MX-10

USER MANUAL

VILTRUS
ELECTRONICS

Table of contents

Introduction

Abbreviations and explanations

Xn – is a number of sockets. This information are provided for manufacturer's purpose and used in data schemas and connection diagrams.
GSM – Global Standard for Mobile Communications. This interface is prepared for remote connections and data bidirectional data transfer over Global Standard Mobile network.
GPRS - a packet oriented mobile data service on the 2G and 3G cellular communication system's global system for mobile communications (GSM).
Ethernet - a family of computer networking technologies for local area networks (LANs) commercially introduced in 1980. Standardized in IEEE 802.3, Ethernet has largely replaced competing wired LAN technologies. This interface is prepared for connection LAN (Local Area Network).
IP address - An Internet Protocol (IP) address is a numerical label that is assigned to devices participating in a network that uses the Internet Protocol for communication between its nodes.
TCP/IP – Transmission Control Protocol is for communication between computers, used as a standard for transmitting data over networks and as the basis for standard Internet protocols.
MAC address –Media Access Control address is a unique identifier assigned to most network adapters.
UART – An Universal Asynchronous Receiver/Transmitter is a type of "asynchronous receiver/transmitter, a part of computer hardware that translates data between parallel and serial forms. UART are commonly used in conjunction with communication standards such as EIA RS-232, RS-422 or RS-485. Record (UARTx) on top of enclosure also is used as serial interface number.
GND – ground wire contact
RS485 - standard defining the electrical characteristics of drivers and receivers for use in balanced digital multipoint systems. The standard is 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 connection of peripherical devices (example energy meters, controllers, machines and etc.).
A+ – contact for positive wire of RS485 socket
B- –contact for negative wire of 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 computer and electronic devices. USB type B socket is prepared for connection to PC(Personal Computer). USB type A socket is prepared for connection to peripherical devices (example memory stick's and etc.).
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 on two wires, making it very cost effective.
MBUS+ – contact for M-Bus positive wire
MBUS- – contact for M-Bus negative wire
Socket – is an endpoint of a bidirectional inter-process communication flow across an Internet Protocol-based computer network, such as the Internet**.**
Status – device status indicating LED
TX/RX – data transfer/receive indicating LED
TXD – data transferring LED indicator
RXD – data receiving LED indicator
100Mbs – Ethernet High speed connection indicating LED
"Alarm mode" – in state of alarm status Controller initiates an event notification for user selected discrete input mode (Alarm mode: unconnected, connected, both events)
Central computer – server or a computer, where data can be sent.

Preface

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

Safety instructions

To install and setup device, special technical knowledges are needed. Call to seller or certified professionals to connect and setup device!

Before connecting to power supply, be sure that:

1. Controller is not damaged (no cracks, melted, broken or exposed areas)
2. Controller is used with right and correct thickness cables.
3. Controller and antenna are installed indoor.
4. The controller is intended for supply from a Limited Power Source (LPS) with current rating of over current protective device not greater than 2A
5. The highest transients on the DC secondary circuit of LPS, derived from AC main supply, shall be less then 71V peak.
6. The associated equipments (AE): PC and PSU (LPS) shall comply with the requirements of Standard EN 69050-1.
7. Controller is dry;
8. Ambient temperature and humidity is in normal range;
9. Other types of devices (counters, etc.) are connected correctly by using manufacturer‘s regulations.
10. The end of stranded conductor shall not be consolidated by soft soldering and must to be terminated
11. Device, PC and other peripherical devices are strictly connected through one double pole breaker (current break less than 5A and space between breaker contacts more than 3mm.) Pole breaker has to be in building‘s wiring and in reachable place with markings

Don‘t use:

1. Device under open water (in rain and if water are splashing on controller or connected devices;
2. Device if enclosure, connected cables, or other connected devices are damaged;

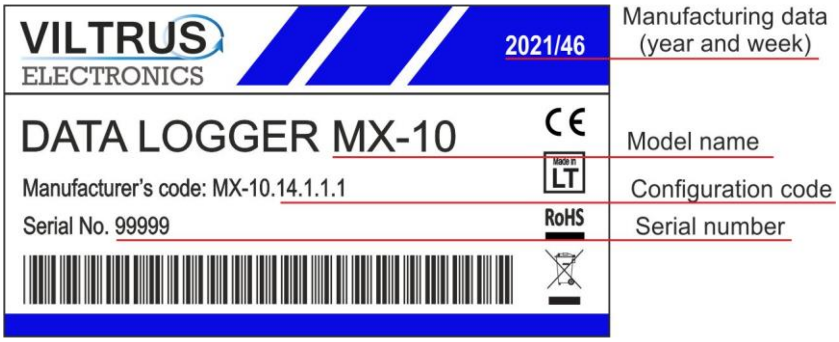
⚠ **Use device by manufacturer‘s regulations otherwise you can damage controller or other devices. In that case manufacturer‘s warranty could not be obtained.**

⚠ **If you suspect that device doesn‘t operate correctly or has visible violations, please contact manufacturer or your distributor to check or run maintenance.**

Top label information

Overview

This label provides common information about controllers and if you will need any consultations from technical center or manufacturer, please let them know full Manufacturer’s code and Serial No.



Connecting to the device

Overview

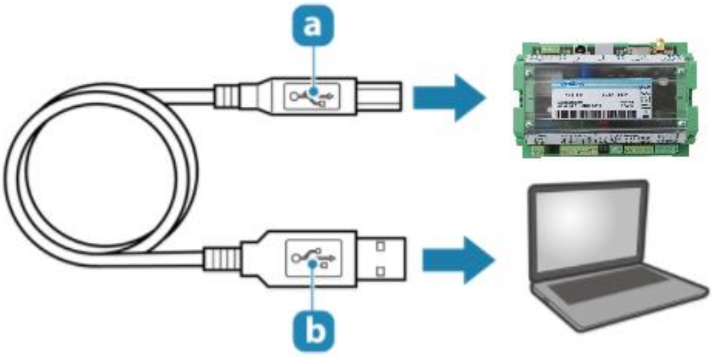
USB port is used for local configure of device. Also it is possible configure device via Ethernet, 3G modem or any of UARTS if them are used as Modbus slaves. All configuration is made using Modbus protocol and using device configuration tool software which can be downloaded from manufacture website.

Connecting over USB

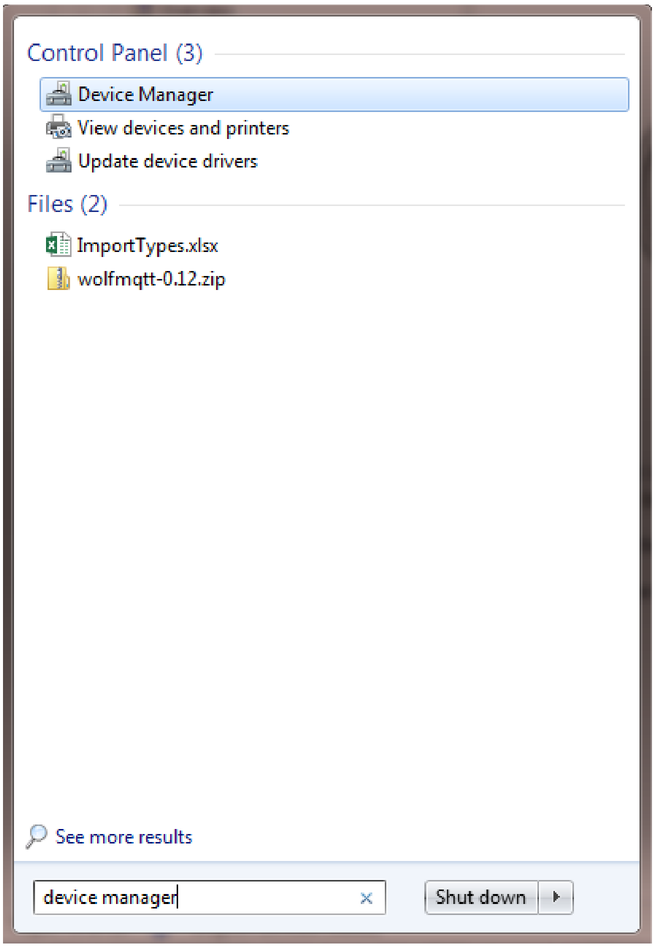
Use USB Type-A to Type-B cable to connect device to computer.

- a) To device USB Type-B

- b) To computer USB Type-A



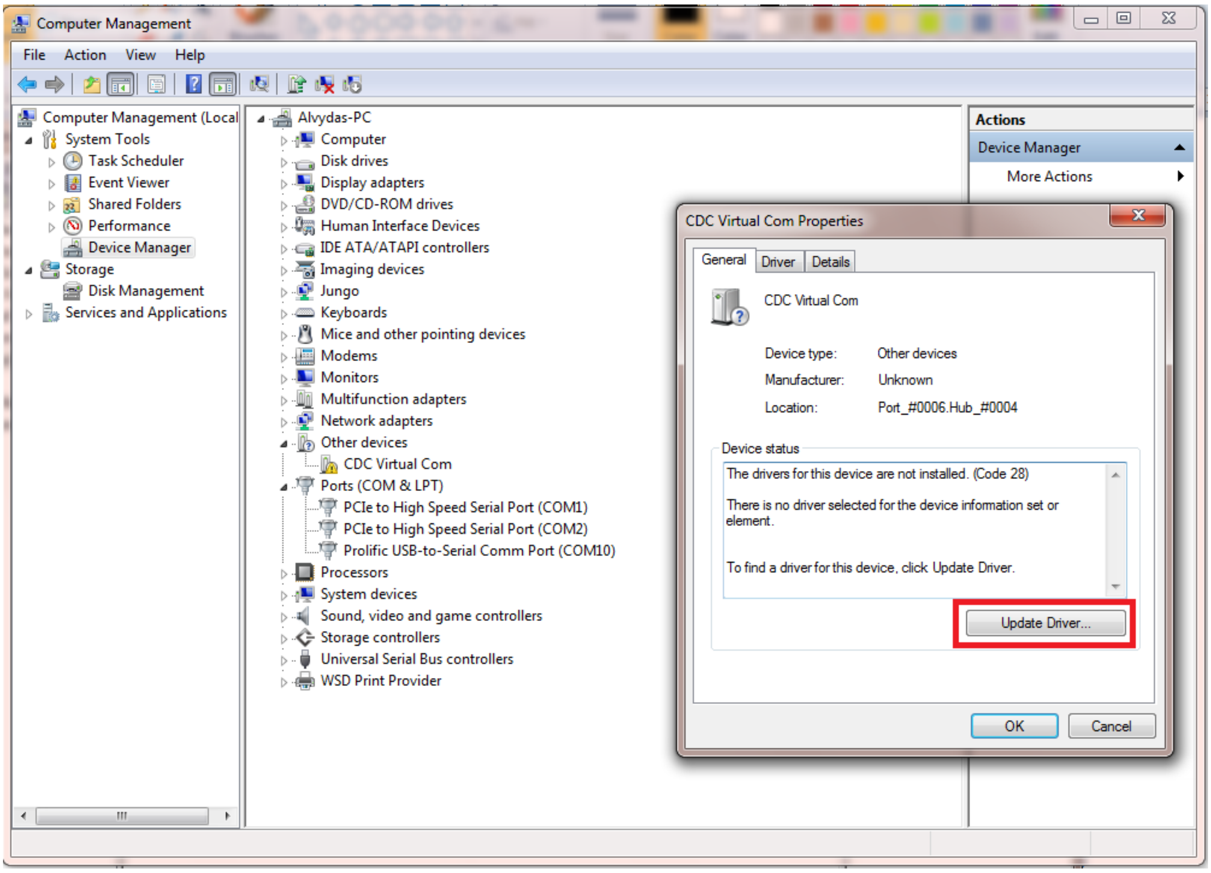
If USB drivers not installed automatically you need to install them manually. Follow these steps:



1. Double-click the device category, and then double-click the device you want.
2. Click Update Driver, and follow the instruction

3. Select "Browse my computer for driver software", click "Browse" and select configuration software folder.

4. Click "Next".



CDC Virtual Com

Device type: Other devices

Manufacturer: Unknown

Location: Port_#0006.Hub_#0004

Device status

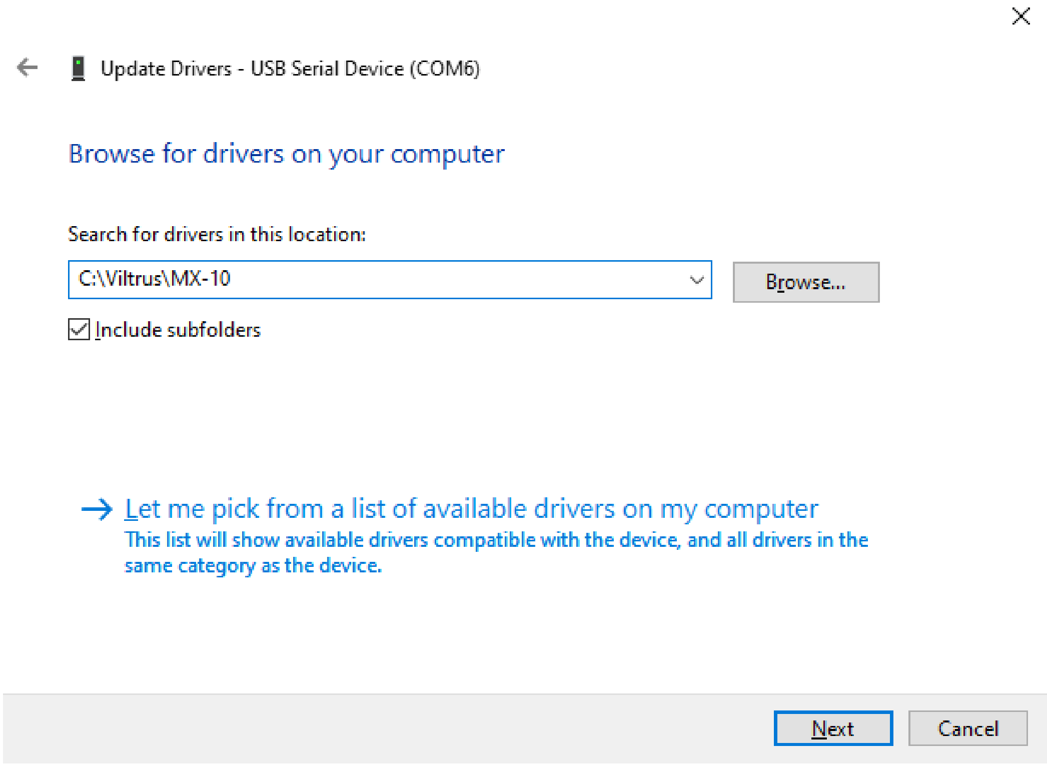
The drivers for this device are not installed. (Code 28)

There is no driver selected for the device information set or element.

To find a driver for this device, click Update Driver.

5. Wait while Windows installs driver. If you see message "Windows can't verify the publisher ...." select "Install this driver software anyway".

6. After installation you will see something like "USB serial device Virtual Com Port" and com port number. Use this com port for connection with configuration tool.



Browse for drivers on your computer

Search for drivers in this location:

C:\Viltrus\MX-10

☑ Include subfolders

→ Let me pick from a list of available drivers on my computer

This list will show available drivers compatible with the device, and all drivers in the same category as the device.

⚠ **On some Windows versions (Windows 8, windows 10) you will need to disable third party driver signature checking before installing device driver. Please check on Internet how to do this.**

Configuration tool software

After USB driver is installed run device configuration tool software.
Select connection type "Modbus RTU" and appropriate COM port. Press "Get all configuration" to read all device configuration.
Other functions:

- "Get all configuration" button - read all configuration from device
- "Save to file" button - save all configuration to file. So later it is possible to load this configuration to device.
- "Load from file" button - load saved configuration from file.
- "Write all configuration" - write loaded configuration to device.
- "Get all at connection" check box - reads all configuration when connecting over TCP/IP connection
- Connection type "Modbus RTU" - connect to device over USB or serial port.
- Connection type "Modbus TCP/IP" - connect to device over 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 device has installed backup battery.
  - "After which time switch off" time - power off device after configured time if main power supply disconnects
- "Float number format" - floating point data byte order, E - exponent, M1, M2, M3 - Mantissa
- "Last restart information" - time of last reset and reset code. Reset code 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
  11 - No TCP packet over ETHERNET in configured time
  12 - ETHERNET task stops working
  13 - All TCP sockets is used (if defined UIP_RESET_ALL_CONN_USED)
  14 - Periodic reset

Program modules

Archive

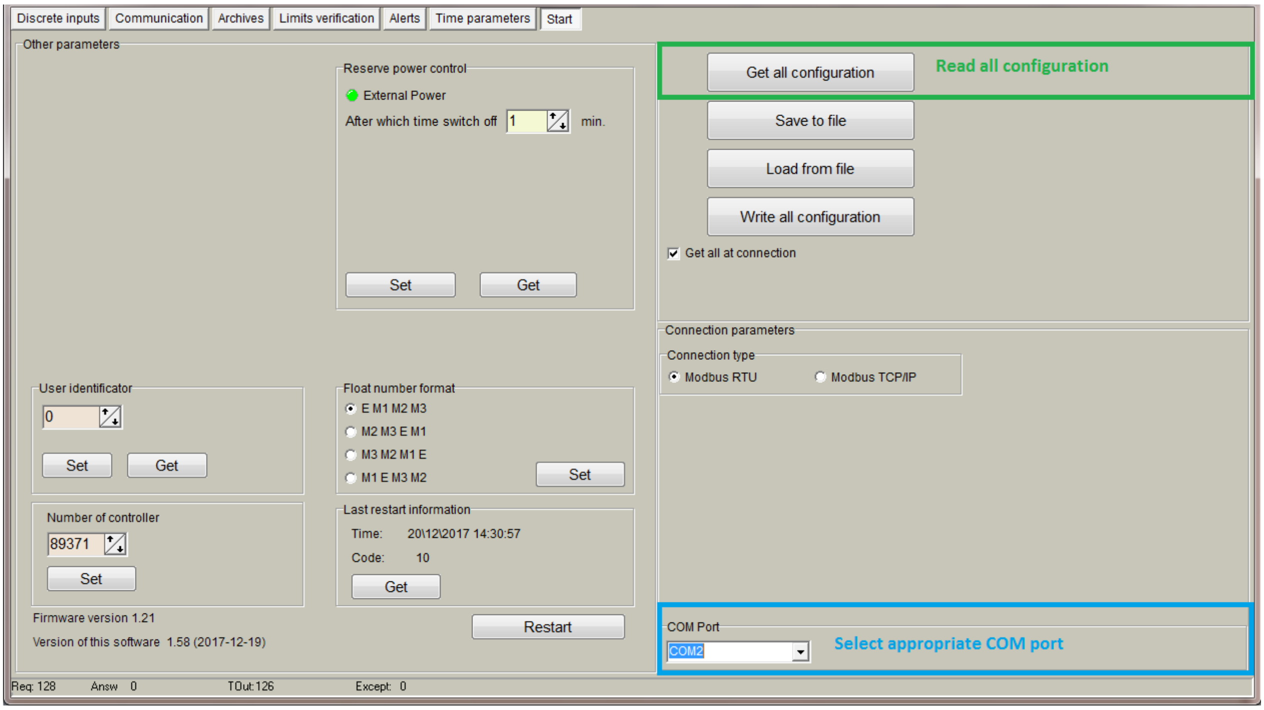
Overview

Device has several types of archives:

- Events archive. All events will be saved here (Analog inputs alarms, Discrete inputs alarms, Limits verification). Events is used to generate SMS messages, MQTT event messages, emails.
- Diagnostic archive. It is a list of changes made in device. It's resets, configuration changes, connection/disconnection to GPRS network and other
- User defined. It's user configured archive, user can add any existing data register archive.

All archives are saved to internal data flash memory. Maximum archive records count depends on memory size and separated place. If Micro SD card presents archives is duplicated to SD card (For every day is created new file with date stamp).

Events archive



Read all configuration

● External Power

After which time switch off 1 min.

☑ Get all at connection

● Modbus RTU ○ Modbus TCP/IP

User identificator: 0

● E M1 M2 M3
○ M2 M3 E M1
○ M3 M2 M1 E
○ M1 E M3 M2

Number of controller: 89371

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

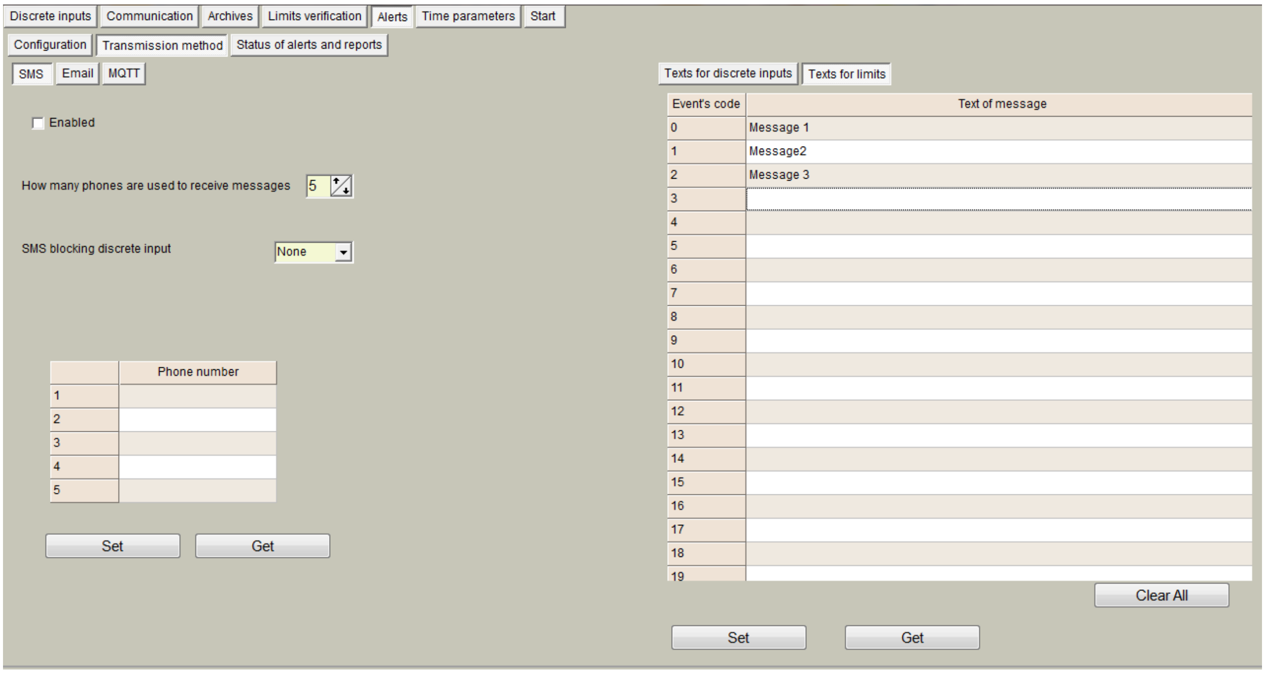
Req: 128 Answ 0 TOut:126 Except: 0

Overview

All events are saved to events archive. Events sources can be:

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

Event archive is used to generate SMS, MQTT messages, emails. For every event ID you can configure event message text.
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:

The 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): 0 – Analog input alarm, 1 – Discrete input alarm, 4 - Limits alarm. Youngest byte value: if alarm source 0 or 1 then channel number; if alarm source 4 then index from limits table (start from 0 to 99)	Int (16 bits)
Type of deviation	If alarm source 0 then: if value 1 –Analog input deviation below the lower alarm value. 2 - Analog input deviation above the upper alarm value. If alarm source 1 then value always 0.	Int (16 bits)

	If alarm source 4 then value is limits alarm ID.	
Deviation value	If alarm source 0 then Analog input value. If alarm source 1 then Discrete input value. If alarm source 4 then Limits value.	Float
		Total 16 bytes

Records transferred from newest to oldest. For example: to read the latest event archive record shall contain the following information:

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

If the 5 oldest then:

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)

Diagnostic archive

Overview

Diagnostic archive is a list of changes made in device. It's useful for debug purpose

Reading diagnostic archive over Modbus file system

Diagnostic 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	900 .. 999	10000	1250	4911

Diagnostic archive record structure:

The 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	Long int (32 bits)

	6. Unable connect to GPRS 7. External pin reset 8. Watchdog reset 9. Brownout reset 10. Power up reset 11. No TCP packet over ETHERNET in configured time 12. ETHERNET task stops working 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	
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 transferred from newest to oldest. For example: to read the latest event archive record shall contain the following information:

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

If the 5 oldest then:

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)

User defined archive

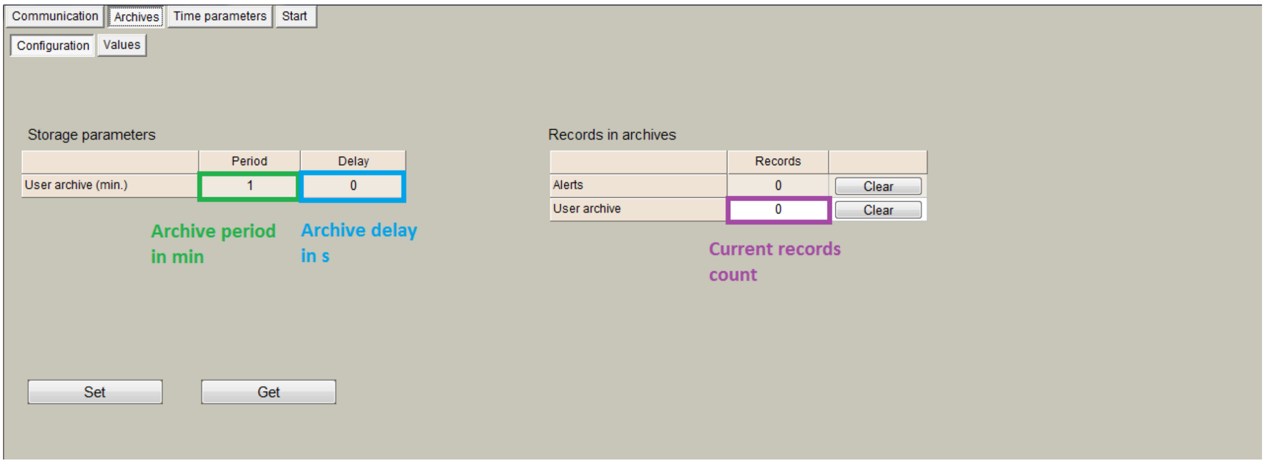
Overview

User can add any device register(value) to user defined archive. So you can make periodic archive of useful values.

Archive period is in minutes and can be from 1 min to 600 min (10 hours). You can configure it in configuration tool in "Archives/Configuration" section.

User archive storage period

Archive period is synchronized with real time. If read period is 1 min, records will be generated every minute (00:00:00, 00:01:00, 00:02:00 ....), If read period is 15 min, records will be generated every 15 minute (00:00:00, 00:15:00, 00:30:00 ....). Delay is used to delay archive time in seconds, for example period 15 minutes, delay 30 seconds so archives will be made at 00:00:30, 00:15:30... Delay is used to delay archive process if values is taken from meters, wait until read process ends.



