MPC-311
User Manual

01/2025

Table of contents

1. Abbreviations 4
2. Preface........ 5
2.1 Symbols 5
2.2 Safety instructions 5
2.3 Connecting to device 6
3. Program modules 8
3.1 Archives 8
3.1.1 Events archive 8
3.1.2 Diagnostic archive 9
3.1.3 User-defined archive........ 11
3.2 TCP modules 13
3.2.1 FTP server 13
3.2.2 FTP client 14
3.3 MQTT client 18
3.3.1 MQTT client configuration 18
3.3.2 Event messages configuration 19
3.3.3 Report data file configuration 20
3.4 TCP/IP connection table........ 21
3.5 Routing TCP/IP - serial (request/answer) 22
3.6 Routing TCP/IP - serial (transparent) 23
3.7 Router 4G/3G/GPRS<->Ethernet 26
3.8 Gateway Modbus TCP<->Modbus RTU 28
4. Hardware 31
4.1 microSD 31
4.2 Time settings 31
4.3 Ethernet configuration........ 31
4.4 4G/3G/GPRS configuration 33
4.5 Serial ports 35
5. Virtual interfaces 38
5.1 Virtual COM clients 38
5.2 Modbus RTU clients 40
5.3 Modbus RTU servers 42
6. Meters........ 44
6.1 M-Bus meters 44
6.2 Modbus devices........ 47
6.2.1 Modbus RTU devices........ 47
6.2.2 Modbus TCP devices........ 50

7. Manufacturer’s warranty 52

1. Abbreviations

Xn – A number representing a socket. This information is provided for the manufacturer's purposes and is used in data schemas and connection diagrams.
GSM – Global Standard for Mobile Communications. This interface is designed for remote connections and bidirectional data transfer over the Global Standard Mobile network.
GPRS – A packet-oriented mobile data service on the 2G and 4G/3G cellular communication systems' global system for mobile communications (GSM).
IP address – An Internet Protocol (IP) address is a numerical label assigned to devices participating in a network that uses the Internet Protocol for communication between its nodes.
TCP/IP – Transmission Control Protocol, used for communication between computers, serves as the standard for transmitting data over networks and as the basis for standard Internet protocols.
UART – A Universal Asynchronous Receiver/Transmitter is a type of "asynchronous receiver/transmitter," a part of computer hardware that translates data between parallel and serial forms. UARTs are commonly used in conjunction with communication standards such as EIA RS-232, RS-422, or RS-485. Records (UARTx) on top of the enclosure are also used as the serial interface number.
GND – Ground wire contact.
RS232 – The traditional name for a series of standards for serial binary single-ended data and control signals connecting a DTE (Data Terminal Equipment) and a DCE (Data Circuit-terminating Equipment). It is commonly used in computer serial ports. The standard defines the electrical characteristics and timing of signals, the meaning of signals, and the physical size and pin-out of connectors. RS232 interfaces are prepared for the connection of peripheral devices (e.g., energy meters, controllers, machines, etc.).
TD – Contact for the transfer data wire of the RS232 socket.
RD – Contact for the read data wire of the RS232 socket.
DTR – Contact for the Data Transmit Ready wire of the RS232 socket.
RS485 – A standard defining the electrical characteristics of drivers and receivers for use in balanced digital multipoint systems. Published by the ANSI Telecommunications Industry Association/Electronic Industries Alliance (TIA/EIA), digital communications networks implementing the EIA-485 standard can be used effectively over long distances and in electrically noisy environments. Multiple receivers may be connected to such a network in a linear, multi-drop configuration. RS485 interfaces are prepared for the connection of peripheral devices (e.g., energy meters, controllers, machines, etc.).
A+ – Contact for the positive wire of the RS485 socket.
B- – Contact for the negative wire of the RS485 socket.
USB – Universal Serial Bus is an industry standard that defines the cables, connectors, and protocols used for connection, communication, and power supply between computers and electronic devices.
M-Bus – A European standard (EN 13757-2 physical and link layer, EN 13757-3 application layer) for the remote reading of gas or electricity meters. The M-Bus interface is made for communication over two wires, making it very cost-effective.
MBUS+ – Contact for the M-Bus positive wire.
MBUS- – Contact for the M-Bus negative wire.
Socket – An endpoint of a bidirectional inter-process communication flow across an Internet Protocol-based computer network, such as the Internet.
Data – Contact for the data wire.
Req – Contact for the request wire.
CL+ – Contact for the current loop positive wire.
CL- – Contact for the current loop negative wire.
Status – Device status indicating LED.
Uoutput – Status of power for the external device indicating LED.
TX/RX – Data transfer/receive indicating LED.
TXD – Data transfer LED indicator.
RXD – Data receiving LED indicator.
Alarm mode – In the alarm status state, the controller initiates an event notification for the user-selected discrete input mode (Alarm mode: unconnected, connected, or both events).
Central computer – A server or computer to which data can be sent.

2. Preface

2.1 Symbols

International electrical symbol list. Some or all symbols can be used on controller marking or in this user manual.

Symbol	Explanation
CE	With the **CE** marking on a product the manufacturer ensures that the product conforms with the essential requirements of the applicable **EC** directives.
⎓	**DC** (Direct Current)
⚠	Caution
⏚	Grounding
●	LED indicator
1	Contact number on plug
RoHS	Directive on the restriction of the use of certain hazardous substances in electrical and electronic equipment 2002/95/EC. Commonly referred to as the Restriction of Hazardous Substances Directive or **RoHS**)
🗑	Waste Electrical and Electronic Equipment Directive

2.2 Safety instructions

To install and set up the device, special technical knowledge is required. Contact the seller or certified professionals to connect and set up the device!

Before connecting to the power supply, ensure that:

1. The controller is not damaged (no cracks, melted, broken, or exposed areas).
2. The controller is used with the correct cables of appropriate thickness.
3. The controller and antenna are installed indoors.
4. The controller is intended for supply from a 100-240VAC (0.25A), 50/60Hz Power Source
5. The associated equipment (AE), such as the PC and PSU (LPS), shall comply with the requirements of Standard EN 60950-1.
6. The controller is dry.
7. Ambient temperature and humidity are within the normal range.
8. Other types of devices (e.g., counters, etc.) are connected correctly according to the manufacturer’s regulations.
9. The end of stranded conductors shall not be consolidated by soft soldering and must be terminated properly.
10. The device, PC, and other peripheral devices must be strictly connected through a double-pole breaker (with a current break less than 5A and a space between breaker contacts greater than 3mm). The pole breaker must be part of the building's wiring and located in an accessible place with proper markings.

Don‘t use:

- The device in open water (in the rain or if water is splashing on the controller or connected devices).
- The device if the enclosure, connected cables, or other connected devices are damaged.

⚠ **Use the device according to the manufacturer’s regulations; otherwise, you may damage the controller or other devices, and in such a case, the manufacturer’s warranty may not be valid.**

⚠ **If you suspect that the device is not operating correctly or has visible issues, please contact the manufacturer or your distributor for inspection or maintenance.**

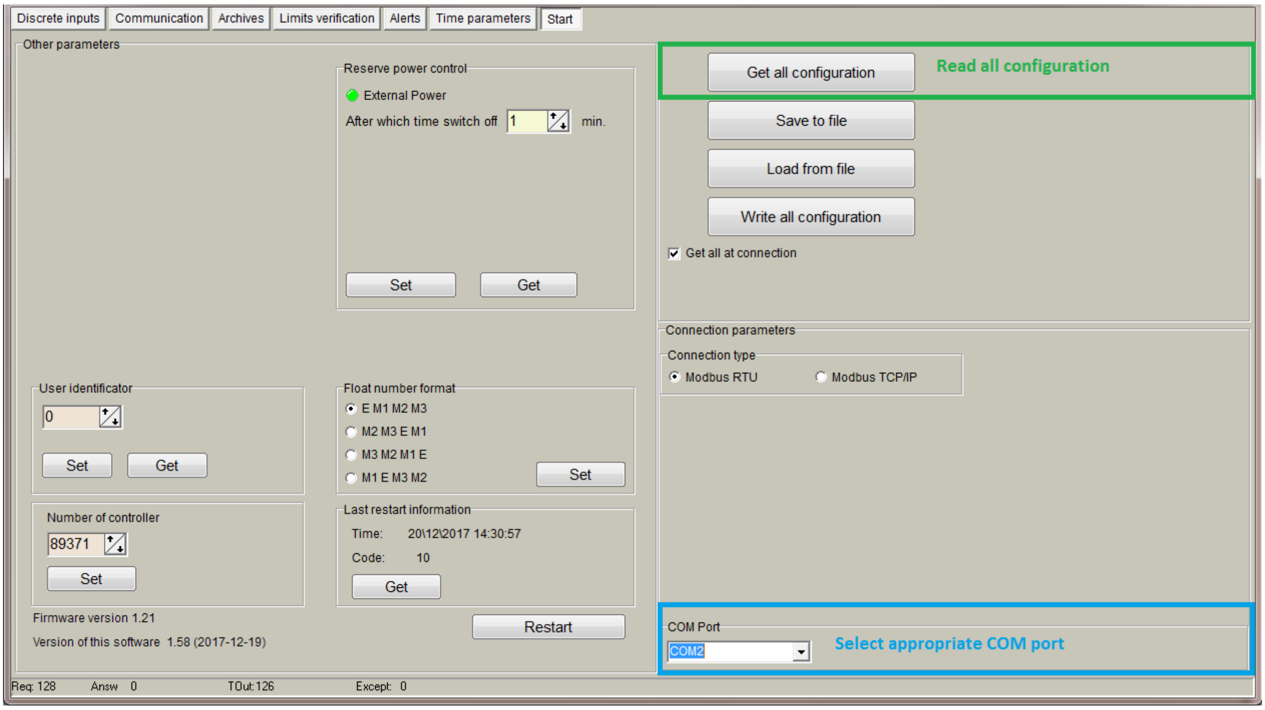
2.3 Connecting to device

The USB port is used for local configuration of the device. It is also possible to configure the device via Ethernet, a 4G/3G modem, or any of the UARTs if they are used as Modbus slaves. All configuration is done using the Modbus protocol and the device configuration tool software, which can be downloaded from the manufacturer’s website.

Configuration tool software
Once the USB driver is installed, run the device configuration tool software. Select the connection type *Modbus RTU* and the appropriate COM port. Press *Get all configuration* to read the entire device configuration.
Other functions include:

- *Get all configuration* button: Reads all configuration from the device.
- *Save to file* button: Saves all configuration to a file, so it can be loaded back to the device later.
- *Load from file* button: Loads saved configuration from a file.
- *Write all configuration*: Writes the loaded configuration to the device.
- *Get all at connection* checkbox: Reads all configuration when connecting over a TCP/IP connection.
- Connection type *Modbus RTU*: Connects to the device over USB or serial port.
- Connection type *Modbus TCP/IP*: Connects to the device over a TCP/IP connection.
- *User identification* section: User-configured device ID.
- *Number of controller* section: Device serial number.
- *Reserve power control* section: This is visible only if the device has a backup battery installed.
    - *After which time switch off* time: Powers off the device after the configured time if the main power supply is disconnected.
- *Float number format*: Floating-point data byte order (E - exponent, M1, M2, M3 - Mantissa).
- *Last restart information*: Time of the last reset and reset code. The reset code values are:

1. No TCP packet over GPRS in the configured time.
2. GPRS task stops working.
3. Not enough heap memory.
4. Firmware update reset.
5. Modbus reset.
6. Unable to connect to GPRS.
7. Watchdog reset.
8. Brownout reset.
9. Power-up reset.
10. All TCP sockets are used (if defined UIP_RESET_ALL_CONN_USED).
11. Periodic reset.



Other parameters

Reserve power control
● External Power
After which time switch off 1 min.

Get all configuration

Read all configuration

☑ Get all at connection

Connection parameters
Connection type
● Modbus RTU ○ Modbus TCP/IP

User identificator 0

Float number format
● E M1 M2 M3
○ M2 M3 E M1
○ M3 M2 M1 E
○ M1 E M3 M2

Number of controller 89371

Last restart information
Time: 20\12\2017 14:30:57
Code: 10

Firmware version 1.21
Version of this software 1.58 (2017-12-19)

COM Port COM2

Select appropriate COM port

Req: 128 Answ 0 TOut:126 Except: 0

3. Program modules

3.1 Archives

The device has several types of archives:

- **Events archive**: All events will be saved here (analog input alarms, discrete input alarms, limits verification). Events are used to generate SMS messages, MQTT event messages, and emails.
- **Diagnostic archive**: This archive contains a list of changes made to the device, such as resets, configuration changes, and connection/disconnection to the GPRS network, among others.
- **User-defined archive**: This is a user-configured archive, allowing the user to add any existing data register to the archive.

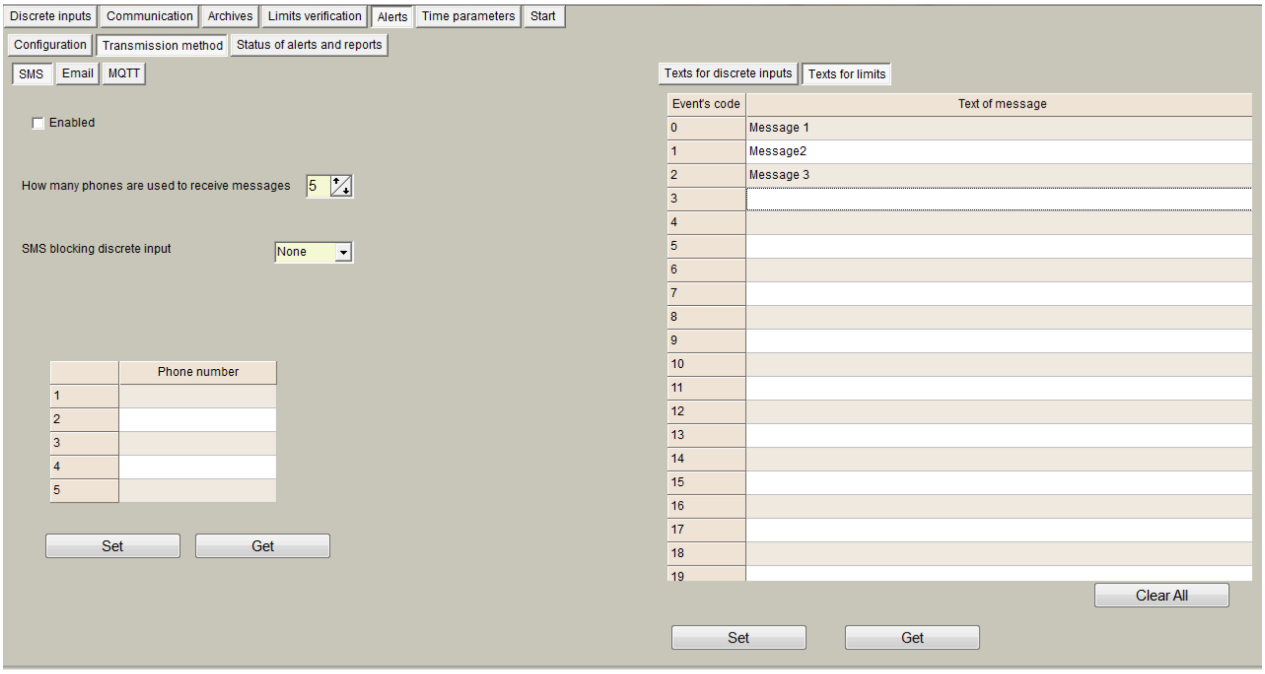
All archives are saved to the internal data flash memory. The maximum archive record count depends on the memory size and available space. If a Micro SD card is present, archives are duplicated to the SD card (a new file is created every day with a date stamp).

3.1.1 Events archive

All events are saved in the events archive. Event sources can include:

- Analog input events
- Discrete input events
- Events generated from the "Limits verification" module

The event archive is used to generate SMS, MQTT messages, and emails. For every event ID, you can configure the event message text. The event message text can be configured in *"Alerts -> Transmission method."*



☐ Enabled

How many phones are used to receive messages 5

SMS blocking discrete input None

	Phone number
1	
2	
3	
4	
5	

Event's code	Text of message
0	Message 1
1	Message2
2	Message 3
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	

Reading event archive over Modbus file system
Event archive can be read using Modbus read file function 20.

Modbus function	Modbus ID	Modbus file address	Max registers in file	Records in file	Current record count register
20 - Read File Record	Modbus RTU - 254 Modbus TCP - 255	400 .. 499	10000	1250	4910

Event archive record structure:

Variable name	Purpose / Value	Type of value
Time	Time of alarm	Long int (32 bits)
Alarm identifier	Every identifier is change +1	Long int (32 bits)
Alarm source	The oldest byte value (alarm source): 4 - Limits alarm. Youngest byte value: if alarm source 4 then index from limits table (start from 0 to 99)	Int (16 bits)
Type of deviation	If alarm source 4 then value is limits alarm ID.	Int (16 bits)
Deviation value	If alarm source 4 then Limits value.	Float
		Total 16 bytes

Records are transferred from the newest to the oldest. For example, to read the latest event archive record, the following information is required:

File address : 400
Register address: 0
Register count : 8 (16/2, event archive structure length/2)

To read the 5th oldest record:

File address : 400 + (5/1250) (record number/record count in file)
Register address: 8*(5-1)
Register count : 8 (16/2, event archive structure length/2)

3.1.2 Diagnostic archive

The **diagnostic archive** is a list of changes made to the device. It is useful for debugging purposes.

Reading diagnostic archive over the Modbus file system
The diagnostic archive can be read using the Modbus read file function 20.

Modbus function	Modbus ID	Modbus file address	Max registers in file	Records in file	Current record count register
20 - Read File Record	Modbus RTU - 254 Modbus TCP - 255	900 .. 999	10000	1250	4911

Diagnostic archive record structure:

Variable name	Purpose / Value	Type of value
Time	Record time. If event type=7 then new set time	Long int (32 bits)
Event type	1 - RESET event 2 - Firmware update event 3 – Archive counter change event 4 – Automatic time correction 5 – Time change over MODBUS 6 – Change of internal parameters 7 – Time correction	Long int (32 bits)
Event value (integer)	If event type: 1. Reason of last reset, values: 1. No TCP packet over GPRS in configured time 2. GPRS task stops working 3. Not enough heap memory 4. Firmware update reset 5. Modbus reset 6. Unable connect to GPRS 7. External pin reset 8. Watchdog reset 9. Brownout reset 10. Power up reset 13. All TCP sockets is used (if defined UIP_RESET_ALL_CONN_USED) 14. Periodic reset 3 Delete of archive index 0 - Alarm archive 1 - Diagnostic archive 2 - User defined archive 4 New time 5 New time 6 Always 0 7 Always 0	Long int (32 bits)
Event value (float)	If event type: 1 – always 0 3 – new set value(mostly 0, if delete all storage) 4 and 5– always 0 6 – always 0 7 – Time correction value (-30s +30s)	Float (32 bits)
		Total 16 bytes

Records are transferred from newest to oldest. For example, to read the latest event archive record, the following information is required:

File address : 900
Register address: 0
Register count : 8 (16/2, diagnostic archive structure length/2)

To read the 5th oldest record:

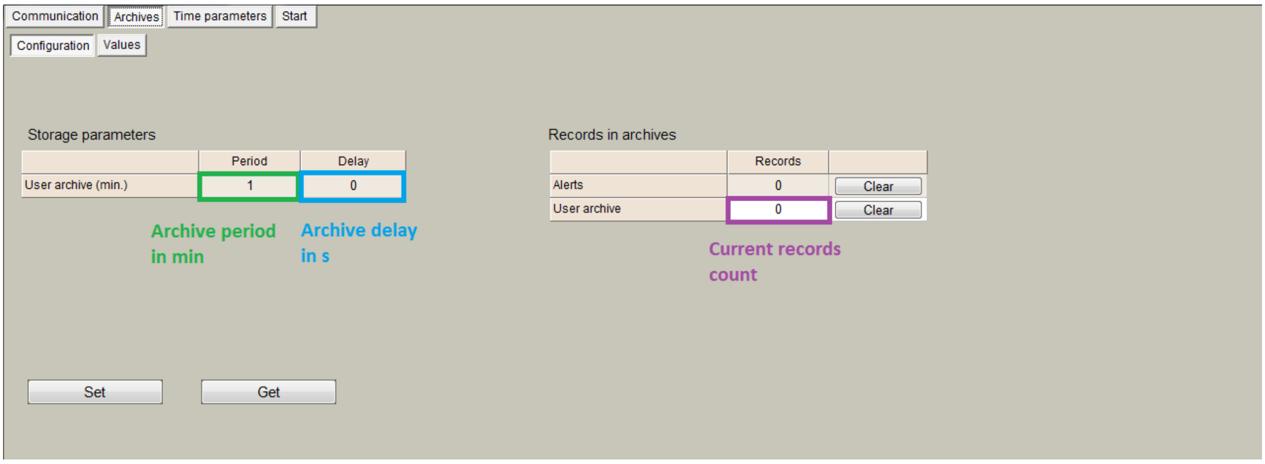
File address : 900 + (5/1250) (record number/record count in file)
Register address: 8*(5-1)
Register count : 8 (16/2, diagnostic archive structure length/2)

3.1.3 User-defined archive

The user can add any device register (value) to the user-defined archive, allowing the periodic archiving of useful values.

The archive period is in minutes and can range from 1 minute to 600 minutes (10 hours). You can configure this in the configuration tool under the "Archives/Configuration" section.

User archive storage period
The archive period is synchronized with real time. If the read period is 1 minute, records will be generated every minute (00:00:00, 00:01:00, 00:02:00, etc.). If the read period is 15 minutes, records will be generated every 15 minutes (00:00:00, 00:15:00, 00:30:00, etc.).
A delay can be configured to delay the archive time in seconds. For example, with a period of 15 minutes and a delay of 30 seconds, archives will be made at 00:00:30, 00:15:30, and so on. The delay is used to delay the archive process if values are being taken from meters, allowing time for the read process to complete.



Storage parameters

	Period	Delay
User archive (min.)	1	0

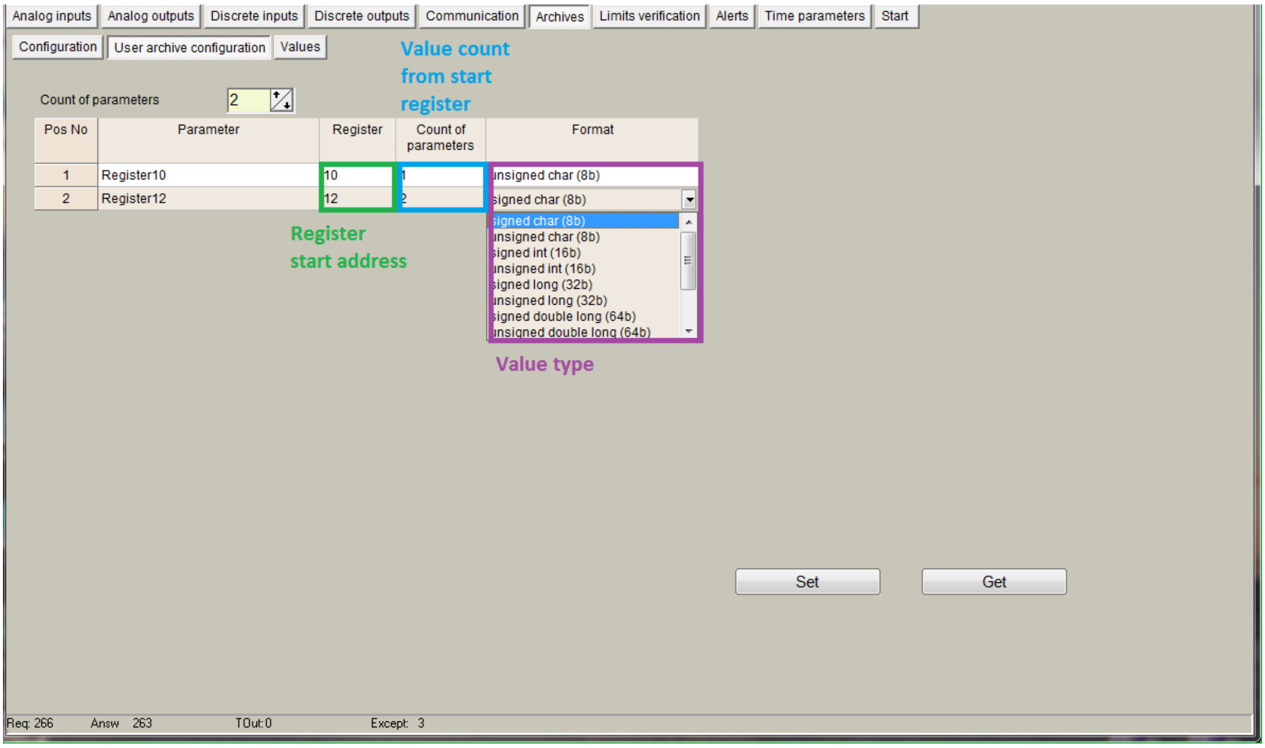
Records in archives

	Records	
Alerts	0	Clear
User archive	0	Clear

User archive configuration with configuration software
You can add the necessary registers to the archive using the table in *"Archives/User Archive Configuration."*

1. **Count of parameters** – The number of configured lines in the table.
2. In the **Register** column, configure the start register of the value.
3. In the **Count of parameters** column, configure how many data values will be taken from the start register.
4. In the **Format** column, configure the data value type for the current line. Value types can include:

- *Signed char (8 bit)*
- *Unsigned char (8 bit)*
- *Signed int (16 bit)*
- *Unsigned int (16 bit)*
- *Signed long (32 bit)*
- *Unsigned long (32 bit)*
- *Signed double long (64 bit)*
- *Unsigned double long (64 bit)*
- *Float (32 bit)*
- *Double float (64 bit)*
- *Siemens float (32 bit), special siemens data format*
- *String*
- *New line:* Adds a new line with the same timestamp. This is used to add a new record with the same time in the CSV file.
- *Unix time*



Count of parameters 2

Pos No	Parameter	Register	Count of parameters	Format
1	Register10	10	1	nsigned char (8b)
2	Register12	12	2	signed char (8b)

Req: 266 Answ 263 TOut:0 Except: 3

If FTP or MQTT clients are used, the archive is configured automatically using the *"Communication/Data Transfer/Common Parameters"* data table.

