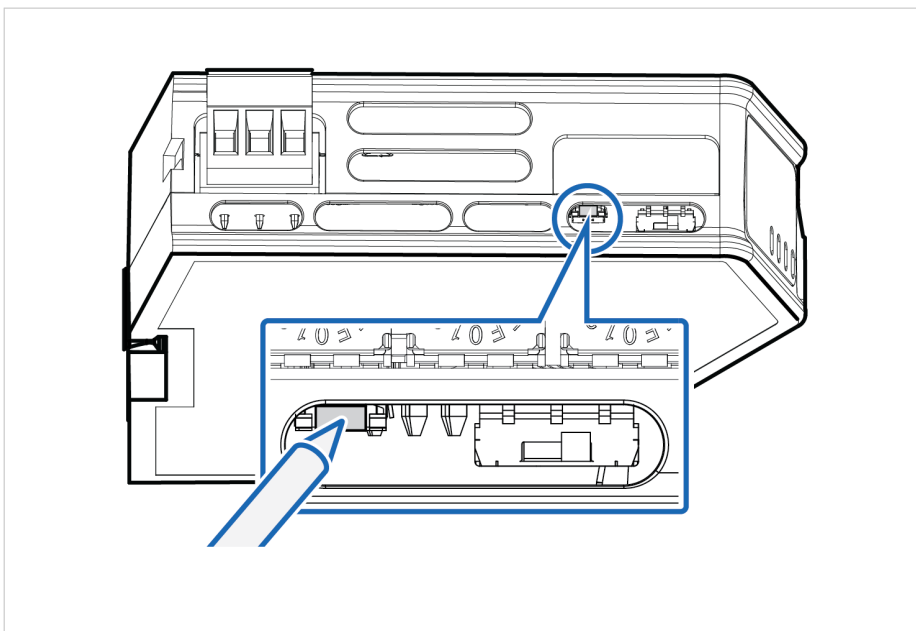


Figure 162. Press and hold **Reset** button

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

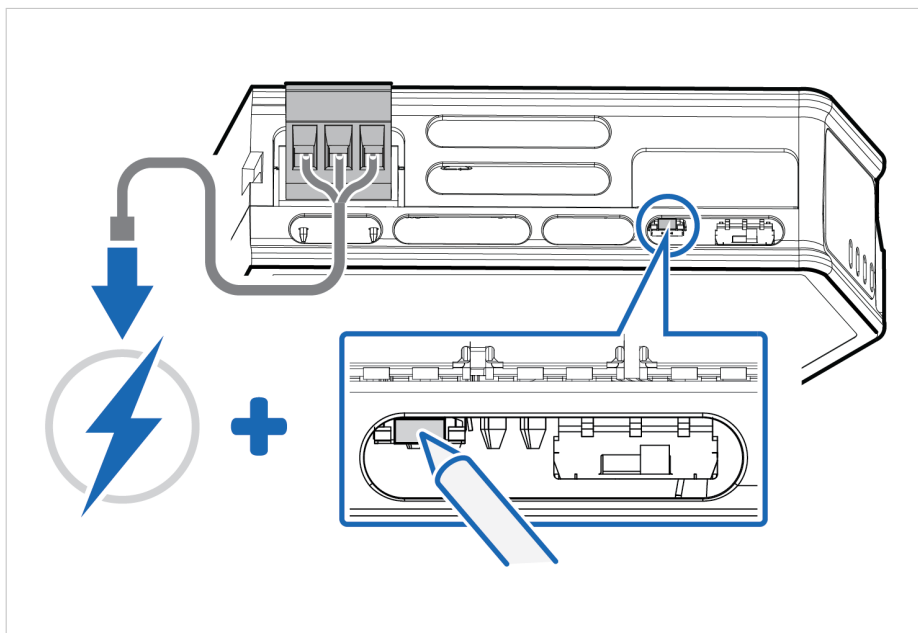


Figure 163. Hold **Reset** button and reconnect power

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 Communicator built-in web interface.