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 needed registers to archive using table in "Archives/User archive configuration"

1. "Count of parameters" - number of configured lines in table
2. In „Register" column configure start register of value
3. In "Count of parameters" column configure how many data values will be from start register.
4. In "Format" column configure data value type for current line. Value types can be:

- *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, add new line with the same timestamp. Used to add new line record with the same time in csv file*
- *Unix time*

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:

The 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	Long int (32 bits)

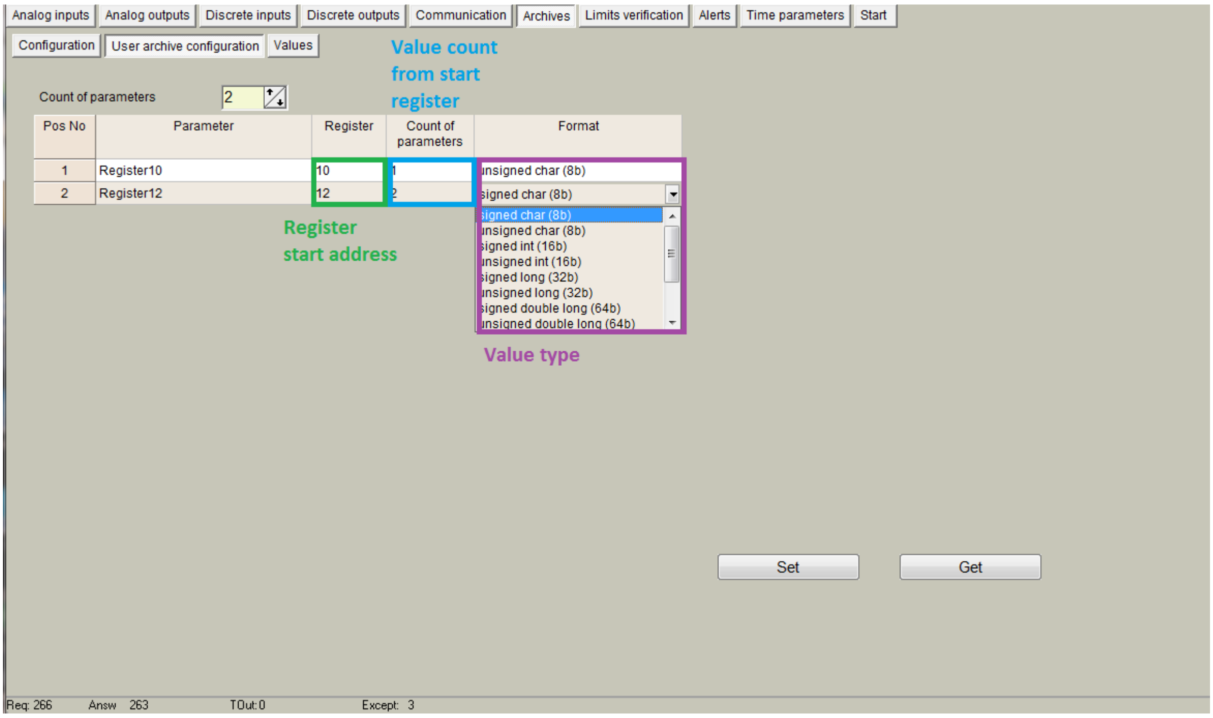
	4929 register or in configuration tool "Archives/User archive configuration" How configure registers check	
		Total 4+2xregister count bytes

Records transferred from newest to oldest. For example: to read the latest user archive record (with 2 registers) shall contain the following information:

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, user archive structure length/2)

If the 5 oldest then:

File address: 800 + (5/2500) (record number/record count in file)
Register address: 4*(5-1)
Register count: 4 (8/2, user archive structure length/2)



Value count from start register

Count of parameters 2

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

Register start address

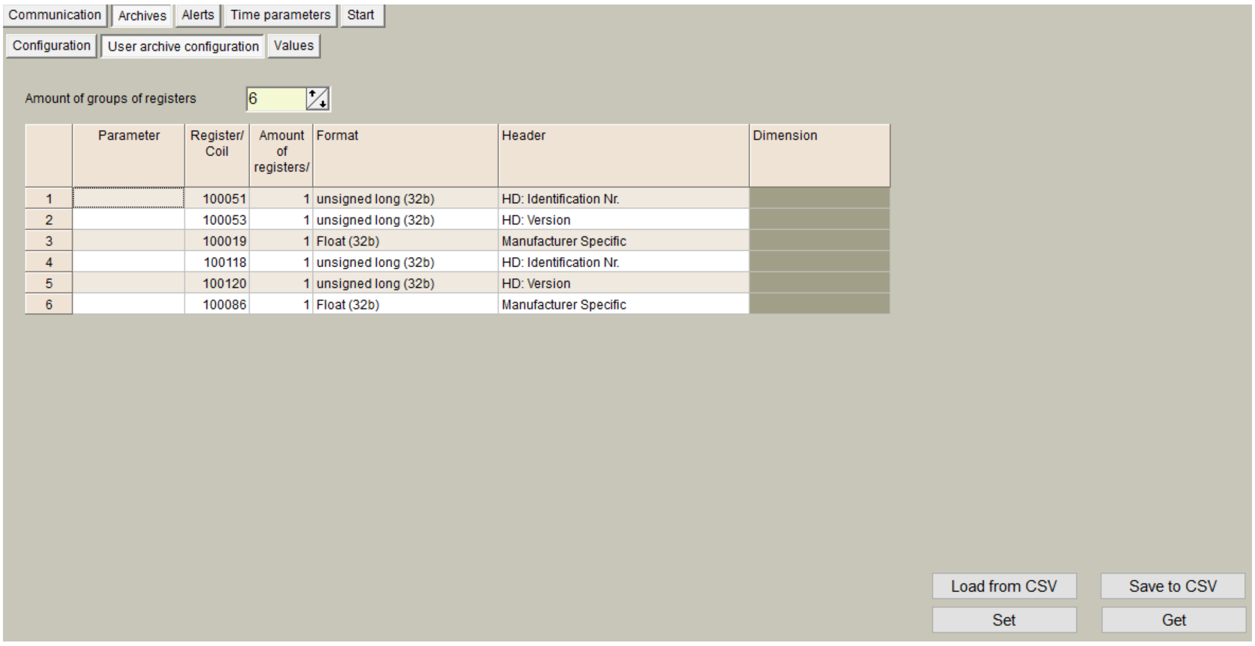
Value type

Req: 266 Answ 263 TOut:0 Except: 3

If FTP or MQTT clients is used, archive is configured automatically using "Communication/Data transfer/Common parameters" data table, then at below.

CSV files examples

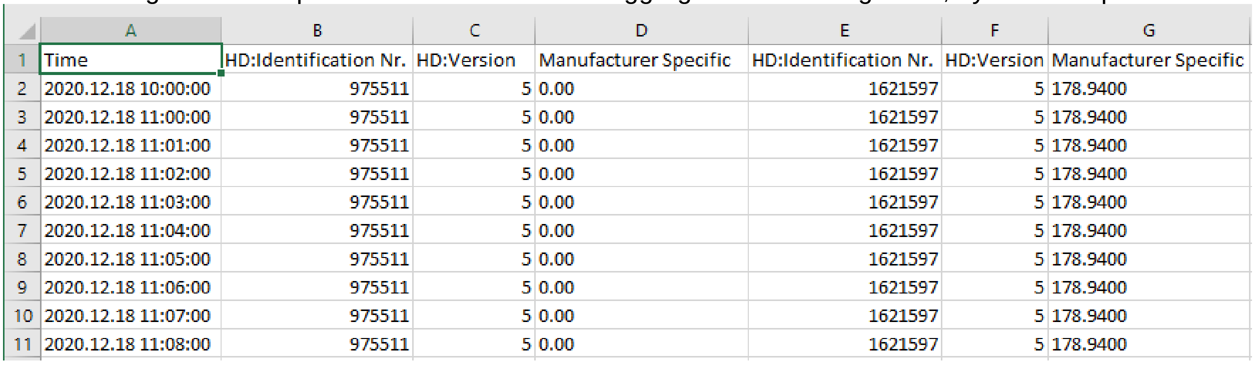
The following is an example of an archive where data is grouped into a single line, by time stamp.



Amount of groups of registers: 6

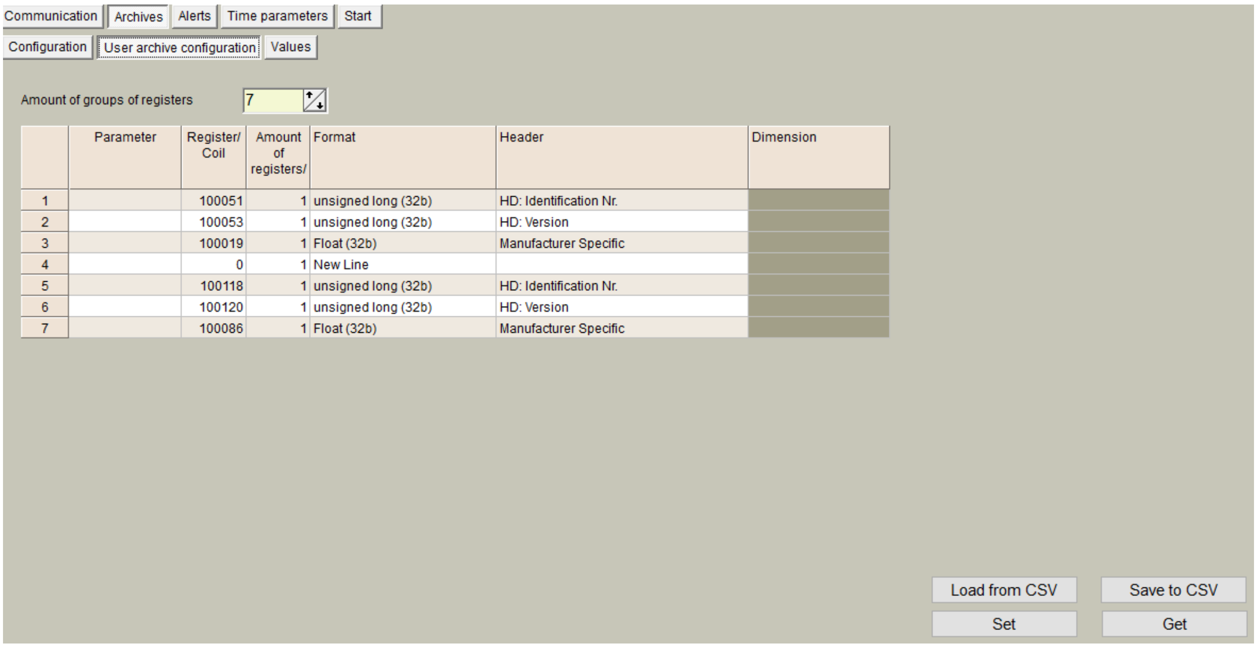
	Parameter	Register/ Coil	Amount of registers/	Format	Header	Dimension
1		100051	1	unsigned long (32b)	HD: Identification Nr.	
2		100053	1	unsigned long (32b)	HD: Version	
3		100019	1	Float (32b)	Manufacturer Specific	
4		100118	1	unsigned long (32b)	HD: Identification Nr.	
5		100120	1	unsigned long (32b)	HD: Version	
6		100086	1	Float (32b)	Manufacturer Specific	

The following is an example CSV file where data is aggregated into a single line, by timestamp.



	A	B	C	D	E	F	G
1	Time	HD:Identification Nr.	HD:Version	Manufacturer Specific	HD:Identification Nr.	HD:Version	Manufacturer Specific
2	2020.12.18 10:00:00	975511	5	0.00	1621597	5	178.9400
3	2020.12.18 11:00:00	975511	5	0.00	1621597	5	178.9400
4	2020.12.18 11:01:00	975511	5	0.00	1621597	5	178.9400
5	2020.12.18 11:02:00	975511	5	0.00	1621597	5	178.9400
6	2020.12.18 11:03:00	975511	5	0.00	1621597	5	178.9400
7	2020.12.18 11:04:00	975511	5	0.00	1621597	5	178.9400
8	2020.12.18 11:05:00	975511	5	0.00	1621597	5	178.9400
9	2020.12.18 11:06:00	975511	5	0.00	1621597	5	178.9400
10	2020.12.18 11:07:00	975511	5	0.00	1621597	5	178.9400
11	2020.12.18 11:08:00	975511	5	0.00	1621597	5	178.9400

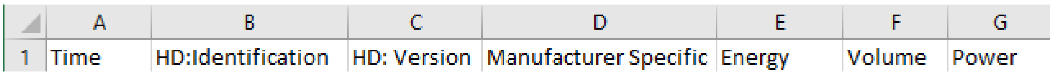
The following is an example of an archive where the data for each meter is added to a new row with common Header.



Amount of groups of registers: 7

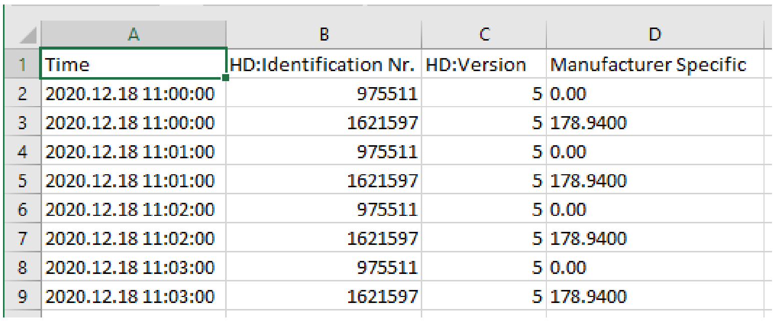
	Parameter	Register/ Coil	Amount of registers/	Format	Header	Dimension
1		100051	1	unsigned long (32b)	HD: Identification Nr.	
2		100053	1	unsigned long (32b)	HD: Version	
3		100019	1	Float (32b)	Manufacturer Specific	
4		0	1	New Line		
5		100118	1	unsigned long (32b)	HD: Identification Nr.	
6		100120	1	unsigned long (32b)	HD: Version	
7		100086	1	Float (32b)	Manufacturer Specific	

This format can be used when all meters parameters names are the same because the file uses common Headers.



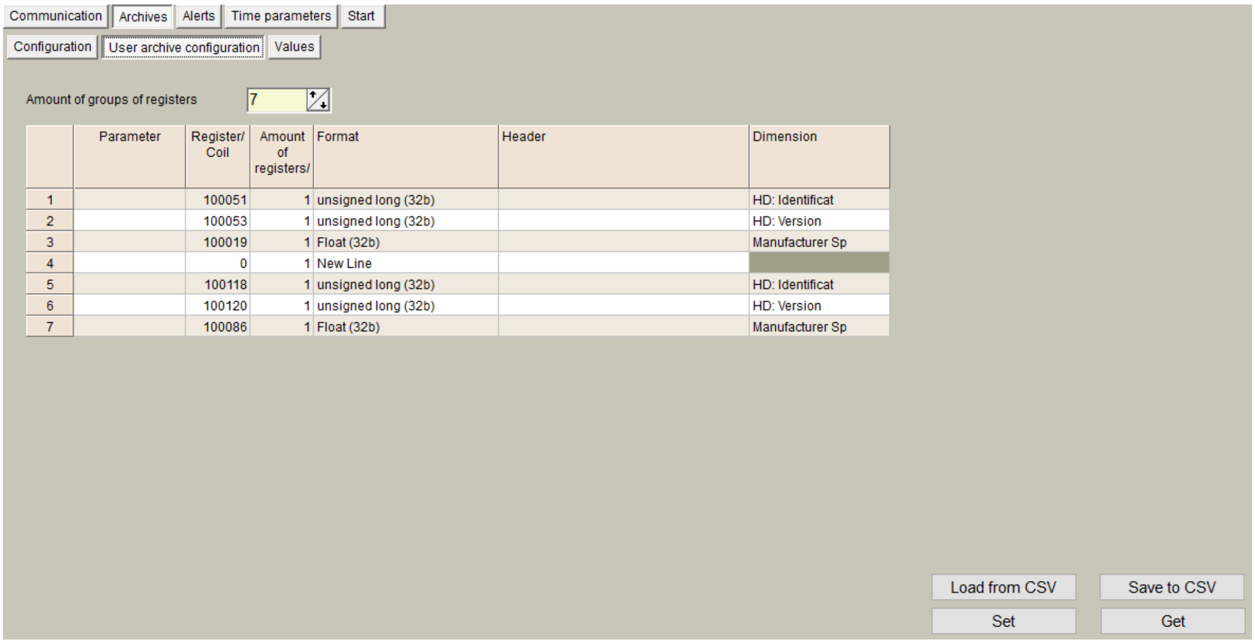
	A	B	C	D	E	F	G
1	Time	HD:Identification	HD: Version	Manufacturer Specific	Energy	Volume	Power

The following is an example CSV file where the data for each meter is added to a new row.



	A	B	C	D
1	Time	HD:Identification Nr.	HD:Version	Manufacturer Specific
2	2020.12.18 11:00:00	975511	5	0.00
3	2020.12.18 11:00:00	1621597	5	178.9400
4	2020.12.18 11:01:00	975511	5	0.00
5	2020.12.18 11:01:00	1621597	5	178.9400
6	2020.12.18 11:02:00	975511	5	0.00
7	2020.12.18 11:02:00	1621597	5	178.9400
8	2020.12.18 11:03:00	975511	5	0.00
9	2020.12.18 11:03:00	1621597	5	178.9400

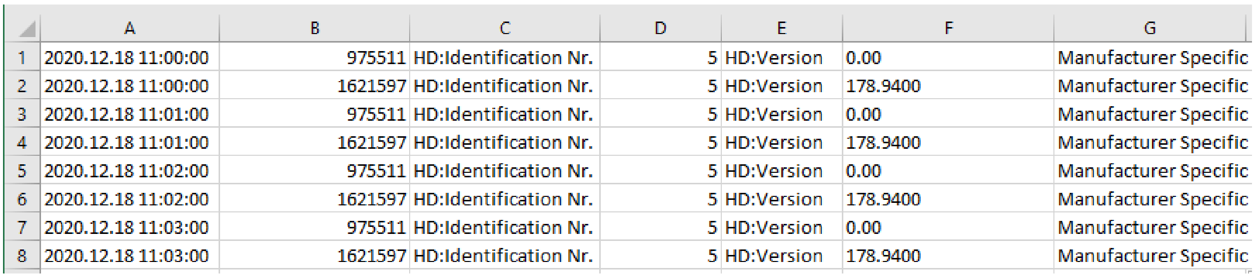
The following is an example of an archive where the data for each meter is added to a new row without a Header but with dimensions.



Amount of groups of registers 7

	Parameter	Register/ Coil	Amount of registers/	Format	Header	Dimension
1		100051	1	unsigned long (32b)		HD: Identificat
2		100053	1	unsigned long (32b)		HD: Version
3		100019	1	Float (32b)		Manufacturer Sp
4		0	1	New Line		
5		100118	1	unsigned long (32b)		HD: Identificat
6		100120	1	unsigned long (32b)		HD: Version
7		100086	1	Float (32b)		Manufacturer Sp

The following is an example CSV file where the data for each meter is added to a new row without a Header but with dimensions.



	A	B	C	D	E	F	G
1	2020.12.18 11:00:00	975511	HD:Identification Nr.	5	HD:Version	0.00	Manufacturer Specific
2	2020.12.18 11:00:00	1621597	HD:Identification Nr.	5	HD:Version	178.9400	Manufacturer Specific
3	2020.12.18 11:01:00	975511	HD:Identification Nr.	5	HD:Version	0.00	Manufacturer Specific
4	2020.12.18 11:01:00	1621597	HD:Identification Nr.	5	HD:Version	178.9400	Manufacturer Specific
5	2020.12.18 11:02:00	975511	HD:Identification Nr.	5	HD:Version	0.00	Manufacturer Specific
6	2020.12.18 11:02:00	1621597	HD:Identification Nr.	5	HD:Version	178.9400	Manufacturer Specific
7	2020.12.18 11:03:00	975511	HD:Identification Nr.	5	HD:Version	0.00	Manufacturer Specific
8	2020.12.18 11:03:00	1621597	HD:Identification Nr.	5	HD:Version	178.9400	Manufacturer Specific

TCP modules

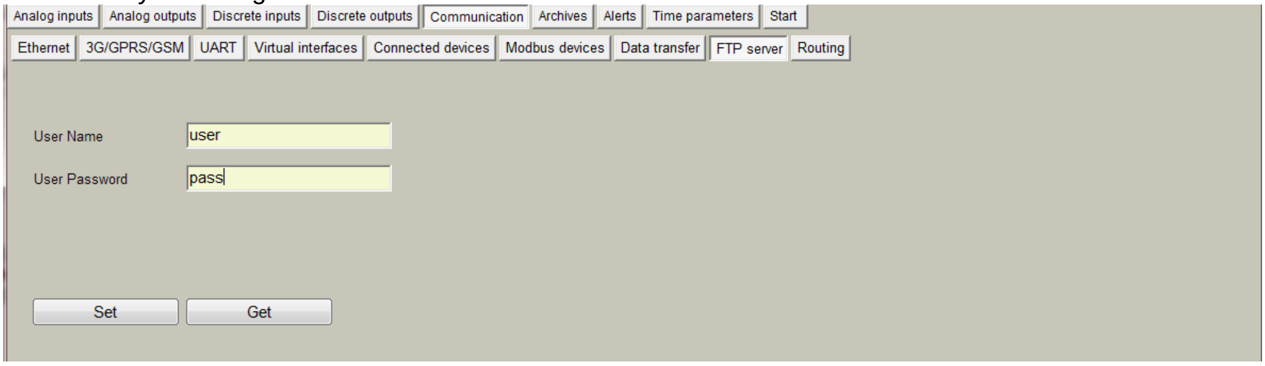
FTP server

Overview

FTP server is used to access internal micro SD card data. You can connect to it using any ftp client program like "FileZilla", "total commander" or any web browser. FTP server runs on standard 21 port

Connection to FTP server

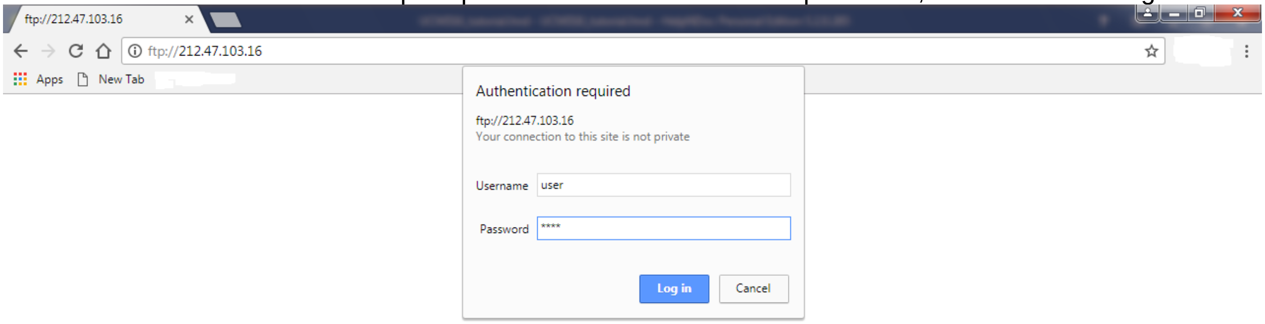
Before connection you need to know your device IP address and FTP user name/password. FTP server user name and password can be configured in "Communication/FTP server" tab. User name and password is max 19 symbol length



User Name: user

User Password: pass

Let’s try to connect to device IP 212.47.103.16 and google chrome browser. Use ftp://212.47.103.16 in browser address line. You will be prompted to enter user name and password, enter them and log in.



ftp://212.47.103.16

Authentication required

ftp://212.47.103.16
Your connection to this site is not private

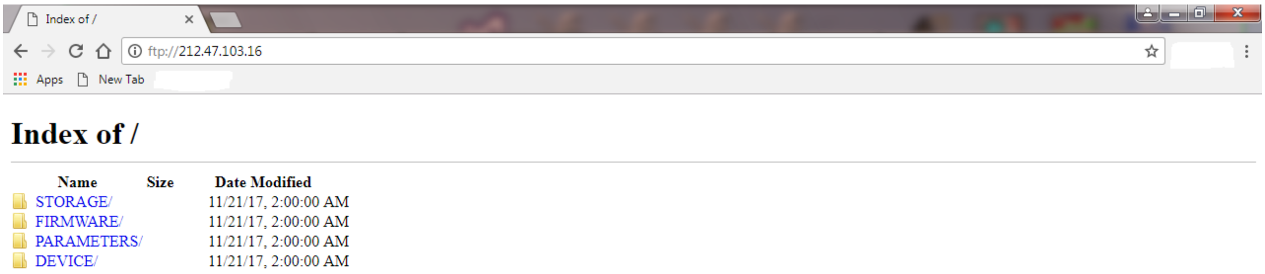
Username: user

Password: ****

Browser will load directory list of SD card. Where are some main directories in device:

- Storage - contains all archives (Contains folders "Alarms", "Diagnostic", "User defined")
- Firmware - reserved for future (firmware update folder)
- Parameters - contains some device parameters (in some firmware version can be saved csv file headers/dimensions and other)

- Device - contains 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

FTP client

Overview

FTP client is used to send user archive files to ftp server. Files are with .csv extension and is generated from saved user archive.

Csv file creator

Overview

Device creates csv report file from user defined archive values, every record in file have it's time stamp (value record time), it can have "Header" for every value, and dimension for every value. All data in file is separated by a ";" symbol and every record is placed in new line. 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> - is configured header, from „Communication/Data transfer/Common parameters" tab.
- <Value data> - is configured parameter value taken from user archive.
- <Value dimension> - is configured dimension, from „Communication/Data transfer/Common parameters" tab.
- <Record Date/Time> - is stored archive record time

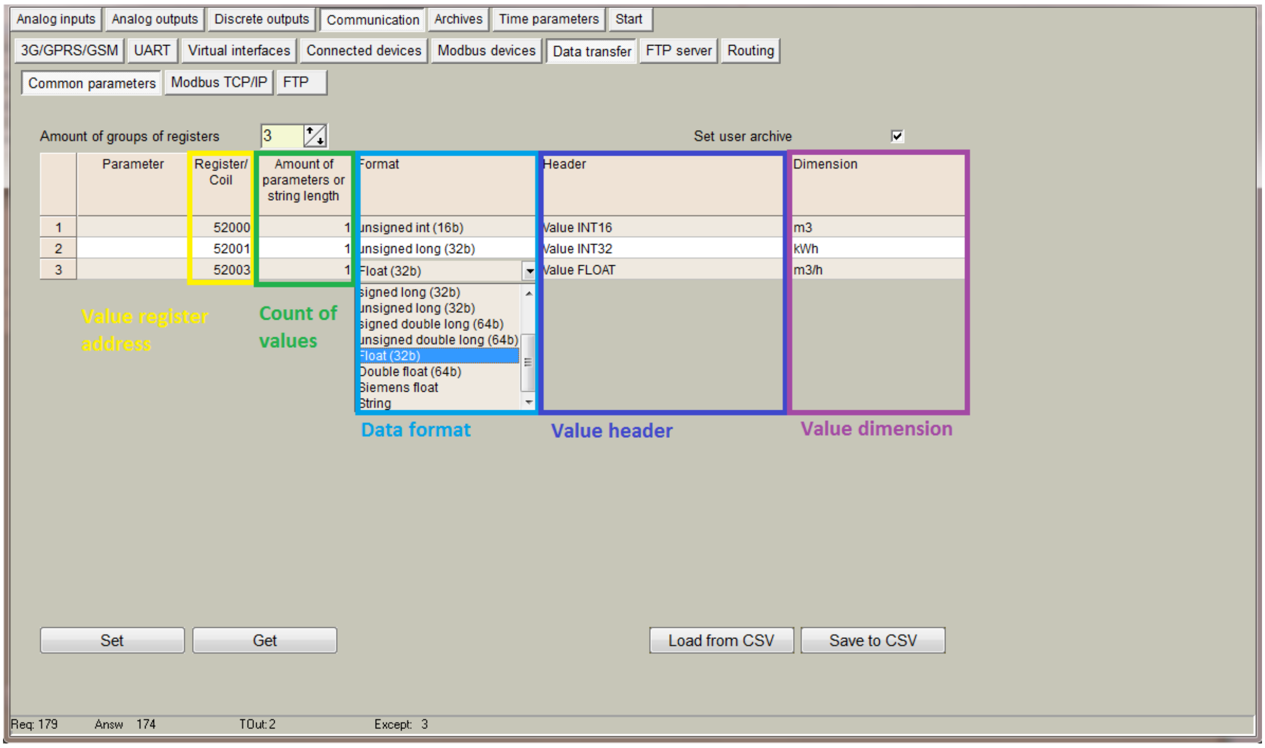
CSV file creator table

Device creates csv report file from user defined archive values and data configured in "Communication/Data transfer/Common parameters" tab or in "Archives/User archive configuration" (from 1.58 version).