Reading user archive over Modbus file system
User defined archive can be read using Modbus read file function 20.

Modbus function	Modbus ID	Modbus file address	Max registers in file	Records in file	Current record count register
20 - Read File Record	Modbus RTU - 254 Modbus TCP - 255	800 .. 899	Depends on structure length	Depends on structure length	4912

Diagnostic archive record structure:

Variable name	Purpose / Value	Type of value
Time	Record time.	Long int (32 bits)
Register values	Values of configured registers. Register amount can be set in 4929 register or in configuration tool "Archives/User archive configuration" How configure registers check	Long int (32 bits)
		Total 4+2xregister count bytes

Records are transferred from newest to oldest. For example, to read the latest user archive record (with 2 registers), the following information is required:

User archive structure length = 4+2 registers*2 = 8 bytes = 4 registers
Records in file = 10000/4 registers = 2500
File address : 800
Register address: 0
Register count : 4 (8 / 2, where 8 is the user archive structure length and 2 is the number of bytes per register)

To read the 5th oldest record:

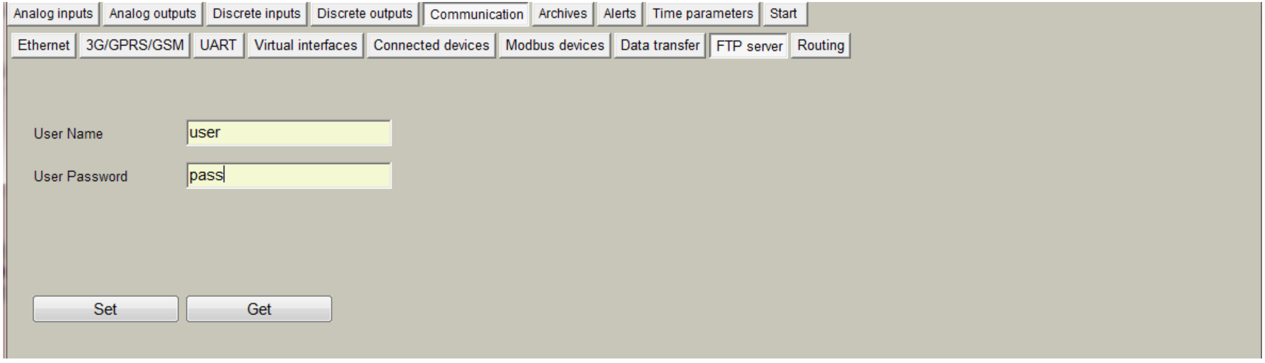
File address : 800 + (5/2500)   (record number/record count in file)
Register address: 4*(5-1)
Register count : 4 (8 / 2, where 8 is the user archive structure length and 2 is the number of bytes per register)

3.2 TCP modules

3.2.1 FTP server

The FTP server is used to access internal micro SD card data. You can connect to it using any FTP client program, such as "FileZilla," "Total Commander," or any web browser. The FTP server runs on the standard port 21.

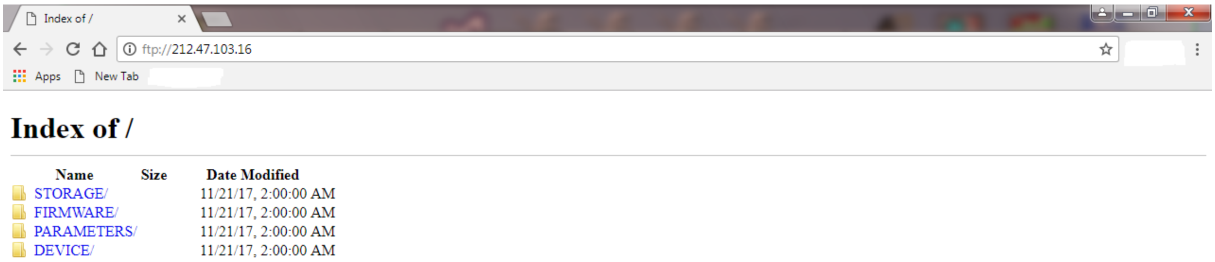
Connection to FTP server
Before connecting, you need to know your device's IP address and FTP username/password. The FTP server username and password can be configured in the *"Communication/FTP Server"* tab. The username and password can be up to 19 characters long.



User Name: user
User Password: pass

Example:
Let's try to connect to the device at IP 212.47.103.16 using Google Chrome. Enter *ftp://212.47.103.16* in the browser address bar. You will be prompted to enter a username and password; provide them and log in.
The browser will load a directory list of the SD card. The main directories in the device are:

- **Storage** – Contains all archives (folders include "Alarms," "Diagnostic," "User Defined").
- **Firmware** – Reserved for future use (firmware update folder).
- **Parameters** – Contains device parameters (in some firmware versions, it may store CSV file headers, dimensions, etc.).
- **Device** – Contains the device description file.



Index of /

Name	Size	Date Modified
STORAGE/		11/21/17, 2:00:00 AM
FIRMWARE/		11/21/17, 2:00:00 AM
PARAMETERS/		11/21/17, 2:00:00 AM
DEVICE/		11/21/17, 2:00:00 AM

3.2.2 FTP client

The FTP client is used to send user archive files to an FTP server. Files have a .csv extension and are generated from saved user archives.

Csv file creator

The device creates a CSV report file from user-defined archive values. Every record in the file has its timestamp (value record time). It can include a "Header" for each value, and a dimension for each value. All data in the file is separated by a ";" symbol, and each record is placed on a new line. A standard file content looks like this:

```
Time;<Value Header 1>;<Value Header 2>;...<Value Header N>;
<Record 1 Date/Time>;<Value 1 data>;<Value 1 dimension>;<Value 2 data>;<Value 2 dimension>;...<Value N data>;<Value N dimension>;
<Record 2 Date/Time>;<Value 1 data>;<Value 1 dimension>;<Value 2 data>;<Value 2 dimension>;...<Value N data>;<Value N dimension>;
...
<Record N Date/Time>;<Value 1 data>;<Value 1 dimension>;<Value 2 data>;<Value 2 dimension>;...<Value N data>;<Value N dimension>;
```

- **<Value Header>** – Configured header from the *"Communication/Data Transfer/Common Parameters"* tab.
- **<Value data>** – Configured parameter value taken from the user archive.
- **<Value dimension>** – Configured dimension from the *"Communication/Data Transfer/Common Parameters"* tab.
- **<Record Date/Time>** – Stored archive record time.

CSV file creator table

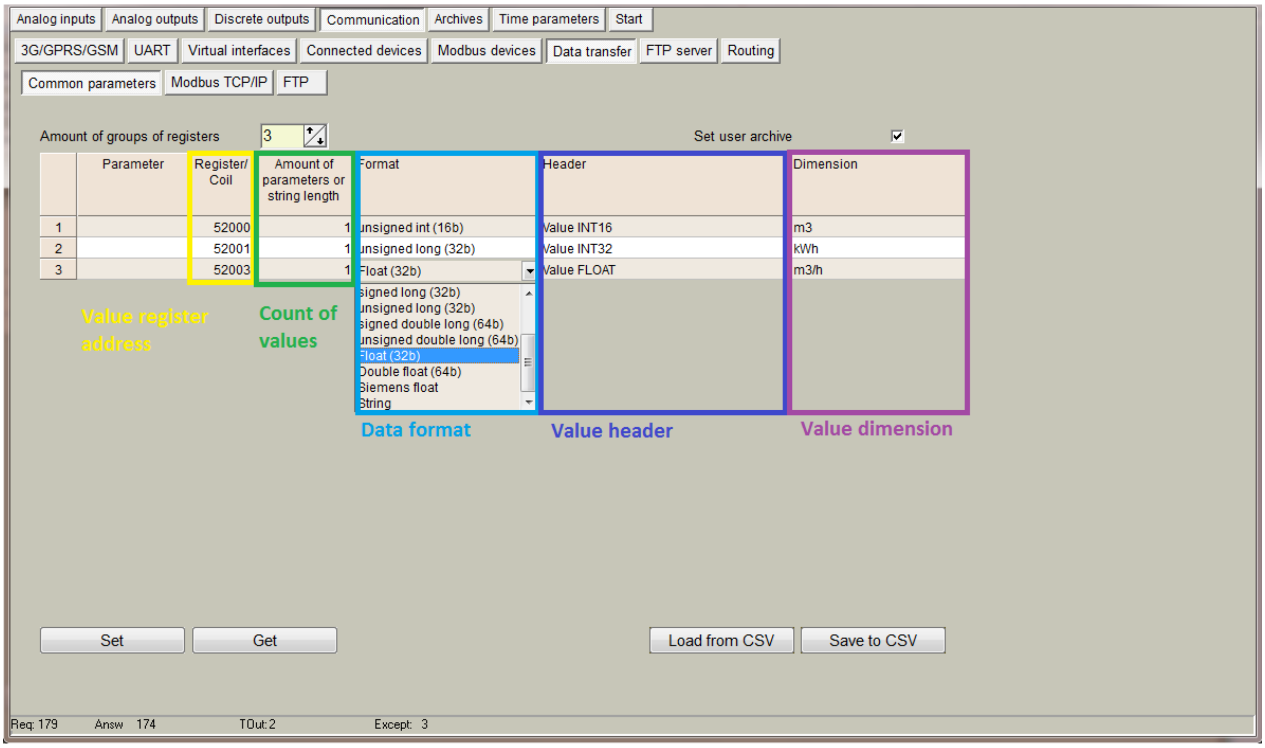
The device creates a CSV report file from user-defined archive values and data configured in the *"Communication/Data Transfer/Common Parameters"* tab or in *"Archives/User Archive Configuration"* (from version 1.58).

- **"Amount of groups of register"** number field – Number of lines in the table.
- **"Set user archive"** checkbox – Always check this; it automatically configures the user-defined archive.
- **"Register/Coil"** column – Value internal start register. You can configure any internal value register from 0 to 65535 (Function 3 - read holding registers) or from 100000 to 165535 (Function 4 - read input registers).
- **"Amount of parameters or string length"** column – The number of values from the start register. It can range from 1 to 50. This can be used to configure up to 50 values from the start register in one line, but note that the same header and dimension will apply to all values.
- **"Format"** column – Value format. In Modbus protocol, you need to know the value format before reading it; otherwise, you will read hexadecimal values that are difficult to interpret. In some firmware versions, not all formats are supported. List of data formats:
  - *Signed char (8 bits)*
  - *Unsigned char (8 bits)*
  - *Signed integer (16 bits)*
  - *Unsigned integer (16 bits)*
  - *Signed long (32 bits)*
  - *Unsigned long (32 bits)*
  - *Float (32 bits)*
  - *Double float (64 bits)*
  - *Siemens float (32 bit), special siemens data format*
  - *String*
  - *New line* Add a new line with the same timestamp. Used to add a new record with the same time in

the CSV file

- *Unix time*

- **"Header"** column – Value header in the CSV file. If more than one value is configured, the same header will be used for all values.
- **"Dimension"** column – Value dimension in the CSV file. If more than one value is configured, the same dimension will be used for all values.
- **"Set"** button – Write the configuration to the device.
- **"Get"** button – Read the configuration from the device.
- **"Load from CSV"** button – Load a saved configuration table from a CSV file.
- **"Save to CSV"** button – Save the configuration table to a CSV file.



Amount of groups of registers: 3

Set user archive ☑

	Parameter	Register/ Coil	Amount of parameters or string length	Format	Header	Dimension
1		52000	1	unsigned int (16b)	Value INT16	m3
2		52001	1	unsigned long (32b)	Value INT32	kWh
3		52003	1	Float (32b)	Value FLOAT	m3/h

signed long (32b)
unsigned long (32b)
signed double long (64b)
unsigned double long (64b)
Float (32b)
Double float (64b)
Siemens float
String

Req: 179 Answ 174 TOut:2 Except: 3

Example:
With this configuration received file will be like this:

Time;Value INT16;;Value INT32;;Value FLOAT;;
2017.11.22 12:00:00;123;m3;123456;kWh;1.0000;m3/h;
2017.11.22 13:00:00;124;m3;123459;kWh;1.0000;m3/h;
2017.11.22 14:00:00;125;m3;123468;kWh;1.0000;m3/h;
2017.11.22 15:00:00;128;m3;123475;kWh;1.0000;m3/h;

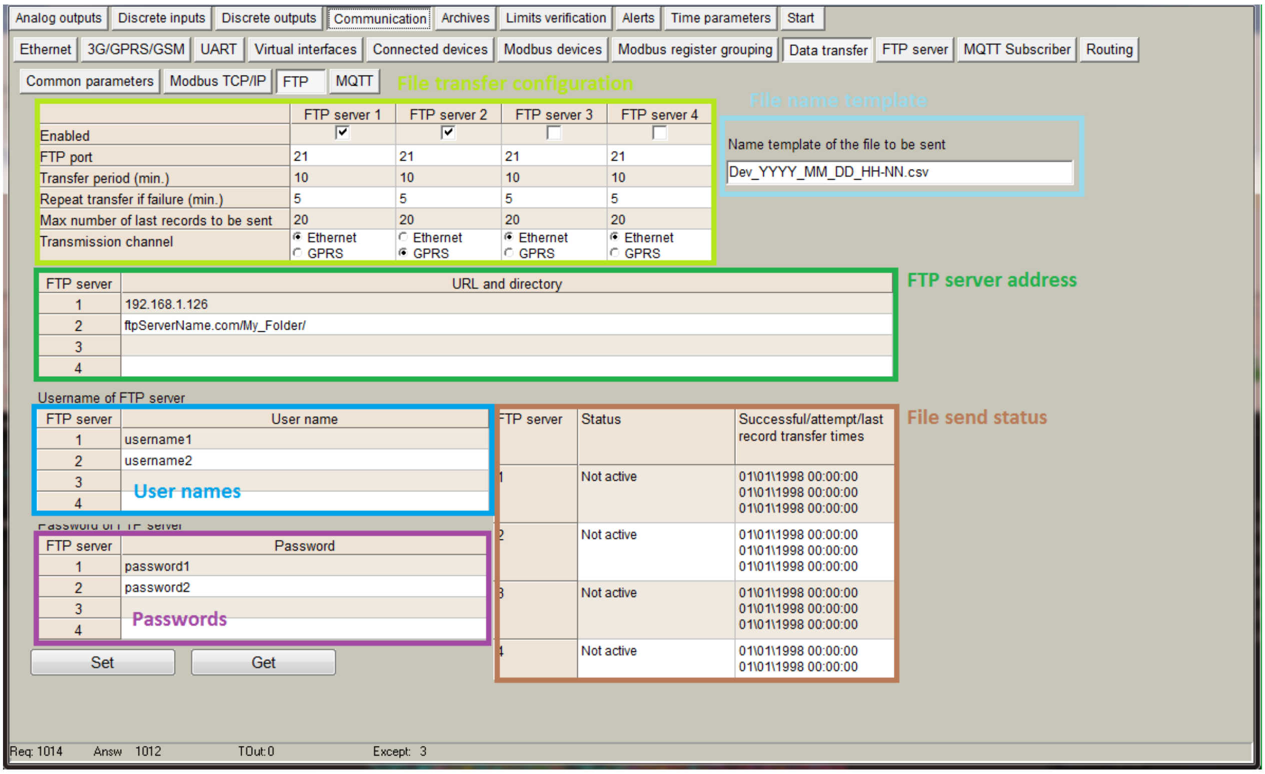
FTP client configuration

A CSV file can be sent to up to 4 FTP servers. Only the FTP protocol is supported using passive mode, user/password authentication, and the CSV file format.

All configuration is done in the *"Communication/Data Transfer/FTP"* tab. Before configuring, you need to have a working FTP server and the following information: its IP address or URL, and the connection username and password. The sections in this tab include:

- "**File Transfer Configuration**" section:
  - **"Enabled"** checkbox: Enables the appropriate FTP server.
  - **"FTP port"**: The appropriate FTP server's TCP port.

  - **"Transfer period (min)"**: File send period. The value is in minutes and can range from 1 minute to 1440 minutes (24 hours).
  - **"Repeat transfer if failure (min)"**: Retries sending the file after the configured time if the file transfer fails. The value is in minutes and can range from 1 minute to 1440 minutes (24 hours). The recommended value is half the transfer period.
  - **"Max number of last records to be sent"**: The maximum number of last records to include in the CSV file. Only new records will be included. If 25 new records are available and the maximum is configured to 20, the file will only contain the 20 newest records, and the last 5 will be lost. If there are 5 new records and the maximum is set to 20, the file will contain 5 new records. Values can range from 1 to 200.
  - **"Transmission channel"**: The device can have 2 transfer channels: Ethernet and GPRS. Select which to use for the appropriate FTP server.
- "**FTP server address**" section:
  - **"URL and directory" column**: The IP or URL address of the FTP server (up to 127 characters). It can be an IP address like "127.0.0.1" or a URL like "www.myftp.com". Directory listings can also be used, e.g., "www.myftp.com/MyFiles/".
- **"Usernames"** section:
  - "User name" column: The username for the appropriate FTP server.
- **"Passwords"** section:
  - "Password" column: The password for the appropriate FTP server.
- **"File Send Status"** section:
  - **"Status"** column: The current status of the FTP client. After the file is sent to the server, the status changes to "File transmitted". There are several other statuses for process checks: "Connecting to server", "Sending user", "Sending password", "Sending data file", and others.
  - **"Successful/Attempt/Last record transfer times"** column: Shows the times of some operations. "Successful" time is the time of the last successfully completed file send to the FTP server. "Attempt" time is the time of the last file send attempt (whether successful or not). "Last record transfer" time is the time the last record was sent.
- **"File Name Template"** section: The file name can be up to 127 characters long, including the ".csv" extension. The file name can include fixed fields that will be replaced with the date and time. Fixed fields include:
  - YYYY – Year
  - MM – Month
  - DD – Day
  - HH – Hour
  - NN – Minute
  - For example, if the file name template is "Dev_YYYY_MM_DD__HH-NN.csv" and the date is 2017.03.25 at 14:25, the file name will be "Dev_2017_03_25__14-25.csv".
- **"Set"** button: Writes the configuration to the device.
- **"Get"** button: Reads the configuration from the device.



File transfer configuration

	FTP server 1	FTP server 2	FTP server 3	FTP server 4
Enabled	☑	☑	□	□
FTP port	21	21	21	21
Transfer period (min.)	10	10	10	10
Repeat transfer if failure (min.)	5	5	5	5
Max number of last records to be sent	20	20	20	20
Transmission channel	● Ethernet ○ GPRS	○ Ethernet ● GPRS	● Ethernet ○ GPRS	● Ethernet ○ GPRS

File name template

Name template of the file to be sent

Dev_YYYY_MM_DD_HH-NN.csv

FTP server address

FTP server	URL and directory
1	192.168.1.126
2	ftpServerName.com/My_Folder/
3	
4	

Username of FTP server

FTP server	User name
1	username1
2	username2
3	
4	

User names

Password of FTP server

FTP server	Password
1	password1
2	password2
3	
4	

Passwords

File send status

TP server	Status	Successful/attempt/last record transfer times
1	Not active	01\01\1998 00:00:00 01\01\1998 00:00:00 01\01\1998 00:00:00
2	Not active	01\01\1998 00:00:00 01\01\1998 00:00:00 01\01\1998 00:00:00
3	Not active	01\01\1998 00:00:00 01\01\1998 00:00:00 01\01\1998 00:00:00
4	Not active	01\01\1998 00:00:00 01\01\1998 00:00:00

Req: 1014 Answ 1012 TOut:0 Except: 3

Example:

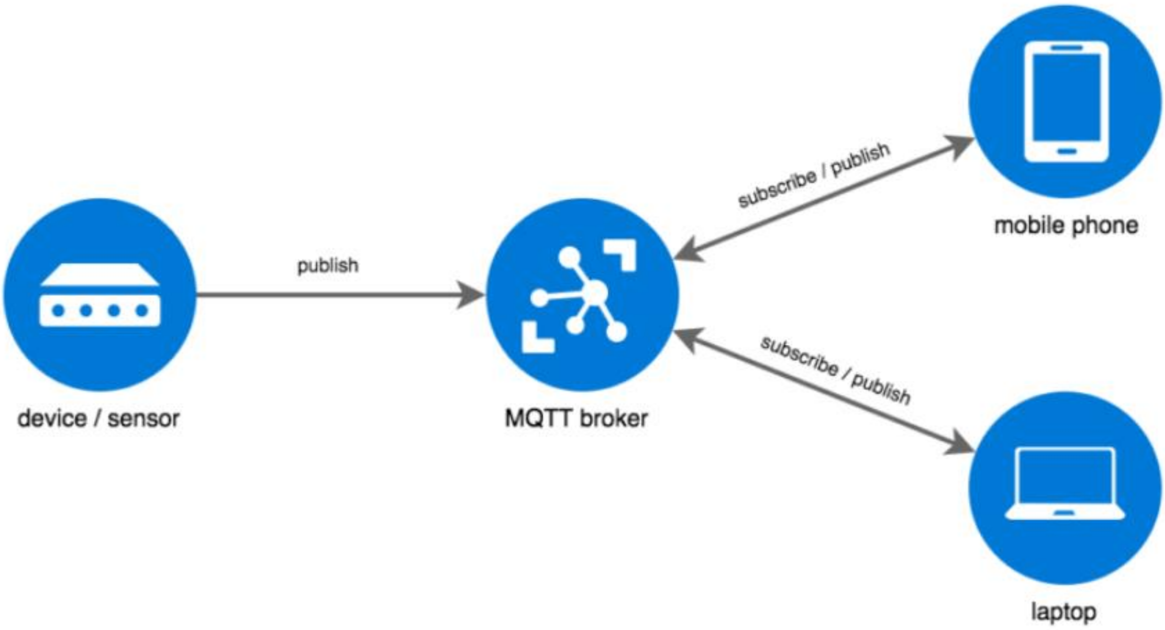
In the picture above, two FTP servers are configured:

- The first will be accessed through an Ethernet connection. Its address is "192.168.1.126", with the username "username1" and password "password1". Files will be sent every 10 minutes, and the process will repeat every 5 minutes if the sending was unsuccessful.
- The second will be accessed through a GPRS connection. Its address is "ftpServerName.com", and the directory where the files will be stored is "My_Folder", with the username "username1" and password "password1". Files will be sent every 10 minutes, and the process will repeat every 5 minutes if the sending was unsuccessful.

3.3 MQTT client

MQTT stands for MQ Telemetry Transport. It is a publish/subscribe, extremely simple, and lightweight messaging protocol designed for constrained devices and low-bandwidth, high-latency, or unreliable networks. The design principles are to minimize network bandwidth and device resource requirements while attempting to ensure reliability and some degree of delivery assurance. These principles make the protocol ideal for the emerging "machine-to-machine" (M2M) or "Internet of Things" (IoT) world of connected devices, as well as for mobile applications where bandwidth and battery power are at a premium.

With MQTT, devices (clients) connect to a broker (server) to publish their status to specific topics. The broker then ensures that all other clients interested in this status topic receive the status.



Our device can send event messages or JSON-type data files via MQTT. Data files can be created from real-time values or archived values, depending on the configuration. Different topics are used for event messages and data files.

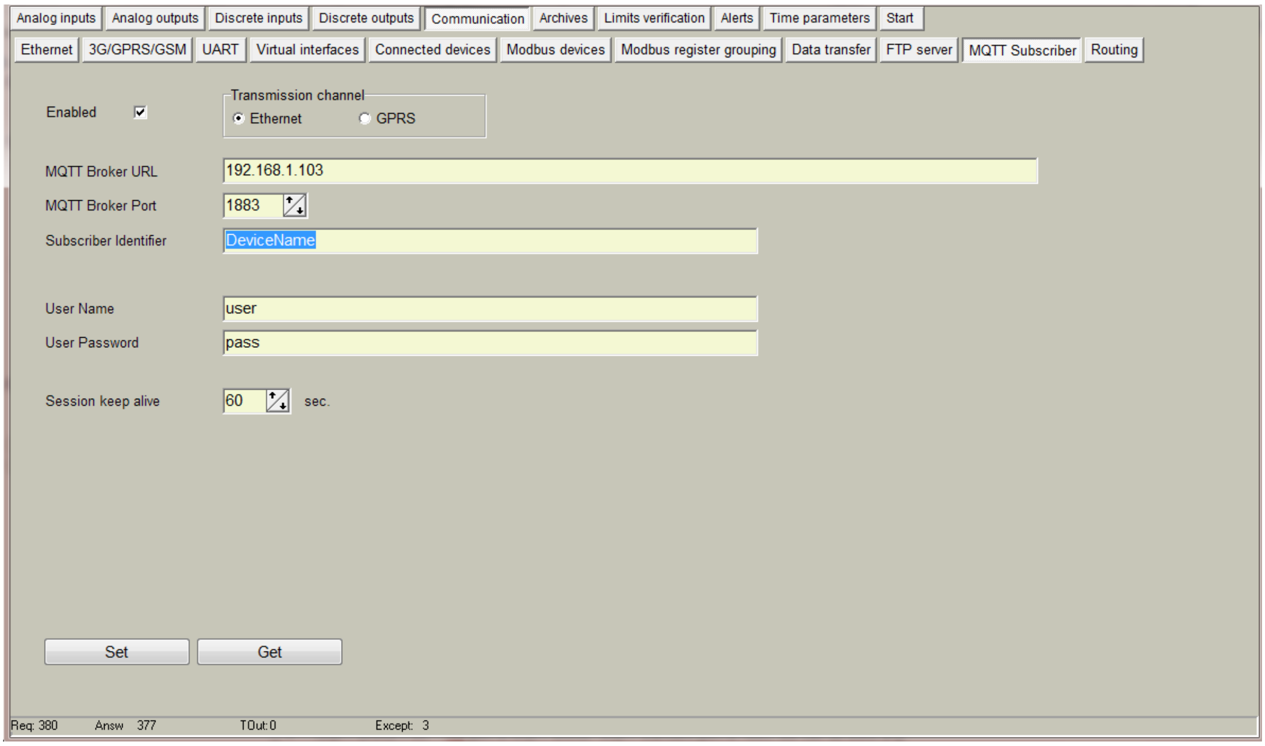
3.3.1 MQTT client configuration

MQTT client configuration is done in the *"Communication/MQTT Subscriber"* tab. Before configuring, you need to have a working MQTT broker and some information, such as its IP address or URL, port, username, and password.