- 2. Navigate to the **Files & firmware** page and click **Revert**.

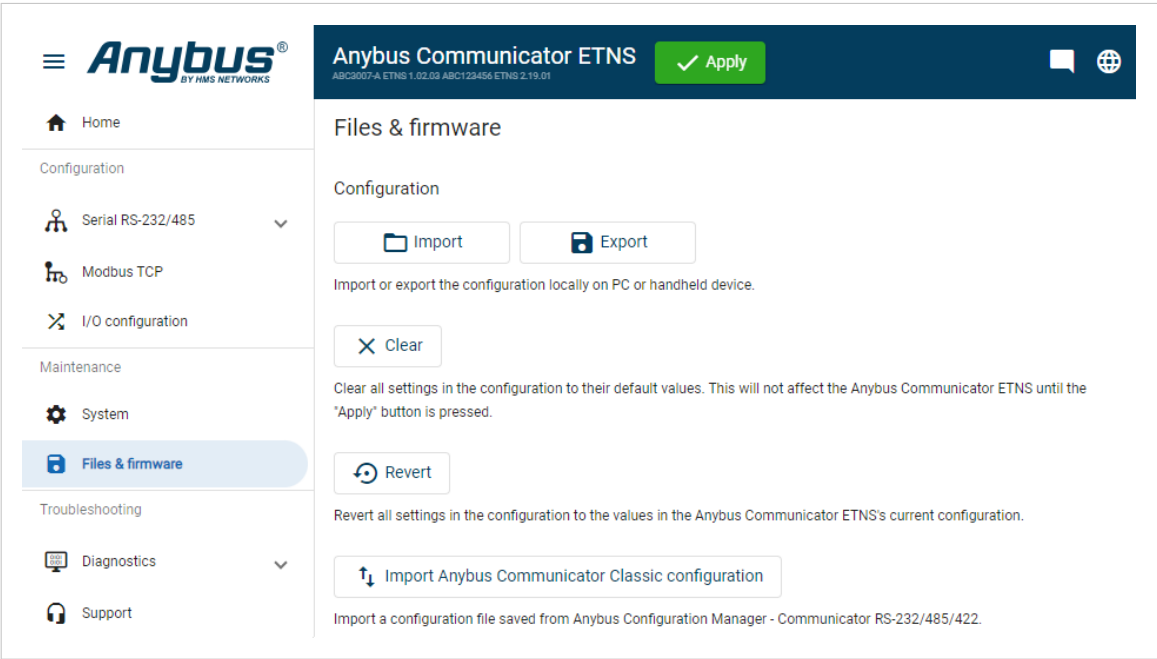


Figure 164. Files & firmware, Revert

15.3. Firmware Upgrade Error Management

Before You Begin

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

The firmware file from the latest update attempt remains in the flash memory, but it is not active.

Procedure

To complete the interrupted firmware update:

1. Disconnect the Communicator from power.

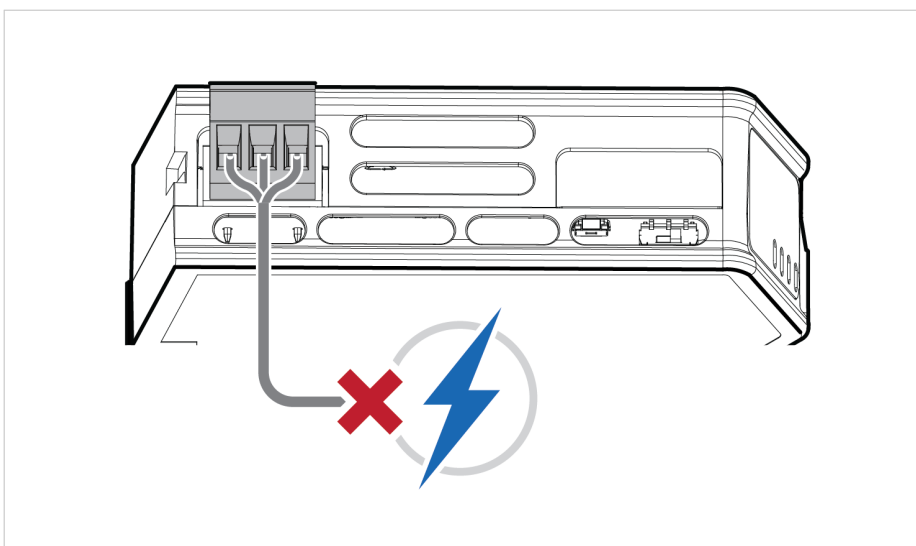


Figure 165. Disconnect power

2. Reconnect the Communicator to power.

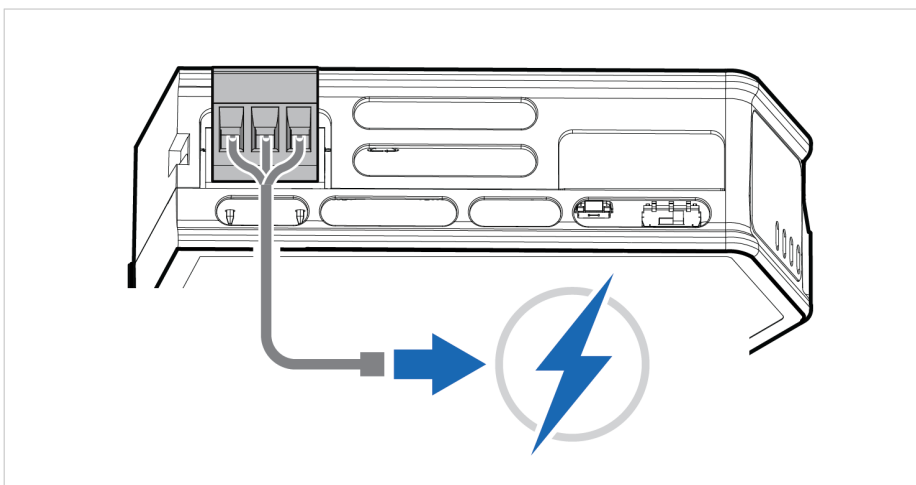


Figure 166. Reconnect power

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

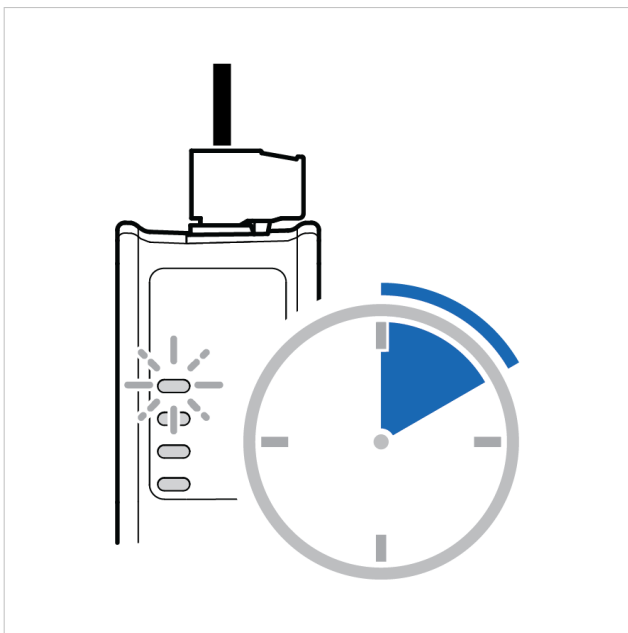


Figure 167. Firmware upgrade LED indication

Result

The Communicator recover and return to normal operation.