- "Amount of groups of register" number field - table lines count
- "Set user archive" checkbox - always check this, it automatically configure 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 - value count from start register. It can be from 1 to 50. This can be used to configure up to 50 values from start register in one line, problem is what you get the same header and the same dimension for every value.
- "Format" column - value format. In Modbus protocol you need know value format before to read it, otherwise you will read hexadecimal values with whom it is difficult to operate. In some firmware version not all formats can be 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 new line with the same timestamp. Used to add new line record with the same time in csv file*
- ○ *Unix time*

- "Header" column - value header in csv file. If count of values is configured more than 1, for all of them is used the same header.
- "Dimension" column - value dimension in csv file. If count of values is configured more than 1, for all of them is used the same dimension.
- "Set" button - write configuration to device.
- "Get" button - read configuration from device.
- "Load from csv" button - load saved configuration table from csv file.
- "Save to csv" button - save configuration table to 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

```
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

Overview

CCV file can be send up to 4 FTP servers. Supported only FTP protocol, passive mode, user/password authentication, file format csv.

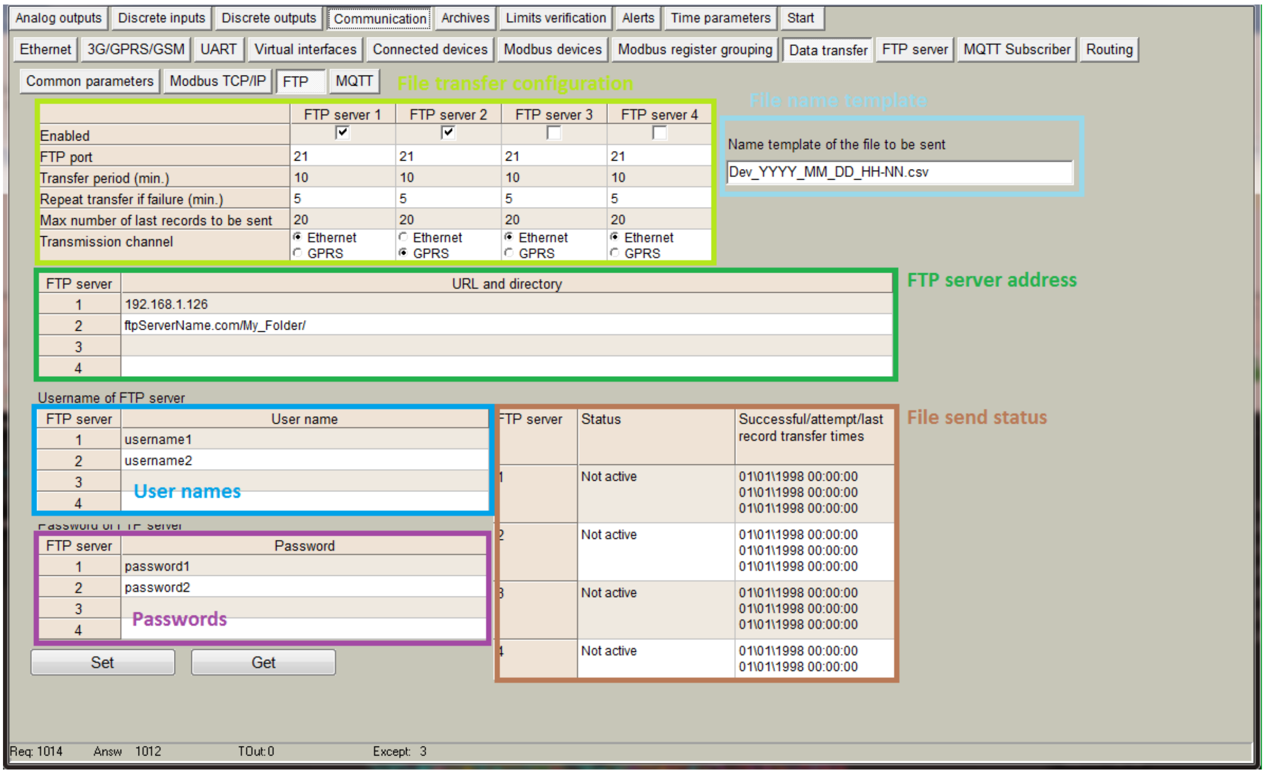
FTP client configuration

All configuration is made in "Communication/Data transfer/FTP" tab. Before configure you need to have working ftp server and have some information like its IP address or URL address and connection user name and password.
Where are some section in this tab:

- "File transfer configuration" section:
  - "Enabled" checkbox - enable appropriate ftp server.
  - "FTP port" - appropriate ftp server TCP port.
  - "Transfer period (min)" - file send period. Value in minutes and can be from 1 min to 1440 min (24 hours).
  - "Repeat transfer if failure (min)" - try resend file after configured time if file transfer failed. Value in minutes and can be from 1 min to 1440 min (24 hours). Recommended value is half transfer period.
  - "Max number of last records to be send" - maximum last records to include in csv file. Only new records will be included in file. If it is time to send file and new records count is 25 and max number of last records configured to 20, in file will be only 20 newest records, last 5 will be lost. If new records count is 5 and max number of last records configured to 20, in file will be 5 new records. Values can be from 1 to 200.
  - "Transmission channel" - device can have 2 transfer channels Ethernet and GPRS. Select which to use for appropriate FTP server.
- "FTP server address" section:
  - "URL and directory" column - IP or URL address of FTP server (Up to 127 symbols). Can be used as IP address - "127.0.0.1" or URL - "www.myftp.com". Also available to use directory listing like : "www.myftp.com/MyFiles/"
- "User names" section:
  - "User name" column - user name for appropriate FTP server.
- "Passwords" section:
  - "Password" column - password for appropriate FTP server.
- "File send status" section:
  - "Status" column - current status of FTP client. After file was send to server, status sets to "File transmitted". Where are several other status for process check: "Connecting to server", "Sending user", "Sending password", "Sending data file" and other.
  - "Successful/attempt/last record transfer times" column - shows times of some operations. "Successful" time is the time of last successful finished file send to FTP server operation. "Attempt" time is the time of last attempt send file to FTP server (can be successful or not). "Last record transfer" time is the last send record time.
- "File name template" section - file name can be up to 127 symbols with extension "csv". In file can be fixed fields which will be changed to date and time. Fixed fields:

  YYYY – year
  MM – month
  DD – day
  HH – hour
  NN – minute

  for example template "Dev_YYYY_MM_DD__HH-NN.csv" and date is 2017.03.25 14:25, so file name will be "Dev_2017_03_25__14-25.csv"
- "Set" button - write configuration to device.
- "Get" button - read configuration from device.



	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

Name template of the file to be sent

Dev_YYYY_MM_DD_HH-NN.csv

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	

Password of FTP server

FTP server	Password
1	password1
2	password2
3	
4	

FTP 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 is configured 2 FTP servers

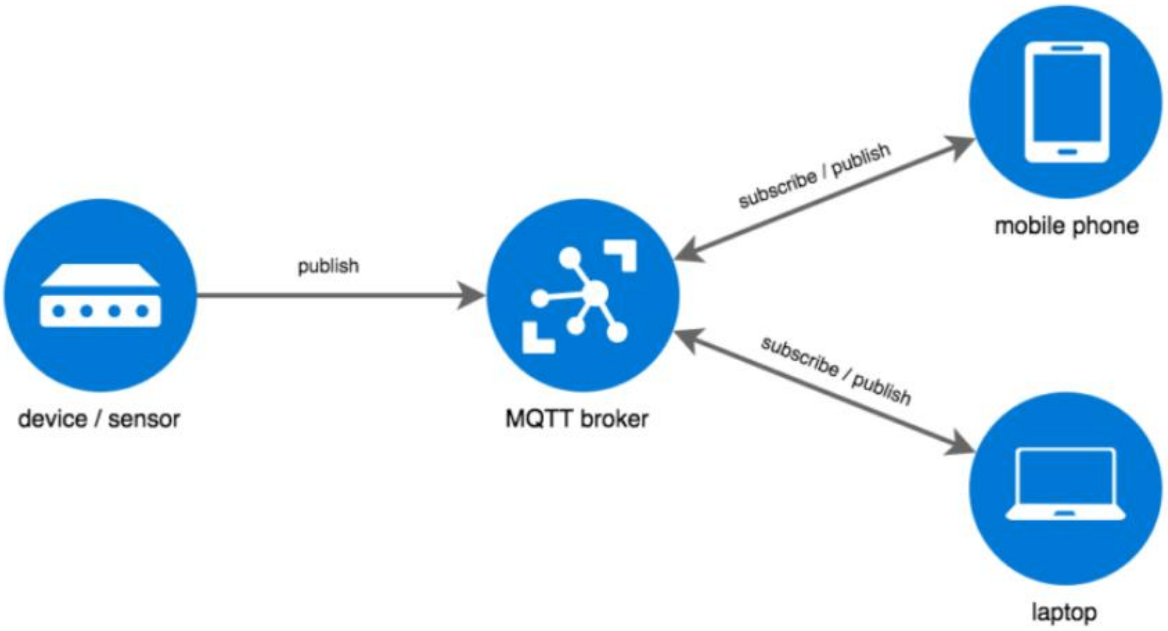
- First will be accessed through Ethernet connection, it's address "192.168.1.126", user name "username1" and password "password1". File will be send every 10min and repeats every 5 min if sending was unsuccessful.
- First will be accessed through GPRS connection, it's address "ftpServerName.com" and directory where to put files "My Folder", user name "username1" and password "password1". File will be sending every 10min and repeats every 5 min if sending was unsuccessful.

MQTT client

Overview

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 whilst also attempting to ensure reliability and some degree of assurance of delivery. These principles also turn out to make the protocol ideal of the emerging “machine-to-machine” (M2M) or “Internet of Things” world of connected devices, and 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 into topics. The broker will then make sure that all other clients that are interested in this status topic will receive the status.

Our device can send event messages or json type data files via MQTT. Data files can be created from real values or archived values depends on configuration. For event messages and data files different topics are used.

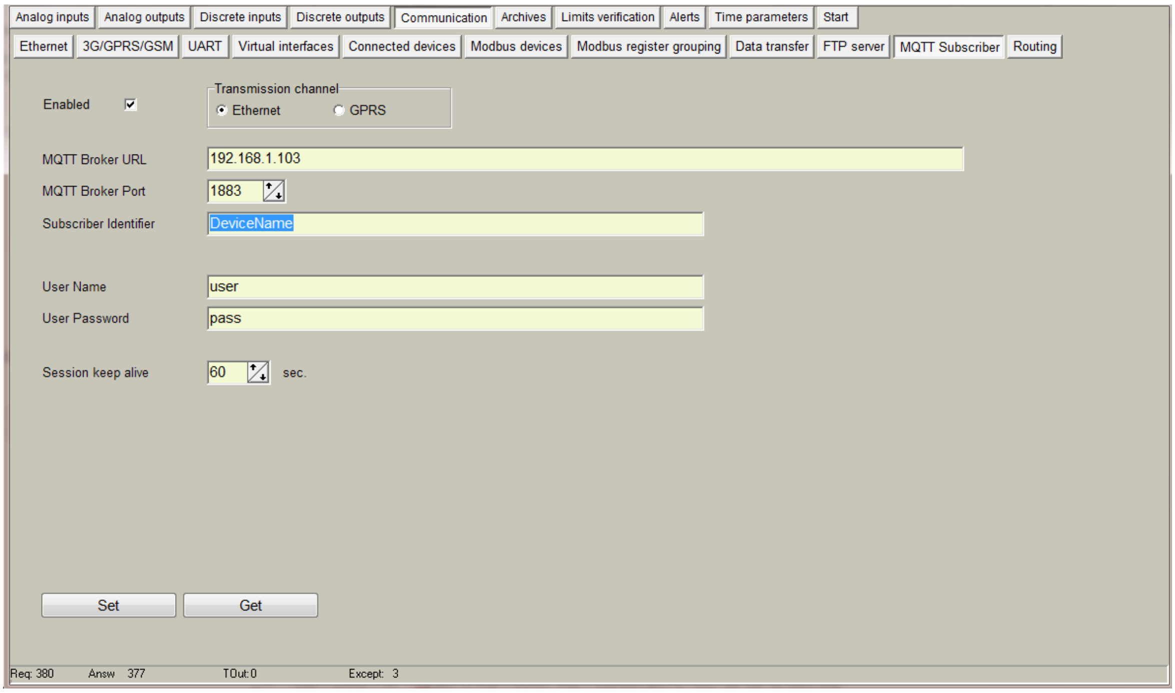


MQTT client configuration

MQTT client configuration is made in "Communication/MQTT Subscriber" tab. Before configure you need to have working MQTT broker and have some information like its IP address or URL address, port, user name and password.

- "Enabled" checkbox - enable MQTT client on device
- "Transmission channel" - device can have 2 transfer channels Ethernet or GPRS. Select which one you want to use for connection to MQTT broker.
- "MQTT Broker URL" - IP or URL address of MQTT broker. Can be up to 63 symbols
- "MQTT Broker Port" - TCP port of MQTT broker. MQTT standard port is 1883, but can be used any.
- "Subscriber Identifier" - The client identifier is an identifier of each MQTT client connecting to a MQTT broker. As the word identifier already suggests, it should be unique per broker. The broker uses it for identifying the client and the current state of the client.
- "User Name" - client username.
- "User Password" - client password.
- "Session keep alive" - send keep alive packet every xx seconds
- "Set" button - write configuration to device.
- "Get" button - read configuration from device.

Device connects with configured parameters and keeps connected until disconnects for any reasons. After disconnection, automatically is established new connection. Still device is connected to MQTT broker it can send event messages or data report files.



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

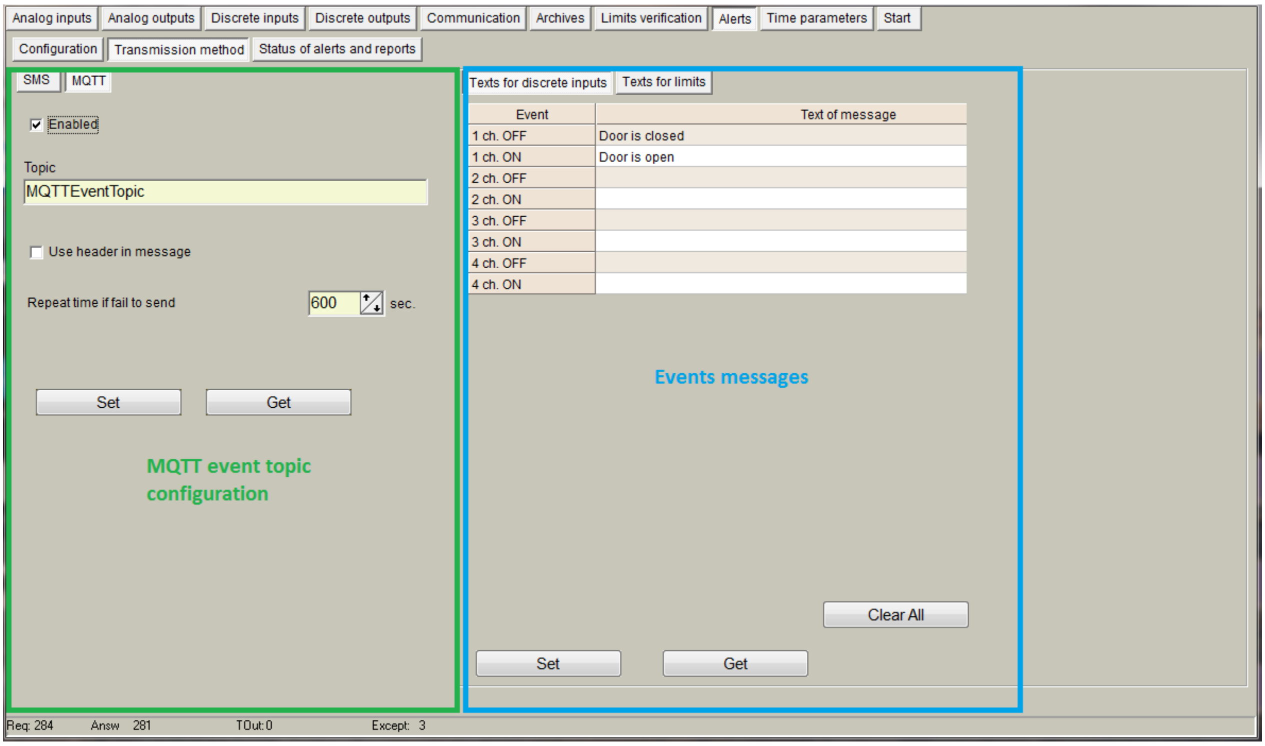
Session keep alive: 60 sec.

Req: 380 Answ 377 TOut:0 Except: 3

Event messages configuration

Event message sending to MQTT broker configure in "Alerts/Transmission method/MQTT" tab. Before sending event messages, message text have to be configured:

- "Events messages" section:
  - "Text for discrete inputs" - every discrete input have two states ON and OFF. For every state can be configured individual message.
  - "Texts for limits" - can be configured up to 100 messages for limits verification module. So, every index have its own message.
  - "Set" button - write configuration to device.
  - "Get" button - read configuration from device.
  - "Clear" button - clear all message texts.
- "MQTT event topic configuration" section:
  - "Enabled" check box - enable event message sending to MQTT broker
  - "Topic" - publish topic for event messages. Device sends event messages with this topic
  - "Use header in message" checkbox - if checked, header is added for every message. It is useful if you want to group messages by device with the same topic, so in header you can write device name, identification number, address or something else. Headers can be configured in "Start" tab "Other parameters" section.
  - "Repeat time if fail to send" - if event message delivery fails, device try repeat it after xx seconds
  - "Set" button - write configuration to device.
  - "Get" button - read configuration from 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	

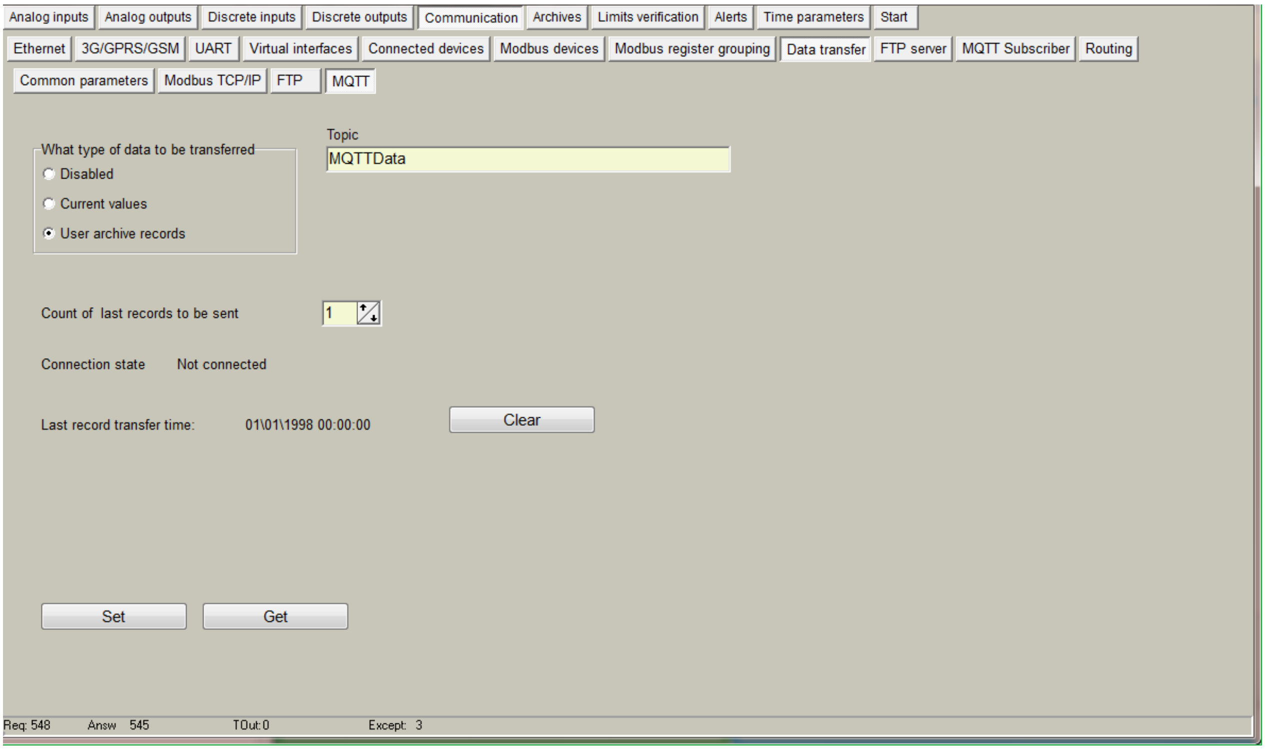
Req: 284 Answ 281 TOut:0 Except: 3

Report data file configuration

Data report file can be created from real values or archived values. Needed to send value configuration is the same like FTP client and is configured in "Communication/Data transfer/Common parameters" only difference FTP client sends in CSV format, MQTT client in JSON format. FTP client and MQTT client use the same data value configuration table.

Configuration:

- "What type of data to be transferred" section:
  - "Disabled" - disables data send to MQTT broker.
  - "Current values" - sends data created from current values. Sends file every configured period from 1s to 86400 seconds (24 hours).
  - "User archive records" - sends data created from archived values. In this mode you need to configure how many records send in one file. File will be created and send when new archive record appears
- "Topic" - MQTT topic for json file transfer
- "Count of last records to be sent" - how many last records to send if was connection problems. This option is available if transfer type is "User archive records"
- "Current values transmission period" - transfer period of current values (period in seconds). This option is available if transfer type is "Current values"
- "Connection state" - current state. This option is available if transfer type is "User archive records"
- "Last record transfer time:" - time of last send record. This option is available if transfer type is "User archive records"
- "Clear" button - reset time of last send record. This option is available if transfer type is "User archive records"
- "Set" button - write configuration to device.
- "Get" button - read configuration from 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

Data logger control via MQTT

MQTT command structure

```
{
  "<Command>": [
    { "<Register 0 address>": "<value>"},
    {   "<Register 1 address>": "<value>"},
...................................................................
    {   "<Register n address>": "<value>"}
  ]
}
```

<Command> - Possible values:
"R_Read" - read register
"R_Write" - write register

<Register x address> - start register address

<value> - If command == "R_Read" means how many registers to read from start register address. At this moment allowed value - 1.
If command == "R_Write" means new register value
Maximum allowed registers in one message - 16.

Answers:
If write successful returns true:
{"A_Write" : true}

if read successful returns register values.

Example:

Write value 12354 to 52000 register, value 5678 to 52001 register. value 9136 to 52002 register.
Command :

```
{
 "R_Write": [
{ "52000": "12354"},
 { "52001": "5678"},
 { "52002": "9136"}
 ]
}
```

Answer if registers was written:

```
{"A_Write" : true}
```

Answer if can't write registers:

```
{"A_Write" : false}
```

//---//

Read 4900 and 4901 registers.

Command:

```
{
 "R_Read": [
{ "4900": "1"},
 { "4901": "1"}
 ]
}
```

Answer if registers is illegal:

```
{"A_Read" : false}
```

Answer:

```
{
 "A_Read": [
{ "4900": "12354"},
 { "4901": "5678"}
  ]
}
```

TCP/IP connection table

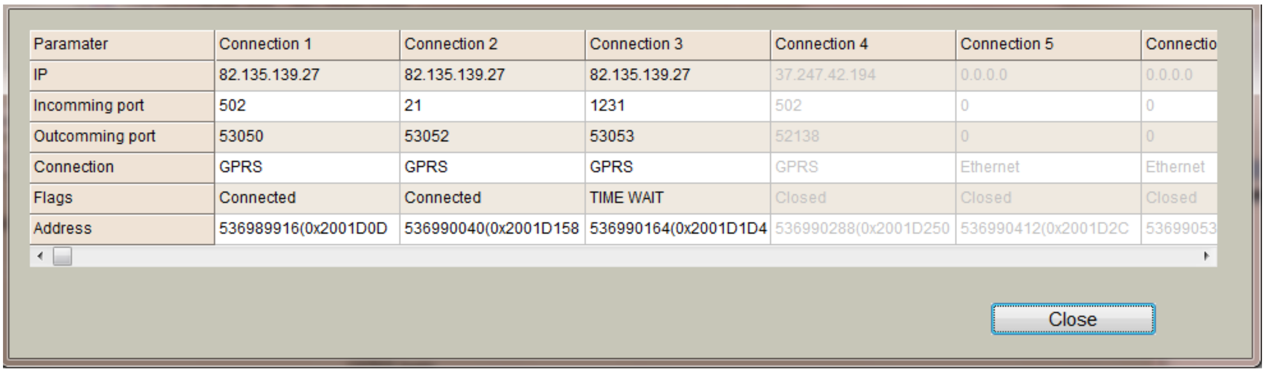
Overview

Internal TCP/IP stack can accept up to 40 connections at the same time. It includes both channels GPRS and Ethernet. It is useful for debug purpose to check current connections, opened ports and etc.

TCP/IP connection table

Open TCP/IP connection table by pressing CTRL+F1. New window opens "Stack Information". Table consist of 40 connections columns and connection parameters:

- "IP" - remote IP address.
- "In coming port" - local TCP port.
- "Out coming port" - remote TCP port
- "Connection" - network channel (GPRS or Ethernet)
- "Flags" - connection state. (Closed, Connecting, Connected, Time Wait)
- "Address" - internal address where is stored connection structure



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	53699053

Example

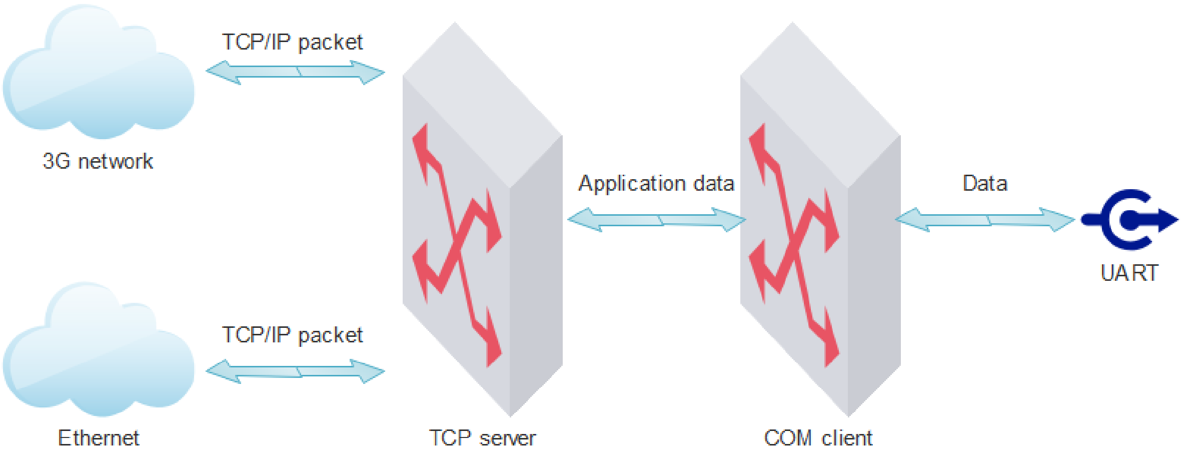
In the picture above 3 connections is established.

- 1 connection remote host 82.135.139.27 connected to 502 port (Modbus TCP/IP connection). Connection established through 3G/GPRS channel.
- 2 connection remote host 82.135.139.27 connected to 21 port (FTP). Connection established through 3G/GPRS channel.
- 3 connection remote host 82.135.139.27 connected to 1231 port. Connection established through 3G/GPRS channel and waiting to close (TIME WAIT)

Routing TCP/IP - serial (request/answer)

Overview

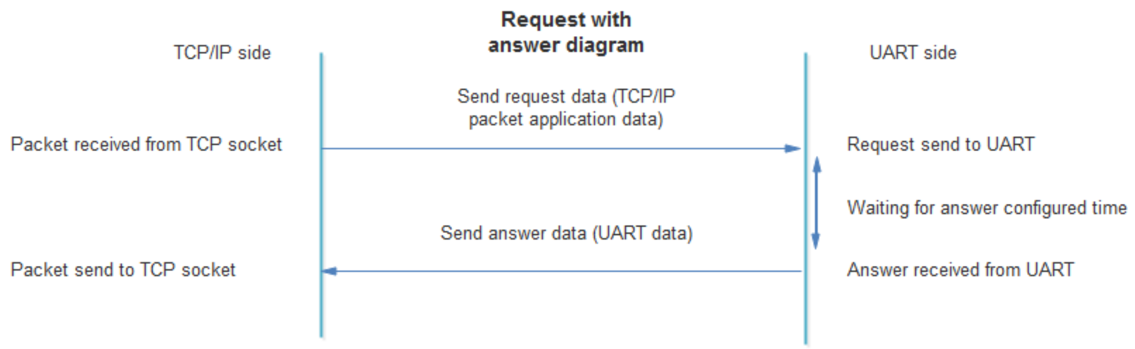
TCP server routes application data (request) from TCP network to serial port and return the serial data (answer) to TCP network. TCP server works through virtual service - COM client so other services can access UART too.TCP server resends application data to UART and back without data modifications.

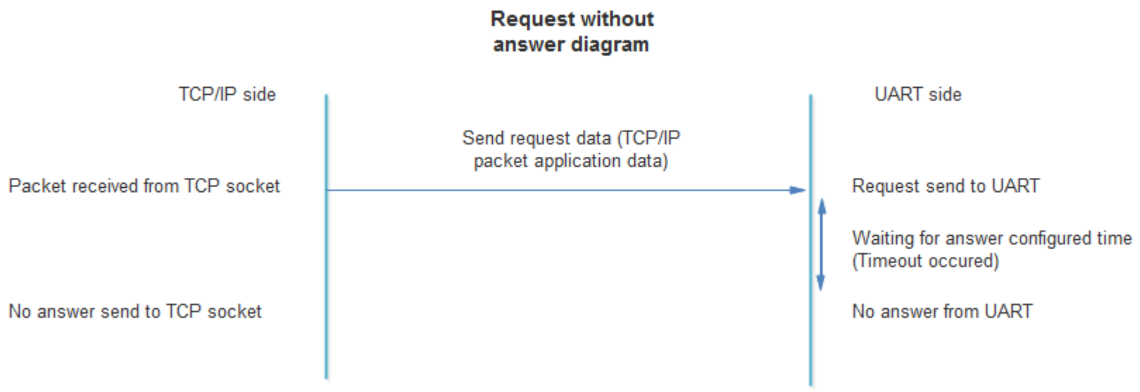


In diagram below we can see how packets are transmitted from TCP/IP sockets to UARTs. After TCP socket is opened all application data through this socket goes to UART. Process like this:

1. TCP socket opened
2. Waiting for TCP/IP packet
3. TCP/IP packet received. it's application data send to COM client and to UART
4. Waiting for answer configured time (COM client timeout)
5. If answer received sends this data to TCP socket
6. Waiting for next request ...

TCP server use virtual COM client to connect to UART so it waits for answer configured time (COM client timeout), if no packets received on UART in this time - nothing is send to 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

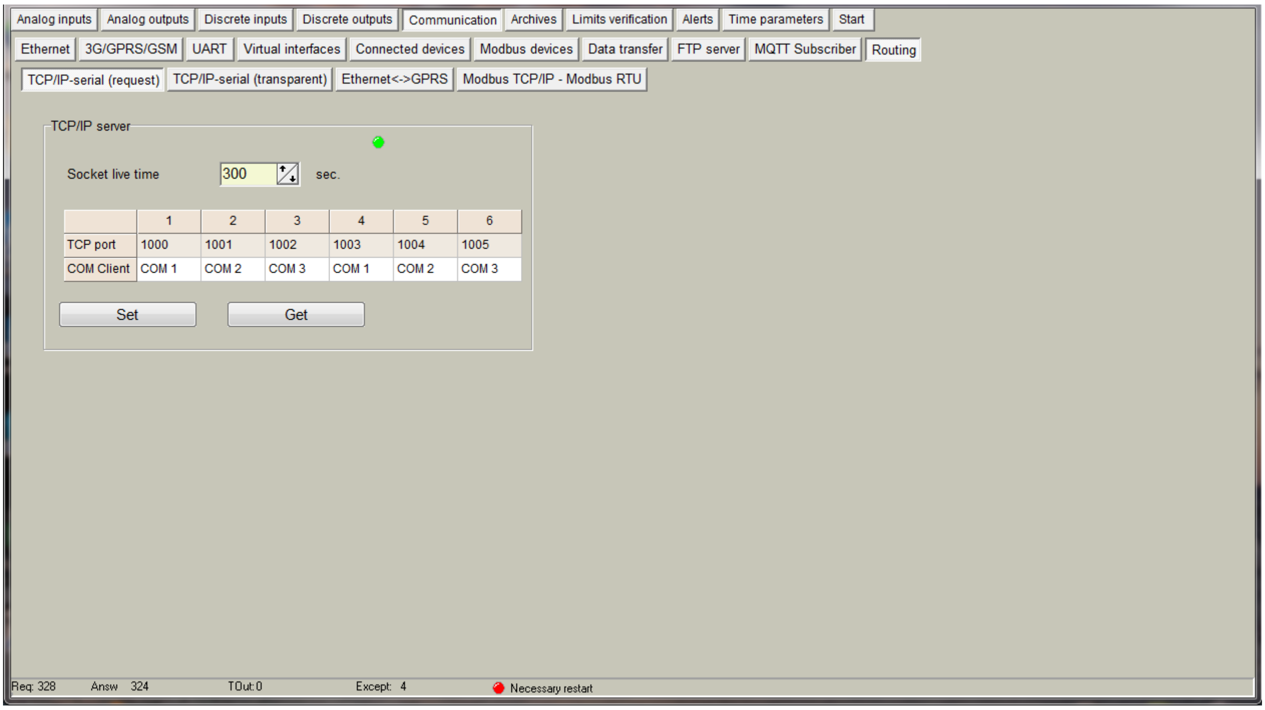

Request without answer diagram
TCP/IP side
UART side
Send request data (TCP/IP packet application data)
Packet received from TCP socket
Request send to UART
Waiting for answer configured time (Timeout occured)
No answer send to TCP socket
No answer from UART

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

TCP server configuration

TCP server configuration is made in "Communication/Routing/TCP/IP-serial(request)" tab. Device can open up to 6 different TCP ports for data transfer to associated virtual COM client (Check more COM client topic how it is connected to physical UART).

- "Socket live time" - socket timeout in seconds (60-65535 seconds). If no data is transferred in configured time, device automatically close socket.
- "TCP port" - internal socket TCP port (1 -65535). Device waiting connection to configured ports and opens data transfer channel with appropriate virtual COM client.
- "COM Client" - virtual COM client associated with the corresponding TCP port (COM1 - COM3).
- "Set" button - write configuration to device.
- "Get" button - read configuration from device.



TCP/IP server

Socket live time 300 sec.

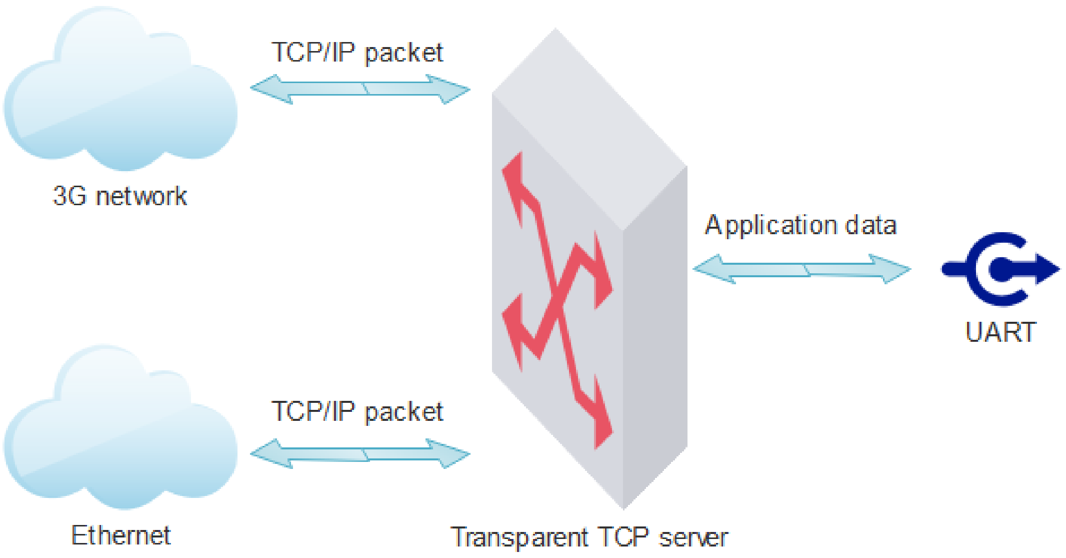
	1	2	3	4	5	6
TCP port	1000	1001	1002	1003	1004	1005
COM Client	COM 1	COM 2	COM 3	COM 1	COM 2	COM 3

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

Routing TCP/IP - serial (transparent)

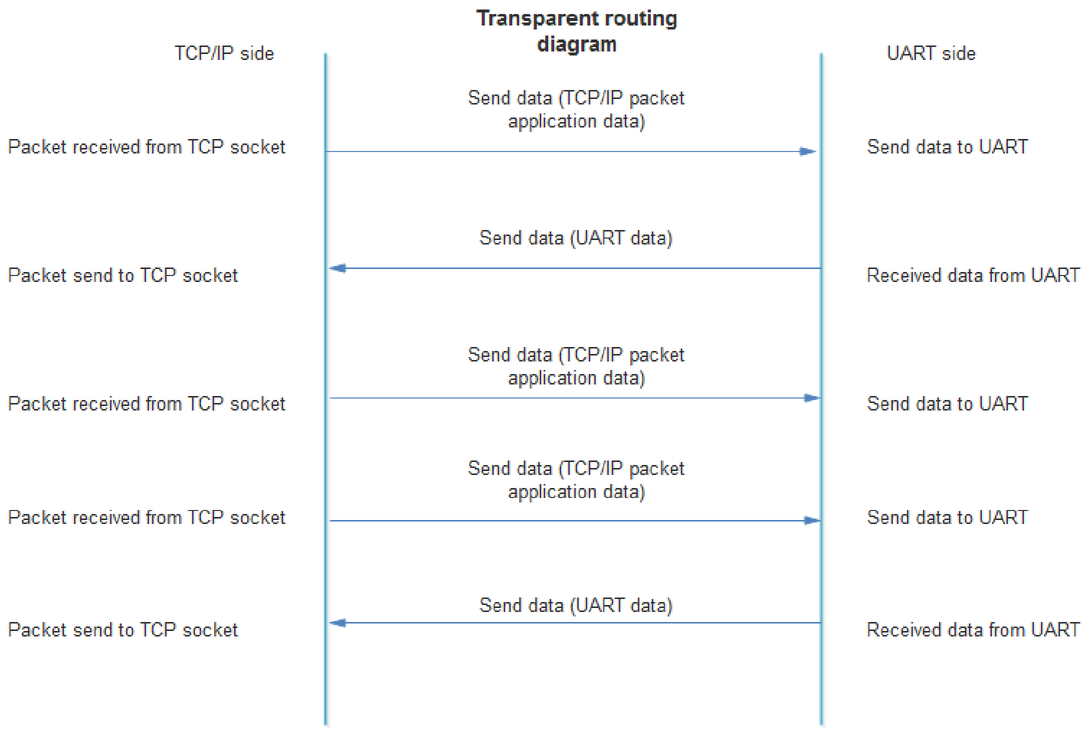
Overview

Transparent TCP server routes application data from TCP/IP network to UART and routes the serial data from UART to TCP/IP network. Difference from TCP server - Transparent TCP server works directly with UARTS and all data is transferred in both direction (no answer timeouts).



In diagram below we can see how packets are transmitted from TCP/IP sockets to UARTs and back. After TCP socket is opened all application data through this socket goes to UART and all data from UART goes to TCP socket. Process like this:

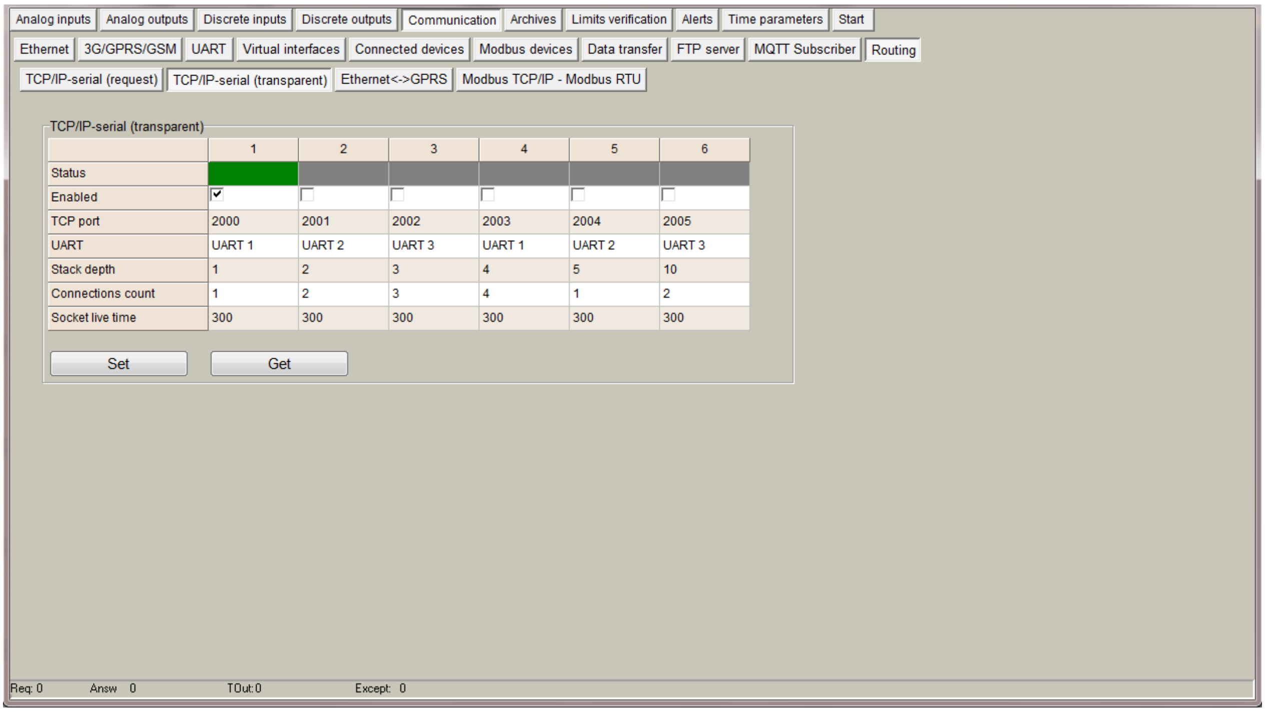
1. TCP socket opened
2. Waiting for TCP/IP packet or data on UART
3. TCP/IP packet received. it's application data send to UART. If UART data is received, data is directly send to TCP socket
4. Waiting for next TCP/IP packet or data on UART


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

Transparent TCP server configuration

Transparent TCP server configuration is made in "Communication/Routing/TCP/IP-serial(transparent)" tab. Device can open up to 6 different TCP ports for data transfer to associated UART.

- "Status" - green - server is working, grey - server is stopped
- "Enabled" check box - enable /disable appropriate transparent TCP server
- "TCP port" - internal socket TCP port (1 -65535). Device waiting connection to configured ports and opens data transfer channel with appropriate UART.
- "UART" - UART associated with the corresponding TCP port (UART1 - UART3).
- "Stack depth" - number of packets can be processed at the same time (1 - 10). TCP/IP network is much faster than serial UART, so packets can be put to queue and send to 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 UART and UART data will be returned to all TCP sockets. if connections reached maximum available connection count, new connection will be refused.
- "Socket live time" - socket timeout in seconds (60-65535 seconds). If no data is transferred in configured time, device automatically close socket.
- "Set" button - write configuration to device.
- "Get" button - read configuration from device.



TCP/IP-serial (transparent)

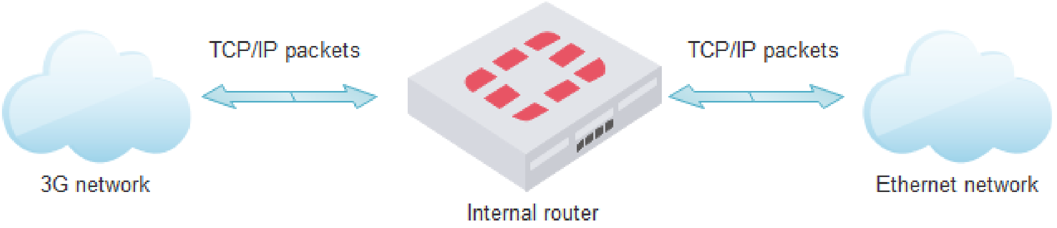
	1	2	3	4	5	6
Status	🟩	▨	▨	▨	▨	▨
Enabled	☑	□	□	□	□	□
TCP port	2000	2001	2002	2003	2004	2005
UART	UART 1	UART 2	UART 3	UART 1	UART 2	UART 3
Stack depth	1	2	3	4	5	10
Connections count	1	2	3	4	1	2
Socket live time	300	300	300	300	300	300

Req: 0 Answ 0 TOut:0 Except: 0

Router 4G/GPRS<->Ethernet

Overview

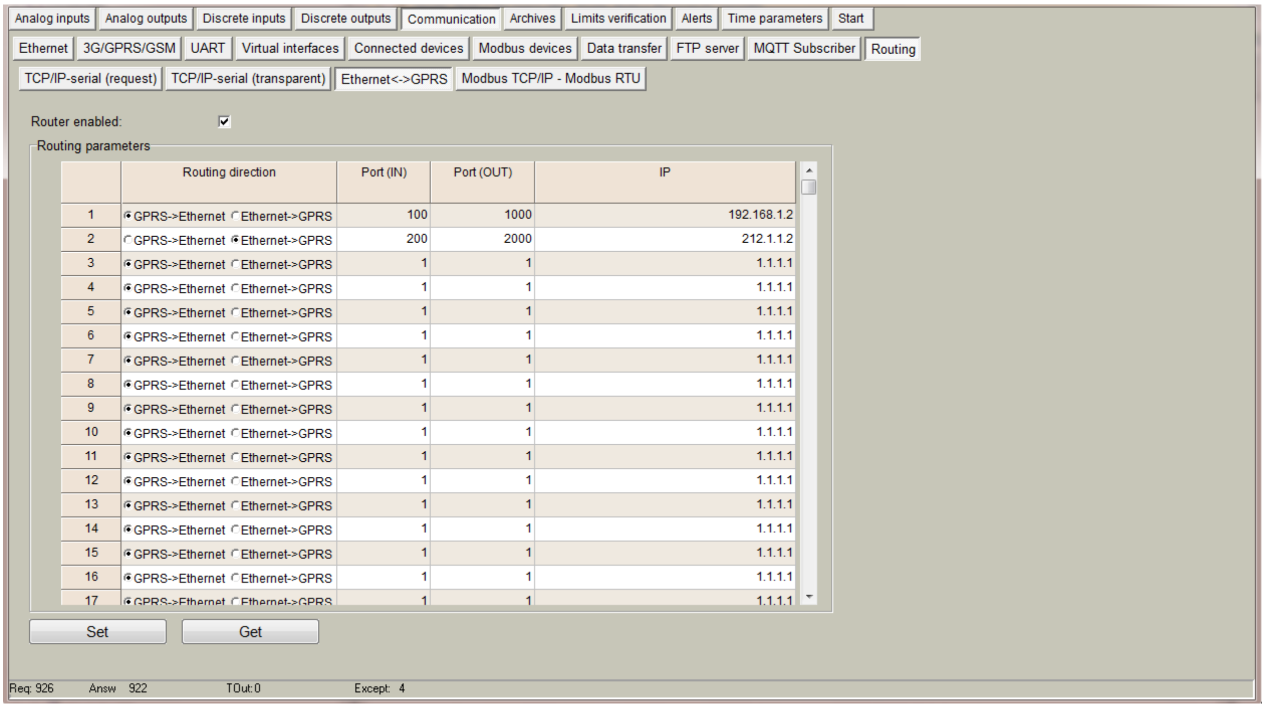
Purpose of 3G/GPRS<->Ethernet router is to access Ethernet connected devices from 3G/GPRS network. It routes TCP/IP packets from 3G/GPRS network to Ethernet and back.



Router configuration

3G/GPRS<->Ethernet router configuration is made in "Communication/Routing/Ethernet<->GPRS" tab. Device can route up to 20 different TCP ports.

- "Routing direction" - indicates how packets are routed from GPRS to Ethernet, or from Ethernet to GPRS. From Direction depends how "Port (IN)", "Port(Out)" and "IP" values are used
- "Port (IN)" - device incoming port. If direction is GPRS to Ethernet this port will be incoming port from 3G/GPRS network. If direction is Ethernet to GPRS this port will be incoming port from Ethernet network.
- "Port (Out)" - device outgoing port. If direction is GPRS to Ethernet this port will be outgoing port to Ethernet network. If direction is Ethernet to GPRS this port will be outgoing port to GPRS network.
- "IP" - device outgoing IP address. If direction is GPRS to Ethernet this IP will be outgoing IP to Ethernet network. If direction is Ethernet to GPRS this IP will be outgoing IP to GPRS network.
- "Set" button - write configuration to device
- "Get" button - read configuration from device



Analog inputs | Analog outputs | Discrete inputs | Discrete outputs | Communication | Archives | Limits verification | Alerts | Time parameters | Start

Ethernet | 3G/GPRS/GSM | UART | Virtual interfaces | Connected devices | Modbus devices | Data transfer | FTP server | MQTT Subscriber | Routing

TCP/IP-serial (request) | TCP/IP-serial (transparent) | Ethernet<->GPRS | Modbus TCP/IP - Modbus RTU

Router enabled: ☑

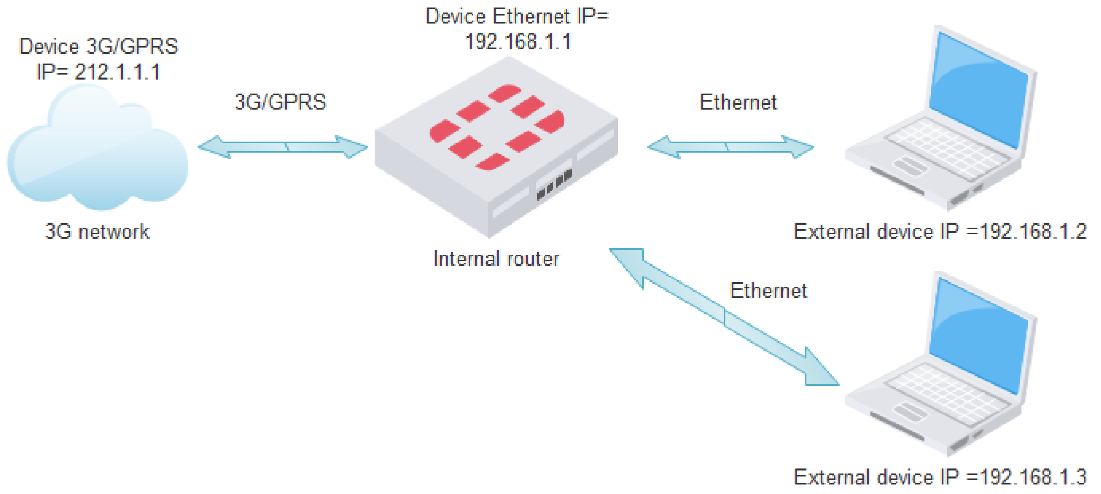
Routing parameters

	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

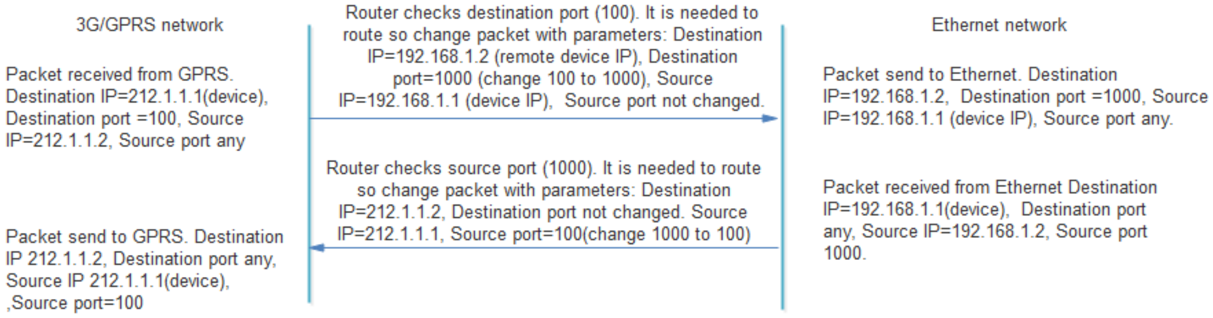
Example

In picture above we can see configuration of two router lines. General connection diagram can be like this:



- First line, routing from GPRS to Ethernet, incoming port 100, outgoing port 1000, and outgoing Ethernet IP 192.168.1.2

Routing diagram



3G/GPRS network

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

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

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.

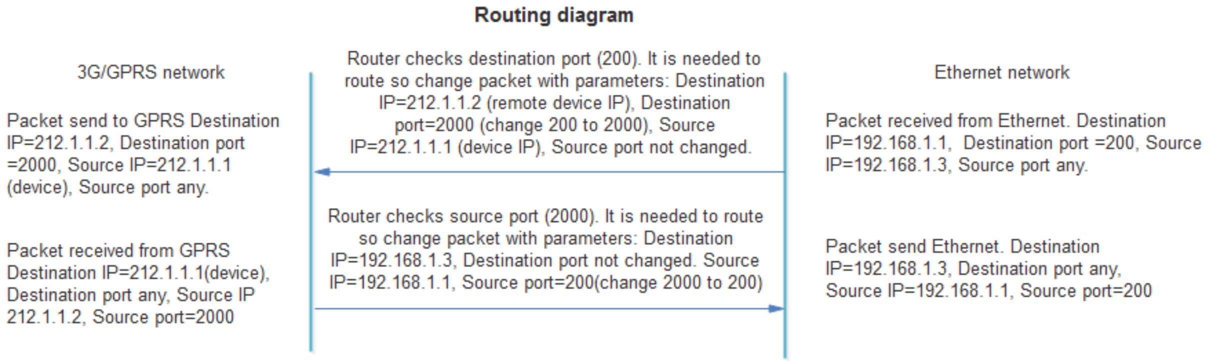
Router checks source port (1000). It is needed to route so change packet with parameters: Destination IP=212.1.1.2, Destination port not changed. Source IP=212.1.1.1, Source port=100(change 1000 to 100)

Ethernet network

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.

- Second line, routing from Ethernet to GPRS, incoming port 200, outgoing port 2000, and outgoing GPRS IP 212.1.1.2



Routing diagram

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

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

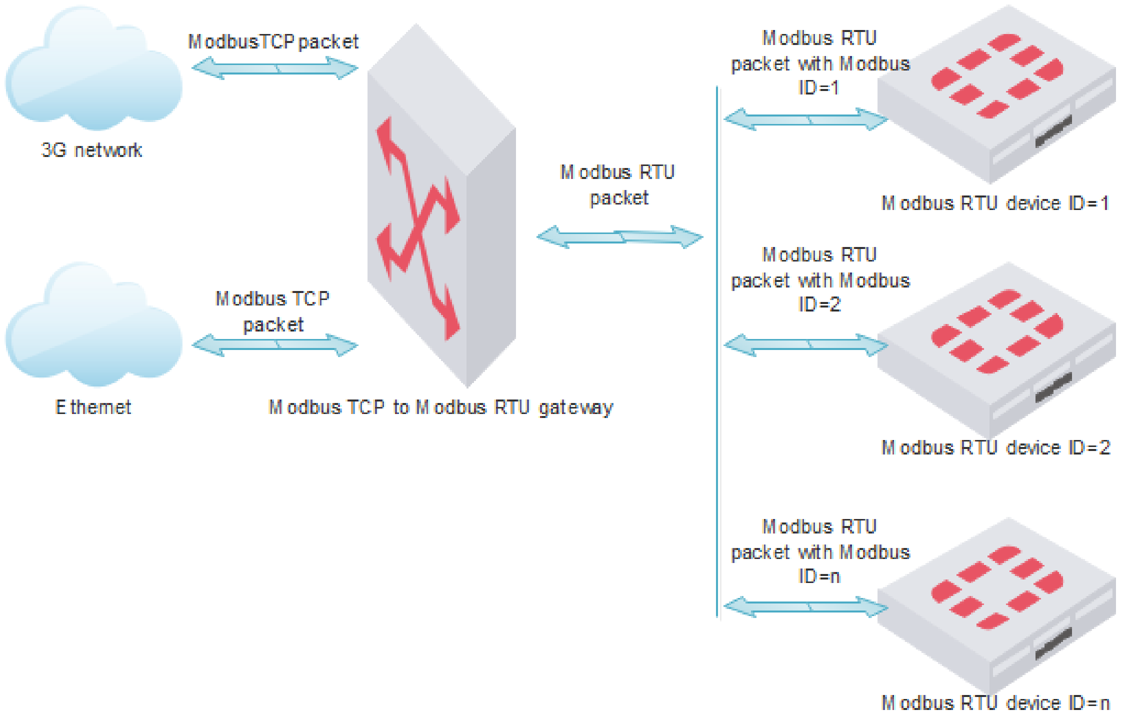
Packet send Ethernet. Destination IP=192.168.1.3, Destination port any, Source IP=192.168.1.1, Source port=200

Gateway Modbus TCP<->Modbus RTU

Overview

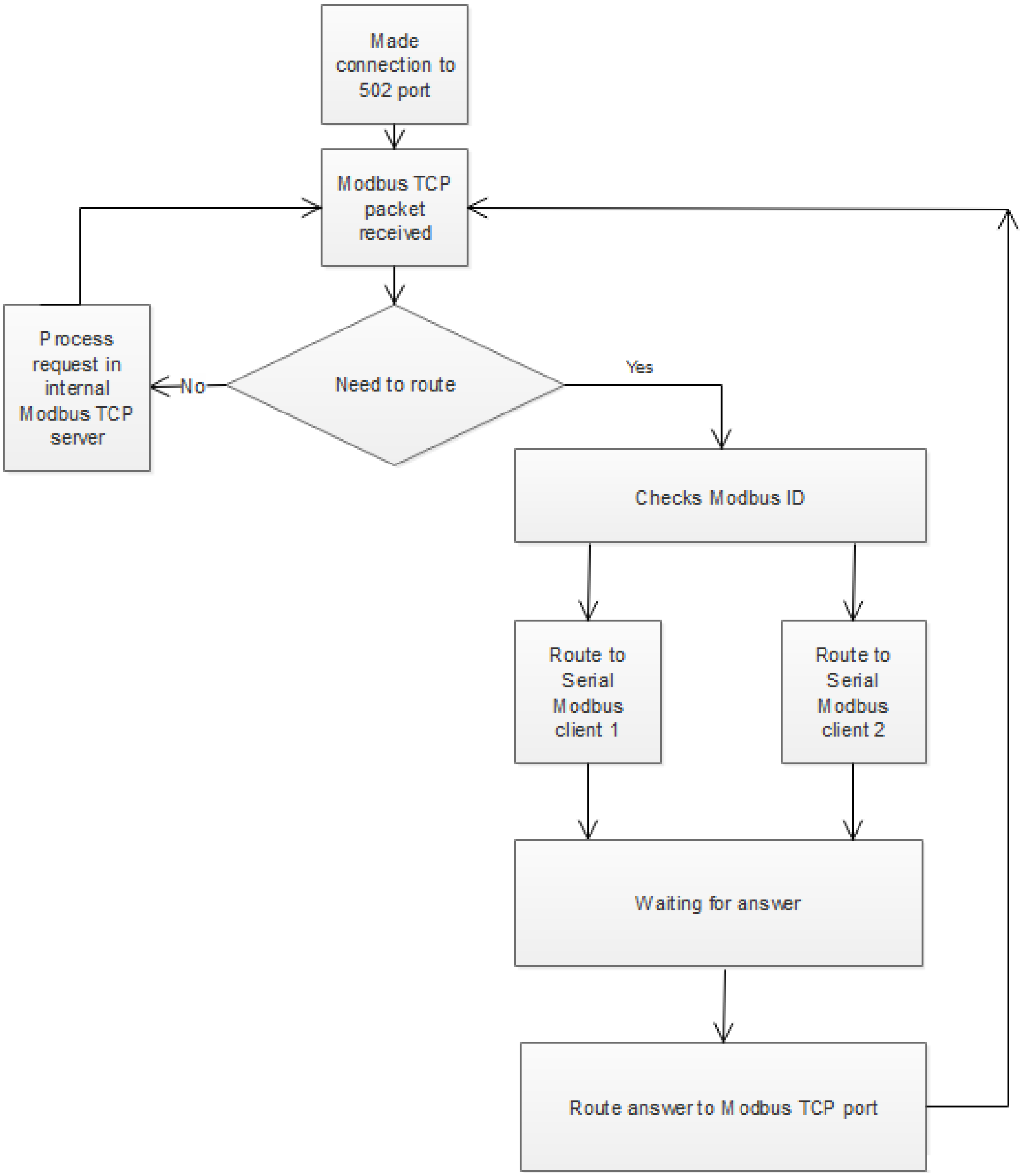
Modbus TCP<->Modbus RTU gateway is used directly to access Modbus RTU device connected to our device using Modbus TCP protocol. If route function is used, received Modbus TCP packet is changed to Modbus RTU packet and send to appropriate Serial Modbus client. In TCP network side device works as Modbus TCP server, in serial side as Modbus RTU master.

Standard network diagram:

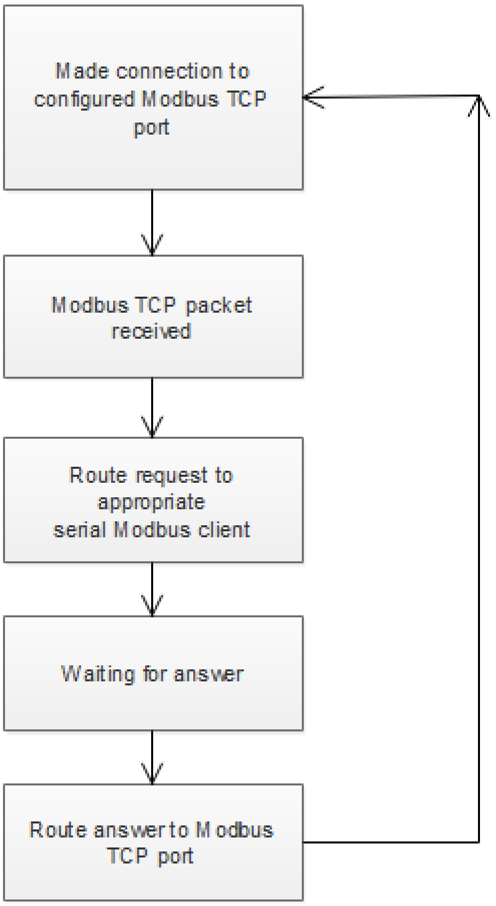

ModbusTCP packet
3G network
Modbus TCP packet
Ethernet
Modbus TCP to Modbus RTU gateway
Modbus RTU packet
Modbus RTU packet with Modbus ID=1
Modbus RTU device ID=1
Modbus RTU packet with Modbus ID=2
Modbus RTU device ID=2
Modbus RTU packet with Modbus ID=n
Modbus RTU device ID=n

Where are two route modes:

- Route by Modbus Device ID. For TCP connection is used standard Modbus TCP port 502 and packets are routed depending on the Modbus ID. Working algorithm:


Made connection to 502 port
Modbus TCP packet received
Need to route
No
Process request in internal Modbus TCP server
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. For TCP connection is used configured TCP port and all Modbus TCP packets are routed to appropriate serial Modbus client. Working algorithm:


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

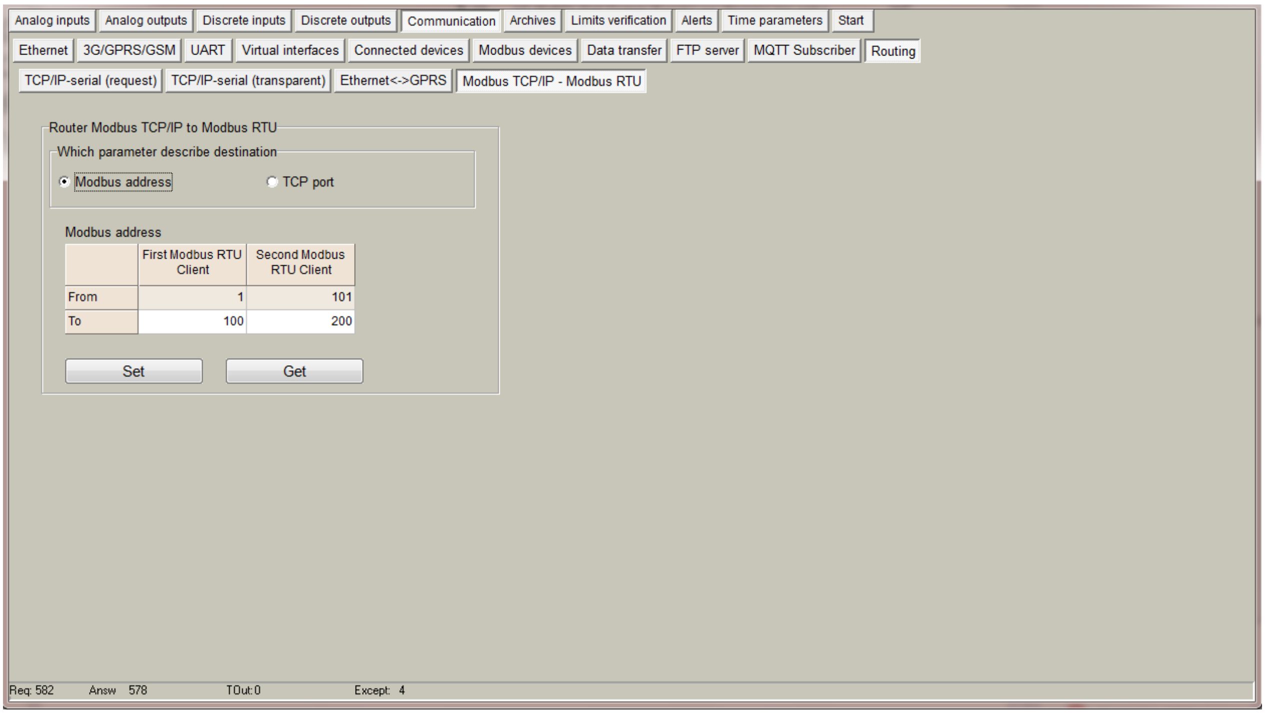
Supported for route Modbus functions:

- 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

Gateway Modbus TCP<->Modbus RTU configuration is made in "Communication/Routing/Modbus TCP/IP<->Modbus RTU" tab. All serial data transmission goes through Modbus RTU clients (Check more Modbus RTU client topic to understand how it is connected to physical UART)

- Route by Modbus Device ID. Select "Modbus address" option in "Which parameter describe destination" section.
  - "Modbus address" table - for every Modbus RTU client can be defined Modbus ID list (From <-> To) and all Modbus packets with Modbus ID from this list is routed to appropriate Modbus RTU client. In picture below all packets with Modbus ID from 1 to 100 will be routed to first Modbus RTU client and all packets with Modbus ID from 101 to 200 will be routed to second Modbus RTU client. All other packets will be processed in internal Modbus server.
  - "Set" button - write configuration to device
  - "Get" button - read configuration from device



Router Modbus TCP/IP to Modbus RTU

Which parameter describe destination

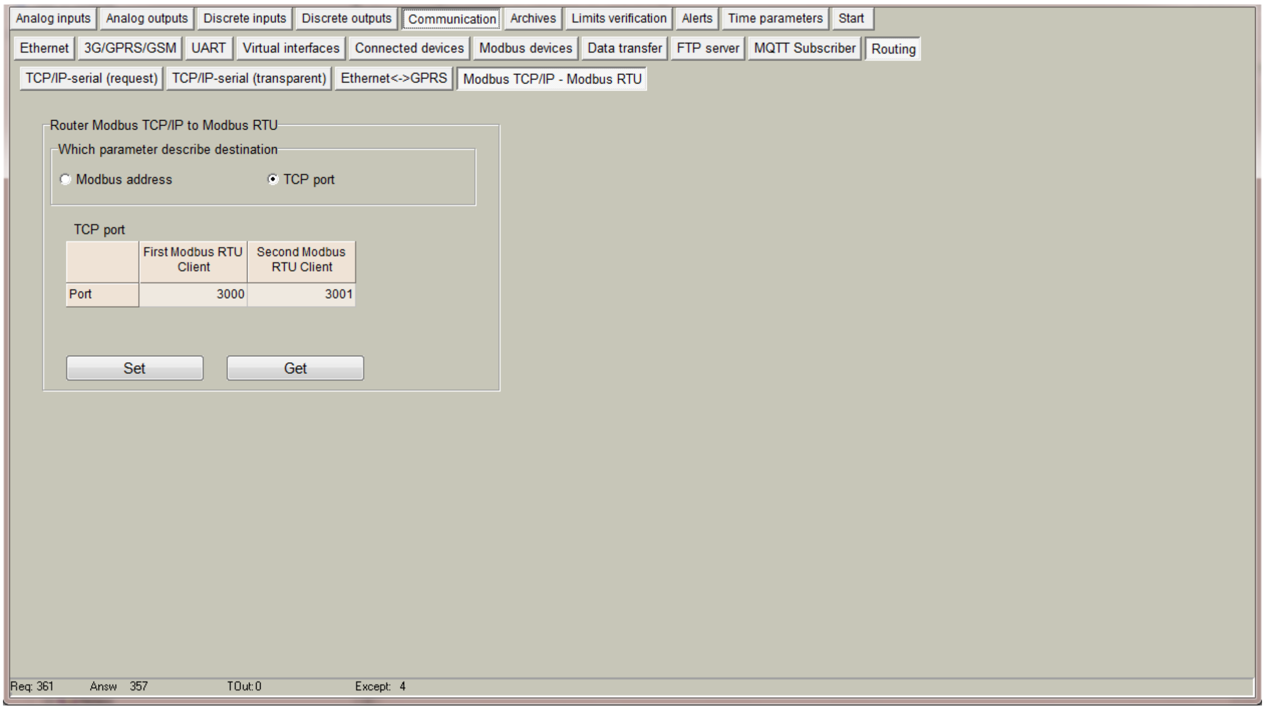
● Modbus address ○ TCP port

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 "TCP port" option in "Which parameter describe destination" section.
  - "TCP port" table - for every Modbus RTU client can be defined individual TCP port. All Modbus packets which are sent to this TCP ports is routed to appropriate Modbus RTU client. In picture below all packets sent to TCP port 3000 will be routed to first Modbus RTU client and all packets sent to TCP port 3001 will be routed to second Modbus RTU client.
  - "Set" button - write configuration to device
  - "Get" button - read configuration from 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

Hardware

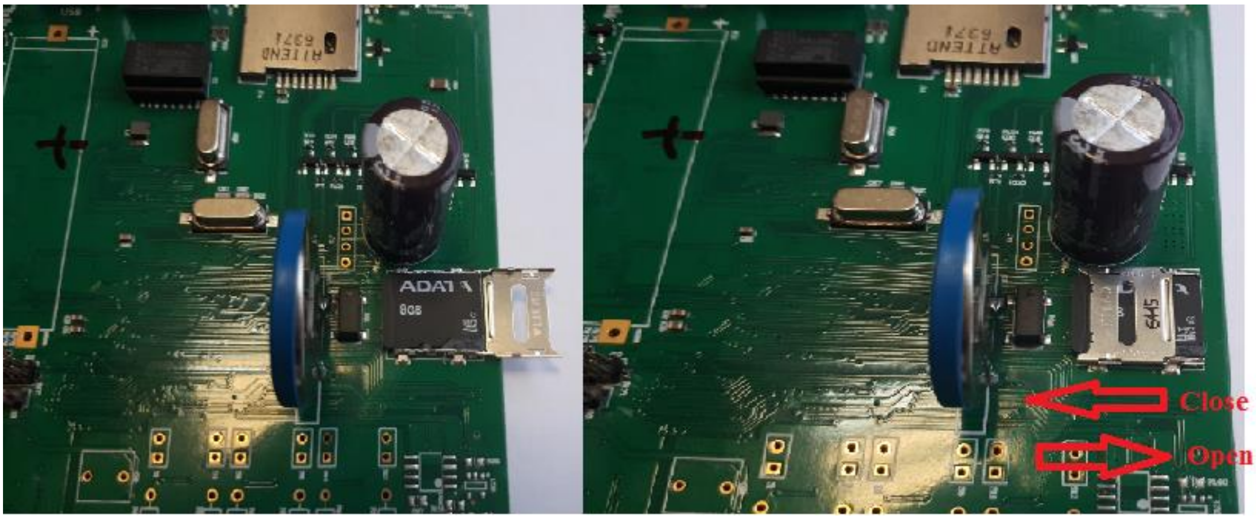
microSD

Overview

If ordered device support microSD card slot, device memory can be extended with microSD card. MicroSD is used to store archives and generate report files for FTP and MQTT clients. Content of micro SD card can be accessed through internal FTP server.

Installing a microSD Card

- Remove upper cover of device to access microSD card slot.
- Push microSD slot cover to right to open it
- Insert microSD card to slot
- Push microSD slot cover to left to close it



Time settings

Overview

Device has integrated battery backed real time clock (RTC) with calendar. RTC works in UTC time and user can configure it return local time by its location (select 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 of the time zones on land are offset from Coordinated Universal Time (UTC) by a whole number 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 on wikipedia.

Setting time with configuration tool

Time settings can be changed in "Time parameters" tab

Ethernet configuration

Overview

Ethernet interface is used to connect device to Local Area Networks (LAN's) and remotely access device. Device support 10 Mbps and 100 Mbps networks. Ethernet interface is used for:

- Data transfer
- Events transfer

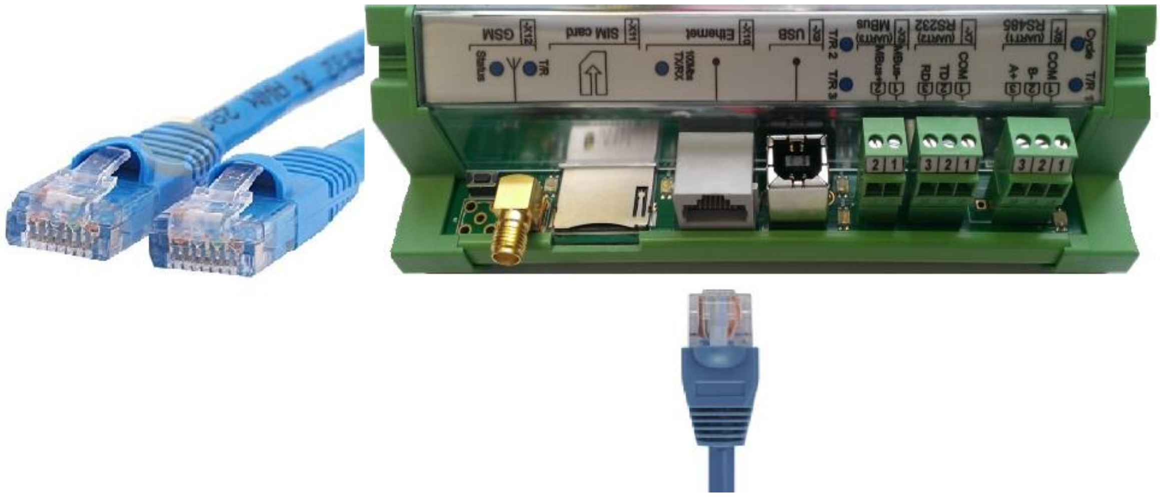
- Clock time synchronization
- Device configuration
- Firmware upgrade
- Etc...

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 GPRS/3G network

Ethernet connection and port pin-outs

Use standard RJ45 cable to connect device to Ethernet router or switch.



Led indicators

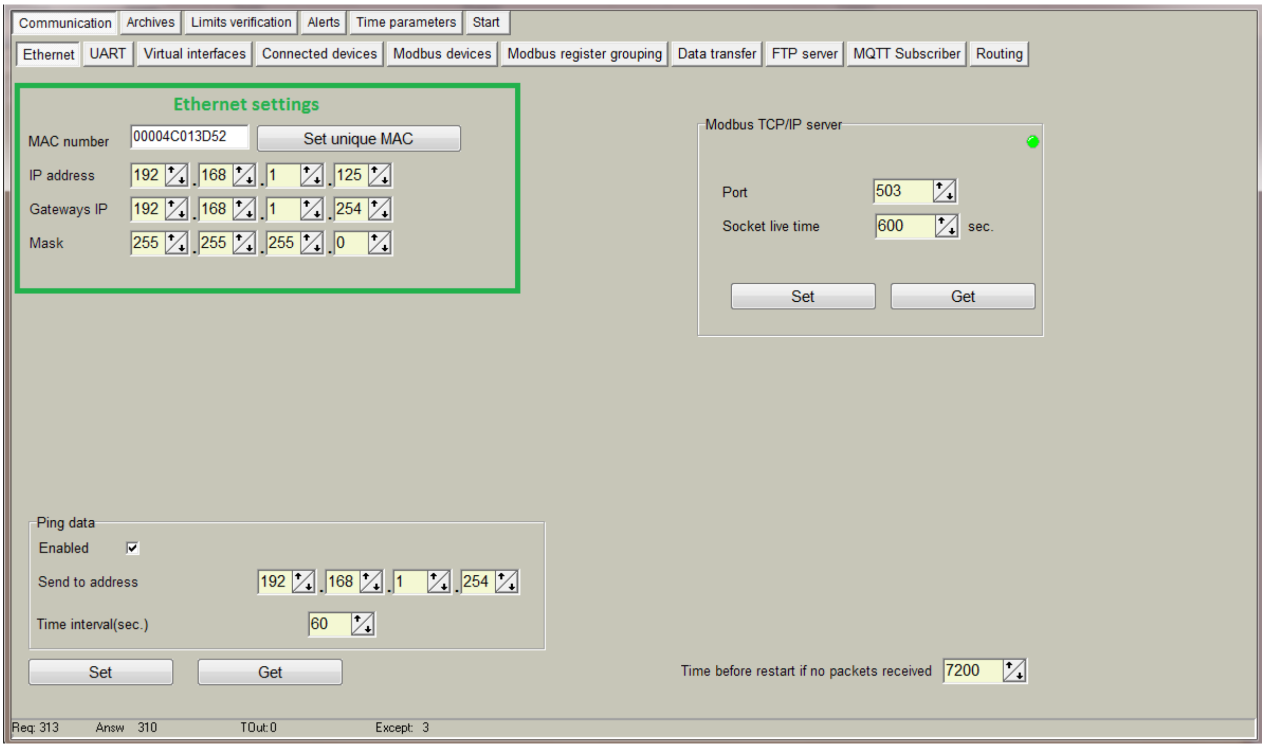
Name	Label and type	Color	Function
100 Mbps	H12, two color	Green	On - 100Mbps bus speed Off - 10Mbps bus speed
TX/RX	H12, two color	Red	Blinks - Data is sending or receiving On - Link is active Off - Link is inactive

Ethernet configuration

Ethernet interface configuration is made in "Communication/Ethernet" tab. Device doesn't support DHCP, so before install you need to set its network settings.

- "MAC number" - device individual MAC address
- "IP address" - device IP address
- "Gateway IP" - gateway IP address

- "Mask" - 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

4G/GPRS configuration

Overview

Device has integrated 4G or 2G modem with standard size SIM card. There are 2 types of modems for 4G - A7600G, 2G – SIM800C.

Modem specification:

- A7600G :
  - LTE-TDD B34/B38/B39/B40/B41
  - LTE-FDD B1/B2/B3/B4/B5/B7/B8/B12/B13/B18/B19/B20/B25/B26/ B28/B6
  - UMTS/HSPA+ B1/B2/B4/B5/B6/B8/B19
  - GPRS/EDGE 850/900/1800/1900MHz
- SIM800C :
  - Quad-Band GSM/GPRS/ 850/900/1800/1900MHz

4G/GPRS interface is used for:

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

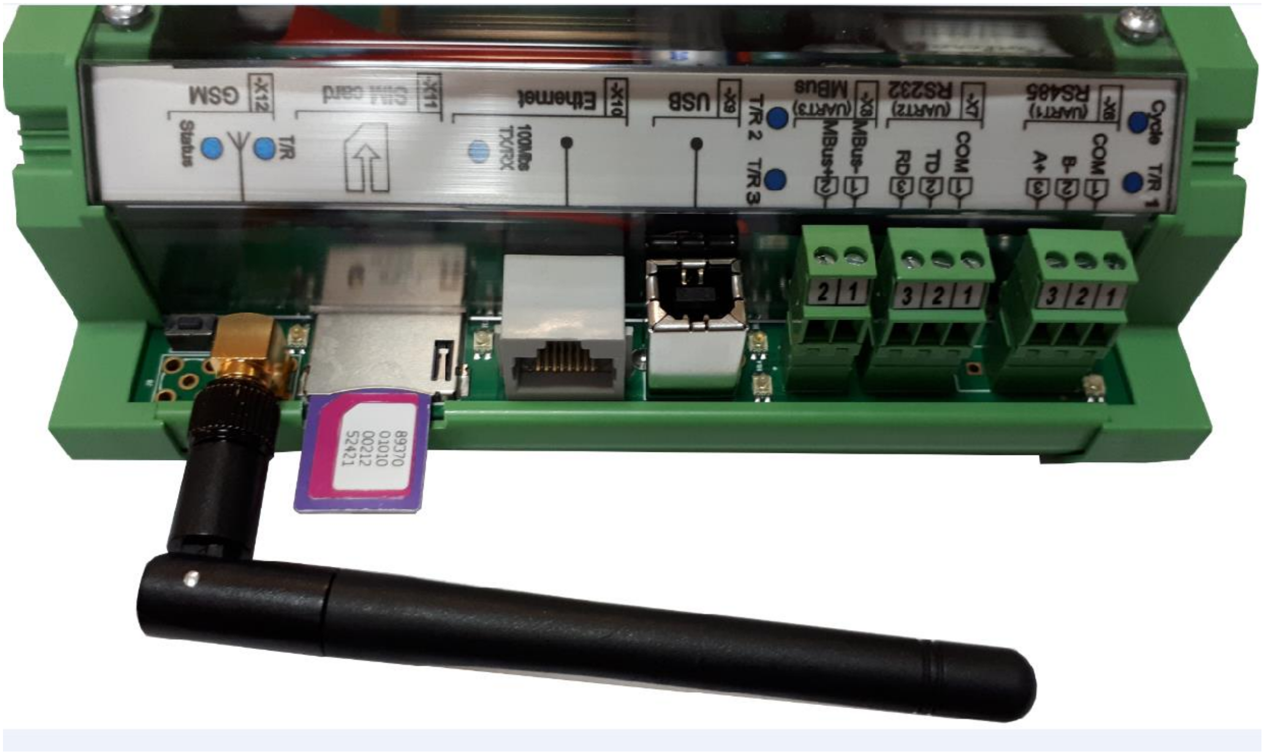
Supported services:

- 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 power and connect GSM SMA male type antenna to antenna connector and insert standard SIM card to SIM card socket.



Led indicators

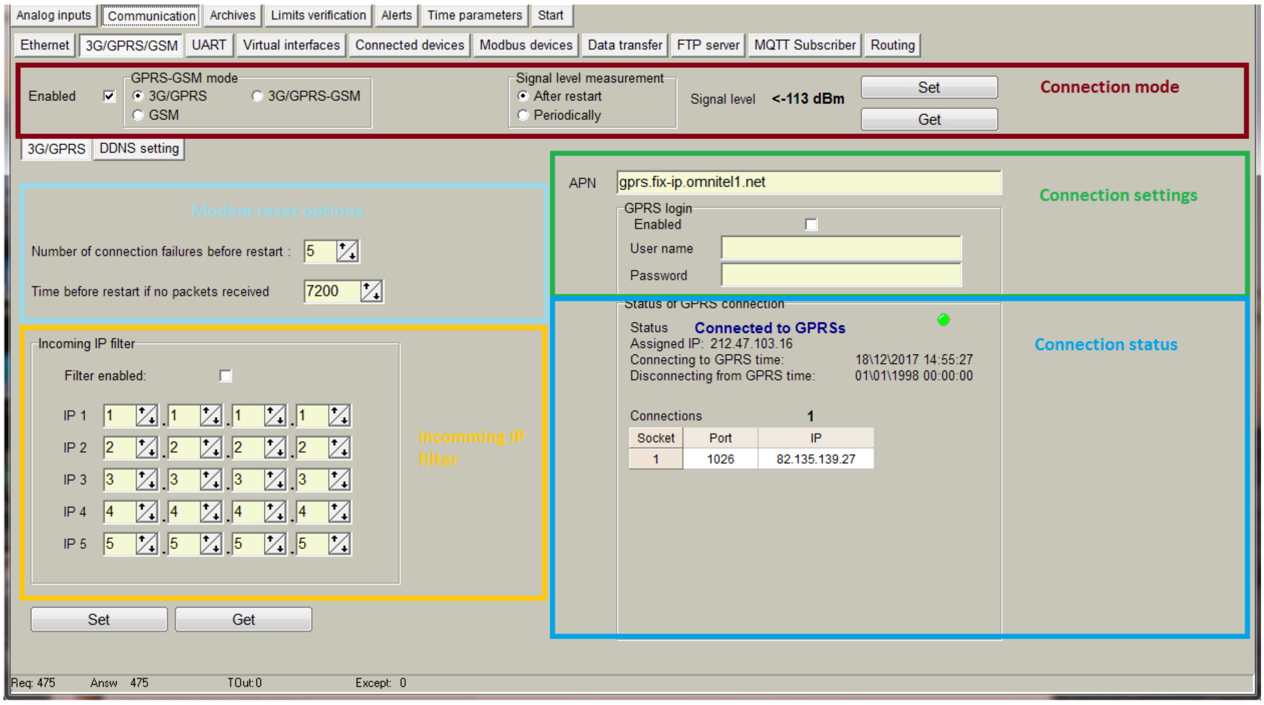
Name	Label and type	Color	Function
Status	H13, two color	Red	Always On - Searching Network/Call Connect 200ms ON, 200ms OFF - Connected to 4G/GPRS network 800ms ON, 800ms OFF - Registered network Off - Power off / Sleep
TX	H2, two color	Red	Blinks - Data is sending to Modem
RX	H2, two color	Green	Blinks - Data is receiving from Modem

4G/GPRS configuration

All configuration is made in "Communication/3G/GPRS/GSM" tab. Before configuring you need to remove PIN code check from your SIM card and have some information like APN address and if present user name and password.

- Connection mode section:

  - "Enabled" check box - enables or disables modem use.
  - GPRS-GSM mode:
    - "3G/GPRS" - connects only in GPRS data mode.
    - "GSM" - connects only in GSM mode (GSM data calls).
    - "3G/GPRS-GSM" - mixed mode, first device try connect to GPRS, if connection failure stays in GSM mode and after timeout tries connect to GPRS again.
  - Signal level measurement"
    - "After reset" - measure signal level once after modem restart.
    - "Periodically" - measure signal level every 2s. Works only in GSM mode.
    - "Signal level" - measured signal level. 51 dBm - best signal, 113dBm - worst signal.