- **"Enabled" checkbox** – Enables the MQTT client on the device.
- **"Transmission channel"** – The device can have 2 transfer channels: Ethernet or GPRS. Select which

one to use for the connection to the MQTT broker.

- **"MQTT Broker URL"** – The IP or URL address of the MQTT broker (up to 63 characters).
- **"MQTT Broker Port"** – The TCP port of the MQTT broker. The standard MQTT port is 1883, but other ports can also be used.
- **"Subscriber Identifier"** – The client identifier, which is a unique identifier for each MQTT client connecting to an MQTT broker. As the term suggests, this identifier should be unique per broker. The broker uses it to identify the client and its current state.
- **"User Name"** – The client's username.
- **"User Password"** – The client's password.
- **"Session keep alive"** – Sends a keep-alive packet every specified number of seconds.
- **"Set" button** – Writes the configuration to the device.
- **"Get" button** – Reads the configuration from the device.
-



Enabled ☑

Transmission channel
● Ethernet ○ GPRS

MQTT Broker URL 192.168.1.103

MQTT Broker Port 1883

Subscriber Identifier DeviceName

User Name user

User Password pass

Session keep alive 60 sec.

Req: 380 Answ 377 TOut:0 Except: 3

The device connects using the configured parameters and remains connected until it disconnects for any reason. After disconnection, a new connection is automatically established. While the device is connected to the MQTT broker, it can send event messages or data report files.

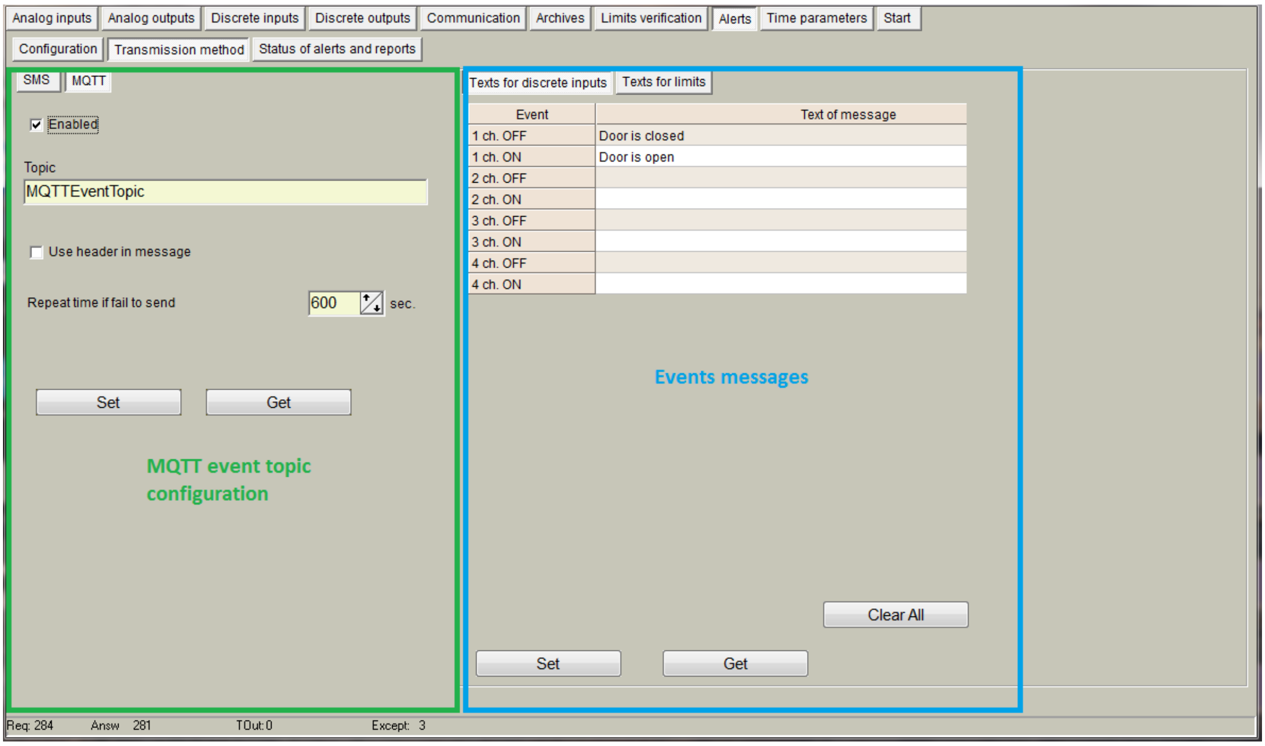
3.3.2 Event messages configuration

Event message sending to the MQTT broker is configured in the *"Alerts/Transmission Method/MQTT"* tab. Before sending event messages, the message text must be configured:

- **"Events Messages" section**:
  - **"Text for discrete inputs"** – Each discrete input has two states: ON and OFF. You can configure an individual message for each state.
  - **"Texts for limits"** – You can configure up to 100 messages for the limits verification module. Each index can have its own message.
  - **"Set" button** – Writes the configuration to the device.
  - **"Get" button** – Reads the configuration from the device.
  - **"Clear" button** – Clears all message texts.
- **"MQTT Event Topic Configuration" section**:
  - **"Enabled" checkbox** – Enables event message sending to the MQTT broker.
  - **"Topic"** – The publish topic for event messages. The device sends event messages using this

topic.

- **"Use header in message" checkbox** – If checked, a header is added to every message. This is useful if you want to group messages by device with the same topic. In the header, you can write the device name, identification number, address, or other relevant details. Headers can be configured in the *"Start" tab, "Other Parameters"* section.
- **"Repeat time if fail to send"** – If event message delivery fails, the device will attempt to resend the message after the configured number of seconds.
- **"Set" button** – Writes the configuration to the device.
- **"Get" button** – Reads the configuration from the device.



☑ Enabled

Topic

MQTTEventTopic

□ Use header in message

Repeat time if fail to send 600 sec.

Event	Text of message
1 ch. OFF	Door is closed
1 ch. ON	Door is open
2 ch. OFF	
2 ch. ON	
3 ch. OFF	
3 ch. ON	
4 ch. OFF	
4 ch. ON	

Req: 284 Answ 281 TOut:0 Except: 3

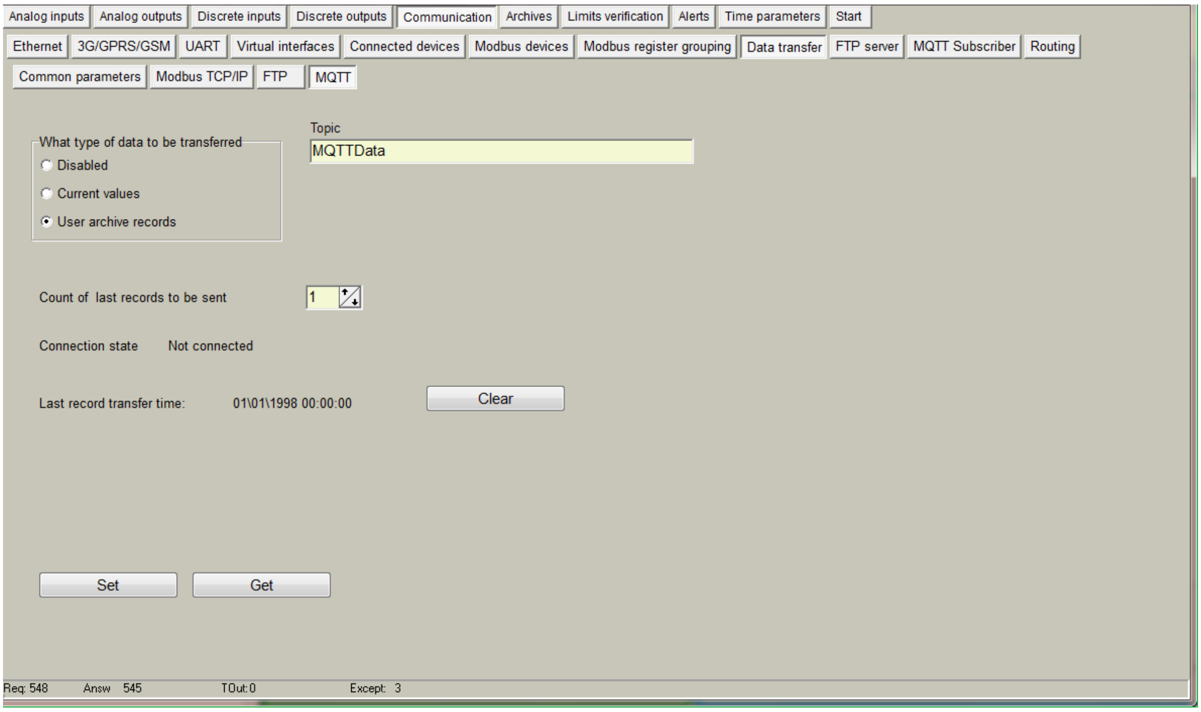
3.3.3 Report data file configuration

Data report files can be created from real-time values or archived values. The value configuration needed to send data is the same as the FTP client and is configured in the *"Communication/Data Transfer/Common Parameters"* tab. The only difference is that the FTP client sends data in CSV format, while the MQTT client sends data in JSON format. Both the FTP client and MQTT client use the same data value configuration table.

Configuration:

- **"What type of data to be transferred" section:**
  - **"Disabled"** – Disables data transfer to the MQTT broker.
  - **"Current values"** – Sends data created from current values. Files are sent every configured period, ranging from 1 second to 86400 seconds (24 hours).
  - **"User archive records"** – Sends data created from archived values. In this mode, you need to configure how many records to send in one file. Files will be created and sent whenever a new archive record appears.
- **"Topic"** – The MQTT topic for JSON file transfer.
- **"Count of last records to be sent"** – Specifies how many of the last records to send in case of connection problems. This option is available if the transfer type is *"User archive records"*.
- **"Current values transmission period"** – Specifies the transfer period for current values (in seconds). This option is available if the transfer type is *"Current values"*.
- **"Connection state"** – Displays the current connection state. This option is available if the transfer type is *"User archive records"*.
- **"Last record transfer time"** – Displays the time of the last sent record. This option is available if the transfer type is *"User archive records"*.

- **"Clear" button** – Resets the time of the last sent record. This option is available if the transfer type is *"User archive records"*.
- **"Set" button** – Writes the configuration to the device.
- **"Get" button** – Reads the configuration from the device.



What type of data to be transferred
○ Disabled
○ Current values
● User archive records

Topic: MQTTData

Count of last records to be sent: 1

Connection state: Not connected

Last record transfer time: 01\01\1998 00:00:00

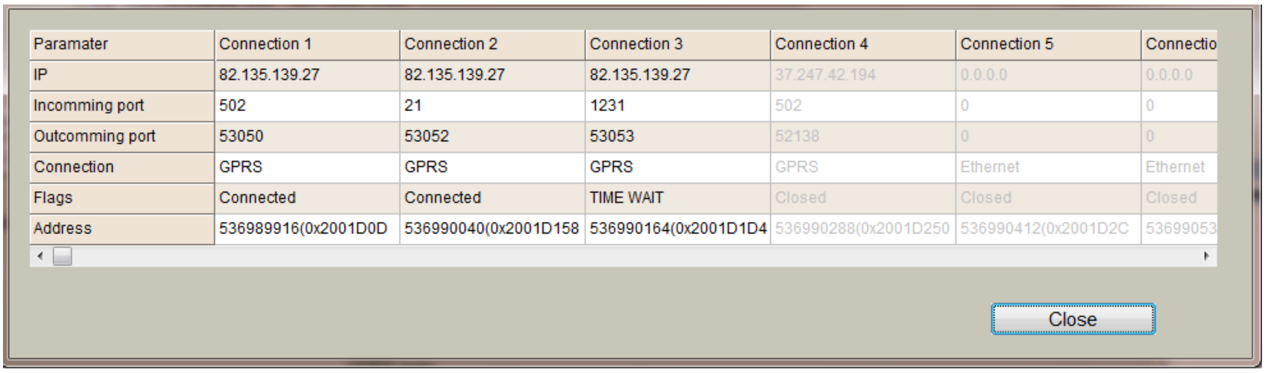
Req: 548 Answ 545 TOut:0 Except: 3

3.4 TCP/IP connection table

The internal TCP/IP stack can accept up to 40 connections simultaneously, including both GPRS and Ethernet channels. It is useful for debugging purposes to check current connections, opened ports, and other relevant details.

Open the TCP/IP connection table by pressing **CTRL+F1**. A new window, "Stack Information," will open. The table consists of 40 connection columns and connection parameters:

- **"IP"** – Remote IP address.
- **"Incoming port"** – Local TCP port.
- **"Outgoing port"** – Remote TCP port.
- **"Connection"** – Network channel (GPRS or Ethernet).
- **"Flags"** – Connection state (Closed, Connecting, Connected, Time Wait).
- **"Address"** – Internal address where the connection structure is stored.



Paramater	Connection 1	Connection 2	Connection 3	Connection 4	Connection 5	Connectio
IP	82.135.139.27	82.135.139.27	82.135.139.27	37.247.42.194	0.0.0.0	0.0.0.0
Incomming port	502	21	1231	502	0	0
Outcomming port	53050	53052	53053	52138	0	0
Connection	GPRS	GPRS	GPRS	GPRS	Ethernet	Ethernet
Flags	Connected	Connected	TIME WAIT	Closed	Closed	Closed
Address	536989916(0x2001D0D	536990040(0x2001D158	536990164(0x2001D1D4	536990288(0x2001D250	536990412(0x2001D2C	5369905З

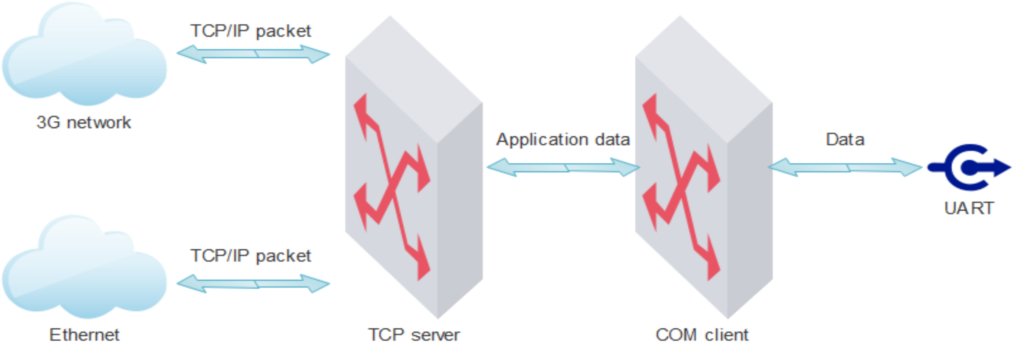
Example

In the picture above, 3 connections are established:

- **1st connection:** Remote host 82.135.139.27 connected to port 502 (Modbus TCP/IP connection). The connection is established through the 4G/3G/GPRS channel.
- **2nd connection:** Remote host 82.135.139.27 connected to port 21 (FTP). The connection is established through the 4G/3G/GPRS channel.
- **3rd connection:** Remote host 82.135.139.27 connected to port 1231. The connection is established through the 4G/3G/GPRS channel and is waiting to close (TIME WAIT).

3.5 Routing TCP/IP - serial (request/answer)

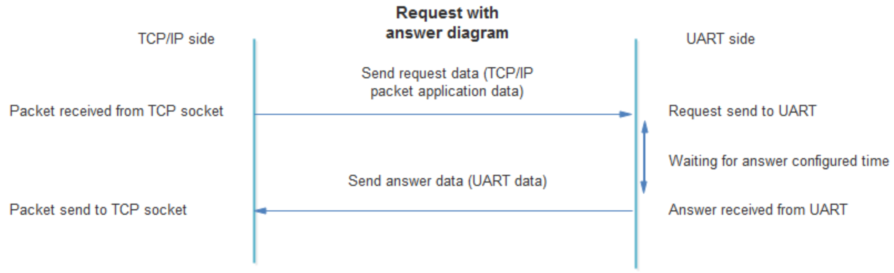
The TCP server routes application data (requests) from the TCP network to the serial port and returns the serial data (answers) to the TCP network. The TCP server works through a virtual service—COM client—so other services can access the UART as well. The TCP server resends application data to the UART and back without modifying the data.

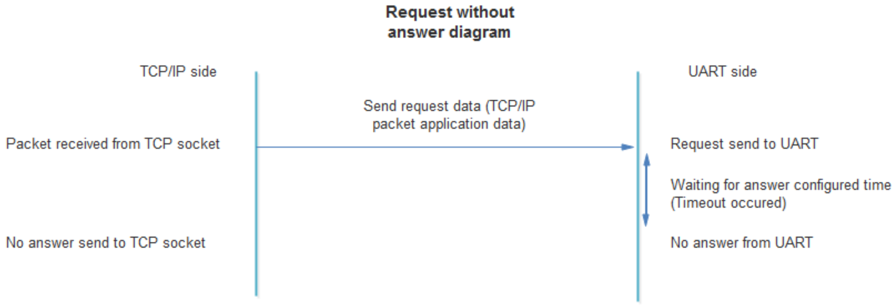


In the diagram below, you can see how packets are transmitted from TCP/IP sockets to UARTs. After the TCP socket is opened, all application data through this socket is sent to the UART. The process works as follows:

1. TCP socket is opened.
2. Waiting for a TCP/IP packet.
3. TCP/IP packet is received; its application data is sent to the COM client and then to the UART.
4. Waiting for an answer for the configured time (COM client timeout).
5. If an answer is received, the data is sent back to the TCP socket.
6. Waiting for the next request...

The TCP server uses a virtual COM client to connect to the UART, so it waits for the configured timeout. If no packets are received on the UART during this time, nothing is sent to the TCP socket.

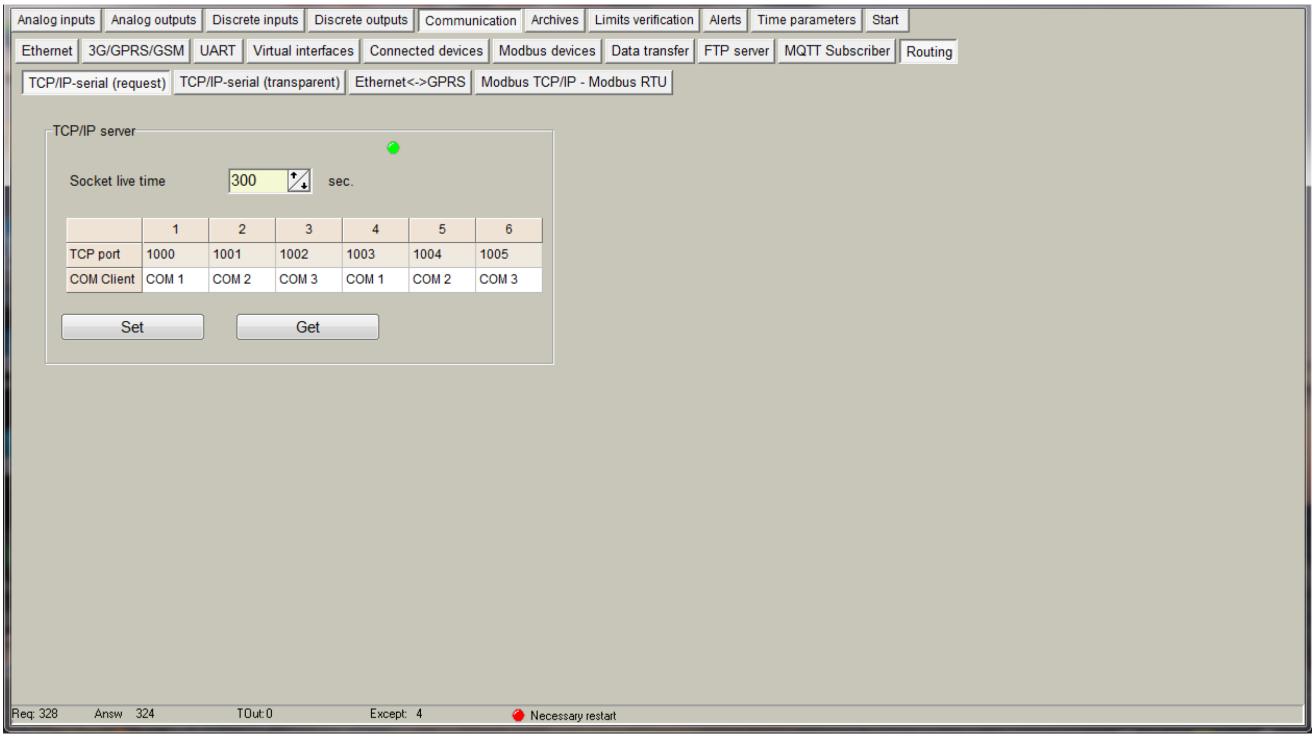

Request with
answer diagram
TCP/IP side
UART side
Send request data (TCP/IP packet application data)
Packet received from TCP socket
Request send to UART
Waiting for answer configured time
Send answer data (UART data)
Packet send to TCP socket
Answer received from UART


Request without
answer diagram
TCP/IP side
UART side
Send request data (TCP/IP packet application data)
Packet received from TCP socket
Request send to UART
Waiting for answer configured time
(Timeout occured)
No answer send to TCP socket
No answer from UART

⚠ **If answer data is received after timeout, it will be lost.**

TCP server configuration
TCP server configuration is done in the *"Communication/Routing/TCP/IP-Serial (Request)"* tab. The device can open up to 6 different TCP ports for data transfer to the associated virtual COM client (refer to the COM client section for details on how it is connected to the physical UART).

- **"Socket live time"** – Socket timeout in seconds (60-65535 seconds). If no data is transferred within the configured time, the device will automatically close the socket.
- **"TCP port"** – Internal socket TCP port (1-65535). The device waits for a connection to the configured ports and opens a data transfer channel with the appropriate virtual COM client.
- **"COM Client"** – The virtual COM client associated with the corresponding TCP port (COM1 - COM3).
- **"Set" button** – Writes the configuration to the device.
- **"Get" button** – Reads the configuration from the device.



TCP/IP server

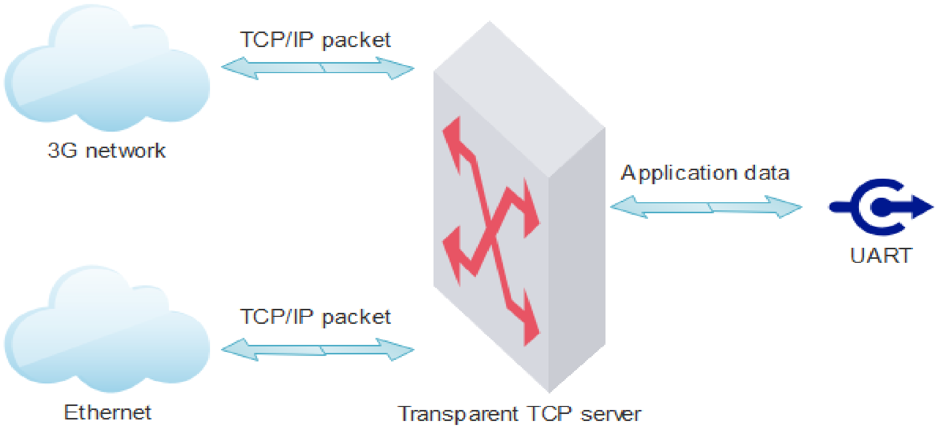
Socket live time 300 sec.

	1	2	3	4	5	6
TCP port	1000	1001	1002	1003	1004	1005
COM Client	COM 1	COM 2	COM 3	COM 1	COM 2	COM 3

Req: 328 Answ 324 TOut:0 Except: 4 ● Necessary restart

3.6 Routing TCP/IP - serial (transparent)

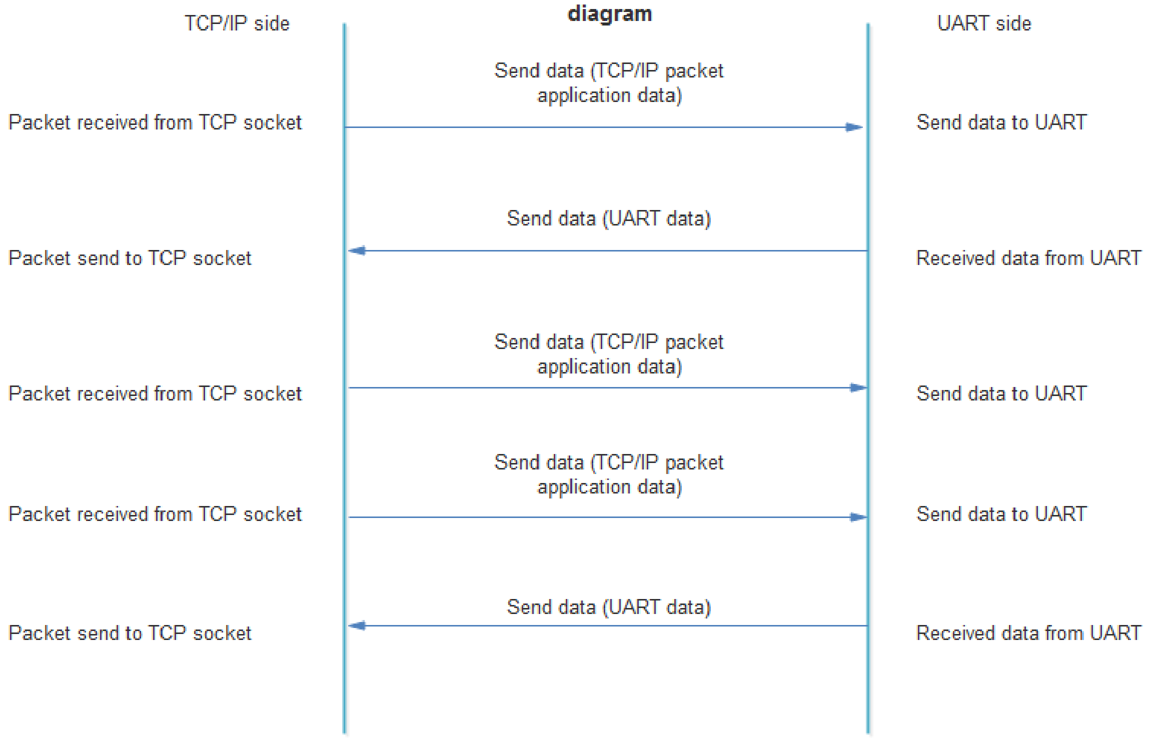
The **Transparent TCP server** routes application data from the TCP/IP network to the UART and routes the serial data from the UART back to the TCP/IP network. The key difference from the standard TCP server is that the Transparent TCP server works directly with UARTs, and all data is transferred in both directions without answer timeouts.