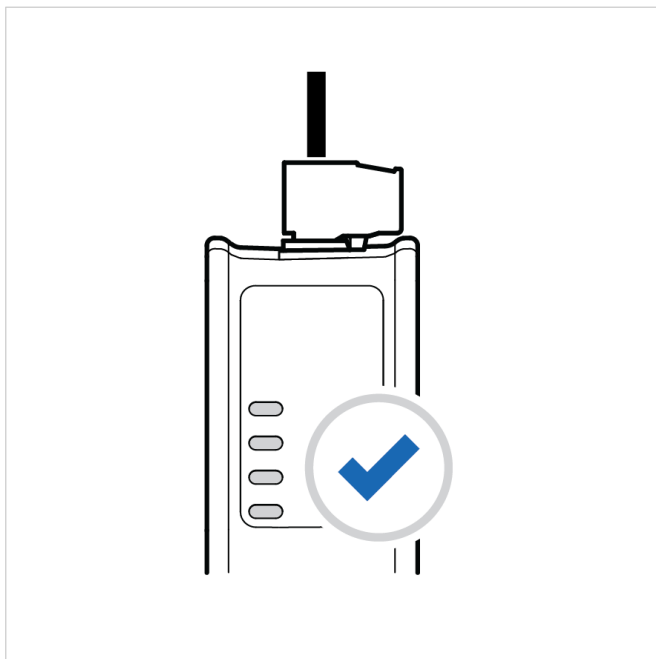


Figure 168. Recover and return to normal operation

See Also

[Connect To Modbus RTU Serial Network Via RS-232/485 \(Page 20\)](#)

[Connect To Modbus TCP Network \(Page 22\)](#)

[Connect To Power \(Page 24\)](#)

[Communicator Status Monitor \(Page 133\)](#)

[Communicator LED Indicators \(Page 135\)](#)

[LED Status \(Page 172\)](#)