  - "Set" button - write configuration to device.
  - "Get" button - read configuration from device.
- Modem reset options section:
  - "Number of connection failures before restart" - tries connect to GPRS configured times, if failure modem is restarted
  - "Time before restart if no packets received" - if configured time no IP packets is received modem is restarted. Time configure in seconds (600-36000s)
- Connection settings section:
  - "APN" - An Access Point Name (APN) is the name of a gateway between a GSM, GPRS, 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" - enable usage of login preferences
  - "User name" - network user name.
  - "Password" - network password
- Connection status section:
  - "Status" - current status. Available values:
  - "Assigned IP" - assigned network IP address.
  - "Connecting to GPRS time" - time of last connection to GPRS network.
  - "Disconnecting from GPRS time" - time of last disconnection from GPRS network.
  - "Connections" table - list of current TCP/IP connections.
- Incoming IP filter section. If IP filter is enabled device accepts only connections from ip addresses which exists in table.
  - "Filter enabled" - enable/disable incoming IP filter
  - IP1 - IP5 - incoming IP addresses
- "Set" button - write configuration to device.
- "Get" button - read configuration from device.



Enabled ☑

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

Signal level measurement: ● After restart ○ Periodically

Signal level <-113 dBm

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

APN gprs.fix-ip.omnitel1.net

GPRS login
Enabled □
User name
Password

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/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

Limits

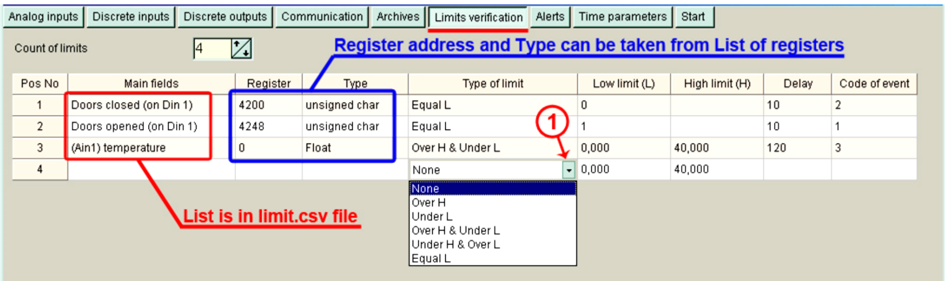
Limits verification

This option is used for customazing of alerts. To setup limits verification, you need:

1. Choose “Type of limit” (and this must to be done first);
2. Enter what register you want to control (see file “List of registers” to find correct register and format) or chose from Main fields list (on how to prepare the list, read down);
3. Enter data type, corresponding to data type of used register;
4. Chose limits;
5. Delay time, if you want to filter accidential ar short time events;
6. And a code of event (value must to be from 0 to 99 and will be used for alerts SMS sending)

If you need to get data not from the first register – you need to count address of register:

- One register takes 2 bytes;
- **Char** type argument takes – 1 register;
- **Int16** type argument takes – 1 regiter;
- **Int32** type argument takes – 2 registers;
- **Float** type argument takes – 2 registers



Count of limits 4

Register address and Type can be taken from List of registers

Pos No	Main fields	Register	Type	Type of limit	Low limit (L)	High limit (H)	Delay	Code of event
1	Doors closed (on Din 1)	4200	unsigned char	Equal L	0		10	2
2	Doors opened (on Din 1)	4248	unsigned char	Equal L	1		10	1
3	(Ain1) temperature	0	Float	Over H & Under L	0,000	40,000	120	3
4				None	0,000	40,000		

List is in limit.csv file

To create and/or edit list of limit’s values use file limits.csv, that is in program’s folder. This file can be edited by MS Excel.

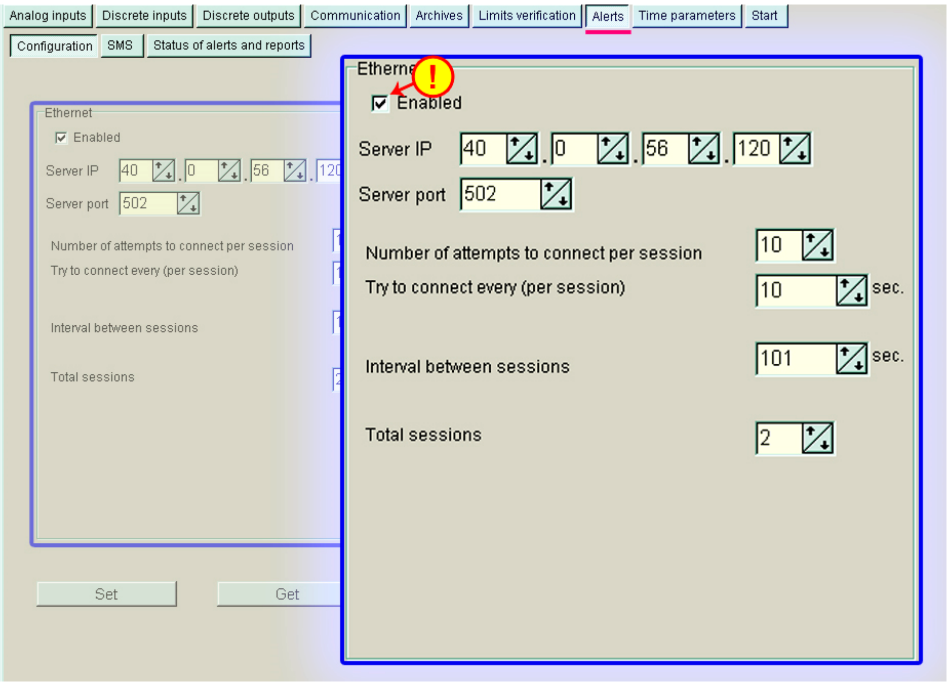
Event code it is not analysed and can be used few tumes per system, so you can set setting for few limits. But form other side, if you will use the same code for a few register, this can mislead you by sending alerts on diggerent events, so use it carefully.

Alert over Ethernet settings [Alerts>Configuration]

Select check box to Enable alert over Ethernet function. Enter destination Modbus TCP/IP servers IP address and Port, change time values if you need and set data.
If you did everything correct, on every deviation, the alert message will be sent.
Information transfer algorithm:

1. Controller makes connection to server, over Ethernet TCP socket.
2. By Modbus TCP/IP protocol, using function nr. 16, deviation report will be sent to server (in register list, information is stored in registers, starting from address nr. 64000). Sent report contains:
    - Unique controller number;
    - Controller IP address;
    - Deviation identifier
3. When answer into function nr. 16 is received, controller disconnects from TCP socket.
4. When server get’s deviation information, it initializes connection with controller and over Modbus TCP/IP read all details about deviation
5. If connection with server was not established over the set (connection to server time) number of tries, controller will try to connect after some time.

If opening of socket is succeeded, but server do not answer into Modbus function 16, controller repeat once and on failure, closes TCP socket. After set time (wait between trial times) amount of time controller will try to send alert again.



Ethernet
☑ Enabled
Server IP 40 . 0 . 56 . 120
Server port 502
Number of attempts to connect per session 10
Try to connect every (per session) 10 sec.
Interval between sessions 101 sec.
Total sessions 2

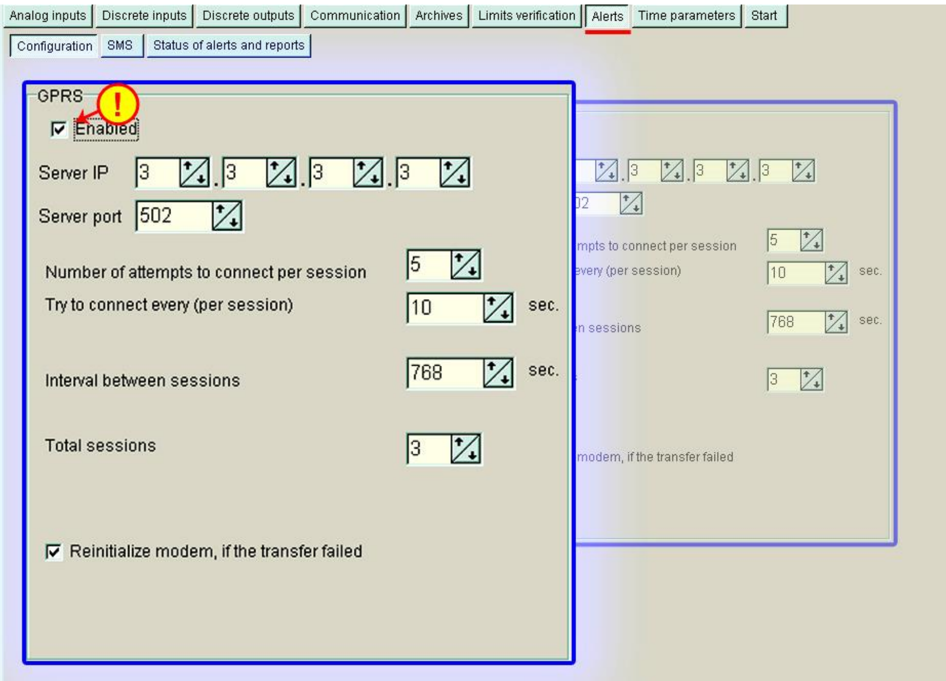
Alert over GPRS [Alerts>Configuration].

On input port change, controller generates event. Through GPRS, controller can send information to remote server (we call this Alert). To use Alerts over GPRS:

1. Check box Enable;
2. Enter remote server IP address (PC that you use to receive alerts IP address);
3. Use default, or change Modbus TCP/IP port address;
4. Setup connection intervals and other settings.

Information transfer algorithm:

1. Controller makes connection to server, over Ethernet TCP socket.
2. By Modbus TCP/IP protocol, using function nr. 16, deviation report will be sent to server. Report contains:
   - Unique controller number;
   - Controller IP address;
   - Deviation identifier
3. When answer into function nr. 16 is received, controller disconnect TCP socket;
4. When server get’s deviation information, he initialize connection with controller and over Modbus TCP/IP read all details about deviation;
5. If connection with server was not established over the set (connection to server time) number of tries, controller will try to connect after some time;
6. If opening of socket is succeeded, but server do not answer into Modbus function 16, controller repeat once and on failed, close TCP socket. After set time (wait between trail times) amount of time controller will try to send alert again.



GPRS

☑ Enabled

Server IP 3 . 3 . 3 . 3

Server port 502

Number of attempts to connect per session 5

Try to connect every (per session) 10 sec.

Interval between sessions 768 sec.

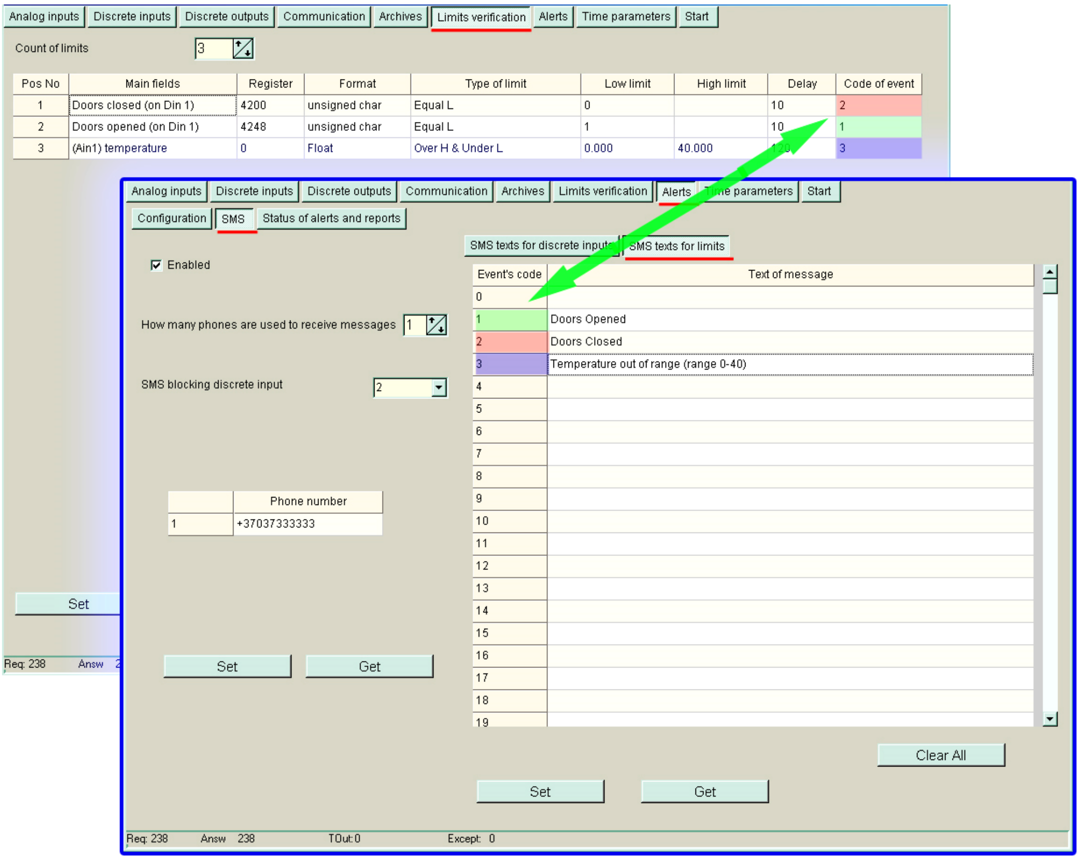
Total sessions 3

☑ Reinitialize modem, if the transfer failed

Sending alert over SMS

Controller has possibility to send short SMS text messages. Mark check box for enabled. Choose how many phone number will get SMS messages (max. 5 phone numbers). Fill table with phone numbers. There is possibility to block SMS sending during controller testing procedures. From discrete input blocking list choose, which discrete input blocks SMS sending (if “none” is selected – SMS blocking is disabled). Closed discrete input disables SMS sending.

SMS Alert on Limit verification



Count of limits: 3

Pos No	Main fields	Register	Format	Type of limit	Low limit	High limit	Delay	Code of event
1	Doors closed (on Din 1)	4200	unsigned char	Equal L	0		10	2
2	Doors opened (on Din 1)	4248	unsigned char	Equal L	1		10	1
3	(Ain1) temperature	0	Float	Over H & Under L	0.000	40.000	120	3

☑ Enabled

How many phones are used to receive messages: 1

SMS blocking discrete input: 2

	Phone number
1	+37037333333

Event's code	Text of message
0	
1	Doors Opened
2	Doors Closed
3	Temperature out of range (range 0-40)
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	

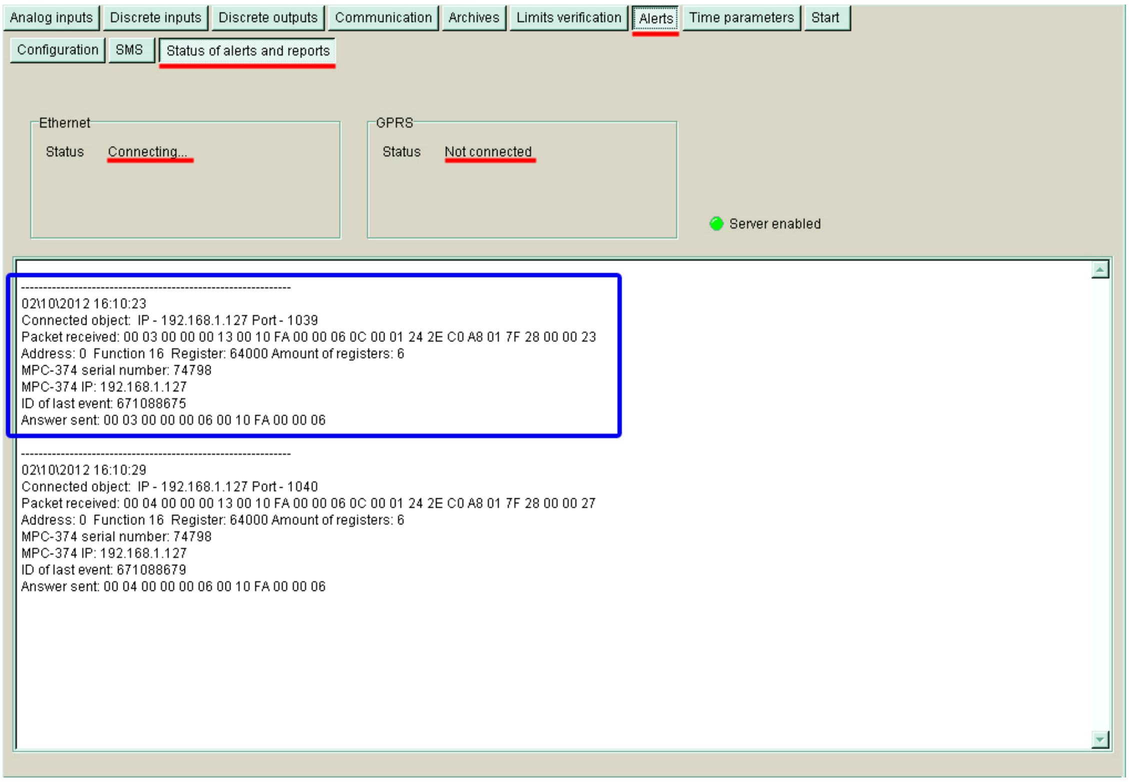
Req: 238 Answ 238 TOut: 0 Except: 0

SMS alerts for limits is used for sending SMS alerts, when alert conditions is set in „Limit verification“ screen. In „Limit verification“ set Codes of events are used as ID (Event‘s code) for alert messages identification.

Status of alerts and reports

In section Status of alerts and reports you can see:

- Current status of alerts over Ethernet;
- Current status of alerts over GPRS;
- Status of Modbus server;
- Modbus server information messages.



Ethernet
Status Connecting...

GPRS
Status Not connected