In the diagram below, you can see how packets are transmitted from TCP/IP sockets to UARTs and back. Once the TCP socket is opened, all application data through the socket goes to the UART, and all data from the UART is sent back to the TCP socket. The process works as follows:

1. TCP socket is opened.
2. Waiting for TCP/IP packet or data from UART.
3. TCP/IP packet is received, and its application data is sent to the UART. If UART data is received, it is directly sent to the TCP socket.
4. Waiting for the next TCP/IP packet or data from UART.

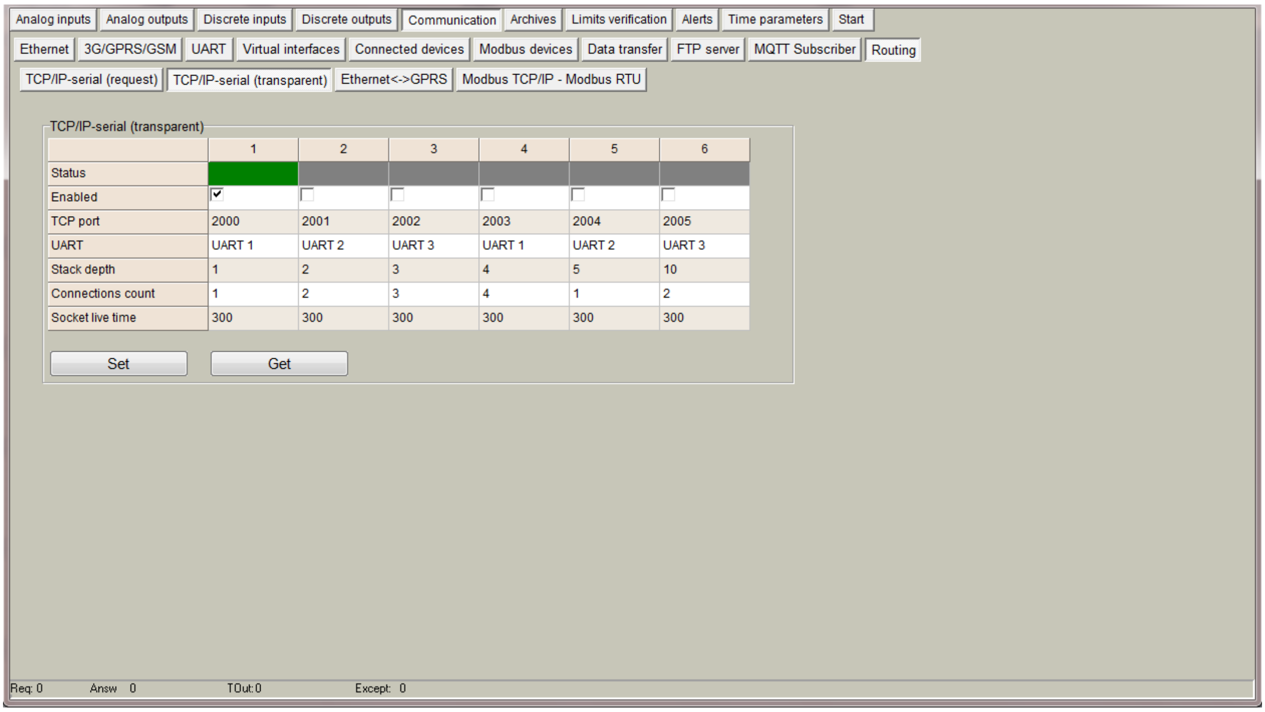
Transparent routing diagram


TCP/IP side
UART side
Packet received from TCP socket
Send data (TCP/IP packet application data)
Send data to UART
Packet send to TCP socket
Send data (UART data)
Received data from UART
Packet received from TCP socket
Send data (TCP/IP packet application data)
Send data to UART
Packet received from TCP socket
Send data (TCP/IP packet application data)
Send data to UART
Packet send to TCP socket
Send data (UART data)
Received data from UART

Transparent TCP server configuration

Transparent TCP server configuration is done in the *"Communication/Routing/TCP/IP-Serial (Transparent)"* tab. The device can open up to 6 different TCP ports for data transfer to the associated UART.

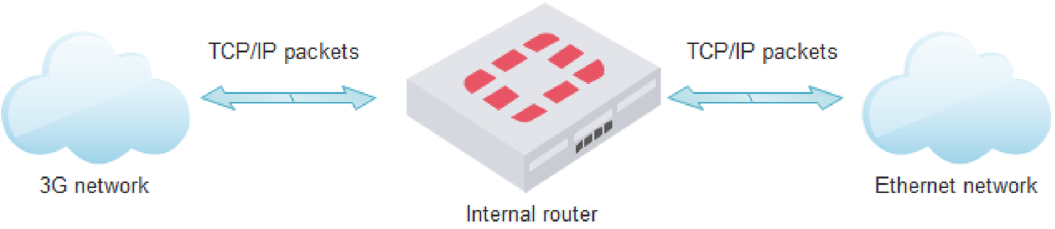
- **"Status"** – Green indicates the server is working; grey indicates the server is stopped.
- **"Enabled" checkbox** – Enables/disables the appropriate transparent TCP server.
- **"TCP port"** – Internal socket TCP port (1-65535). The device waits for a connection to the configured ports and opens a data transfer channel with the appropriate UART.
- **"UART"** – The UART associated with the corresponding TCP port (UART1 - UART3).
- **"Stack depth"** – The number of packets that can be processed simultaneously (1-10). The TCP/IP network is much faster than the serial UART, so packets can be queued and sent to the UART one by one. This parameter indicates how many packets can be processed at the same time.
- **"Connection count"** – Indicates how many connections can be established to the appropriate transparent TCP server (1-4). TCP/IP application data will be transmitted from all TCP sockets to the UART, and UART data will be returned to all TCP sockets. If the maximum connection count is reached, new connections will be refused.
- **"Socket live time"** – Socket timeout in seconds (60-65535 seconds). If no data is transferred within the configured time, the device will automatically close the socket.
- **"Set" button** – Writes the configuration to the device.
- **"Get" button** – Reads the configuration from the device.



	1	2	3	4	5	6
Status	🟩	▨	▨	▨	▨	▨
Enabled	☑	□	□	□	□	□
TCP port	2000	2001	2002	2003	2004	2005
UART	UART 1	UART 2	UART 3	UART 1	UART 2	UART 3
Stack depth	1	2	3	4	5	10
Connections count	1	2	3	4	1	2
Socket live time	300	300	300	300	300	300

3.7 Router 4G/3G/GPRS<->Ethernet

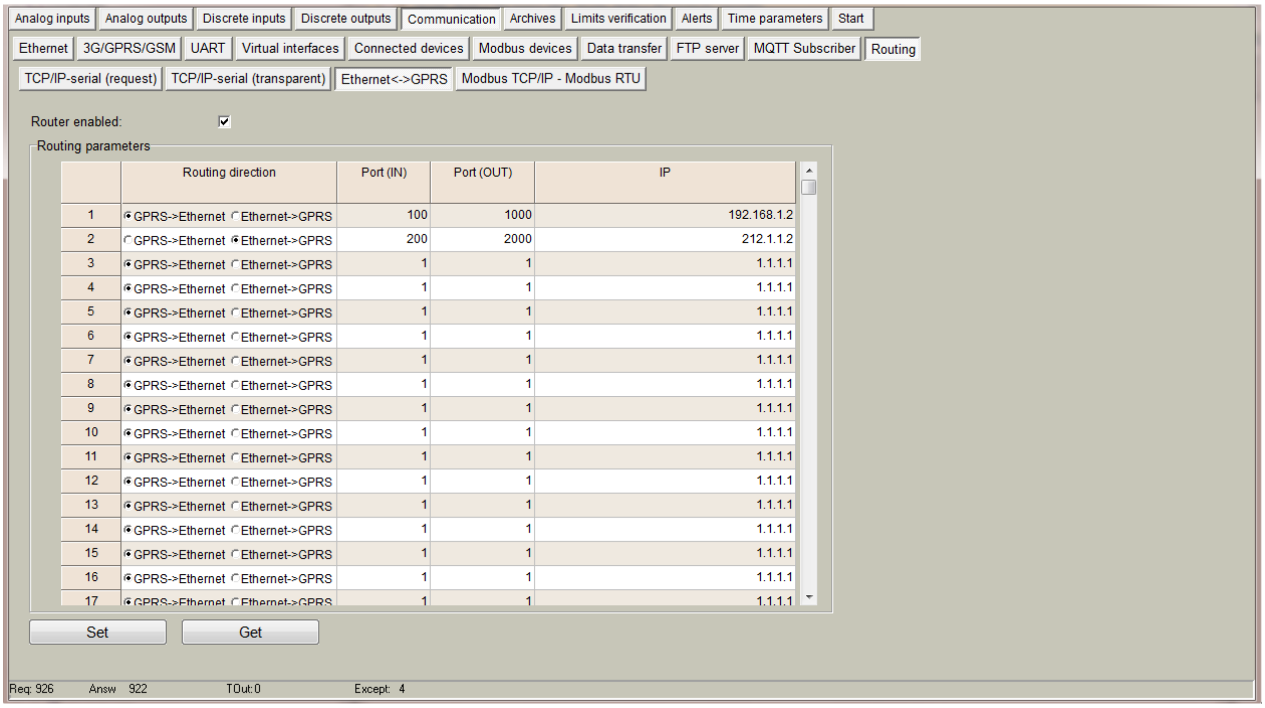
The purpose of the 4G/3G/GPRS <-> Ethernet router is to access Ethernet-connected devices from the 4G/3G/GPRS network. It routes TCP/IP packets from the 4G/3G/GPRS network to Ethernet and back.



Router configuration

4G/3G/GPRS <-> Ethernet router configuration is done in the *"Communication/Routing/Ethernet <-> GPRS"* tab. The device can route up to 20 different TCP ports.

- **"Routing direction"** – Indicates how packets are routed, either from GPRS to Ethernet or from Ethernet to GPRS. The routing direction determines how the *"Port (IN)"*, *"Port (Out)"*, and *"IP"* values are used.
- **"Port (IN)"** – The device's incoming port. If the direction is GPRS to Ethernet, this port will be the incoming port from the 4G/3G/GPRS network. If the direction is Ethernet to GPRS, this port will be the incoming port from the Ethernet network.
- **"Port (Out)"** – The device's outgoing port. If the direction is GPRS to Ethernet, this port will be the outgoing port to the Ethernet network. If the direction is Ethernet to GPRS, this port will be the outgoing port to the GPRS network.
- **"IP"** – The device's outgoing IP address. If the direction is GPRS to Ethernet, this IP will be the outgoing IP to the Ethernet network. If the direction is Ethernet to GPRS, this IP will be the outgoing IP to the GPRS network.
- **"Set" button** – Writes the configuration to the device.
- **"Get" button** – Reads the configuration from the device.



Router enabled: ☑

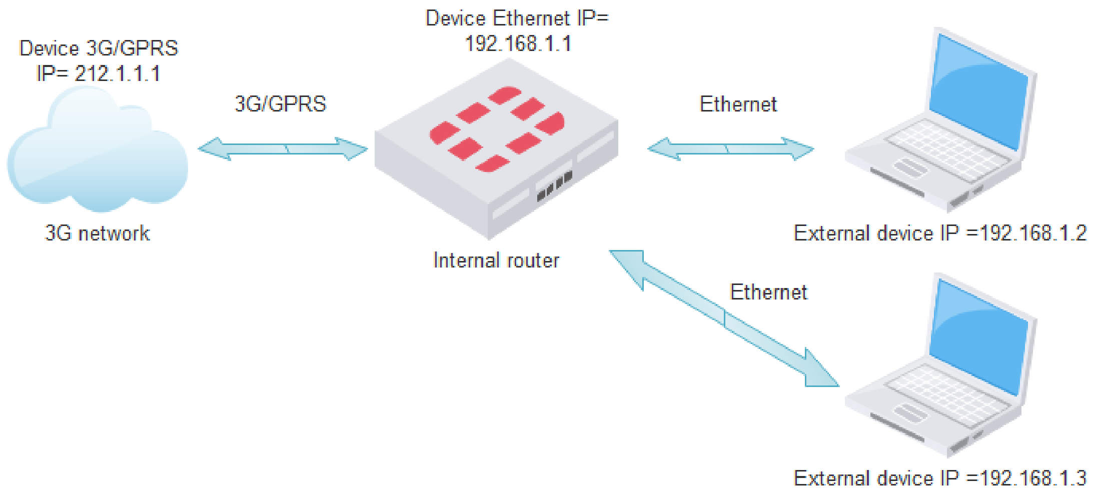
Routing parameters

	Routing direction	Port (IN)	Port (OUT)	IP
1	● GPRS->Ethernet ○ Ethernet->GPRS	100	1000	192.168.1.2
2	○ GPRS->Ethernet ● Ethernet->GPRS	200	2000	212.1.1.2
3	● GPRS->Ethernet ○ Ethernet->GPRS	1	1	1.1.1.1
4	● GPRS->Ethernet ○ Ethernet->GPRS	1	1	1.1.1.1
5	● GPRS->Ethernet ○ Ethernet->GPRS	1	1	1.1.1.1
6	● GPRS->Ethernet ○ Ethernet->GPRS	1	1	1.1.1.1
7	● GPRS->Ethernet ○ Ethernet->GPRS	1	1	1.1.1.1
8	● GPRS->Ethernet ○ Ethernet->GPRS	1	1	1.1.1.1
9	● GPRS->Ethernet ○ Ethernet->GPRS	1	1	1.1.1.1
10	● GPRS->Ethernet ○ Ethernet->GPRS	1	1	1.1.1.1
11	● GPRS->Ethernet ○ Ethernet->GPRS	1	1	1.1.1.1
12	● GPRS->Ethernet ○ Ethernet->GPRS	1	1	1.1.1.1
13	● GPRS->Ethernet ○ Ethernet->GPRS	1	1	1.1.1.1
14	● GPRS->Ethernet ○ Ethernet->GPRS	1	1	1.1.1.1
15	● GPRS->Ethernet ○ Ethernet->GPRS	1	1	1.1.1.1
16	● GPRS->Ethernet ○ Ethernet->GPRS	1	1	1.1.1.1
17	● GPRS->Ethernet ○ Ethernet->GPRS	1	1	1.1.1.1

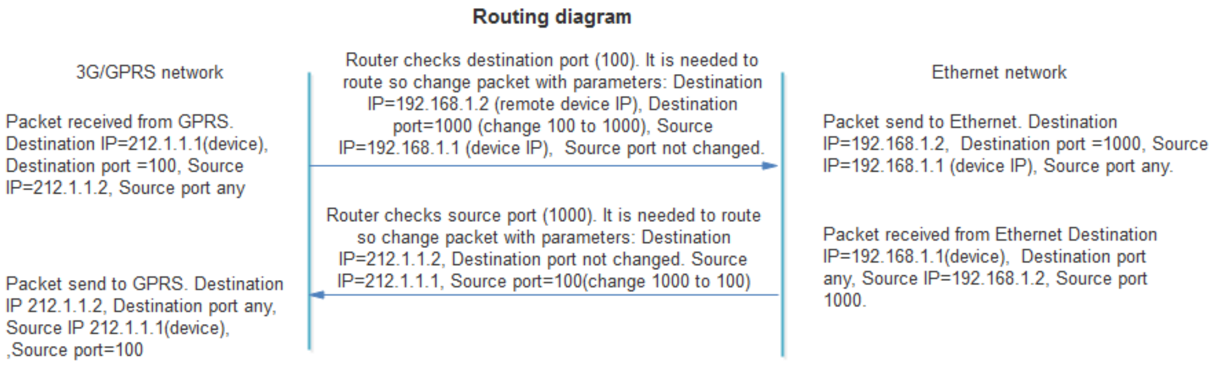
Req: 926 Answ 922 TOut:0 Except: 4

Example

In the picture above, two router lines are configured. The general connection diagram can look like this:



- **First line**: Routing from GPRS to Ethernet, with incoming port 100, outgoing port 1000, and outgoing Ethernet IP 192.168.1.2.



Routing diagram

3G/GPRS network

Ethernet network

Packet received from GPRS. Destination IP=212.1.1.1(device), Destination port =100, Source IP=212.1.1.2, Source port any

Router checks destination port (100). It is needed to route so change packet with parameters: Destination IP=192.168.1.2 (remote device IP), Destination port=1000 (change 100 to 1000), Source IP=192.168.1.1 (device IP), Source port not changed.

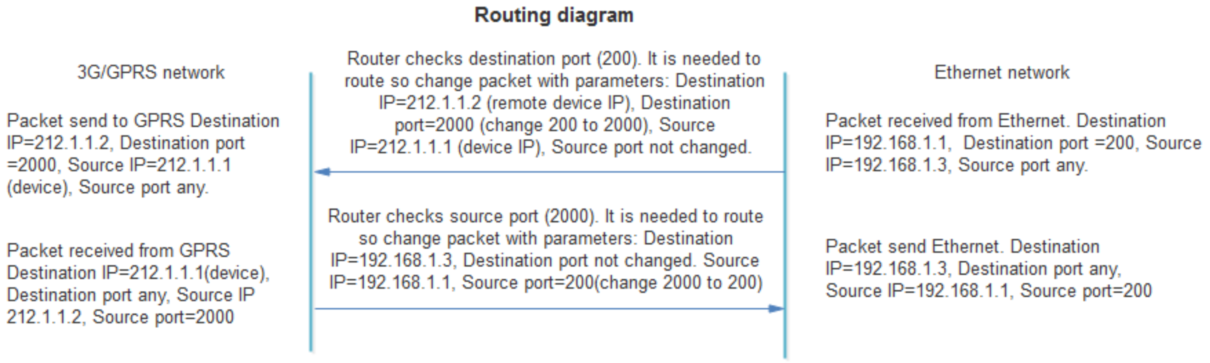
Packet send to Ethernet. Destination IP=192.168.1.2, Destination port =1000, Source IP=192.168.1.1 (device IP), Source port any.

Packet received from Ethernet Destination IP=192.168.1.1(device), Destination port any, Source IP=192.168.1.2, Source port 1000.

Router checks source port (1000). It is needed to route so change packet with parameters: Destination IP=212.1.1.2, Destination port not changed. Source IP=212.1.1.1, Source port=100(change 1000 to 100)

Packet send to GPRS. Destination IP 212.1.1.2, Destination port any, Source IP 212.1.1.1(device), ,Source port=100

- **Second line**: Routing from Ethernet to GPRS, with incoming port 200, outgoing port 2000, and outgoing GPRS IP 212.1.1.2.



Routing diagram

3G/GPRS network

Ethernet network

Packet received from Ethernet. Destination IP=192.168.1.1, Destination port =200, Source IP=192.168.1.3, Source port any.

Router checks destination port (200). It is needed to route so change packet with parameters: Destination IP=212.1.1.2 (remote device IP), Destination port=2000 (change 200 to 2000), Source IP=212.1.1.1 (device IP), Source port not changed.

Packet send to GPRS Destination IP=212.1.1.2, Destination port =2000, Source IP=212.1.1.1 (device), Source port any.

Packet received from GPRS Destination IP=212.1.1.1(device), Destination port any, Source IP 212.1.1.2, Source port=2000

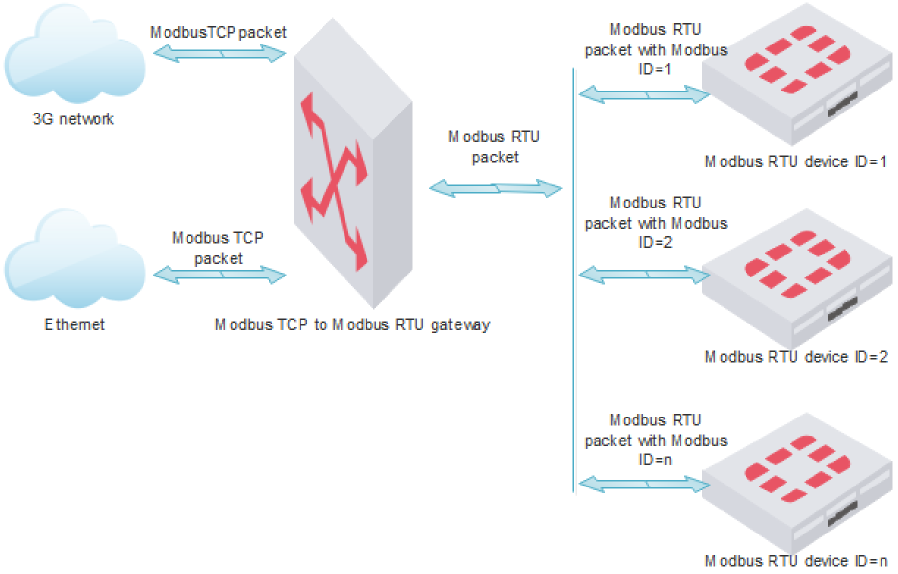
Router checks source port (2000). It is needed to route so change packet with parameters: Destination IP=192.168.1.3, Destination port not changed. Source IP=192.168.1.1, Source port=200(change 2000 to 200)

Packet send Ethernet. Destination IP=192.168.1.3, Destination port any, Source IP=192.168.1.1, Source port=200

3.8 Gateway Modbus TCP<->Modbus RTU

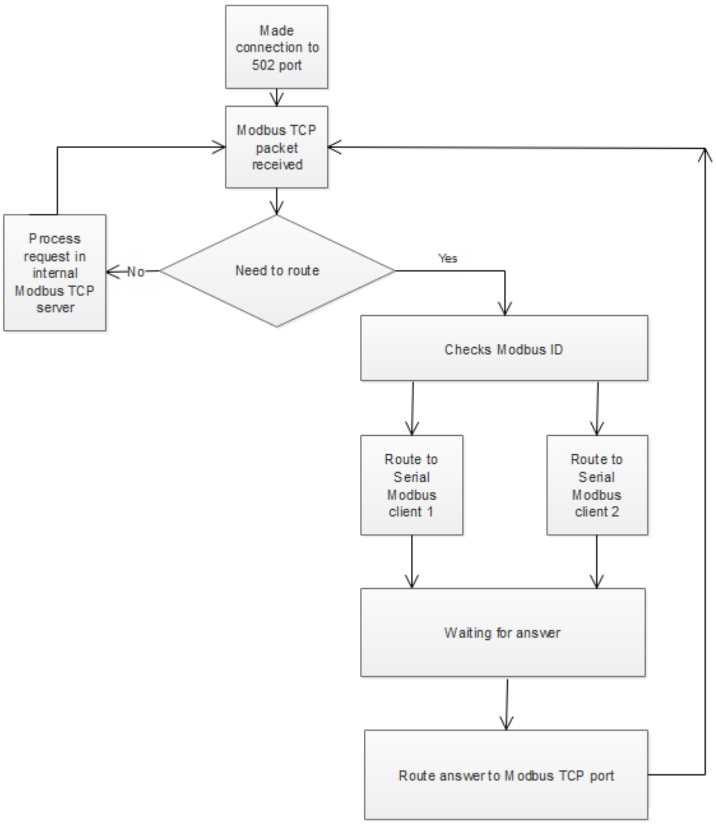
The Modbus TCP <-> Modbus RTU gateway is used to directly access a Modbus RTU device connected to our device using the Modbus TCP protocol. When the route function is used, the received Modbus TCP packet is converted to a Modbus RTU packet and sent to the appropriate Serial Modbus client. On the TCP network side, the device works as a Modbus TCP server, while on the serial side, it acts as a Modbus RTU master.