15.4. Support

15.4.1. Support Package

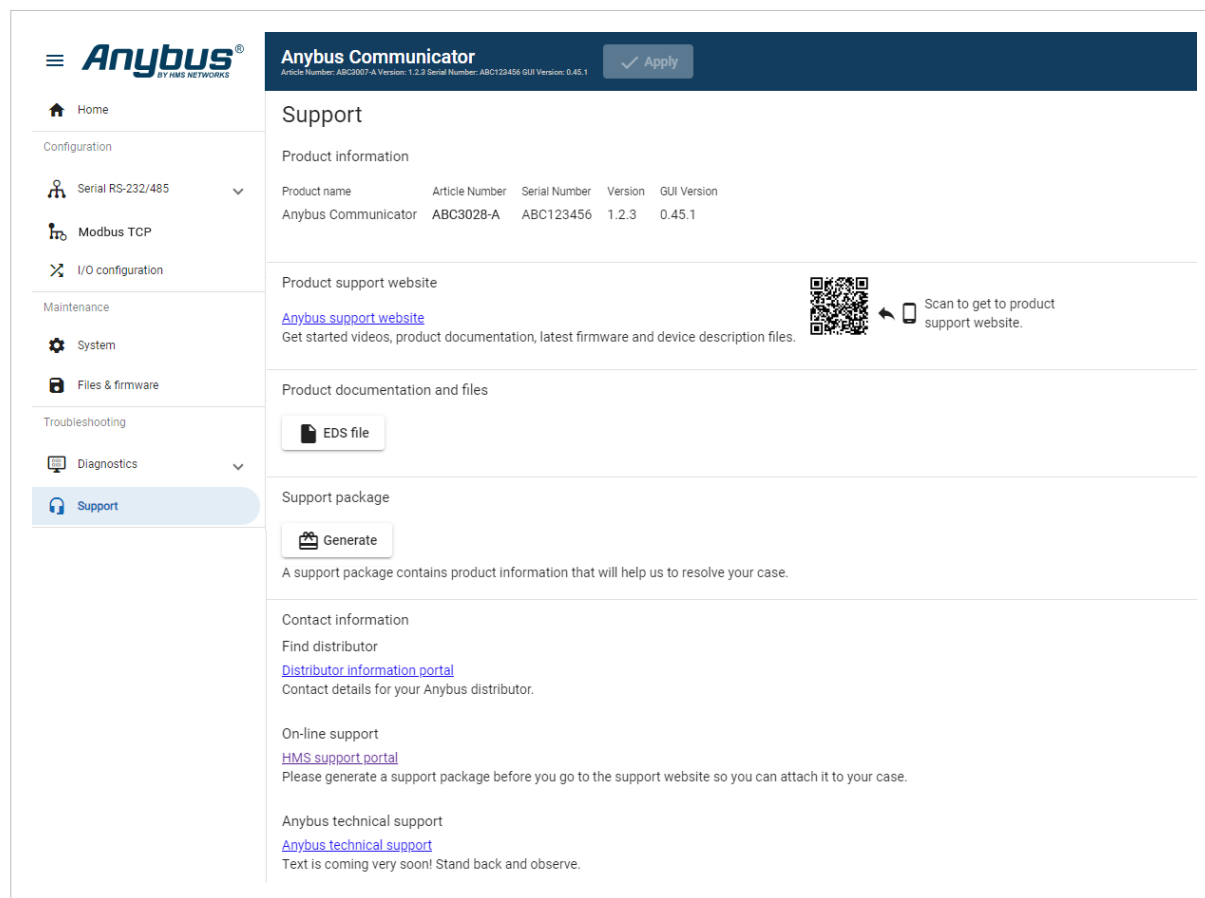


Figure 169. Support page example

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 is 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 **HMS Networks** home page, navigate to the **Support** main menu and click **Support portal**.
2. In the **Support portal**, create a support ticket and upload the support package.

16. Technical Data

For complete technical specifications and regulatory compliance information, please visit www.hms-networks.com.

16.1. Technical Specifications

Article identification	ABC3028
Configuration connector	RJ45
Communication connector	RJ45 x 2
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: 90 mA @ 24 V (2.2 W) Max: 3 W
Storage temperature	-40 to +85 °C
Operating temperature	-25 to +70 °C
Humidity	EN 60068-2-78: 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 (W x H x D) with connectors included
Mounting	DIN-rail

17. End Product Life Cycle

17.1. Secure Data Disposal

**IMPORTANT**

To avoid exposure of sensitive data, always perform a factory reset before decommissioning the equipment.

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

See [Reset to Factory Settings \(page 173\)](#).

18. Reference

18.1. 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.



TIP
If Register is used, add +1 to the Address value.

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.

18.2. Modbus Data Model

Discrete Inputs	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 organization website.

18.3. Modbus Function Codes

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

For more information refer to the Modbus organization website.

Nr	Function	Area	Function Code	Description
1	Read Coils	Coils	0x01	Read from 1 to 2000 contiguous status of coils in a remote device.
2	Read Discrete Inputs	Discrete inputs	0x02	Read from 1 to 2000 contiguous status of discrete inputs in a remote device.
3	Read Holding Registers	Holding registers	0x03	Read the contents of a contiguous block of holding registers in a remote device.
4	Read Input Registers	Input registers	0x04	Read from 1 to 125 contiguous input registers in a remote device.
5	Write Single Coil	Coils	0x05	Write a single output to ON or OFF in a remote device.
6	Write Single Register	Holding registers	0x06	Write a single holding register in a remote device.
15	Write Multiple Coils	Coils	0x0F	In a sequence of coils, force each coil to either ON or OFF in a remote device.
16	Write Multiple Registers	Holding registers	0x10	Write a block of contiguous registers in a remote device.
23	Read/Write Multiple Registers	Holding registers	0x17	Performs a combination of one read operation and one write operation. The write operation is performed before the read.
43/14	Read Device Identification	N/A	0x2B/0x0E	Read the vendor name, product code (article number), firmware version, vendor URL and product name.

18.4. Modbus 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.
03	Illegal Data Value	A value in the query data field is not allowed by the server. This indicates there is a problem with the structure of the rest of the request, such as an incorrect implied length.

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

For more information refer to the Modbus organization website.

18.5. 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

18.6. RS232/RS485 Electrical Connection

18.6.1. RS485 Typical Connection



IMPORTANT

The Communicator has a typical RS485 connection and there are no internal terminations.

To ensure signal integrity and prevent signal reflections on the bus, add termination resistors externally at both ends of the RS485 transmission line.