```
---------------------------------------------------------------
02\10\2012 16:10:23
Connected object:  IP - 192.168.1.127 Port - 1039
Packet received: 00 03 00 00 00 13 00 10 FA 00 00 06 0C 00 01 24 2E C0 A8 01 7F 28 00 00 23
Address: 0  Function 16  Register: 64000 Amount of registers: 6
MPC-374 serial number: 74798
MPC-374 IP: 192.168.1.127
ID of last event: 671088675
Answer sent: 00 03 00 00 00 06 00 10 FA 00 00 06

---------------------------------------------------------------
02\10\2012 16:10:29
Connected object:  IP - 192.168.1.127 Port - 1040
Packet received: 00 04 00 00 00 13 00 10 FA 00 00 06 0C 00 01 24 2E C0 A8 01 7F 28 00 00 27
Address: 0  Function 16  Register: 64000 Amount of registers: 6
MPC-374 serial number: 74798
MPC-374 IP: 192.168.1.127
ID of last event: 671088679
Answer sent: 00 04 00 00 00 06 00 10 FA 00 00 06
```

Serial ports

Overview

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

Port number	Available options	Description
UART 1	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 250 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 - 19200	Even, Odd, Mark, Space, None	5,6,7,8	1,2

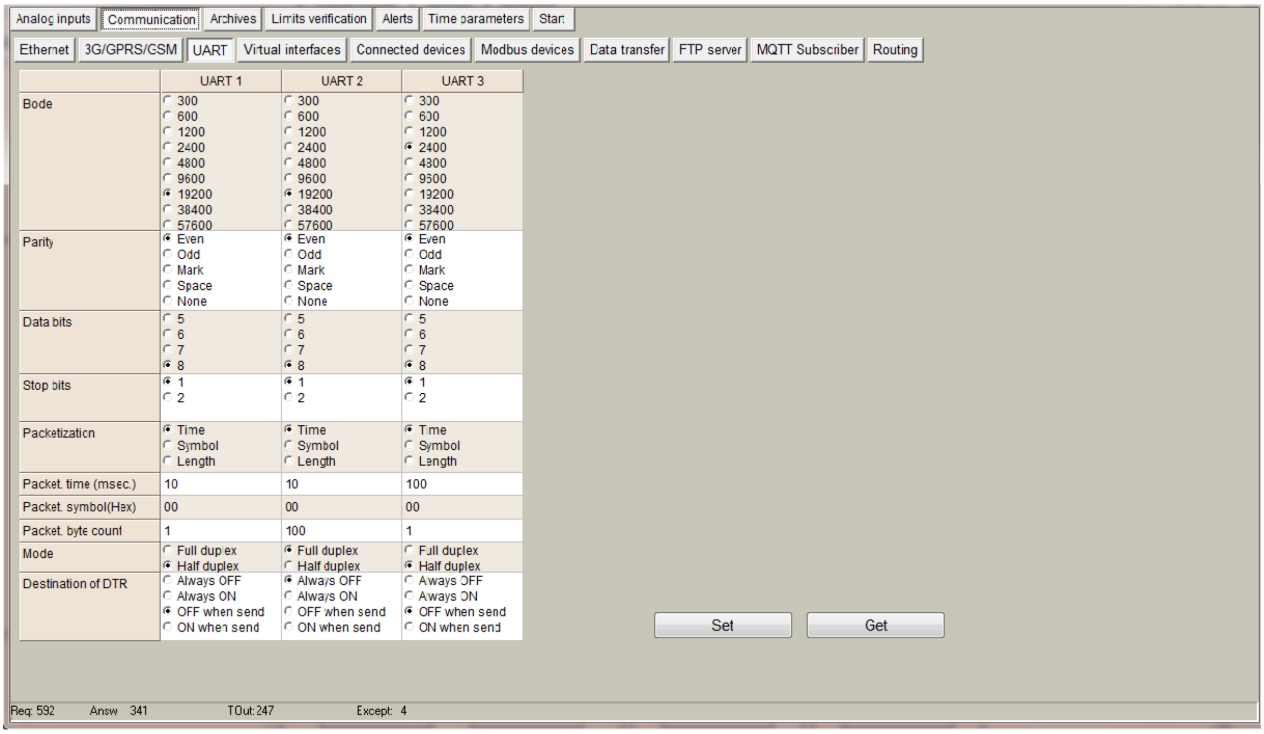
Led indicators

Name	Label and type	Color	Function
RS TX	H1, two color	Green	Blinks - Data is receiving on UART1
RS RX	H1, two color	Red	Blinks - Data is sending on UART1
Mbus TX	H5, two color	Green	Blinks - Data is receiving on UART2
Mbus RX	H5, two color	Red	Blinks – Data is sending on UART2
TX/RX	H8, two color	Green	Ethernet data incoming/outcoming
100Mbs	H8, two color	Red	Ethernet port working
Overload	H4, two color	Red	Always On - MBUS line is shorted, overloaded
Line off	H4, two color	Green	Always On - MBUS line is not connected

UART settings

UART's interface configuration is made in "Communication/UART" tab.

- "Bode" - 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 serial interface principle:
  - "Time" - captures the accepted package if timeout after last received byte is bigger than configured "Packet time (msec)". Time in milliseconds.
  - "Symbol" - captures the accepted package if last received byte equals configured "Packet symbol (Hex)".
  - "Length" - captures the accepted package if received byte count equals configured "Packet byte count".
- "Packet time (msec)" - packetization timeout in milliseconds. Used if is selected time packetization.
- "Packet symbol (Hex)" - packetization end symbol. Used if is selected symbol packetization.
- "Packet byte count" - packetization received packet count. Used if is selected length packetization.
- "Mode" - types of duplex communication system:
  - "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; the communication is one direction at a time.
- "Destination of DTR" - purpose of extra UART signal DTR. **For RS485 always set this signal to "OFF when sending" !**
  - "Always OFF" - DTR signal always in OFF state.
  - "Always ON" - DTR signal always in ON state.
  - "OFF when send" - DTR signal set to OFF when data is sending, other time DTR is in On state
  - "ON when send" - DTR signal set to ON when data is sending, other time DTR is in OFF state
- "Set" button - write configuration to device.
- "Get" button - read configuration from 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 ○ 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)	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

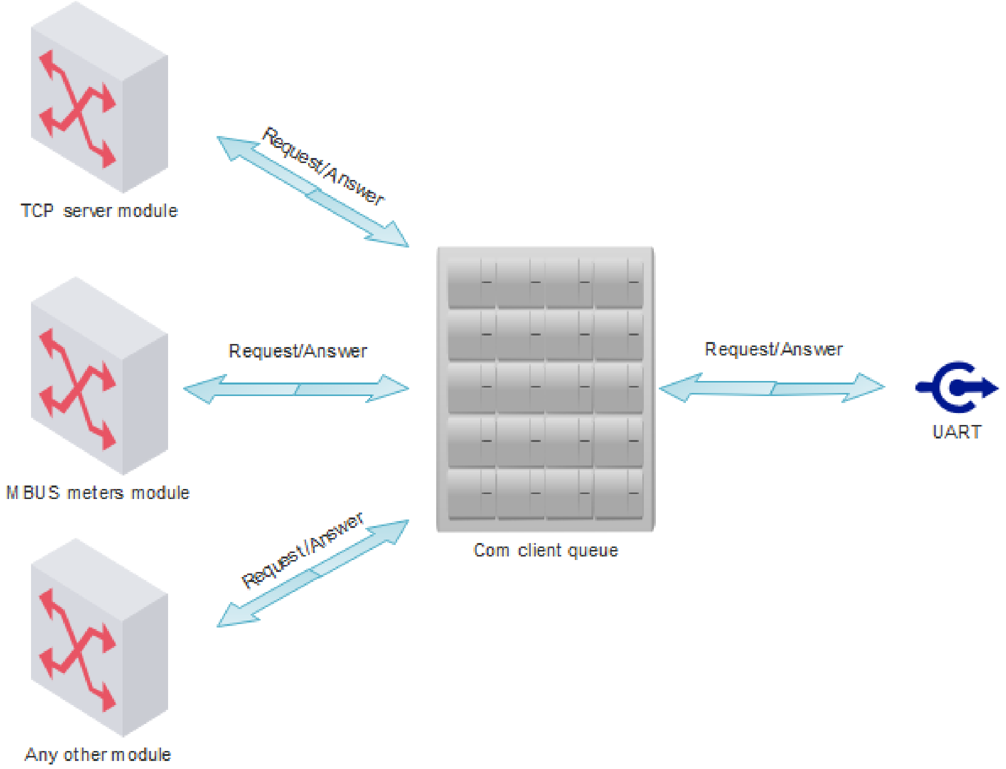
Req: 592 Answ 341 TOut: 247 Except: 4

Virtual interfaces

Virtual COM clients

Overview

COM client is virtual interface between program modules and physical UARTs.COM client allows more than one module access UART at the same time, send data and receive answer.



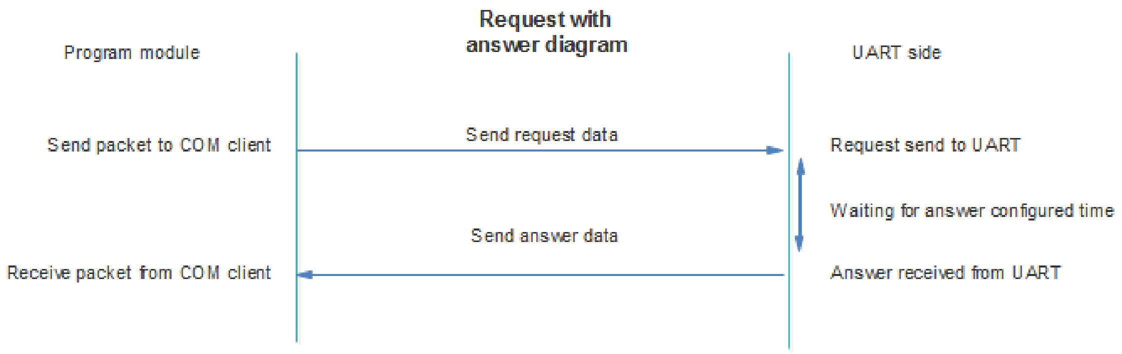
COM client is used with this modules:

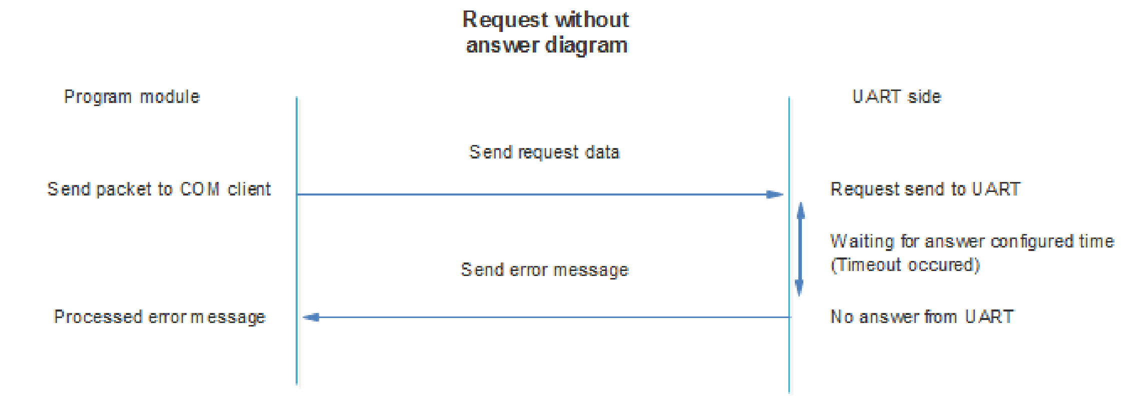
- TCP server

- MBUS meters read module
- Heat meters read module
- Electricity meters read module
- Other

COM client put request to the queue and send it to UART when it is free, after request is send COM client waits for answer configured time and returns it to the source module. If answer didn't received COM client informs source module about error (No data received).

If answer is received after COM client timeout, this data is lost.

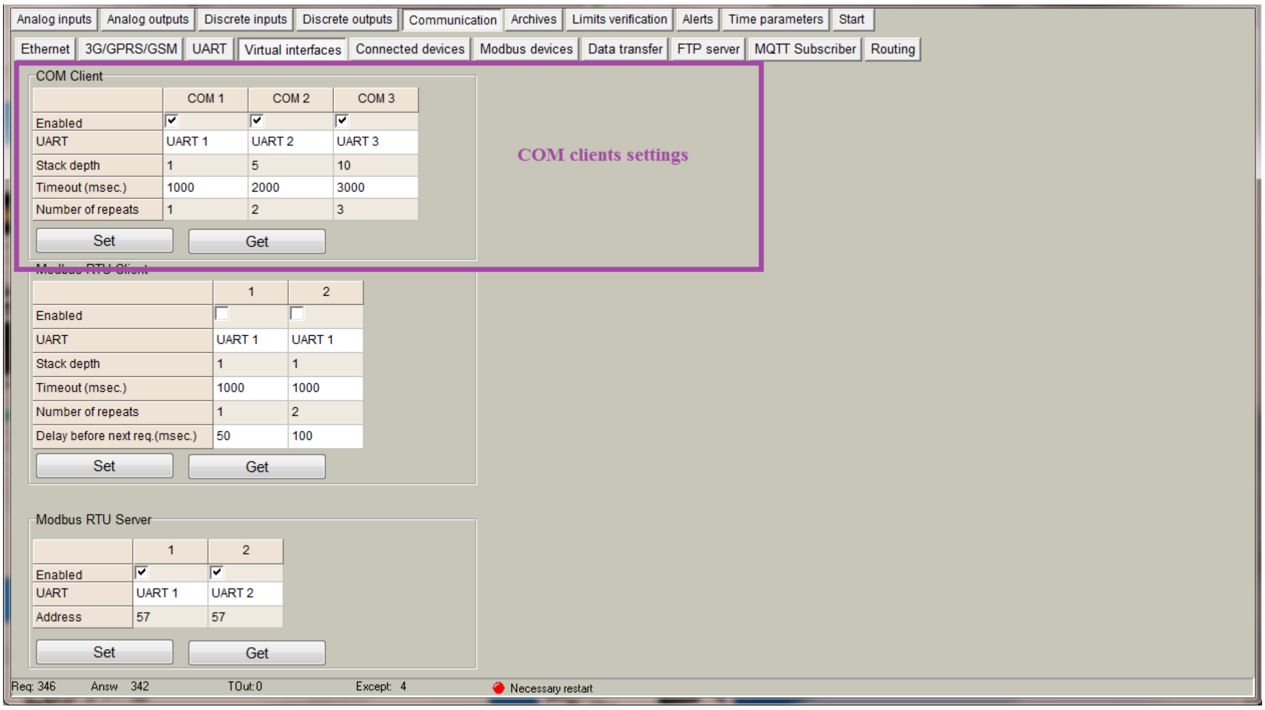

Request with
answer diagram
Program module
UART side
Send packet to COM client
Send request data
Request send to UART
Waiting for answer configured time
Send answer data
Receive packet from COM client
Answer received from UART



Virtual COM client's configuration

Virtual COM clients configuration is made in "Communication/Virtual interfaces" tab. Device can have up to 3 COM clients associated with different physical UARTs.

- "Enabled" check box - enables/disables appropriate COM client
- "UART" - physical UART associated with appropriate COM client (UART1, UART2, UART3). **The same UART can't be used in other virtual interface (Other COM client, Modbus RTU client or Modbus RTU server). In picture below we see bad configuration, because UART1 and UART2 is used in COM clients and Modbus RTU servers. If COM clients are used Modbus RTU servers have to be disabled.**
- "Stack depth" - COM client queue length (1-10). This parameter decides how many packets can be processed at the same time.
- "Timeout (msec)" - answer from UART wait time (Time in milliseconds, 1-30000).
- "Number of repeats" - decides how many times send request if answer not received
- "Set" button - write configuration to device
- "Get" button - read configuration from 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

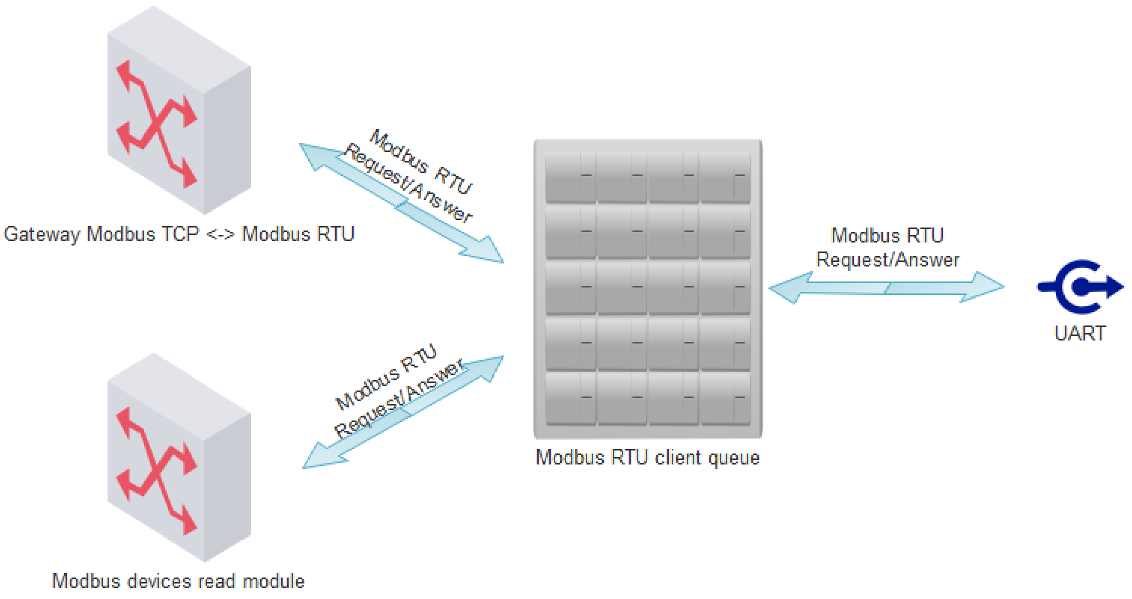
	1	2
Enabled	☑	☑
UART	UART 1	UART 2
Address	57	57

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

Modbus RTU clients

Overview

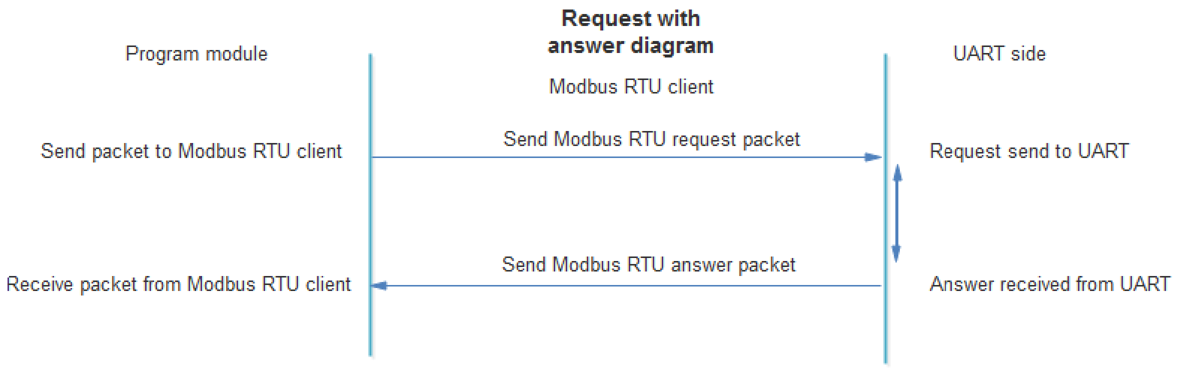
Modbus RTU client is virtual interface used to associate Modbus RTU devices connected to physical UART with internal Modbus modules. It allows more than one module access to UART at the same time to send Modbus requests and receive answers.

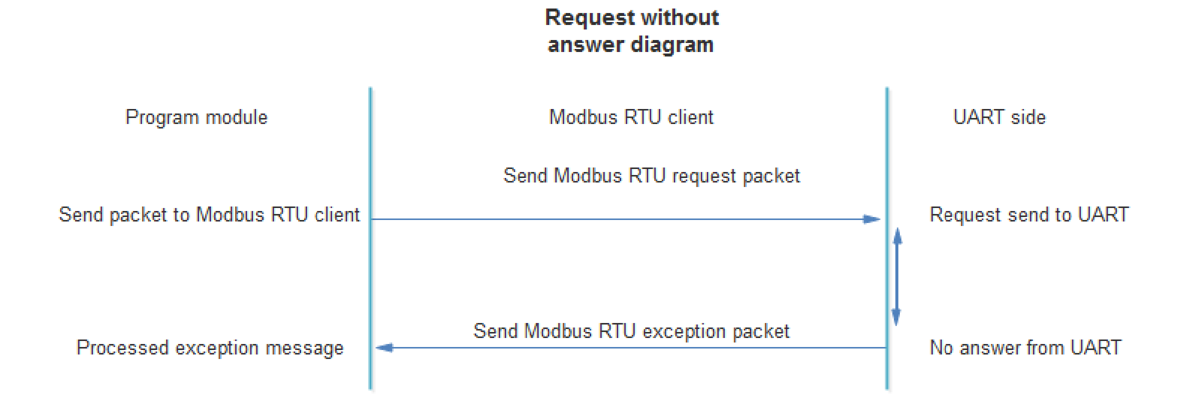

Gateway Modbus TCP <-> Modbus RTU
Modbus RTU Request/Answer
Modbus RTU Request/Answer
Modbus devices read module
Modbus RTU client queue
Modbus RTU Request/Answer
UART

Modbus RTU client is used with these modules:

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

Modbus RTU client put request to the queue and send it to UART when it is free. After request is send Modbus RTU client waits for answer configured time and returns it to the source module. If answer didn't received Modbus RTU client returns exception to 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
Send Modbus RTU answer packet
Receive packet from Modbus RTU client
Answer received from UART

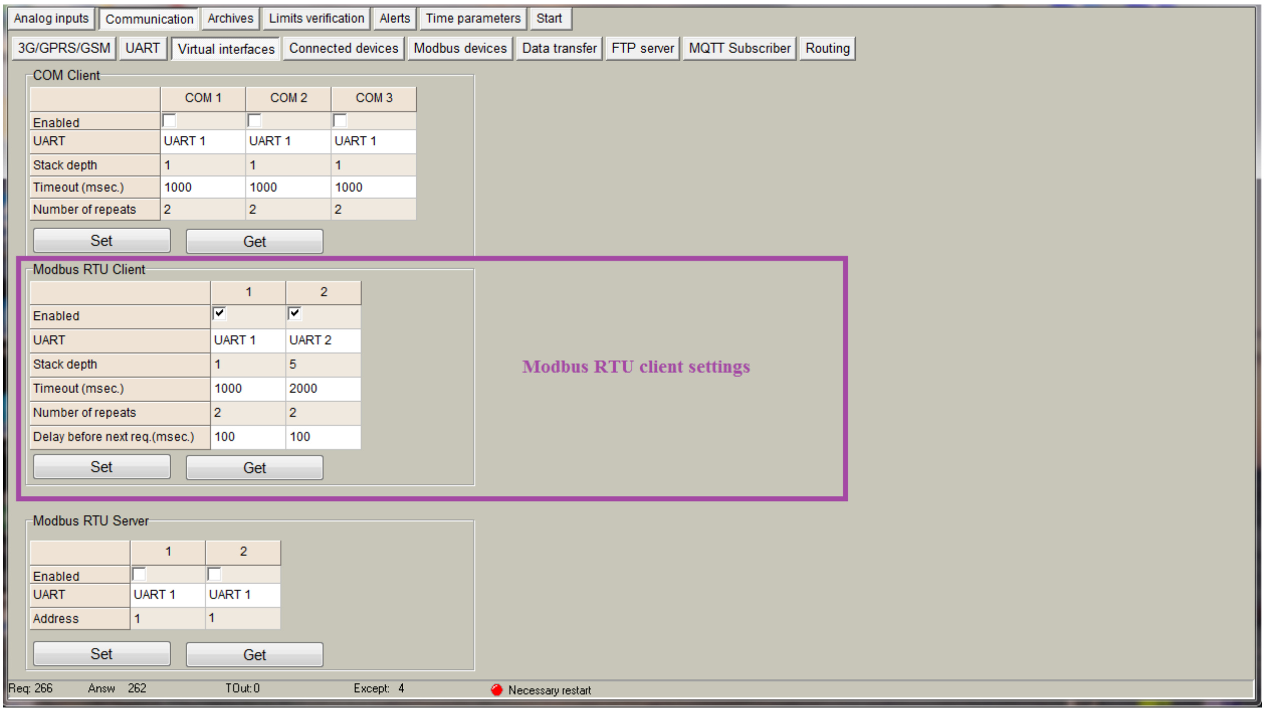

Request without answer diagram
Program module
Modbus RTU client
UART side
Send Modbus RTU request packet
Send packet to Modbus RTU client
Request send to UART
Send Modbus RTU exception packet
Processed exception message
No answer from UART

⚠ **If answer is received after Modbus RTU client timeout, this data is lost.**

Modbus RTU client’s configuration

Modbus RTU clients configuration is made in "Communication/Virtual interfaces" tab. Device can have up to 2 Modbus RTU clients associated with different physical UARTs.

- "Enabled" check box - enables/disables appropriate Modbus RTU client
- "UART" - physical UART associated with appropriate Modbus RTU client (UART1, UART2, UART3). **The same UART can't be used in other virtual interface (Other COM client, Modbus RTU client or Modbus RTU server).**