Standard network diagram:

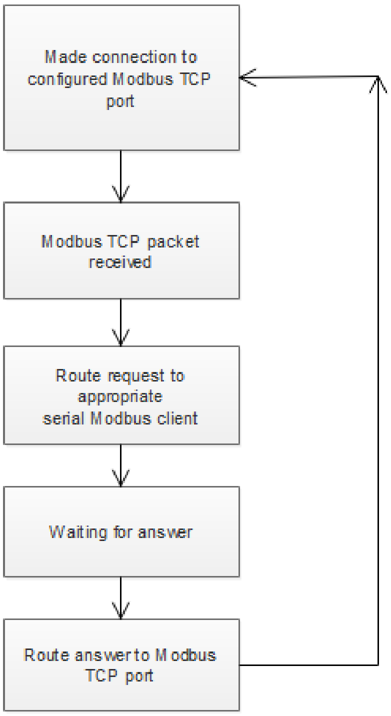

ModbusTCP packet
3G network
Modbus TCP packet
Ethernet
Modbus TCP to Modbus RTU gateway
Modbus RTU packet
Modbus RTU packet with Modbus ID=1
Modbus RTU device ID=1
Modbus RTU packet with Modbus ID=2
Modbus RTU device ID=2
Modbus RTU packet with Modbus ID=n
Modbus RTU device ID=n

There are two routing modes:

- **Route by Modbus Device ID**: The standard Modbus TCP port 502 is used for TCP connections, and packets are routed based on the Modbus ID. The working algorithm is as follows:


Made connection to 502 port
Modbus TCP packet received
Process request in internal Modbus TCP server
No
Need to route
Yes
Checks Modbus ID
Route to Serial Modbus client 1
Route to Serial Modbus client 2
Waiting for answer
Route answer to Modbus TCP port

- **Route by TCP port**: A configured TCP port is used for TCP connections, and all Modbus TCP packets are routed to the appropriate serial Modbus client. The working algorithm is as follows:


Made connection to configured Modbus TCP port
Modbus TCP packet received
Route request to appropriate serial Modbus client
Waiting for answer
Route answer to Modbus TCP port

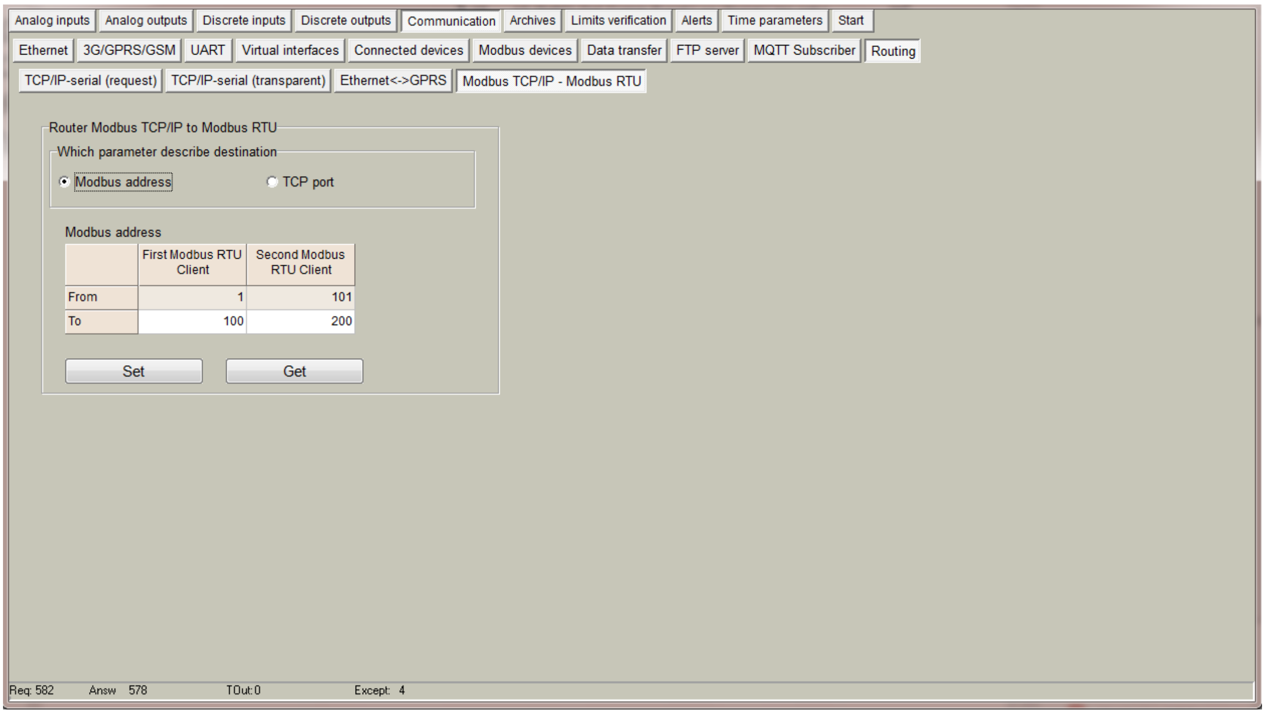
Supported Modbus functions for routing:

- **1** – Read Coils
- **2** – Read Discrete Inputs
- **3** – Read Holding Registers
- **4** – Read Input Registers
- **5** – Write Single Coil
- **6** – Write Single Register
- **16** – Write Multiple Registers
- **20** – Read File Record
- **21** – Write File Record

Gateway Modbus TCP<->Modbus RTU configuration

The Modbus TCP <-> Modbus RTU gateway configuration is done in the *"Communication/Routing/Modbus TCP/IP <-> Modbus RTU"* tab. All serial data transmission goes through the Modbus RTU clients (refer to the Modbus RTU client section to understand how it is connected to the physical UART).

- **Route by Modbus Device ID**: Select the *"Modbus Address"* option in the *"Which parameter describes the destination"* section.
  - **Modbus Address table**: For each Modbus RTU client, you can define a Modbus ID list (From <-> To). All Modbus packets with a Modbus ID from this list will be routed to the appropriate Modbus RTU client. For example, packets with Modbus IDs from 1 to 100 will be routed to the first Modbus RTU client, while packets with Modbus IDs from 101 to 200 will be routed to the second Modbus RTU client. All other packets will be processed by the internal Modbus server.
  - **Set button**: Writes the configuration to the device.
  - **Get button**: Reads the configuration from the device.



Router Modbus TCP/IP to Modbus RTU

Which parameter describe destination

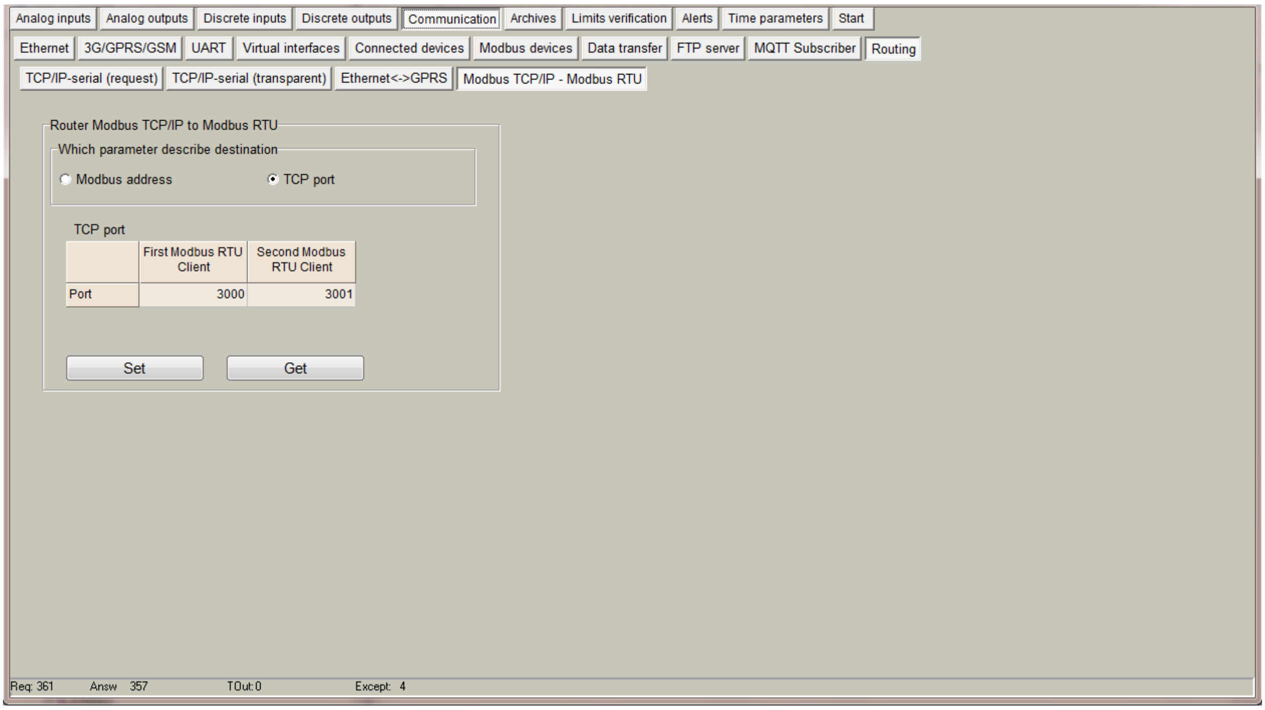
● Modbus address ○ TCP port

Modbus address

	First Modbus RTU Client	Second Modbus RTU Client
From	1	101
To	100	200

Req: 582 Answ 578 TOut:0 Except: 4

- **Route by TCP port**: Select the *"TCP Port"* option in the *"Which parameter describes the destination"* section.
    - **TCP Port table**: For each Modbus RTU client, you can define an individual TCP port. All Modbus packets sent to these TCP ports will be routed to the appropriate Modbus RTU client. For example, packets sent to TCP port 3000 will be routed to the first Modbus RTU client, and packets sent to TCP port 3001 will be routed to the second Modbus RTU client.
    - **Set button**: Writes the configuration to the device.
    - **Get button**: Reads the configuration from the device.



Router Modbus TCP/IP to Modbus RTU

Which parameter describe destination

○ Modbus address ● TCP port

TCP port

	First Modbus RTU Client	Second Modbus RTU Client
Port	3000	3001

Req: 361 Answ 357 TOut:0 Except: 4

4. Hardware

4.1 microSD

If the ordered device supports a microSD card slot, the device memory can be extended using a microSD card. The microSD card is used to store archives and generate report files for FTP and MQTT clients. The content of the microSD card can be accessed through the internal FTP server.

Installing a microSD Card

1. Remove the upper cover of the device to access the microSD card slot.
2. Push the microSD slot cover to the right to open it.
3. Insert the microSD card into the slot.
4. Push the microSD slot cover to the left to close it.

4.2 Time settings

The device has an integrated battery-backed real-time clock (RTC) with a calendar. The RTC works in UTC time, and the user can configure it to return local time based on their location (selecting the time zone and summer/winter time usage).

A time zone is a region of the globe that observes a uniform standard time for legal, commercial, and social purposes. Time zones tend to follow the boundaries of countries and their subdivisions because it is convenient for areas in close commercial or other communication to keep the same time. Most time zones on land are offset from Coordinated Universal Time (UTC) by whole numbers of hours (UTC−12 to UTC+14), but a few zones are offset by 30 or 45 minutes (e.g., Newfoundland Standard Time is UTC−03:30, Nepal Standard Time is UTC+05:45, and Indian Standard Time is UTC+05:30). For more information, check Wikipedia.

Setting Time with Configuration Tool

Time settings can be changed in the *"Time Parameters"* tab.

4.3 Ethernet configuration

The Ethernet interface is used to connect the device to Local Area Networks (LANs) and to remotely access the device. The device supports both 10 Mbps and 100 Mbps networks. The Ethernet interface is used for:

- Data transfer
- Event transfer
- Clock time synchronization
- Device configuration
- Firmware upgrade

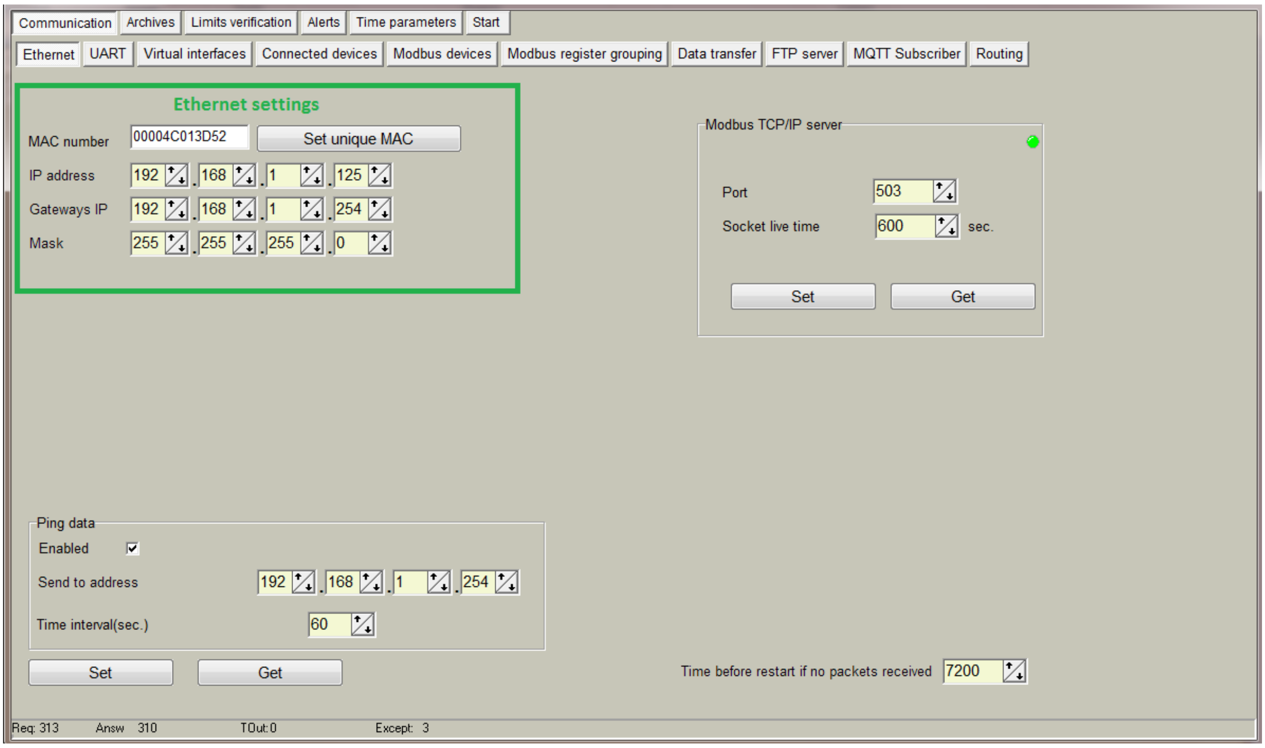
Supported Services:

- Modbus TCP/IP server
- Modbus TCP/IP client
- FTP client
- FTP server
- MQTT client
- DNS client
- SNTP
- ICMP
- Request/Answer to UART channel
- Transparent to UART channel
- Router to GPRS/4G/3G network

Ethernet configuration

Ethernet interface configuration is done in the *"Communication/Ethernet"* tab. The device does not support DHCP, so before installation, you need to set its network settings manually:

- **"MAC number"** – The device's individual MAC address.
- **"IP address"** – The device's IP address.
- **"Gateway IP"** – The gateway IP address.
- **"Mask"** – The network mask.



Ethernet settings

MAC number 00004C013D52

IP address 192.168.1.125

Gateways IP 192.168.1.254

Mask 255.255.255.0

Modbus TCP/IP server

Port 503

Socket live time 600 sec.

Ping data

Enabled ☑

Send to address 192.168.1.254

Time interval(sec.) 60

Time before restart if no packets received 7200

Req: 313 Answ 310 TOut:0 Except: 3

Default settings

Parameter name	Default value
IP address	192.168.1.125
Gateway IP	192.168.1.254
Mask	255.255.255.0

4.4 4G/3G/GPRS configuration

The 4G/3G/GPRS interface is used for:

- Data transfer
- Event transfer
- Clock time synchronization
- Device configuration
- Firmware upgrade
- Etc.

Supported Services:

- Modbus TCP/IP server
- Modbus TCP/IP client
- FTP client
- FTP server
- MQTT client
- DNS client
- SNTP
- ICMP
- Request/Answer to UART channel
- Transparent to UART channel
- Router to Ethernet network

Antenna Connection and SIM Card

Disconnect the power and connect the GSM SMA male-type antenna to the antenna connector. Then, insert a standard SIM card into the SIM card socket.

LED indicators

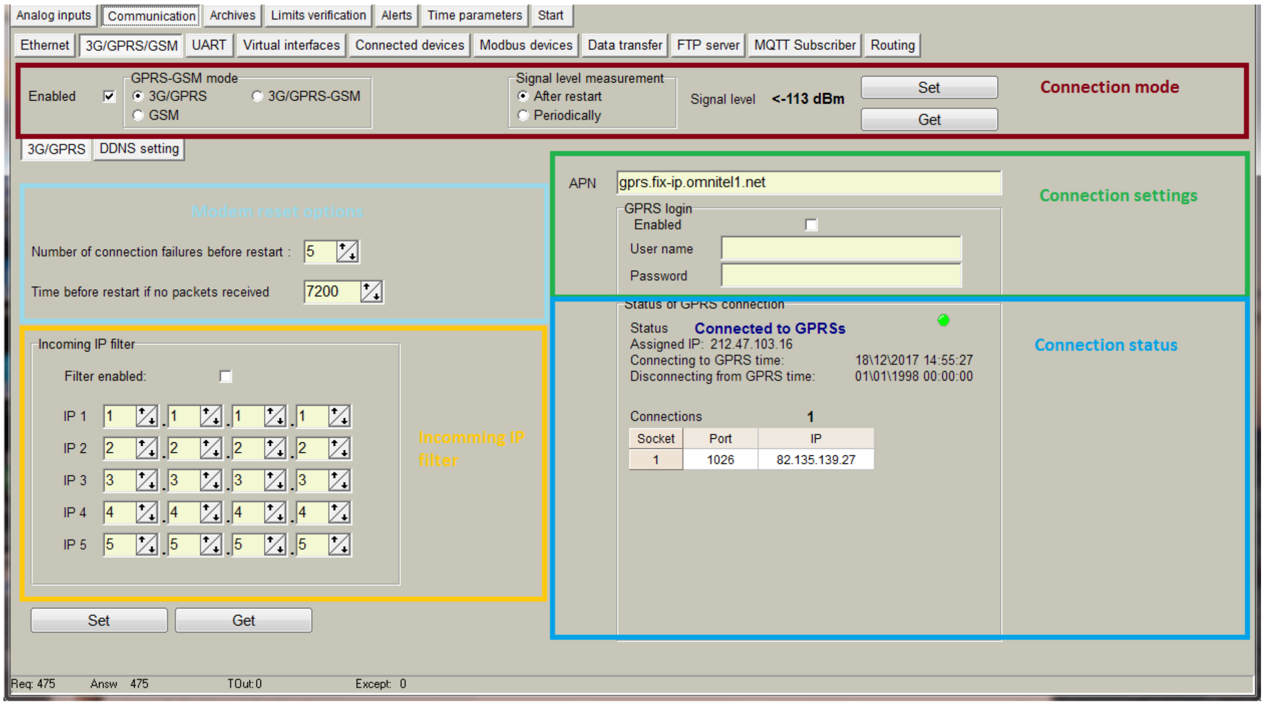
Name	Label and type	Color	Function
Status	H13, one color	Red	Always On - Searching Network/Call Connect 200ms ON, 200ms OFF - Connected to 4G/3G/GPRS network 800ms ON, 800ms OFF - Registered network Off - Power off / Sleep
TX	H9, two color	Red	Blinks - Data is sending to Modem
RX	H9, two color	Green	Blinks - Data is receiving from Modem

4G/3G/GPRS configuration

All configuration is done in the *"Communication/4G/3G/GPRS/GSM"* tab. Before configuring, you need to remove the PIN code check from your SIM card and have information such as the APN address, and if applicable, the username and password.

- **Connection Mode Section**:
  - **"Enabled" checkbox** – Enables or disables modem use.
  - **GPRS-GSM Mode**:
    - **"4G/3G/GPRS"** – Connects only in GPRS data mode.
    - **"GSM"** – Connects only in GSM mode (GSM data calls).
    - **"4G/3G/GPRS-GSM"** – Mixed mode, where the device first tries to connect to GPRS. If the connection fails, it stays in GSM mode and after a timeout, tries to connect to GPRS again.
  - **Signal Level Measurement**:
    - **"After reset"** – Measures the signal level once after modem restart.
    - **"Periodically"** – Measures the signal level every 2 seconds. This works only in GSM mode.
    - **"Signal level"** – The measured signal level. 51 dBm is the best signal, and 113 dBm is the worst signal.
  - **"Set" button** – Writes the configuration to the device.
  - **"Get" button** – Reads the configuration from the device.
- **Modem Reset Options Section**:
  - **"Number of connection failures before restart"** – Specifies how many times to attempt connecting to GPRS before restarting the modem.
  - **"Time before restart if no packets received"** – Specifies the time before restarting the modem if no IP packets are received. The time is configured in seconds (600–36000 seconds).
- **Connection Settings Section**:
  - **"APN"** – The Access Point Name (APN) is the name of a gateway between a GSM, GPRS, 4G/3G, or 4G mobile network and another computer network, frequently the public Internet. A device making a data connection must be configured with an APN to present to the carrier.
  - **"GPRS login enabled"** – Enables the use of login preferences.
  - **"User name"** – The network username.
  - **"Password"** – The network password.
- **Connection Status Section**:
  - **"Status"** – The current connection status. Available values:
    - **"Assigned IP"** – The assigned network IP address.
    - **"Connecting to GPRS time"** – The time of the last connection to the GPRS network.
    - **"Disconnecting from GPRS time"** – The time of the last disconnection from the GPRS network.
  - **"Connections" table** – Lists the current TCP/IP connections.
- **Incoming IP Filter Section**:
  If the IP filter is enabled, the device only accepts connections from IP addresses listed in the table.
  - **"Filter enabled"** – Enables/disables the incoming IP filter.
  - **IP1 - IP5** – The list of incoming IP addresses.
- **"Set" button** – Writes the configuration to the device.

- **"Get" button** – Reads the configuration from the device.



Enabled ☑

GPRS-GSM mode: ● 3G/GPRS ○ 3G/GPRS-GSM ○ GSM

Signal level measurement: ● After restart ○ Periodically

Signal level <-113 dBm

APN gprs.fix-ip.omnitel1.net

GPRS login

Enabled □

User name

Password

Number of connection failures before restart : 5

Time before restart if no packets received 7200

Incoming IP filter

Filter enabled: □

IP 1 1.1.1.1

IP 2 2.2.2.2

IP 3 3.3.3.3

IP 4 4.4.4.4

IP 5 5.5.5.5

Status of GPRS connection

Status Connected to GPRSs

Assigned IP: 212.47.103.16

Connecting to GPRS time: 18\12\2017 14:55:27

Disconnecting from GPRS time: 01\01\1998 00:00:00

Connections 1

Socket	Port	IP
1	1026	82.135.139.27

Req: 475 Answ 475 TOut:0 Except: 0

Default settings

Parameter name	Default value
4G/3G/GPRS enabled	Enabled
APN	"EnterAPN"
Login enabled	Disabled
User	""
Password	""
Incoming IP filter enabled	Disabled
Incoming IPs	0.0.0.0
Number of connection failures before restart	5
Time before restart if no packets received	7200

4.5 Serial ports

Three serial bus connections are available for the connection of RS485, RS232 or MBUS meters, Modbus devices and other devices.

Port number	Available options	Description
UART 1	RS232 or RS485	Can be used as: ● Modbus slave ● Modbus master ● Mbus meter reading (with RS232/RS485 <-> MBUS converter) ● Request/Answer channel ● Transparent channel
UART 2	MBUS up to 20 devices	Can be used as: ● MBUS meter reading ● Request/Answer channel ● Transparent channel

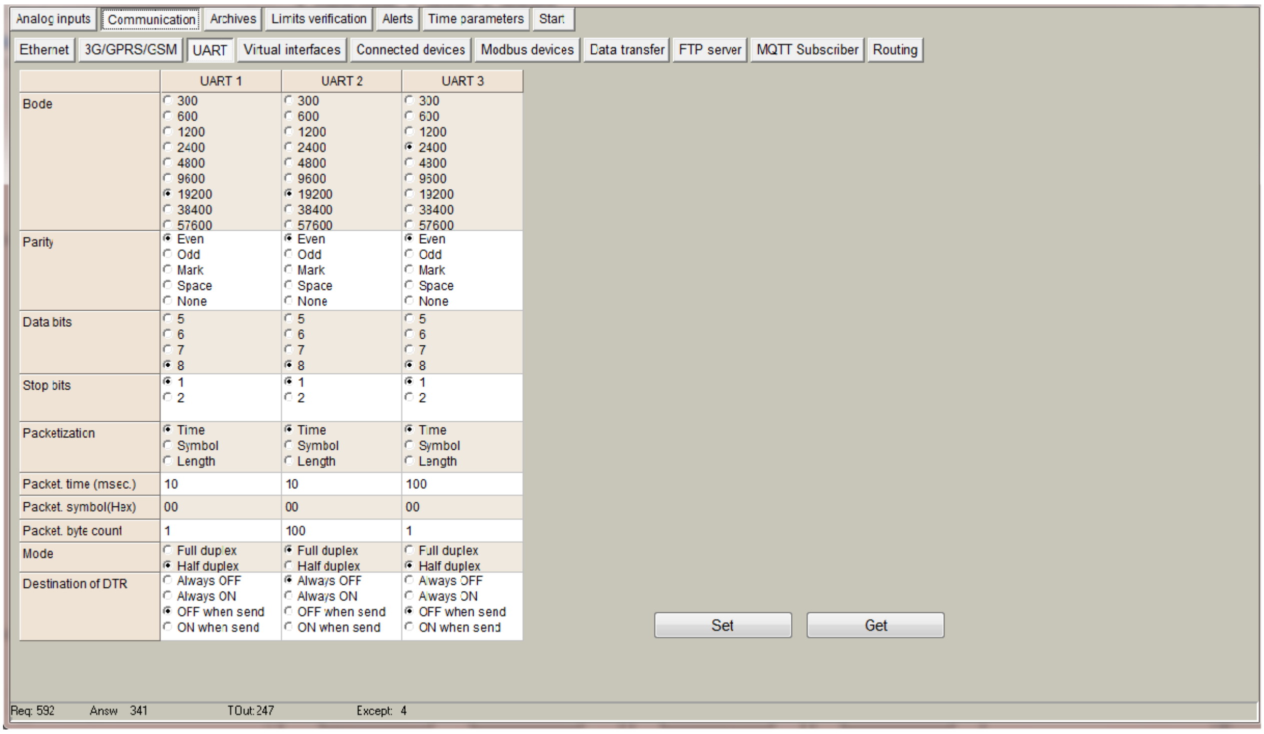
UART characteristics:

Port number	Supported baud rates	Supported parity	Supported data bits	Supported stop bits
UART 1	300 - 57600	Even, Odd, Mark, Space, None	5,6,7,8	1,2
UART 2	300 - 57600	Even, Odd, Mark, Space, None	5,6,7,8	1,2

UART settings

The UART interface configuration is done in the *"Communication/UART"* tab.

- **"Baud"** – The appropriate UART baud rate.
- **"Parity"** – The appropriate UART parity.
- **"Data bits"** – The appropriate UART data bits.
- **"Stop bits"** – The appropriate UART stop bits.
- **"Packetization"** – Data collection through the serial interface based on the following principles:
  - **"Time"** – Captures the accepted packet if the timeout after the last received byte is greater than the configured *"Packet time (msec)"*. Time is in milliseconds.
  - **"Symbol"** – Captures the accepted packet if the last received byte equals the configured *"Packet symbol (Hex)"*.
  - **"Length"** – Captures the accepted packet if the received byte count equals the configured *"Packet byte count"*.
- **"Packet time (msec)"** – Packetization timeout in milliseconds. Used when time-based packetization is selected.
- **"Packet symbol (Hex)"** – Packetization end symbol. Used when symbol-based packetization is selected.
- **"Packet byte count"** – The number of bytes in the packet. Used when length-based packetization is selected.
- **"Mode"** – Types of duplex communication systems:
  - **"Full duplex"** – In a full-duplex system, both parties can communicate with each other simultaneously.
  - **"Half duplex"** – In a half-duplex system, each party can communicate with the other, but not simultaneously; communication occurs in one direction at a time.
- **"Destination of DTR"** – The purpose of the extra UART signal DTR. For RS485, always set this signal to *"OFF when sending"*:
  - **"Always OFF"** – DTR signal is always in the OFF state.
  - **"Always ON"** – DTR signal is always in the ON state.
  - **"OFF when send"** – DTR signal is set to OFF when data is being sent; otherwise, DTR remains in the ON state.
  - **"ON when send"** – DTR signal is set to ON when data is being sent; otherwise, DTR remains in the OFF state.
- **"Set" button** – Writes the configuration to the device.
- **"Get" button** – Reads the configuration from the device.