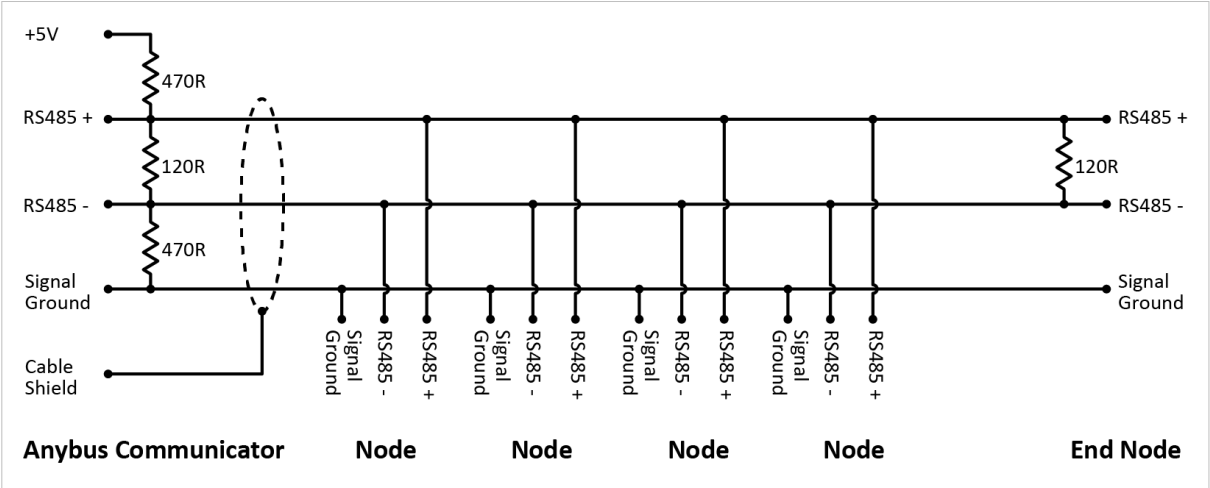


Figure 170. RS485

18.6.2. RS232 Typical Connection

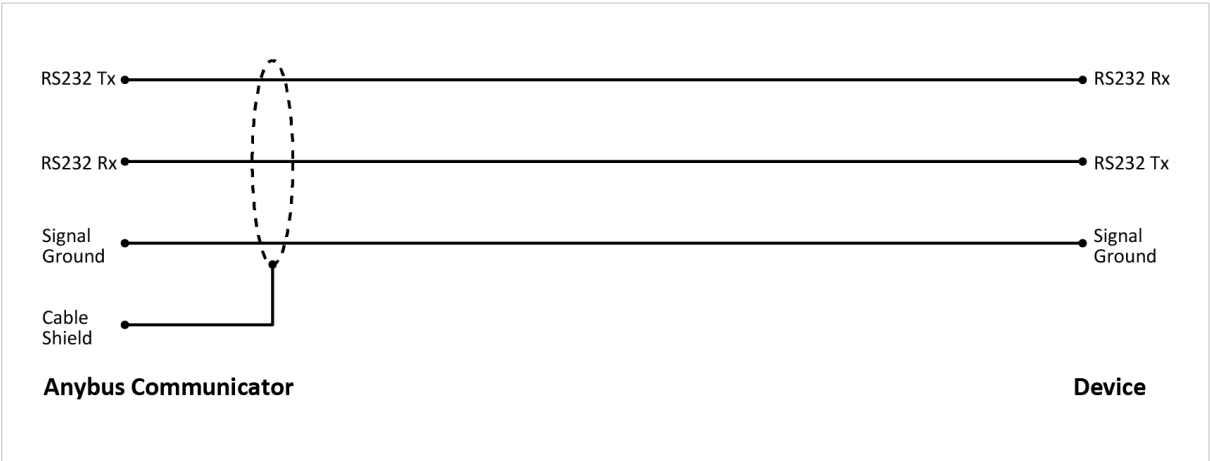


Figure 171. RS232