	UART 1	UART 2	UART 3
Bode	○ 300 ○ 600 ○ 1200 ○ 2400 ○ 4800 ○ 9600 ● 19200 ○ 38400 ○ 57600	○ 300 ○ 600 ○ 1200 ○ 2400 ○ 4800 ○ 9600 ● 19200 ○ 38400 ○ 57600	○ 300 ○ 600 ○ 1200 ● 2400 ○ 4300 ○ 9500 ○ 19200 ○ 33400 ○ 57600
Parity	● Even ○ Odd ○ Mark ○ Space ○ None	● Even ○ Odd ○ Mark ○ Space ○ None	● Even ○ Odd ○ Mark ○ Space ○ None
Data bits	○ 5 ○ 6 ○ 7 ● 8	○ 5 ○ 6 ○ 7 ● 8	○ 5 ○ 6 ○ 7 ● 8
Stop bits	● 1 ○ 2	● 1 ○ 2	● 1 ○ 2
Packetization	● Time ○ Symbol ○ Length	● Time ○ Symbol ○ Length	● Time ○ Symbol ○ Length
Packet. time (msec.)	10	10	100
Packet. symbol(Hex)	00	00	00
Packet. byte count	1	100	1
Mode	○ Full duplex ● Half duplex	● Full duplex ○ Half duplex	○ Full duplex ● Half duplex
Destination of DTR	○ Always OFF ○ Always ON ● OFF when send ○ ON when send	● Always OFF ○ Always ON ○ OFF when send ○ ON when send	○ Always OFF ○ Always ON ● OFF when send ○ ON when send

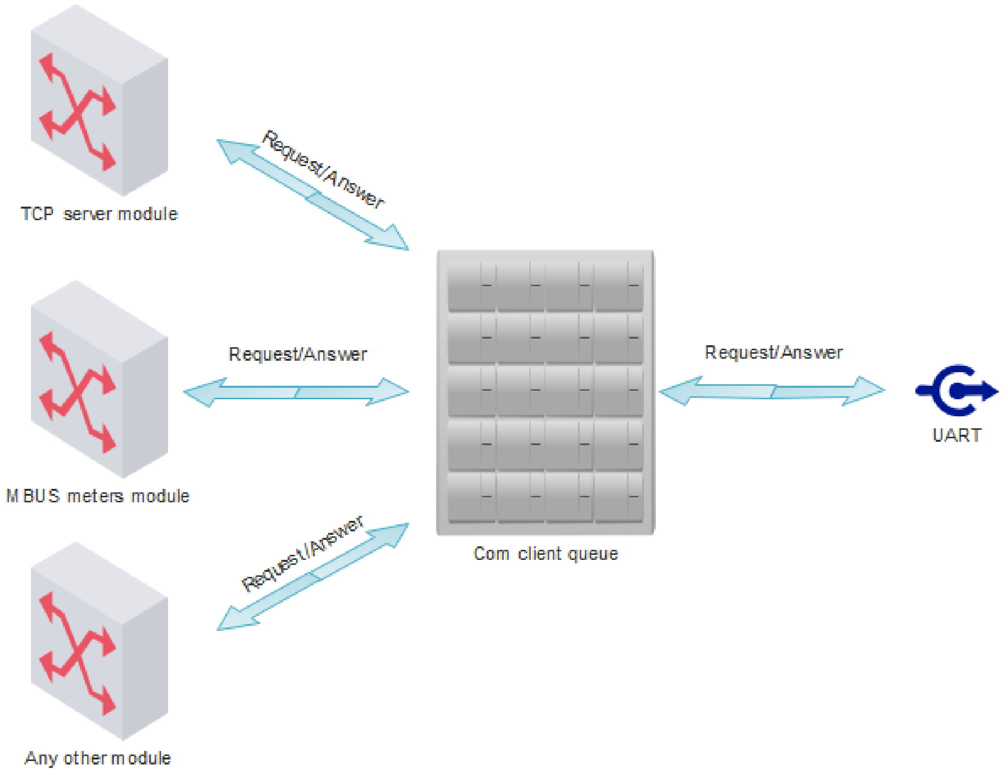
Req: 592 Answ 341 TOut:247 Except: 4

5. Virtual interfaces

5.1 Virtual COM clients

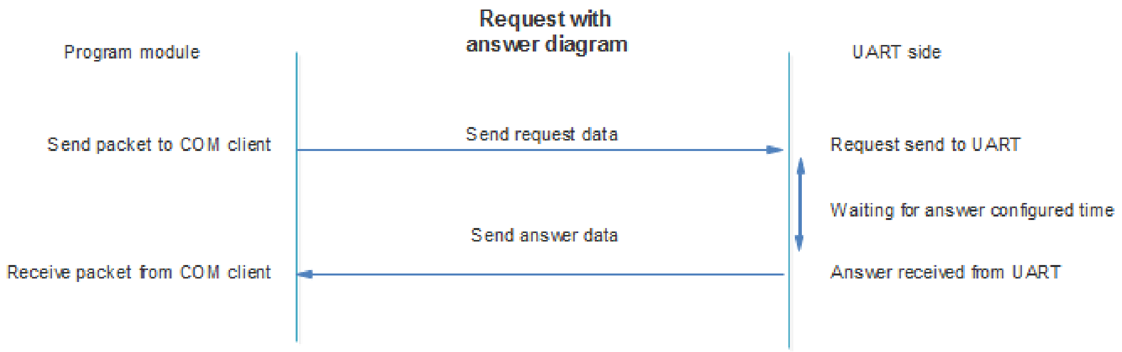
A COM client is a virtual interface between program modules and physical UARTs. The COM client allows more than one module to access the UART simultaneously, send data, and receive a response.
COM clients are used with the following modules:

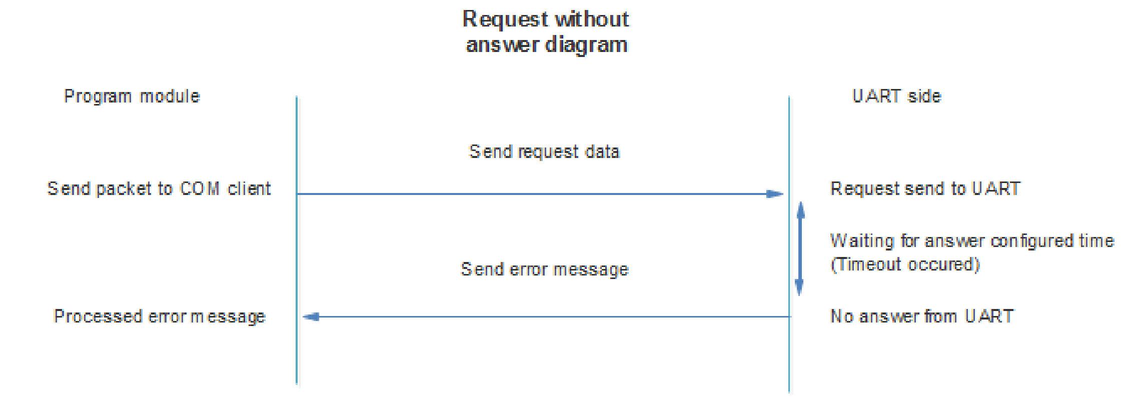
- TCP server
- MBUS meters read module
- Heat meters read module
- Electricity meters read module
- Other modules



The COM client places requests in a queue and sends them to the UART when it is free. After the request is sent, the COM client waits for a response for the configured time and then returns it to the source module. If no response is received, the COM client informs the source module of the error (No data received).

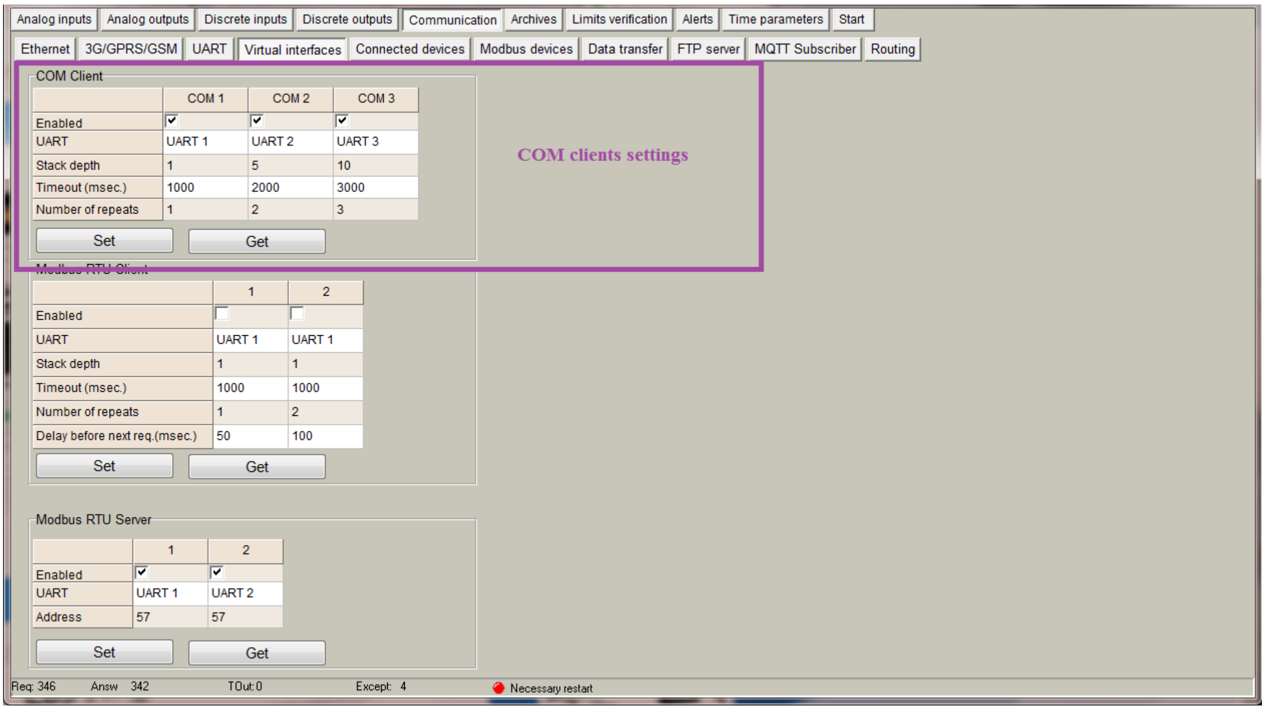
⚠ **If the response is received after the COM client's timeout, the data is lost.**


Request with answer diagram
Program module
UART side
Send packet to COM client
Send request data
Request send to UART
Waiting for answer configured time
Send answer data
Receive packet from COM client
Answer received from UART



Virtual COM clients configuration
Virtual COM clients configuration is done in the *"Communication/Virtual Interfaces"* tab. The device can have up to 3 COM clients associated with different physical UARTs.

- **"Enabled" checkbox** – Enables/disables the appropriate COM client.
- **"UART"** – The physical UART associated with the appropriate COM client (UART1, UART2, UART3). The same UART cannot be used in other virtual interfaces (e.g., another COM client, Modbus RTU client, or Modbus RTU server). In the example below, we see a bad configuration where UART1 and UART2 are used in both COM clients and Modbus RTU servers. If COM clients are used, Modbus RTU servers must be disabled.
- **"Stack depth"** – The COM client queue length (1-10). This parameter determines how many packets can be processed at the same time.
- **"Timeout (msec)"** – The wait time for a response from the UART (in milliseconds, 1-30000).
- **"Number of repeats"** – Determines how many times the request will be sent if no response is received.
- **"Set" button** – Writes the configuration to the device.
- **"Get" button** – Reads the configuration from the device.



COM Client

	COM 1	COM 2	COM 3
Enabled	☑	☑	☑
UART	UART 1	UART 2	UART 3
Stack depth	1	5	10
Timeout (msec.)	1000	2000	3000
Number of repeats	1	2	3

Modbus RTU Client

	1	2
Enabled	☐	☐
UART	UART 1	UART 1
Stack depth	1	1
Timeout (msec.)	1000	1000
Number of repeats	1	2
Delay before next req.(msec.)	50	100

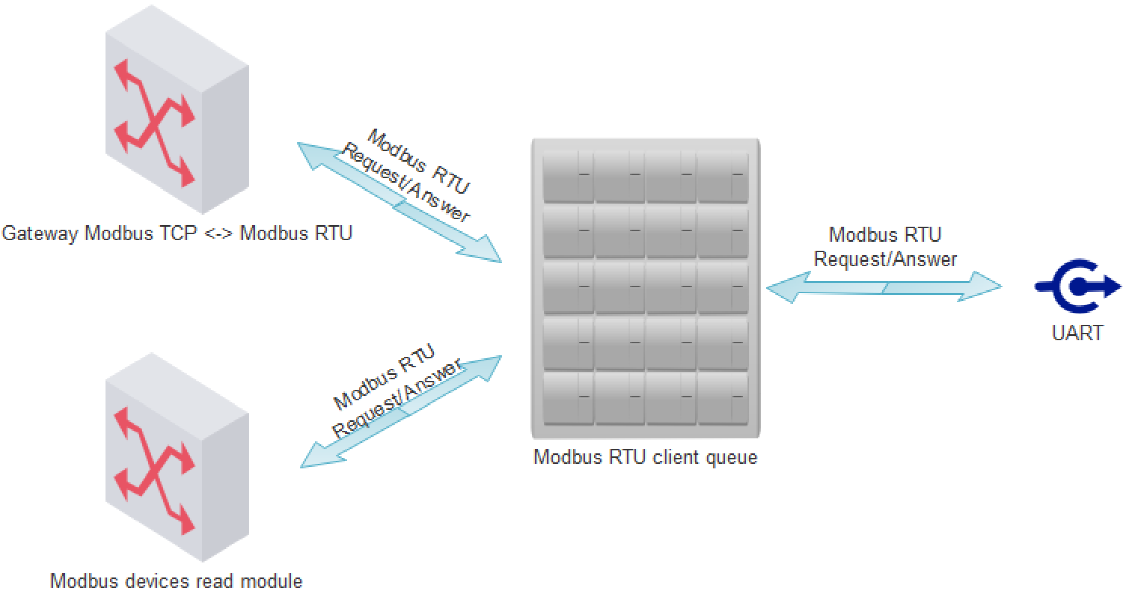
Modbus RTU Server

	1	2
Enabled	☑	☑
UART	UART 1	UART 2
Address	57	57

Req: 346 Answ 342 TOut:0 Except: 4 ● Necessary restart

5.2 *Modbus RTU clients*

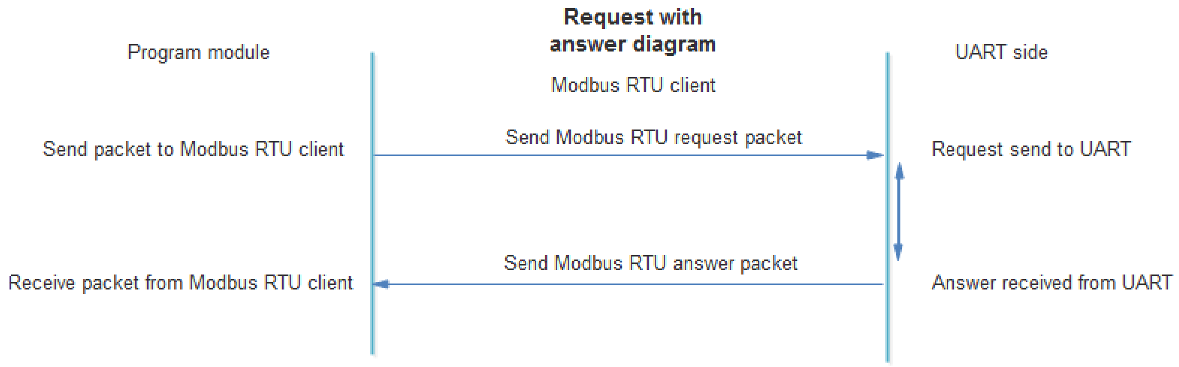
A Modbus RTU client is a virtual interface used to associate Modbus RTU devices connected to a physical UART with internal Modbus modules. It allows more than one module to access the UART simultaneously to send Modbus requests and receive responses.

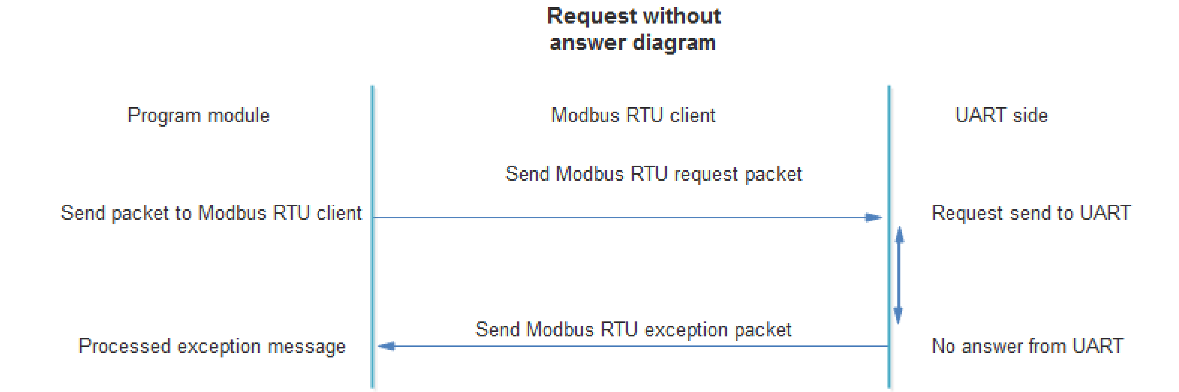

Gateway Modbus TCP <-> Modbus RTU
Modbus RTU Request/Answer
Modbus RTU Request/Answer
Modbus RTU client queue
Modbus RTU Request/Answer
UART
Modbus devices read module

Modbus RTU clients are used with the following modules:

- Gateway Modbus TCP <-> Modbus RTU
- Modbus devices read module

The Modbus RTU client places requests in a queue and sends them to the UART when it is free. After the request is sent, the Modbus RTU client waits for a response for the configured time and then returns it to the source module. If no response is received, the Modbus RTU client returns an exception to the source module.


Request with answer diagram
Program module
Modbus RTU client
UART side
Send packet to Modbus RTU client
Send Modbus RTU request packet
Request send to UART
Receive packet from Modbus RTU client
Send Modbus RTU answer packet
Answer received from UART

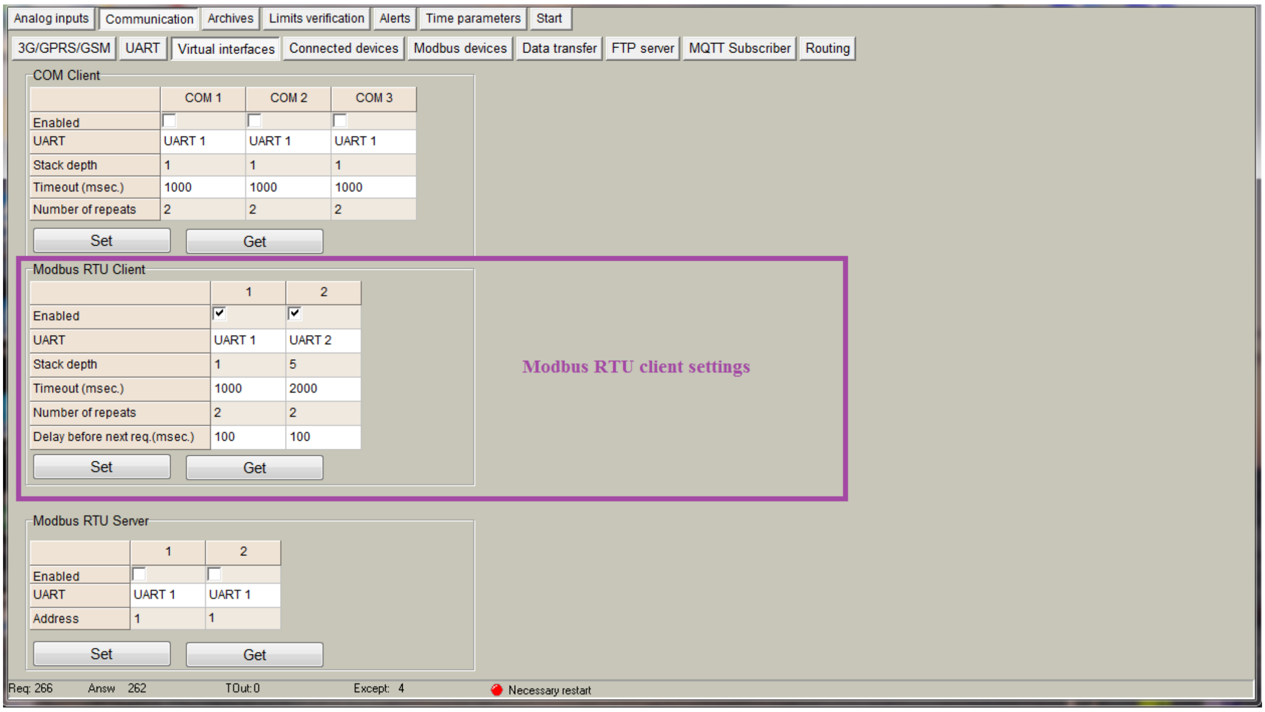

Request without answer diagram
Program module
Modbus RTU client
UART side
Send Modbus RTU request packet
Send packet to Modbus RTU client
Request send to UART
Processed exception message
Send Modbus RTU exception packet
No answer from UART

If the response is received after the Modbus RTU client's timeout, the data is lost.

Modbus RTU clients configuration

Modbus RTU clients configuration is done in the *"Communication/Virtual Interfaces"* tab. The device can have up to 2 Modbus RTU clients associated with different physical UARTs.

- **"Enabled" checkbox** – Enables/disables the appropriate Modbus RTU client.
- **"UART"** – The physical UART associated with the appropriate Modbus RTU client (UART1, UART2, UART3). The same UART cannot be used in another virtual interface (e.g., another COM client, Modbus RTU client, or Modbus RTU server).
- **"Stack depth"** – The Modbus RTU client queue length (1-10). This parameter determines how many packets can be processed at the same time.
- **"Timeout (msec)"** – The wait time for a response from the UART (in milliseconds, 1-30000).
- **"Number of repeats"** – Determines how many times the request will be sent if no response is received.
- **"Delay before next req (msec)"** – The time between requests (in milliseconds, 1-10000). The next request will be sent only after the configured timeout.
- **"Set" button** – Writes the configuration to the device.
- **"Get" button** – Reads the configuration from the device.



COM Client

	COM 1	COM 2	COM 3
Enabled	□	□	□
UART	UART 1	UART 1	UART 1
Stack depth	1	1	1
Timeout (msec.)	1000	1000	1000
Number of repeats	2	2	2

Modbus RTU Client

	1	2
Enabled	☑	☑
UART	UART 1	UART 2
Stack depth	1	5
Timeout (msec.)	1000	2000
Number of repeats	2	2
Delay before next req.(msec.)	100	100

Modbus RTU client settings

Modbus RTU Server

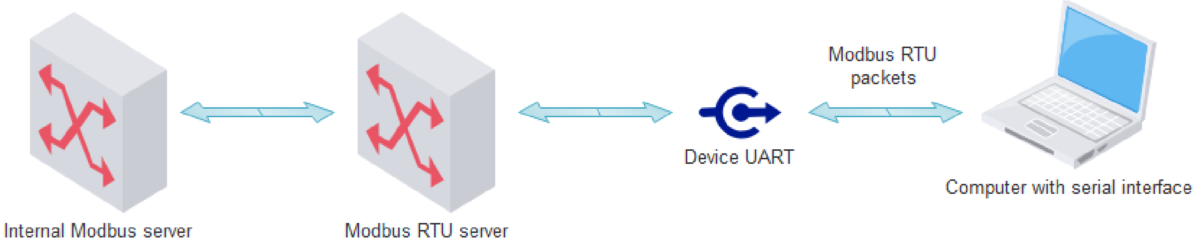
	1	2
Enabled	□	□
UART	UART 1	UART 1
Address	1	1

Req: 266 Answ 262 TOut:0 Except: 4 ● Necessary restart

5.3 Modbus RTU servers

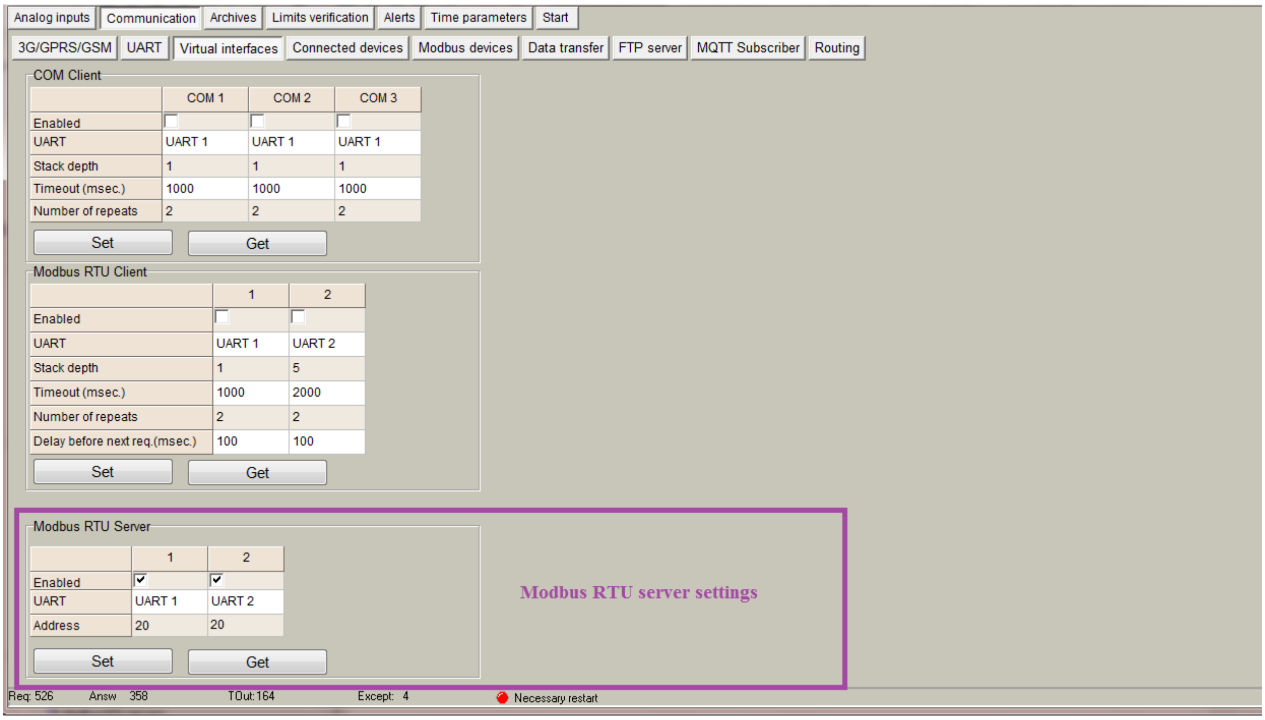
A Modbus RTU server is a virtual interface between the internal Modbus server and physical UARTs. It allows users to read the device's internal Modbus register area using a serial connection.
Modbus RTU servers can be used for:

- Reading the device's internal Modbus registers from any Modbus RTU master.
- Configuring the device using configuration tool software.



Modbus RTU server configuration
Modbus RTU server configuration is done in the *"Communication/Virtual Interfaces"* tab. The device can have up to 2 Modbus RTU servers associated with different physical UARTs.

- **"Enabled" checkbox** – Enables/disables the appropriate Modbus RTU server.
- **"UART"** – The physical UART associated with the appropriate Modbus RTU server (UART1, UART2, UART3). The same UART cannot be used in another virtual interface (e.g., another COM client, Modbus RTU client, or Modbus RTU server).
- **"Address"** – The accepted Modbus ID. The device will respond to the configured Modbus ID (e.g., if Modbus ID = 20, the device will always respond to Modbus ID = 254).
- **"Set" button** – Writes the configuration to the device.
- **"Get" button** – Reads the configuration from the device.



COM Client

	COM 1	COM 2	COM 3
Enabled	□	□	□
UART	UART 1	UART 1	UART 1
Stack depth	1	1	1
Timeout (msec.)	1000	1000	1000
Number of repeats	2	2	2

Modbus RTU Client

	1	2
Enabled	□	□
UART	UART 1	UART 2
Stack depth	1	5
Timeout (msec.)	1000	2000
Number of repeats	2	2
Delay before next req.(msec.)	100	100

Modbus RTU Server

	1	2
Enabled	☑	☑
UART	UART 1	UART 2
Address	20	20

Modbus RTU server settings

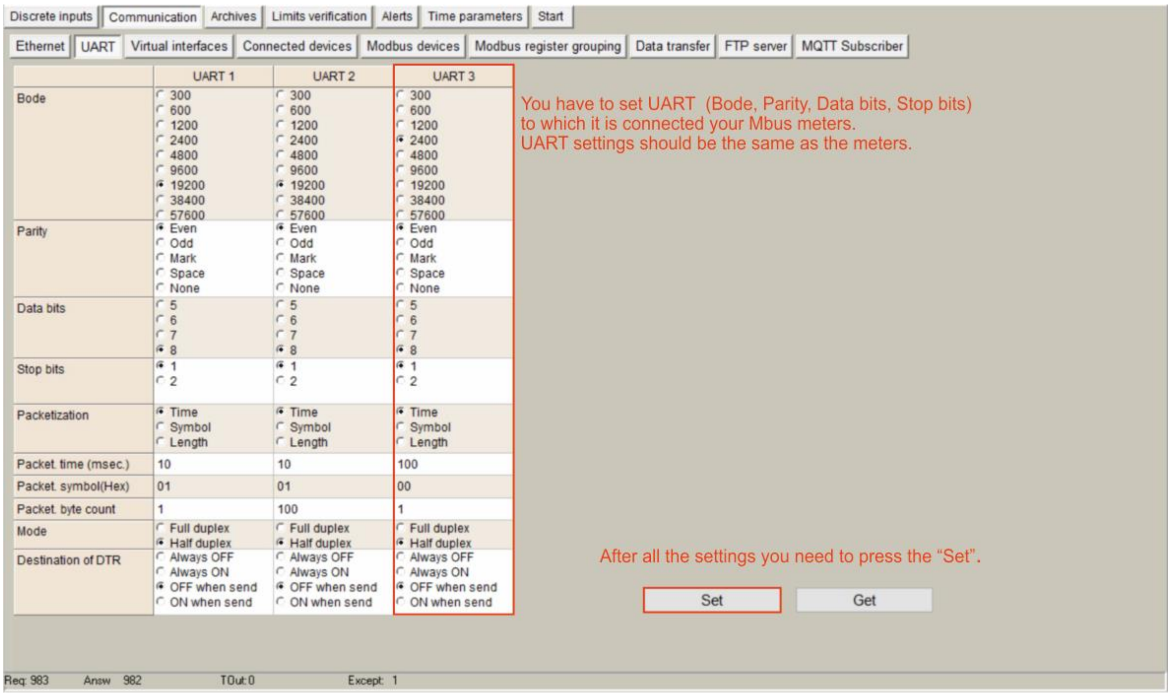
Req: 526 Answ 358 TOut:164 Except: 4 ● Necessary restart

6. Meters

6.1 M-Bus meters

How to set up controller for M-Bus devices reading.

1. You need to specify the UART settings to which the M-Bus line is connected, including baud rate, parity, data bits, etc.



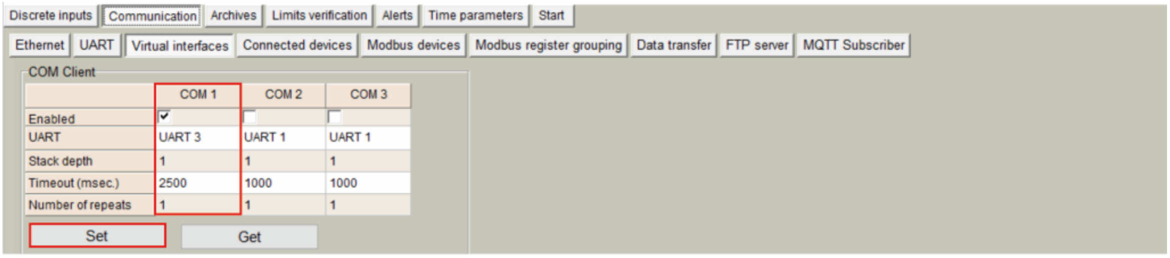
	UART 1	UART 2	UART 3
Bode	○ 300 ○ 600 ○ 1200 ○ 2400 ○ 4800 ○ 9600 ● 19200 ○ 38400 ○ 57600	○ 300 ○ 600 ○ 1200 ○ 2400 ○ 4800 ○ 9600 ● 19200 ○ 38400 ○ 57600	○ 300 ○ 600 ○ 1200 ● 2400 ○ 4800 ○ 9600 ○ 19200 ○ 38400 ○ 57600
Parity	● Even ○ Odd ○ Mark ○ Space ○ None	● Even ○ Odd ○ Mark ○ Space ○ None	● Even ○ Odd ○ Mark ○ Space ○ None
Data bits	○ 5 ○ 6 ○ 7 ● 8	○ 5 ○ 6 ○ 7 ● 8	○ 5 ○ 6 ○ 7 ● 8
Stop bits	● 1 ○ 2	● 1 ○ 2	● 1 ○ 2
Packetization	● Time ○ Symbol ○ Length	● Time ○ Symbol ○ Length	● Time ○ Symbol ○ Length
Packet. time (msec.)	10	10	100
Packet. symbol(Hex)	01	01	00
Packet. byte count	1	100	1
Mode	○ Full duplex ● Half duplex	○ Full duplex ● Half duplex	○ Full duplex ● Half duplex
Destination of DTR	○ Always OFF ○ Always ON ● OFF when send ○ ON when send	○ Always OFF ○ Always ON ● OFF when send ○ ON when send	○ Always OFF ○ Always ON ● OFF when send ○ ON when send

You have to set UART (Bode, Parity, Data bits, Stop bits) to which it is connected your Mbus meters.
UART settings should be the same as the meters.

After all the settings you need to press the "Set".

Req: 983 Answ 982 TOut:0 Except: 1

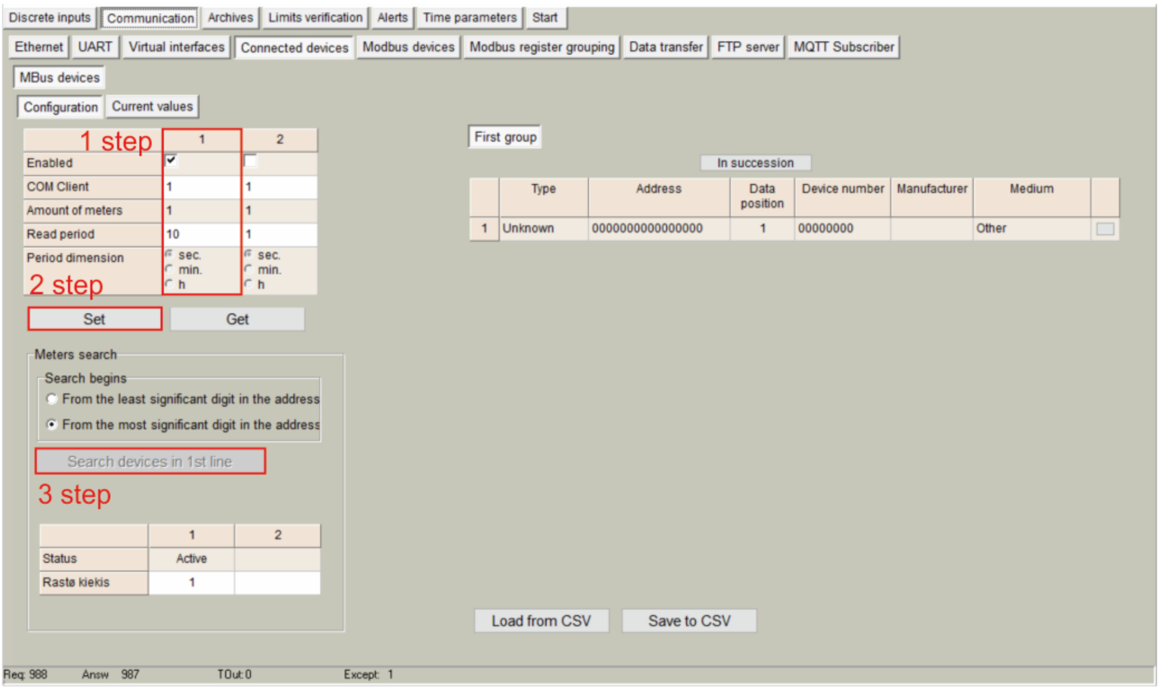
2. Enable the virtual COM port (COM) and select the UART to which the M-Bus line is connected.



COM Client

	COM 1	COM 2	COM 3
Enabled	☑	□	□
UART	UART 3	UART 1	UART 1
Stack depth	1	1	1
Timeout (msec.)	2500	1000	1000
Number of repeats	1	1	1

3. Enable the M-Bus line, enter the COM client number, and transfer (SET) these settings to the controller.
4. After that, you need to perform an M-Bus meters search. If you have your own M-Bus meters address list in a CSV file, you can upload it without performing a search.
   NOTE: If a large number of meters are connected, the search can take a while.



1 step	1	2
Enabled	☑	□
COM Client	1	1
Amount of meters	1	1
Read period	10	1
Period dimension	● sec. ○ min. ○ h	● sec. ○ min. ○ h

Meters search

Search begins

○ From the least significant digit in the address

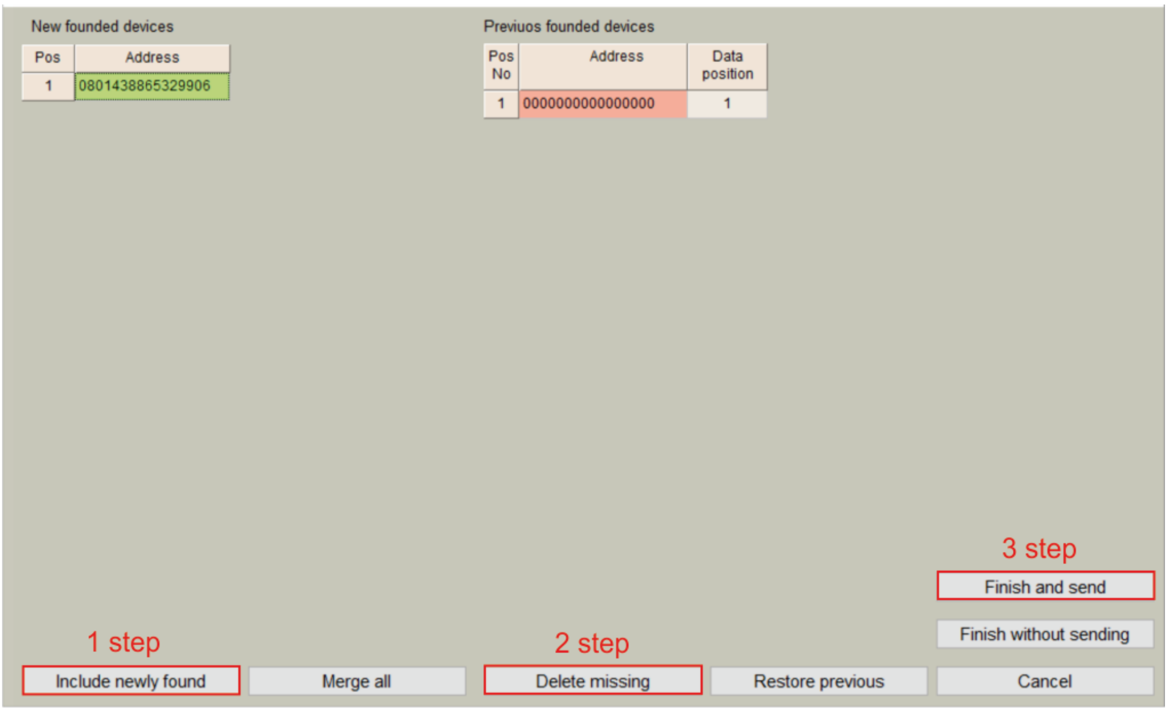
● From the most significant digit in the address

	1	2
Status	Active	
Rastø kiekis	1	

	Type	Address	Data position	Device number	Manufacturer	Medium	
1	Unknown	0000000000000000	1	00000000		Other	□

Req: 988 Answ 987 TOut:0 Except: 1

5. Once the search is complete, you will see a window displaying newly found M-Bus devices. You will need to include these newly found devices and send the information to the controller. See the picture below for guidance.



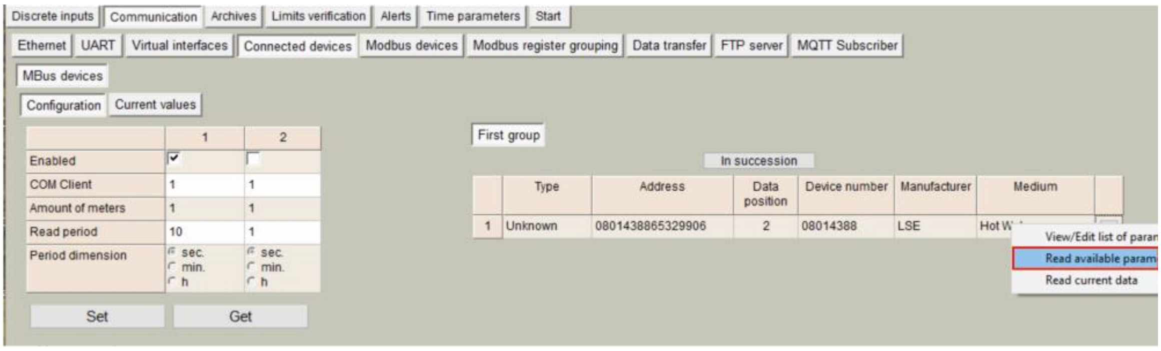
New founded devices

Pos	Address
1	0801438865329906

Previuos founded devices

Pos No	Address	Data position
1	0000000000000000	1

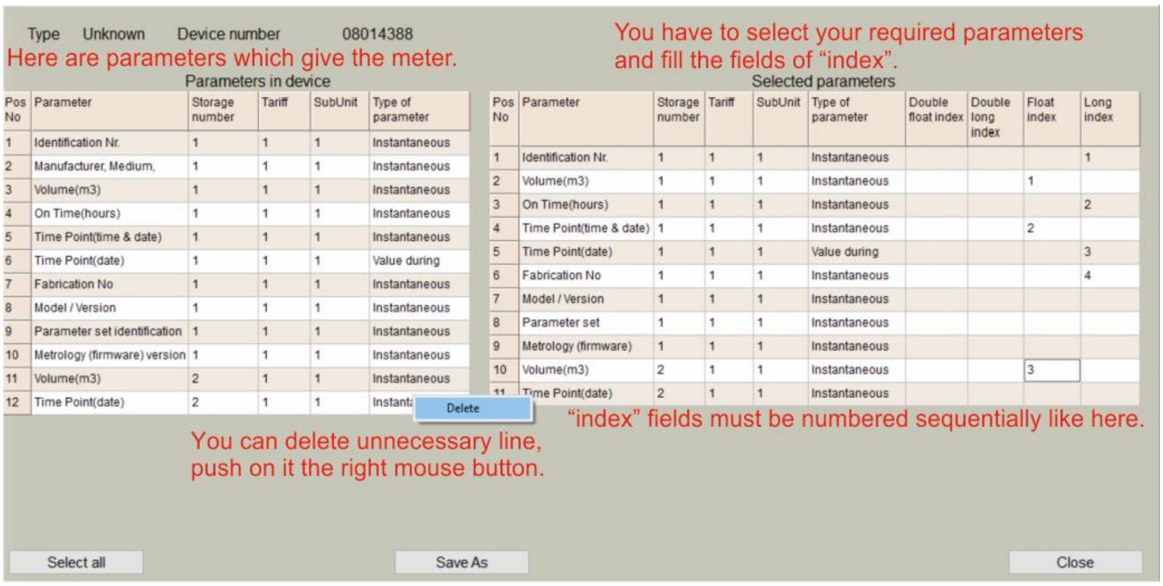
6. After adding the M-Bus meter list to the controller, you need to create a description for each meter so the controller knows which parameters to read from the M-Bus meters.



	1	2
Enabled	☑	□
COM Client	1	1
Amount of meters	1	1
Read period	10	1
Period dimension	● sec. ○ min. ○ h	● sec. ○ min. ○ h

	Type	Address	Data position	Device number	Manufacturer	Medium
1	Unknown	0801438865329906	2	08014388	LSE	Hot W

- The left side of the table shows all the parameters provided by the meter.
- On the right side, you will need to select the required parameters. If a parameter needs to be read, you must fill in the *"Index"* field. The index should be written in the column with the number format you need. The *"Index"* in the column must be numbered sequentially.
- Unnecessary parameters can be deleted from the list by right-clicking.



Type Unknown Device number 08014388

Here are parameters which give the meter.

You have to select your required parameters and fill the fields of "index".

Parameters in device

Pos No	Parameter	Storage number	Tariff	SubUnit	Type of parameter
1	Identification Nr.	1	1	1	Instantaneous
2	Manufacturer, Medium,	1	1	1	Instantaneous
3	Volume(m3)	1	1	1	Instantaneous
4	On Time(hours)	1	1	1	Instantaneous
5	Time Point(time & date)	1	1	1	Instantaneous
6	Time Point(date)	1	1	1	Value during
7	Fabrication No	1	1	1	Instantaneous
8	Model / Version	1	1	1	Instantaneous
9	Parameter set identification	1	1	1	Instantaneous
10	Metrology (firmware) version	1	1	1	Instantaneous
11	Volume(m3)	2	1	1	Instantaneous
12	Time Point(date)	2	1	1	Instant

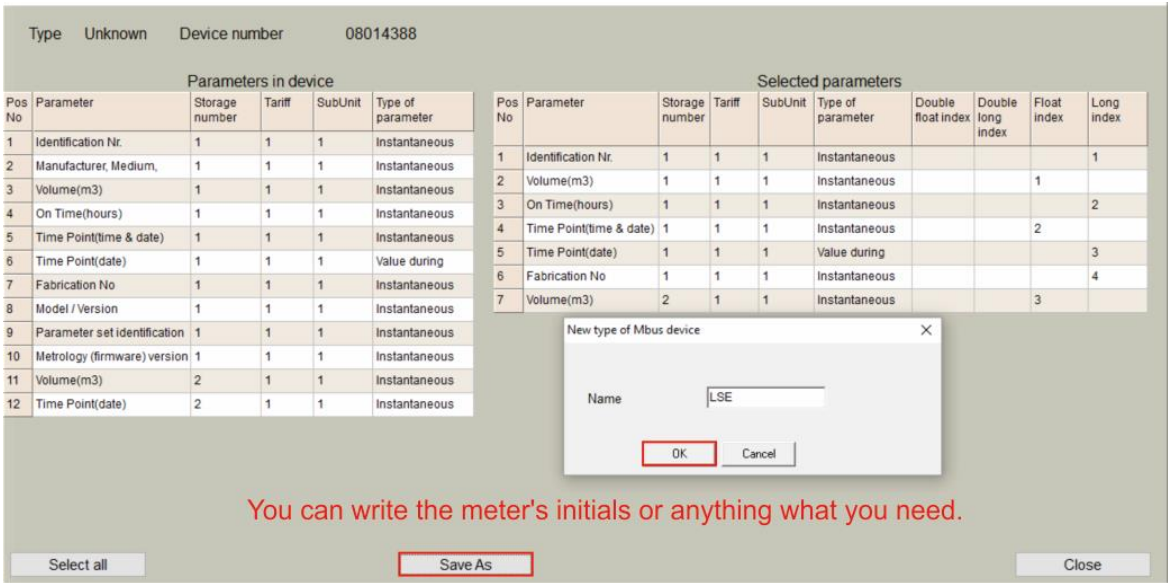
Selected parameters

Pos No	Parameter	Storage number	Tariff	SubUnit	Type of parameter	Double float index	Double long index	Float index	Long index
1	Identification Nr.	1	1	1	Instantaneous				1
2	Volume(m3)	1	1	1	Instantaneous			1	
3	On Time(hours)	1	1	1	Instantaneous				2
4	Time Point(time & date)	1	1	1	Instantaneous			2	
5	Time Point(date)	1	1	1	Value during				3
6	Fabrication No	1	1	1	Instantaneous				4
7	Model / Version	1	1	1	Instantaneous				
8	Parameter set	1	1	1	Instantaneous				
9	Metrology (firmware)	1	1	1	Instantaneous				
10	Volume(m3)	2	1	1	Instantaneous			3	
11	Time Point(date)	2	1	1	Instantaneous				

You can delete unnecessary line, push on it the right mouse button.

"index" fields must be numbered sequentially like here.

7. After selecting the required parameters, save the configuration. It is recommended to use the meter name as the description name.



Type Unknown Device number 08014388

Parameters in device

Pos No	Parameter	Storage number	Tariff	SubUnit	Type of parameter
1	Identification Nr.	1	1	1	Instantaneous
2	Manufacturer, Medium,	1	1	1	Instantaneous
3	Volume(m3)	1	1	1	Instantaneous
4	On Time(hours)	1	1	1	Instantaneous
5	Time Point(time & date)	1	1	1	Instantaneous
6	Time Point(date)	1	1	1	Value during
7	Fabrication No	1	1	1	Instantaneous
8	Model / Version	1	1	1	Instantaneous
9	Parameter set identification	1	1	1	Instantaneous
10	Metrology (firmware) version	1	1	1	Instantaneous
11	Volume(m3)	2	1	1	Instantaneous
12	Time Point(date)	2	1	1	Instantaneous

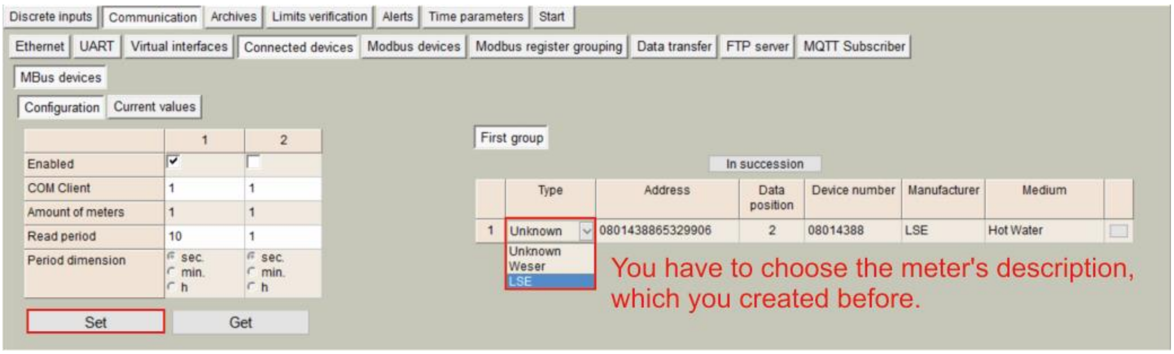
Selected parameters

Pos No	Parameter	Storage number	Tariff	SubUnit	Type of parameter	Double float index	Double long index	Float index	Long index
1	Identification Nr.	1	1	1	Instantaneous				1
2	Volume(m3)	1	1	1	Instantaneous			1	
3	On Time(hours)	1	1	1	Instantaneous				2
4	Time Point(time & date)	1	1	1	Instantaneous			2	
5	Time Point(date)	1	1	1	Value during				3
6	Fabrication No	1	1	1	Instantaneous				4
7	Volume(m3)	2	1	1	Instantaneous			3	

Name: LSE

You can write the meter's initials or anything what you need.

8. When the meter description is created, select it from the list to enable data reading from the meter.



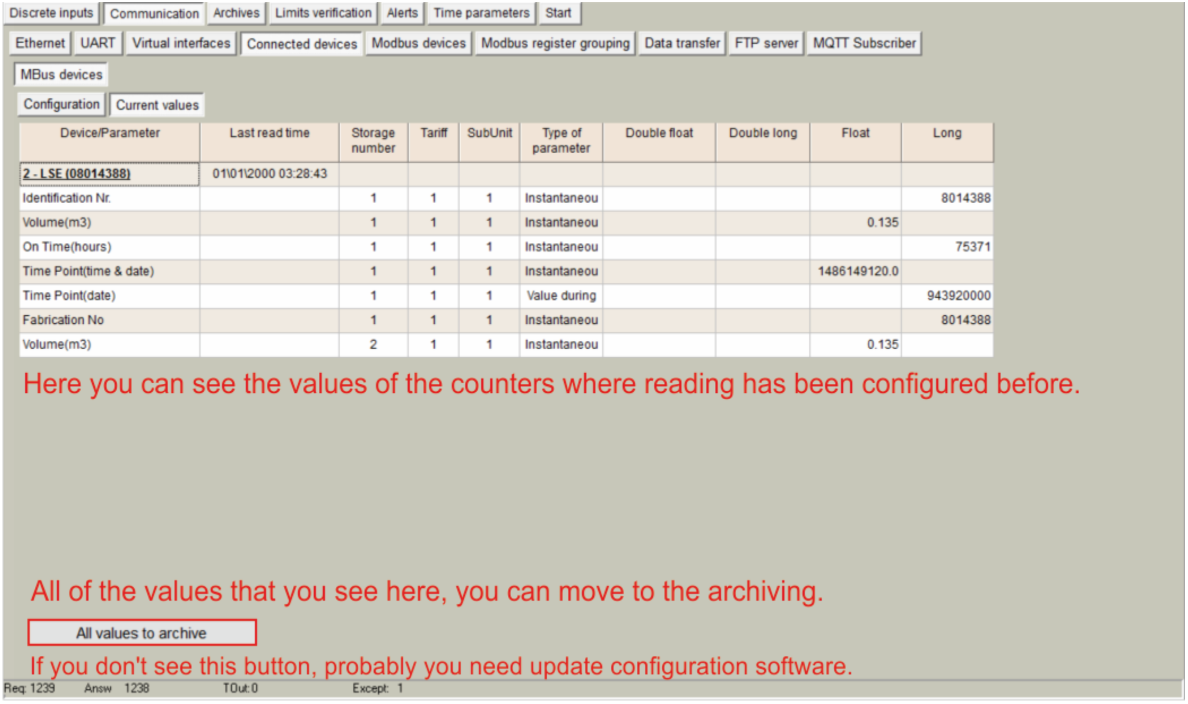
	1	2
Enabled	☑	□
COM Client	1	1
Amount of meters	1	1
Read period	10	1
Period dimension	● sec. ○ min. ○ h	● sec. ○ min. ○ h

	Type	Address	Data position	Device number	Manufacturer	Medium	
1	Unknown	0801438865329906	2	08014388	LSE	Hot Water	□

Unknown
Weser
LSE

You have to choose the meter's description, which you created before.

If your settings are correct, you should see the values from the meter in the configuration tool's *"Current values"* tab. Now, every value from the meter is recorded to the controller's registers. These registers can be archived or read over Modbus RTU or Modbus TCP/IP.
Archiving is used to send a CSV file to the FTP server.



Device/Parameter	Last read time	Storage number	Tariff	SubUnit	Type of parameter	Double float	Double long	Float	Long
2 - LSE (08014388)	01\01\2000 03:28:43								
Identification Nr.		1	1	1	Instantaneou				8014388
Volume(m3)		1	1	1	Instantaneou			0.135	
On Time(hours)		1	1	1	Instantaneou				75371
Time Point(time & date)		1	1	1	Instantaneou			1486149120.0	
Time Point(date)		1	1	1	Value during				943920000
Fabrication No		1	1	1	Instantaneou				8014388
Volume(m3)		2	1	1	Instantaneou			0.135	

Here you can see the values of the counters where reading has been configured before.

All of the values that you see here, you can move to the archiving.

If you don't see this button, probably you need update configuration software.

Req: 1239 Answ 1238 TOut 0 Except 1

6.2 *Modbus devices*

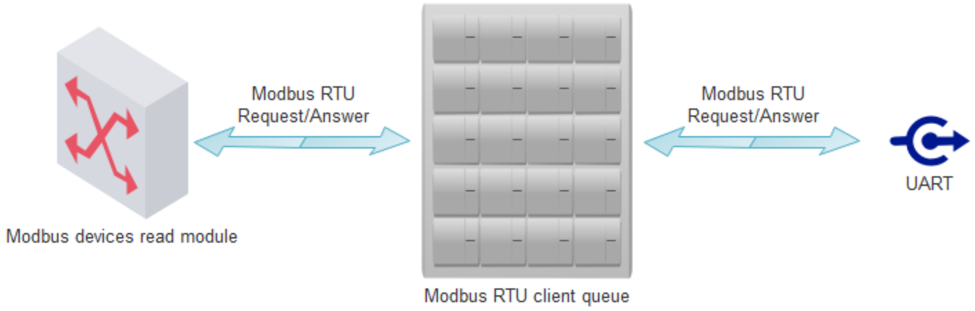
6.2.1 Modbus RTU devices

The device is capable of reading any standard Modbus RTU slave device connected to one of the UART ports.
Supported Modbus Functions:

- **1** – Read Coils
- **2** – Read Discrete Inputs
- **3** – Read Multiple Holding Registers
- **4** – Read Input Registers
- **5** – Write Single Coil
- **6** – Write Single Holding Register
- **15** – Write Multiple Coils
- **16** – Write Multiple Holding Registers

Supported Modbus device IDs range from 1 to 240.

The Modbus devices read module generates Modbus RTU requests and sends them to the appropriate Modbus RTU client. The Modbus RTU client forwards these requests to the configured UART. For details on configuring Modbus requests, refer to the *"Modbus RTU Devices Configuration"* section.


Modbus devices read module
Modbus RTU Request/Answer
Modbus RTU client queue
Modbus RTU Request/Answer
UART

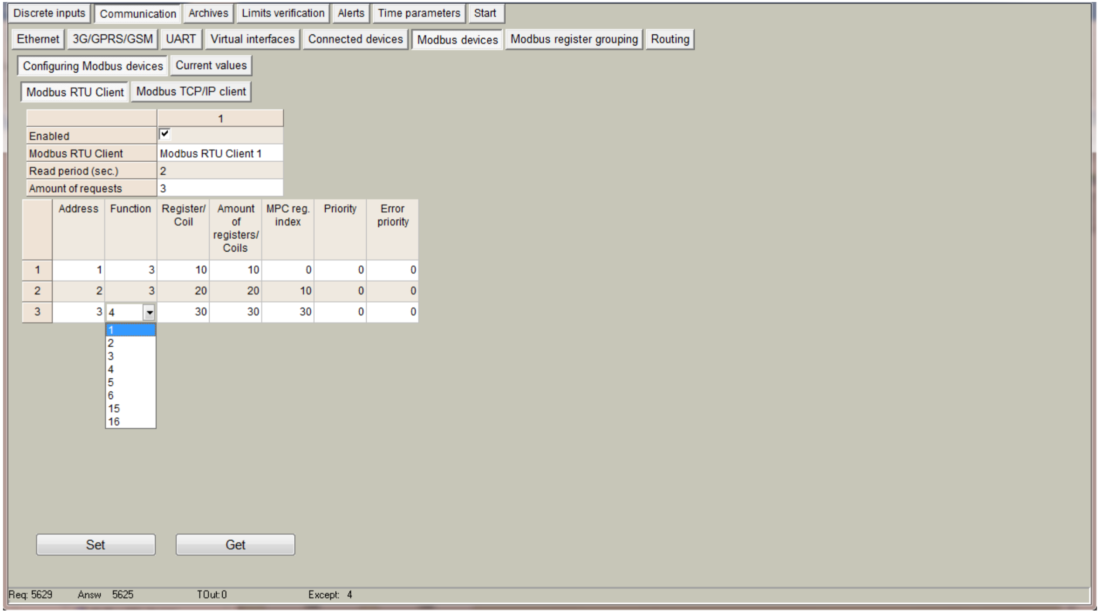
Modbus RTU devices configuration

Modbus RTU devices configuration is done in the *"Communication/Modbus Devices/Configuring Modbus Devices/Modbus RTU Client"* tab. Before configuring your Modbus RTU devices, first configure the Modbus RTU clients, which will link the physical UART with the Modbus RTU devices module.

- **"Enabled" checkbox** – Enables/disables Modbus RTU devices.
- **"Modbus RTU client"** – Selects one of the Modbus RTU clients (Modbus RTU client 1, Modbus RTU client 2).
- **"Read period (sec)"** – Specifies the read period time in seconds. The value can range from 2 to 3600 seconds.
- **"Amount of requests"** – The number of configured requests, with up to 50 requests allowed.

Modbus requests table:

- **"Address"** – Modbus RTU device ID.
- **"Function"** – Modbus function for the current request (see the list of supported functions above).
- **"Register/Coil"** – Start register or coil address.
- **"Amount of registers/Coils"** – Indicates how many registers or coils to read starting from the specified register/coil.
- **"Reg index"** – Indicates where to store the response data in internal registers. Registers from 52000 to 52999 are reserved for Modbus devices data. The *"Reg index"* indicates the data index in this area. For example, *"Reg index" = 0* means data will be stored starting from register 52000. *"Reg index" = 10* means data will be stored starting from register 52010.
- **"Priority"** – Request send priority. A higher number indicates higher priority.
- **"Error priority"** – Always set this to 0.
- **"Set" button** – Writes the configuration to the device.
- **"Get" button** – Reads the configuration from the device.



	1
Enabled	☑
Modbus RTU Client	Modbus RTU Client 1
Read period (sec.)	2
Amount of requests	3

	Address	Function	Register/ Coil	Amount of registers/ Coils	MPC reg. index	Priority	Error priority
1	1	3	10	10	0	0	0
2	2	3	20	20	10	0	0
3	3	4	30	30	30	0	0

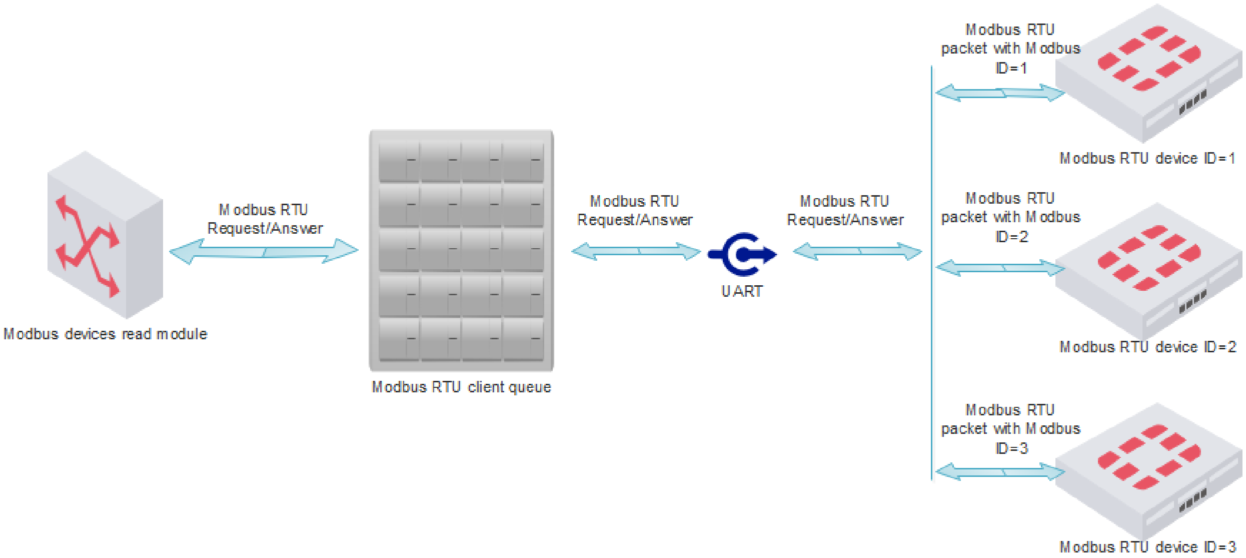
1
2
3
4
5
6
15
16

Req: 5629 Answ 5625 TOut:0 Except: 4

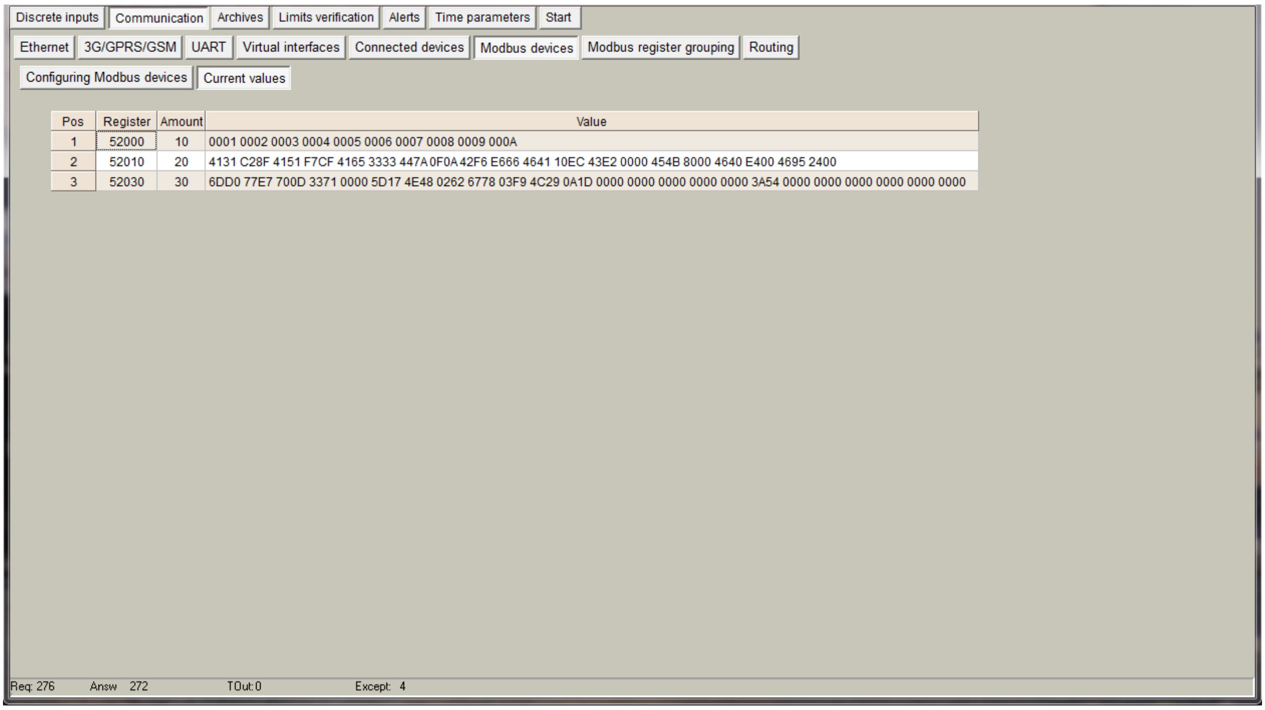
Example

We have 3 Modbus RTU devices connected to one UART, and we need to read some data from them. Let's look at the configuration:

- **Request 1**: Read from Modbus RTU device with ID=1, function = 3 (Read Multiple Holding Registers), start register = 10, and number of registers to read = 10. Device values will be stored in internal registers 52000...52009.
- **Request 2**: Read from Modbus RTU device with ID=2, function = 3 (Read Multiple Holding Registers), start register = 20, and number of registers to read = 20. Device values will be stored in internal registers 52010...52029.
- **Request 3**: Read from Modbus RTU device with ID=3, function = 4 (Read Input Registers), start register = 30, and number of registers to read = 30. Device values will be stored in internal registers 52030...52059.


Modbus devices read module
Modbus RTU Request/Answer
Modbus RTU client queue
Modbus RTU Request/Answer
UART
Modbus RTU Request/Answer
Modbus RTU packet with Modbus ID=1
Modbus RTU device ID=1
Modbus RTU packet with Modbus ID=2
Modbus RTU device ID=2
Modbus RTU packet with Modbus ID=3
Modbus RTU device ID=3

You can check the current values in the *"Communication/Modbus Devices/Current Values"* tab. Each request's data is shown in a separate line, and values are displayed in HEX format.



Pos	Register	Amount	Value
1	52000	10	0001 0002 0003 0004 0005 0006 0007 0008 0009 000A
2	52010	20	4131 C28F 4151 F7CF 4165 3333 447A 0F0A 42F6 E666 4641 10EC 43E2 0000 454B 8000 4640 E400 4695 2400
3	52030	30	6DD0 77E7 700D 3371 0000 5D17 4E48 0262 6778 03F9 4C29 0A1D 0000 0000 0000 0000 0000 3A54 0000 0000 0000 0000 0000 0000

Req: 276 Answ 272 TOut:0 Except: 4

6.2.2 Modbus TCP devices

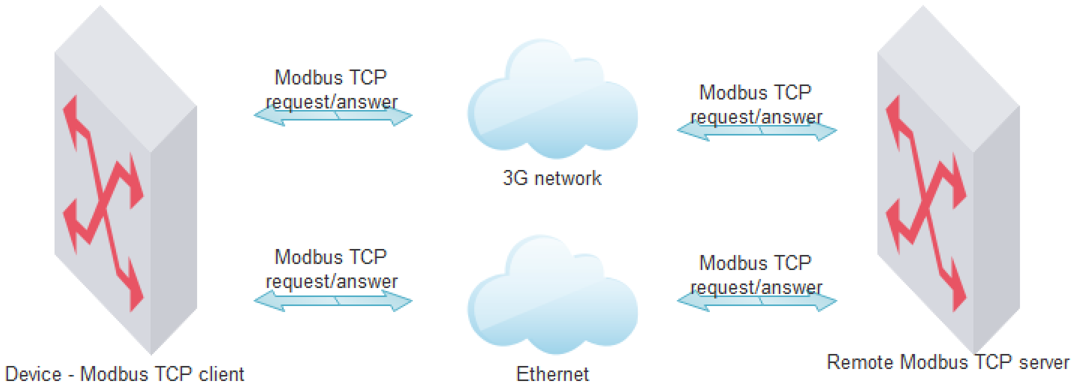
The device is capable of reading any standard Modbus TCP server that can be reached over the 4G/3G/GPRS or LAN network.

Supported Modbus Functions:

- **1** – Read Coils
- **2** – Read Discrete Inputs
- **3** – Read Multiple Holding Registers
- **4** – Read Input Registers
- **5** – Write Single Coil
- **6** – Write Single Holding Register
- **15** – Write Multiple Coils
- **16** – Write Multiple Holding Registers

Supported Modbus device IDs range from 1 to 255.

The Modbus TCP client generates Modbus TCP requests to read from a remote Modbus TCP server and sends these requests via the 4G/3G/GPRS or LAN network. The received data is stored in the internal register area (Registers from 52000 to 52999 are reserved for Modbus devices data).


Modbus TCP request/answer
3G network
Modbus TCP request/answer
Modbus TCP request/answer
Ethernet
Modbus TCP request/answer
Device - Modbus TCP client
Remote Modbus TCP server

Modbus TCP client configuration

Modbus TCP client configuration is done in the *"Communication/Modbus Devices/Configuring Modbus Devices/Modbus TCP/IP Client"* tab.

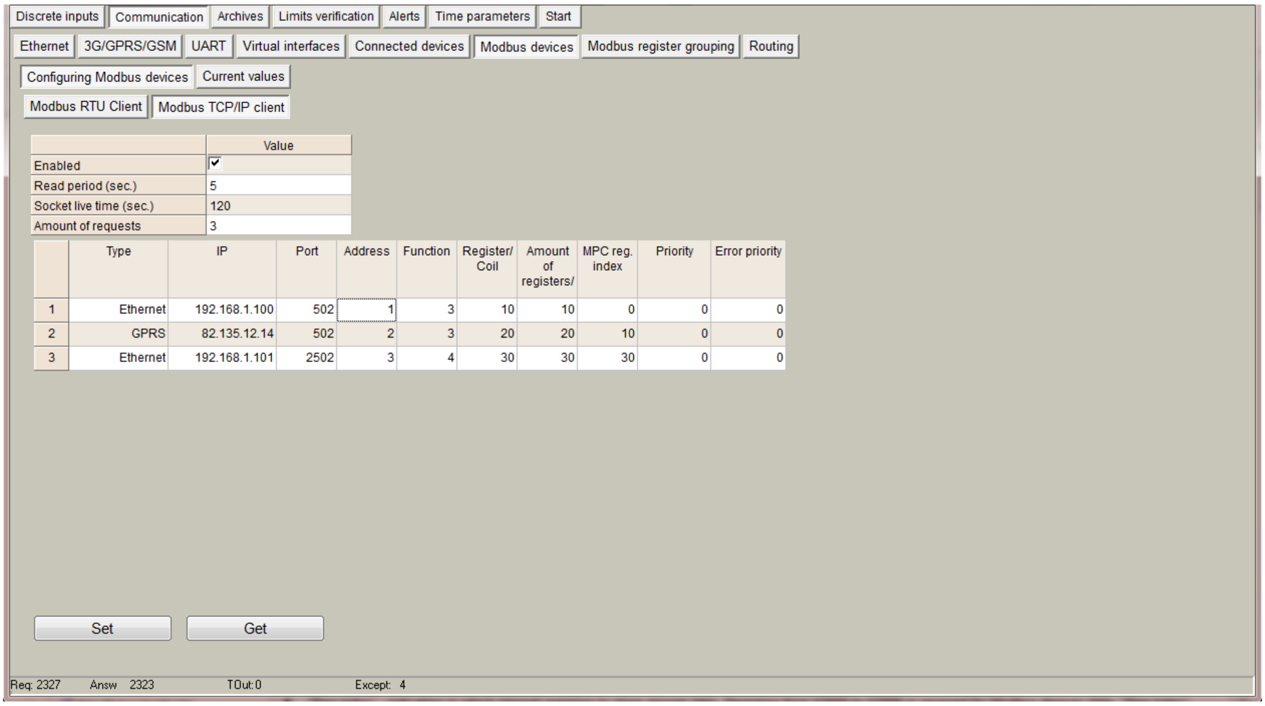
- **"Enabled" checkbox** – Enables/disables the Modbus TCP client.
- **"Read period (sec)"** – Specifies the read period time in seconds. Values can range from 2 to 3600 seconds.
- **"Socket lifetime (sec)"** – The timeout before closing the socket if no data is transferred.
- **"Amount of requests"** – The number of configured requests, with up to 20 requests allowed.

Modbus TCP requests table:

- **"Type"** – Specifies the TCP connection channel (Ethernet or 4G/3G/GPRS).
- **"IP"** – The remote Modbus TCP server's IP address.
- **"Port"** – The remote Modbus TCP server's TCP port. The standard Modbus TCP port is 502.
- **"Address"** – Modbus device ID.
- **"Function"** – The Modbus function for the current request (see the list of supported functions above).
- **"Register/Coil"** – The start register or coil address.
- **"Amount of registers/Coils"** – Indicates how many registers or coils to read from the start register/coil.
- **"Reg index"** – Indicates where to store the response data in internal registers. Registers from 52000 to 52999 are reserved for Modbus devices data. The *"Reg index"* indicates the data index in this area. For example, *"Reg index" = 0* means data will be stored starting from register 52000. *"Reg index" = 10* means

data will be stored starting from register 52010.

- **"Priority"** – Request send priority. A higher number indicates higher priority.
- **"Error priority"** – Always set this to 0.
- **"Set" button** – Writes the configuration to the device.
- **"Get" button** – Reads the configuration from the device.



	Value
Enabled	☑
Read period (sec.)	5
Socket live time (sec.)	120
Amount of requests	3

	Type	IP	Port	Address	Function	Register/ Coil	Amount of registers/	MPC reg. index	Priority	Error priority
1	Ethernet	192.168.1.100	502	1	3	10	10	0	0	0
2	GPRS	82.135.12.14	502	2	3	20	20	10	0	0
3	Ethernet	192.168.1.101	2502	3	4	30	30	30	0	0

Req: 2327 Answ 2323 TOut:0 Except: 4

7. Manufacturer’s warranty

ADVANTICSYS guarantees that all its products are free from defects in materials and workmanship under normal use and service for a period of two years from the date of shipment. This warranty excludes any damage resulting from accidents, misuse, or unauthorized modifications to the product.

This warranty supersedes all other warranties, whether expressed or implied, including implied warranties of merchantability or fitness for a particular purpose, whether arising by law, custom, or conduct. The remedies provided under this warranty are exclusive and replace any other rights or remedies. ADVANTIC SISTEMAS Y SERVICIOS S.L. shall not, under any circumstances, be held liable for any consequential or incidental damages.

If you believe your product is defective and still under warranty, please contact ADVANTICSYS at info@advanticsys.com or by phone at +34 914221023. After confirmation from our support team that the product is defective, we will issue a Return Merchandise Authorization (RMA) number and arrange for the replacement of your product.

This warranty covers the cost of repair, including labor and materials, for any manufacturing defect that impedes the proper operation of the product. Replacement of any component or equipment does not extend the original warranty period. If, upon inspection by ADVANTICSYS, the product is found to be defective, we will cover the shipping costs to return the product to the customer, as well as all costs associated with the inspection. If the product is found not to be defective, the customer will be responsible for the return shipping costs.

Advantic Sistemas y Servicios S.L
C/ Ponzano, 80, Bajo 2
28003 (Madrid) - Spain
www.advanticsys.com
info@advanticsys.com
+34 91 4221023