- "Stack depth" - Modbus RTU client queue length (1-10). This parameter decides how many packets can be processed at the same time.
- "Timeout (msec)" - answer from UART wait time (Time in milliseconds, 1-30000).
- "Number of repeats" - decides how many times send request if answer not received
- "Delay before next req (msec)" - time between requests (Time in milliseconds, 1-10000). Next request will be send only after configured timeout
- "Set" button - write configuration to device
- "Get" button - read configuration from 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 1
Address	1	1

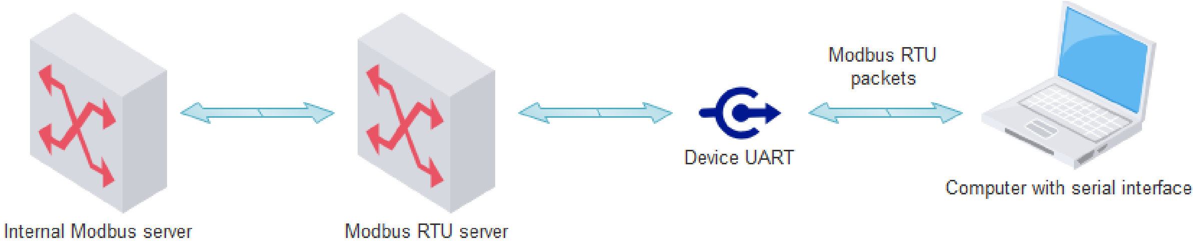
Req: 266 Answ 262 TOut:0 Except: 4 ● Necessary restart

Modbus RTU servers

Overview

Modbus RTU server is virtual interface between internal Modbus server and physical UARTs. It lets user to read device internal Modbus register area using serial connection.
Modbus RTU server can be used:

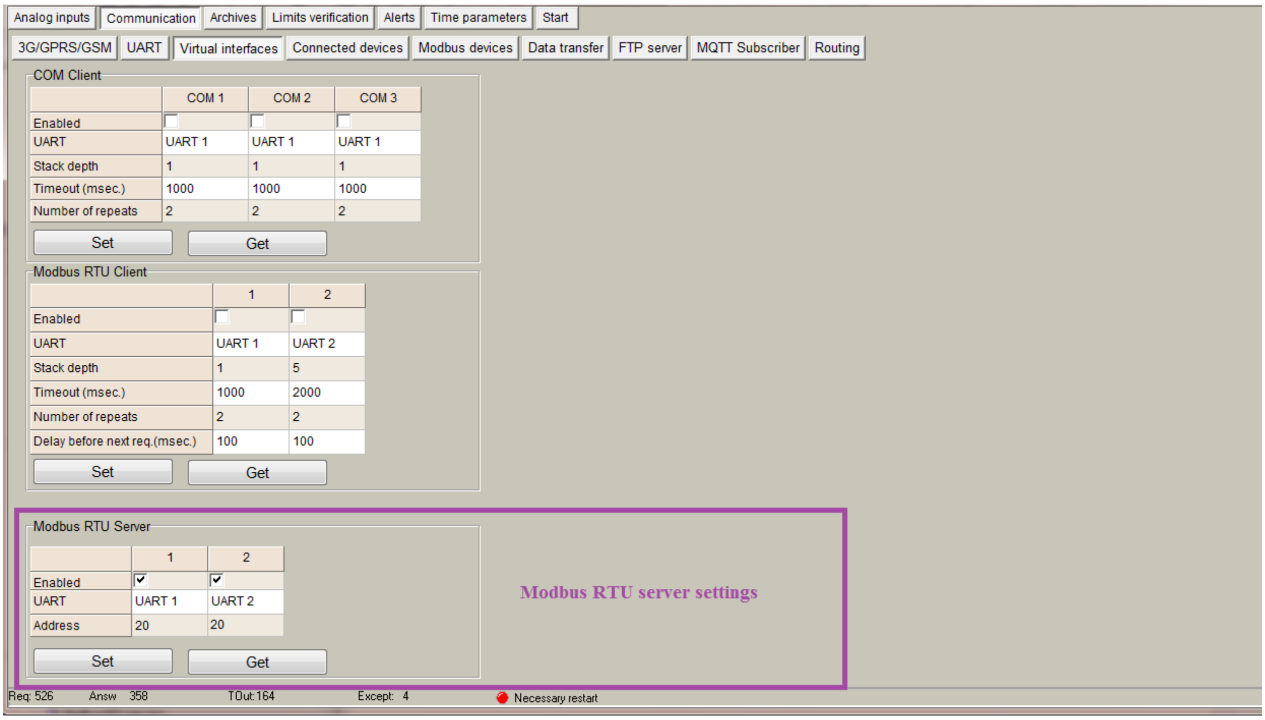
- Read device internal Modbus registers from any Modbus RTU master
- Configure device using configuration tool software



Modbus RTU server configuration

Modbus RTU server configuration is made in "Communication/Virtual interfaces" tab. Device can have up to 2 Modbus RTU servers associated with different physical UARTs.

- "Enabled" check box - enables/disables appropriate Modbus RTU server
- "UART" - physical UART associated with appropriate Modbus RTU server (UART1, UART2, UART3). **The same UART can't be used in other virtual interface (Other COM client, Modbus RTU client or Modbus RTU server).**
- "Address" - accepted Modbus ID. Device will answer to configured Modbus ID (In picture below Modbus ID=20 and always will answer to Modbus ID=254).
- "Set" button - write configuration to device
- "Get" button - read configuration from 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

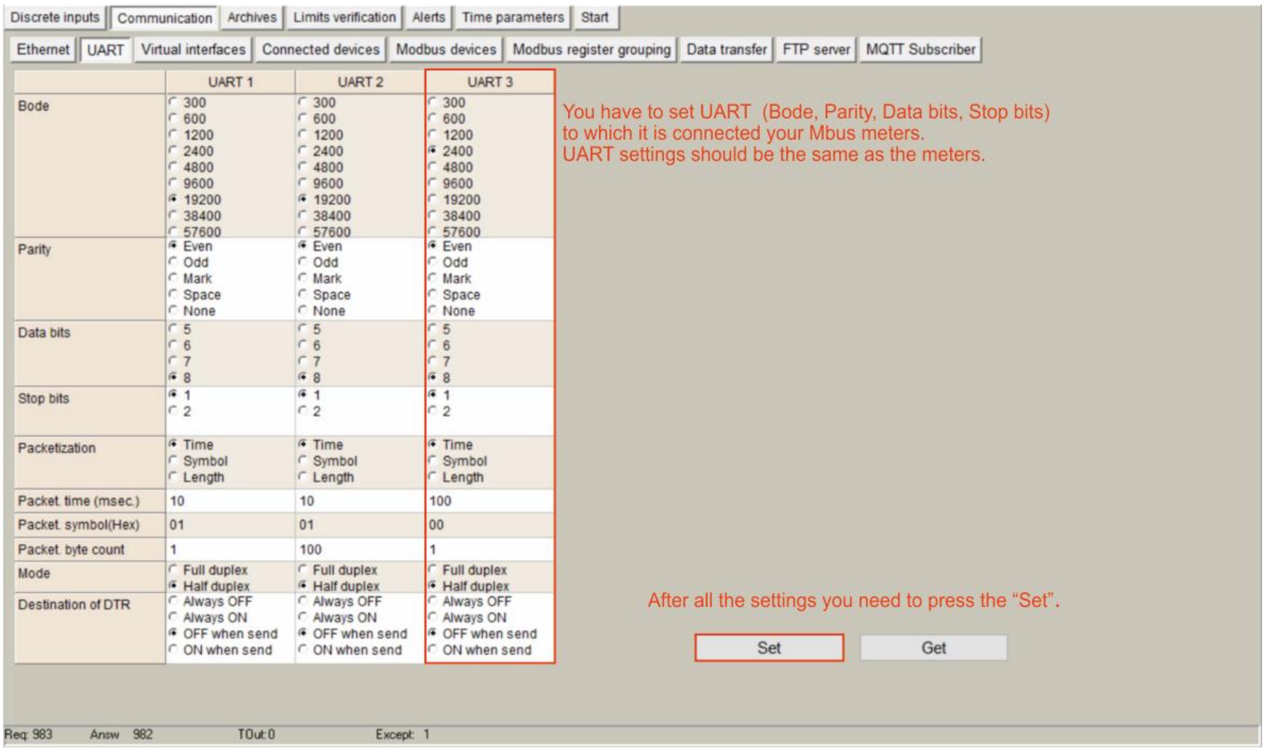
Req: 526 Answ 358 TOut:164 Except: 4 ● Necessary restart

Meters

M-bus meters

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

You have to specify the UART settings to which the M-Bus line is connected. Bode rate, Parity, Data bits...



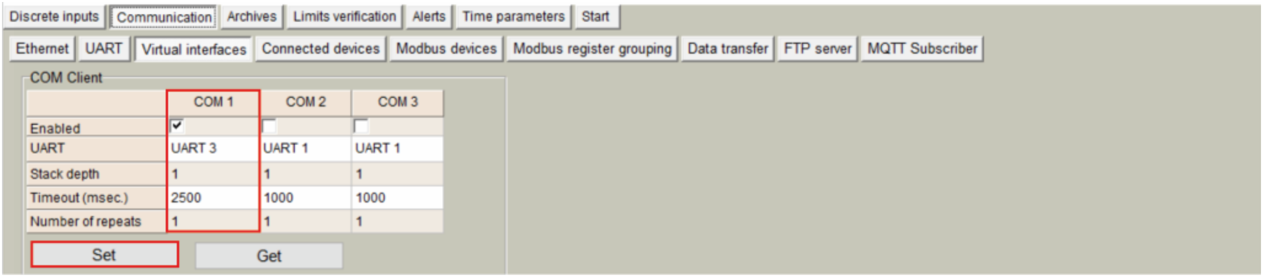
	UART 1	UART 2	UART 3
Bode	○ 300 ○ 600 ○ 1200 ○ 2400 ○ 4800 ○ 9600 ● 19200 ○ 38400 ○ 57600	○ 300 ○ 600 ○ 1200 ○ 2400 ○ 4800 ○ 9600 ● 19200 ○ 38400 ○ 57600	○ 300 ○ 600 ○ 1200 ● 2400 ○ 4800 ○ 9600 ○ 19200 ○ 38400 ○ 57600
Parity	● Even ○ Odd ○ Mark ○ Space ○ None	● Even ○ Odd ○ Mark ○ Space ○ None	● Even ○ 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

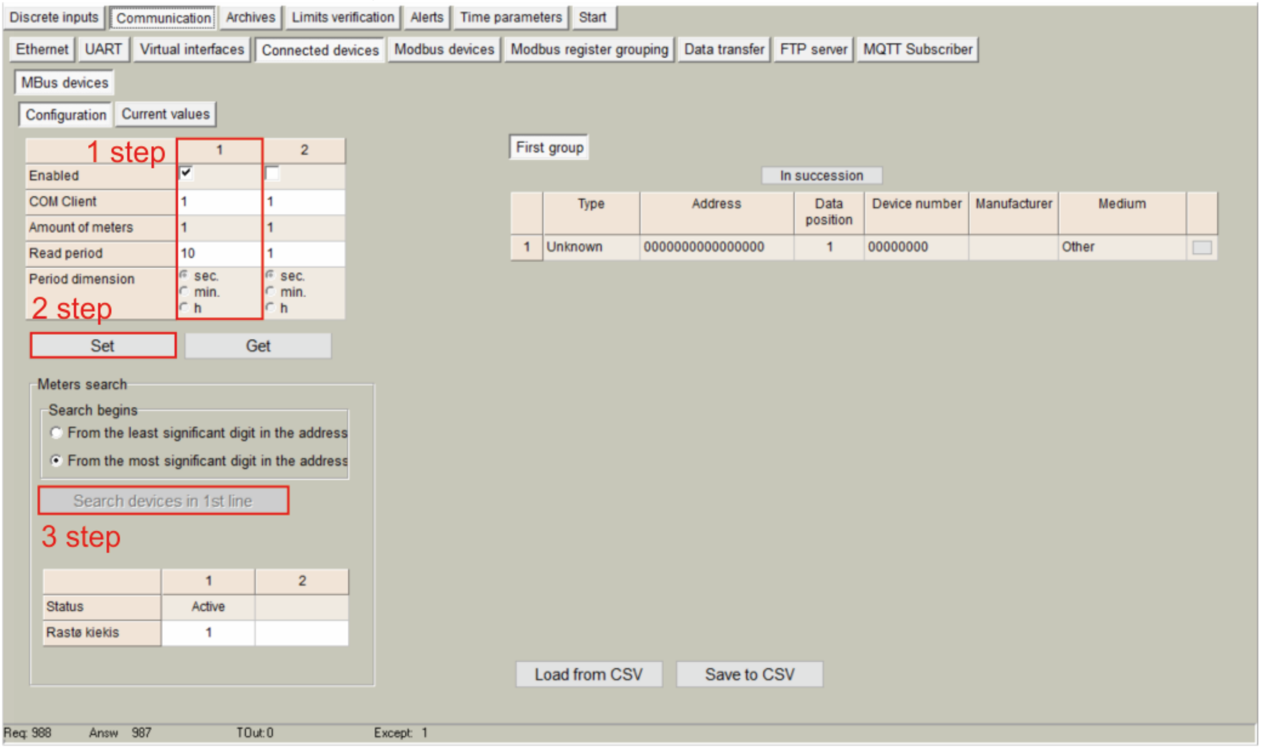
You have to enable virtual comport (COM) and select UART to which the M-Bus line is connected.



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

You have to enable M-Bus line, enter the COM client number and transfer (SET) these settings to controller. After that you have to perform M-Bus meters search. If you have your own M-Bus meters address list in csv file, you can upload it without M-Bus meters searching. NOTE: If you have connected a lot meters, their 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 2 step	● 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

Search devices in 1st line

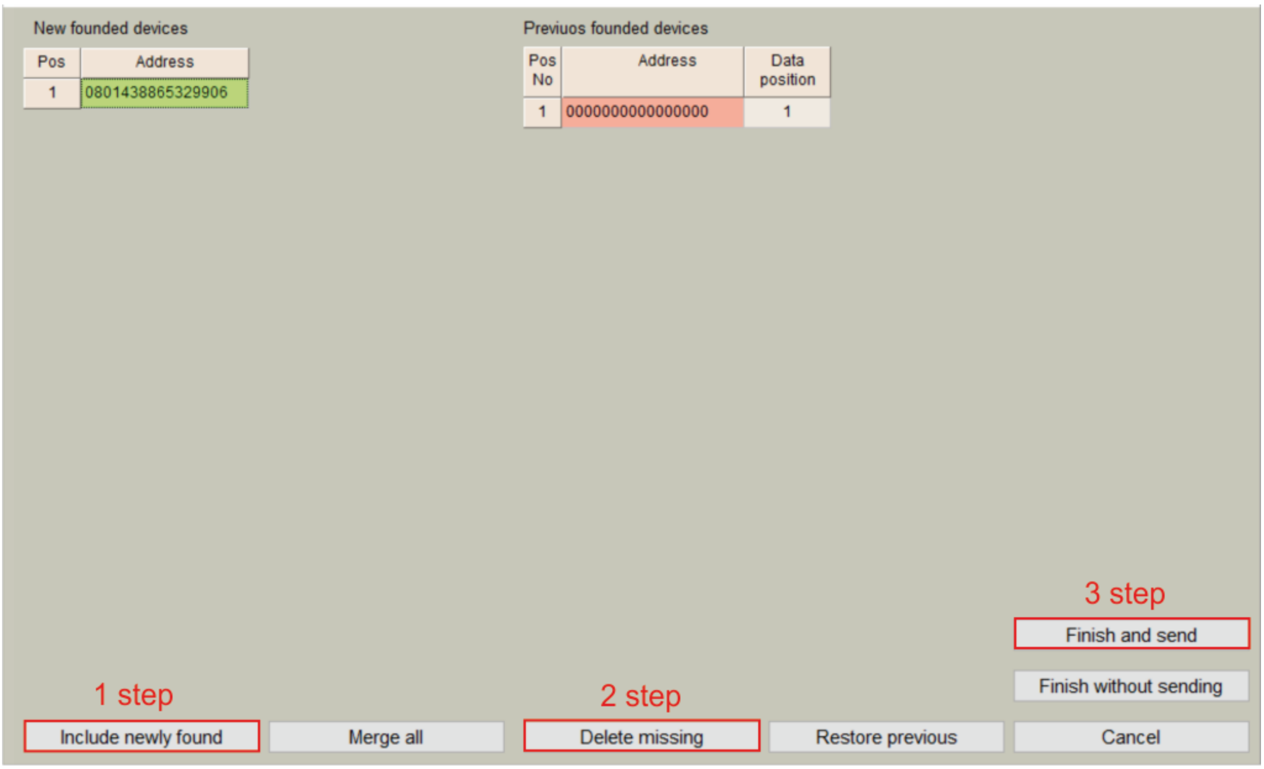
3 step

	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

When the search is end you will see this window. There you will see new found M-Bus devices. You have to include newly found devices and send it to controller. How to do it, you can see at the picture below.



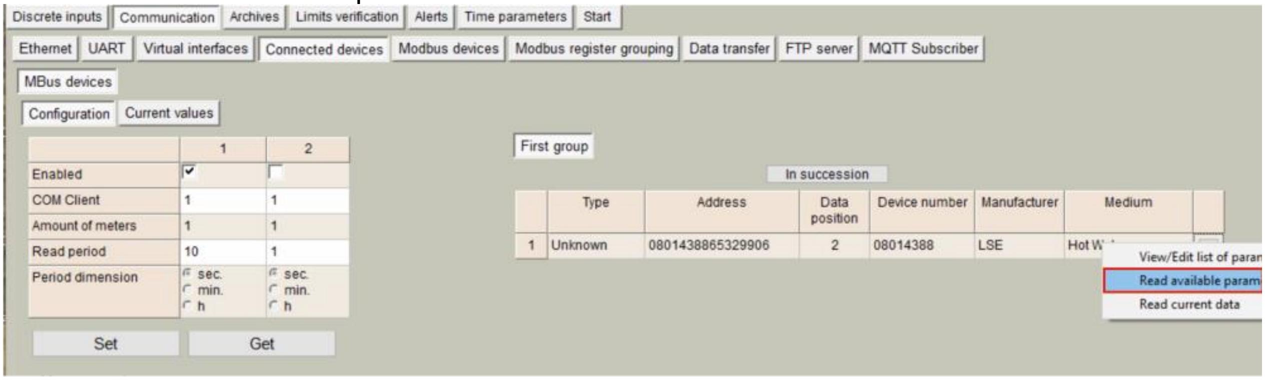
New founded devices

Pos	Address
1	0801438865329906

Previuos founded devices

Pos No	Address	Data position
1	0000000000000000	1

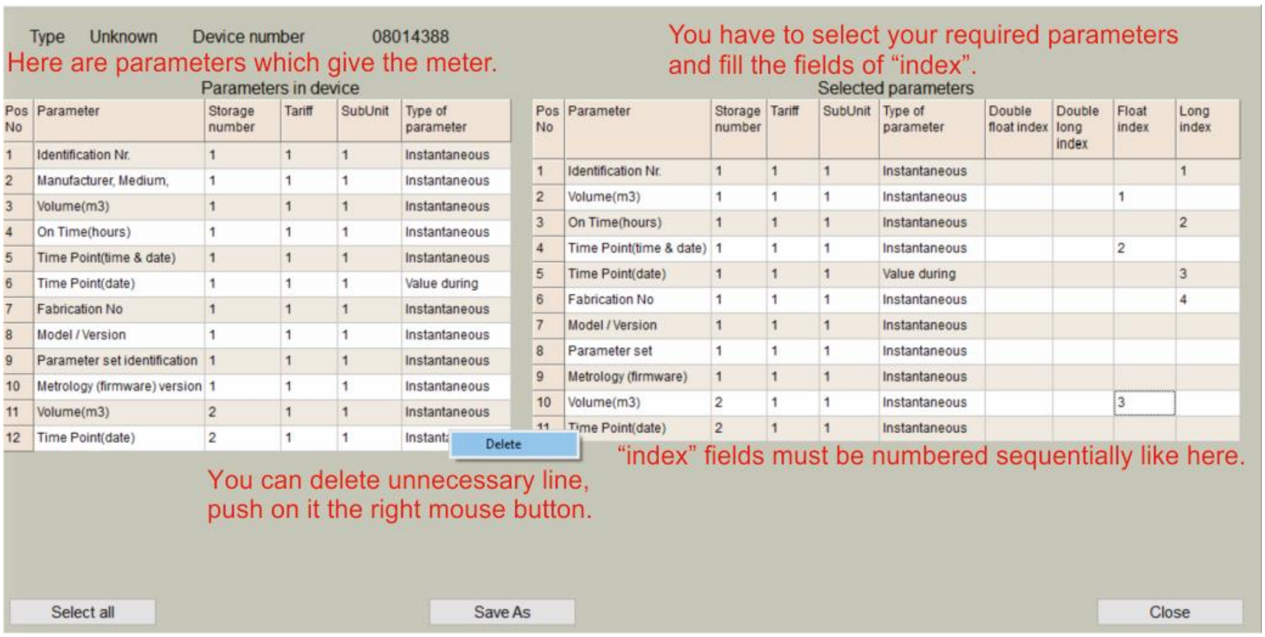
After you added M-Bus meter list to controller, you have to create a description of the meter, that the controller knows which parameters to read from the M-Bus meters.



	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 given by the meter. The right side of the table you have to choose the required parameters. If the parameter is need to read you have to fill "Index" field. Index should be written in column which number format you need. "Index" in column must be numbered sequentially. Unnecessary parameters can be deleted from list with mouse right button.



Type Unknown Device number 08014388

Here are parameters which give the meter.

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

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

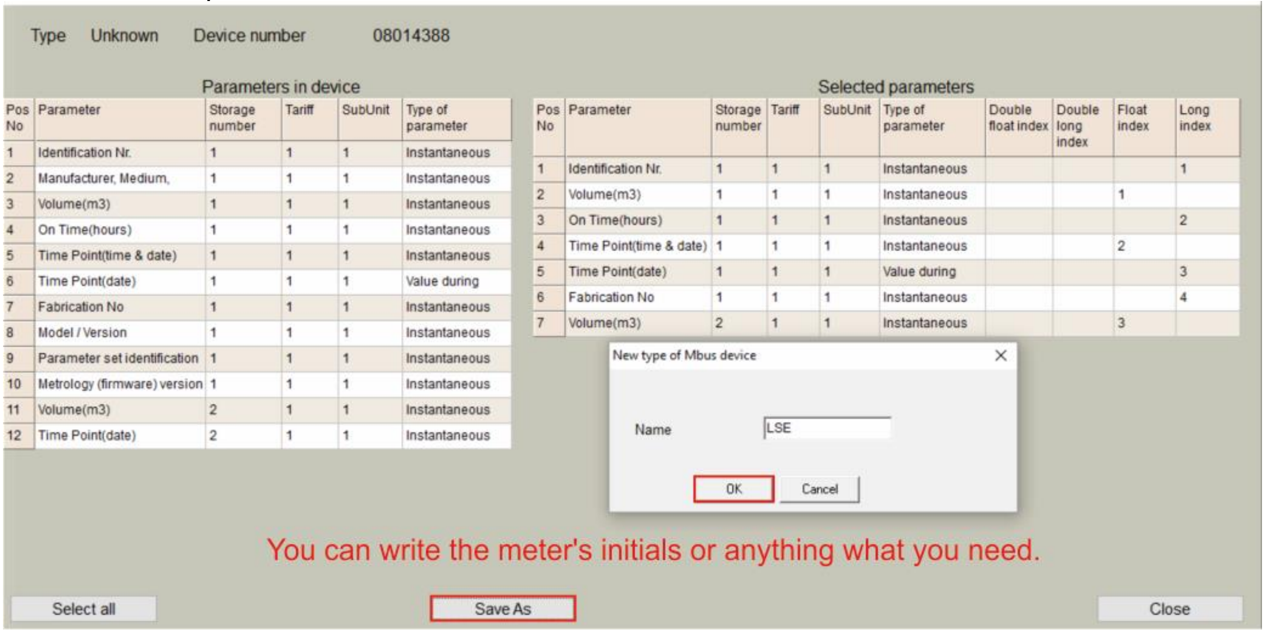
Selected parameters

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				

“index” fields must be numbered sequentially like here.

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

When the required parameters are selected, it has to be saved. We recommend use the meter name for 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

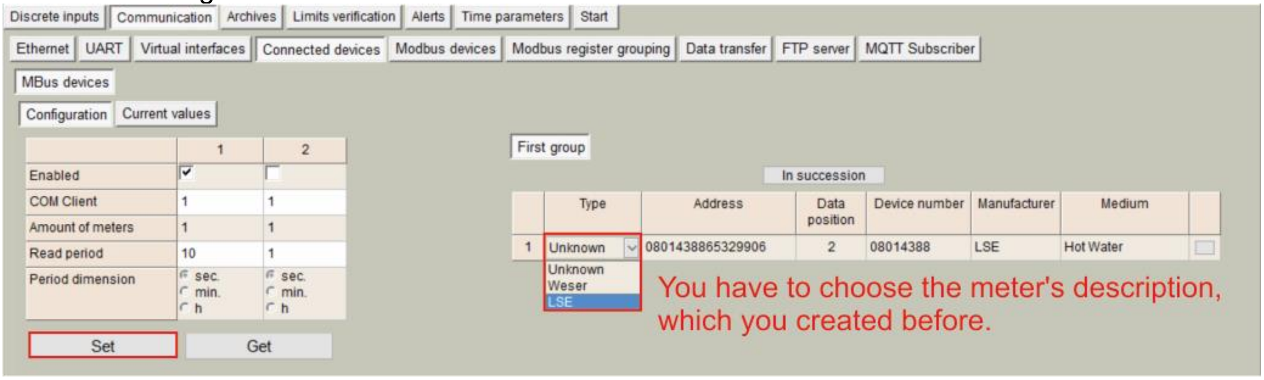
Pos No	Parameter	Storage number	Tariff	SubUnit	Type of parameter	Double float index	Double long index	Float index	Long index
1	Identification Nr.	1	1	1	Instantaneous				1
2	Volume(m3)	1	1	1	Instantaneous			1	
3	On Time(hours)	1	1	1	Instantaneous				2
4	Time Point(time & date)	1	1	1	Instantaneous			2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

New type of Mbus device

Name: LSE

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

When the meter describe is created, it has to be select from list, which is need for the meter for the data reading.



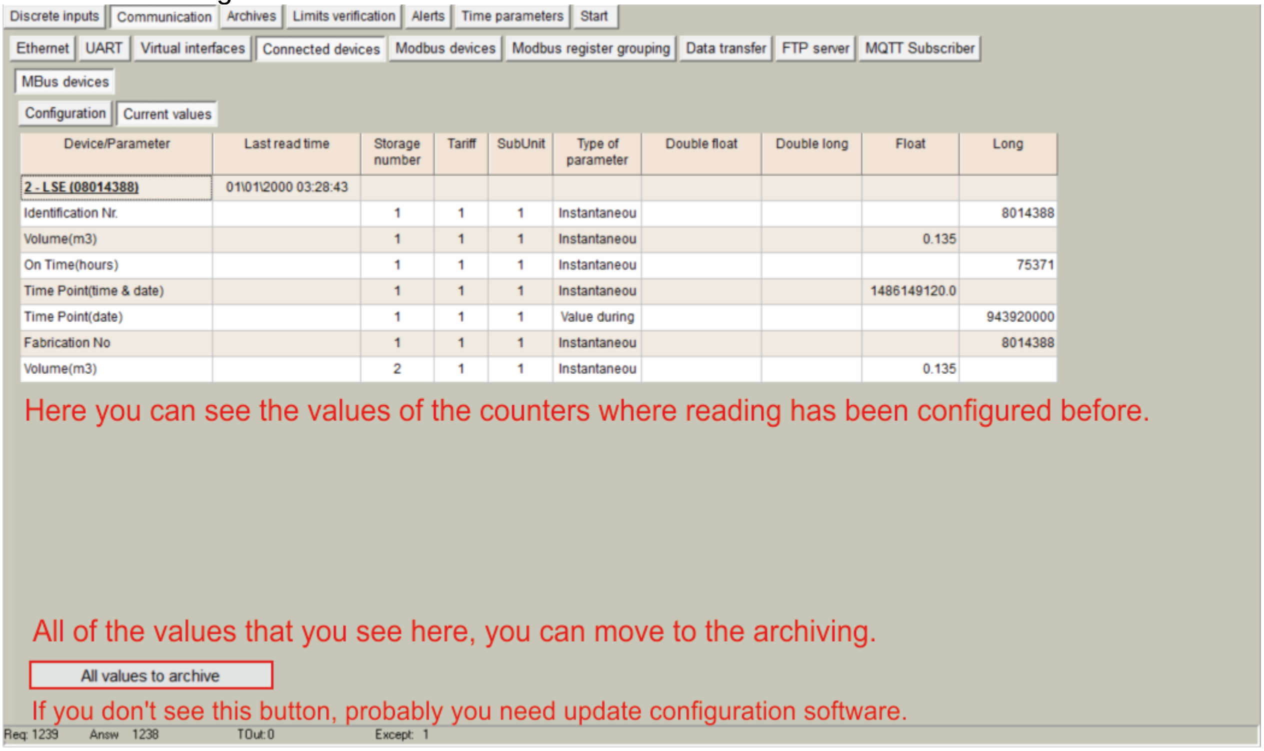
	1	2
Enabled	☑	☐
COM Client	1	1
Amount of meters	1	1
Read period	10	1
Period dimension	● sec. ○ min. ○ h	● sec. ○ min. ○ h

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

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

If your settings are correct, you have to see the values from the meter in configuration tool

"Current values". Now every value from the meter is recorded to registers of controller. That registers can be archived or read over Modbus RTU or Modbus TCP/IP. Archiving is used to send a CSV file to 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.

Modbus devices

Modbus RTU devices

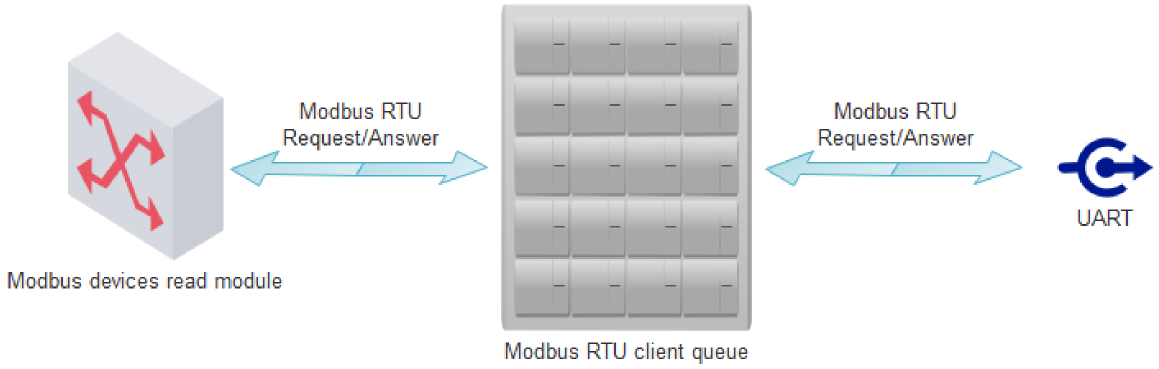
Overview

Device is able to read any standard Modbus RTU slave device which is connected to one of 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 ID from 1 to 240.

Modbus devices read module generates Modbus RTU requests and sends them to appropriate Modbus RTU client. Modbus RTU client sends these requests to configured UART. How configure Modbus requests check "Modbus RTU devices configuration" topic.


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

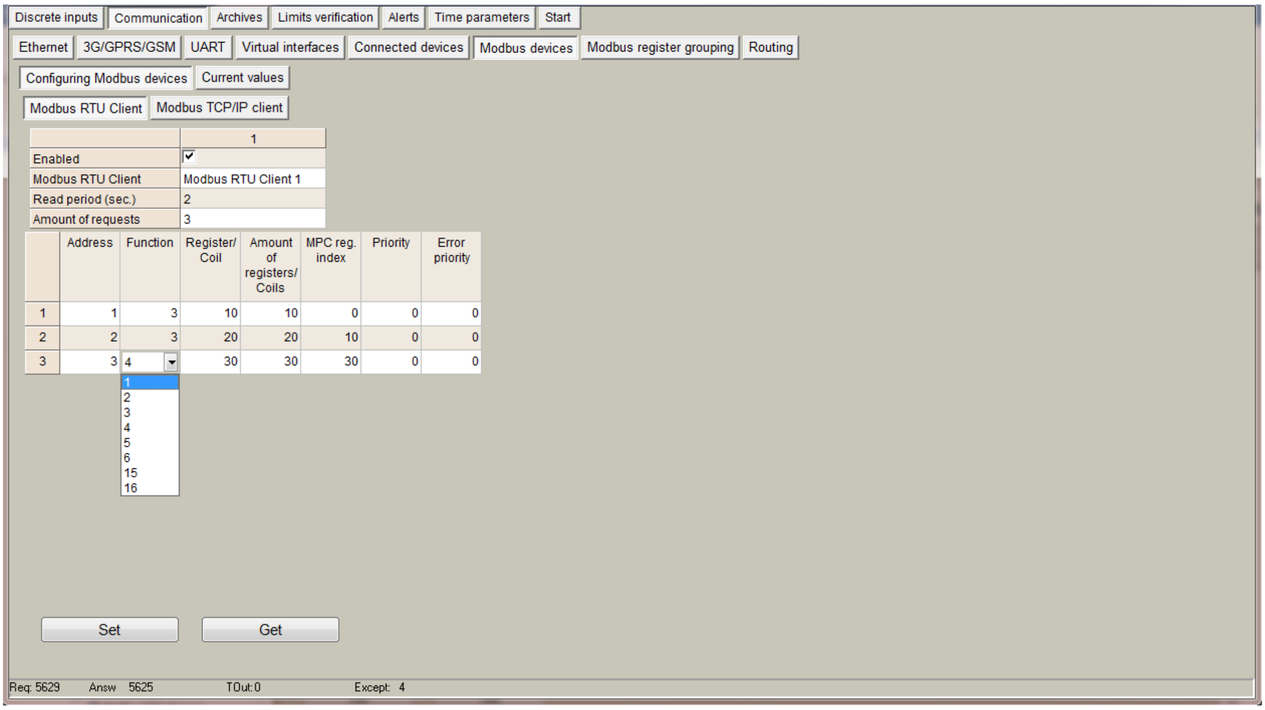
Modbus RTU devices configuration

Modbus RTU devices configuration is made in "Communication/Modbus Devices/Configuring Modbus devices/Modbus RTU client" tab. Before configure your Modbus RTU devices please first make configuration of Modbus RTU clients. Modbus RTU client will link physical UART with Modbus RTU devices module.

- "Enabled" check box - enables/disables Modbus RTU devices
- "Modbus RTU client" - selects one of Modbus RTU client (Modbus RTU client 1, Modbus RTU client 2)
- "Read period (sec)" - read period time in seconds. Value can be from 2 to 3600 s
- "Amount of requests" - number of configured requests. Can be configured up to 50 requests

Modbus requests table:

- "Address" - Modbus RTU device ID
- "Function" - Modbus function for current request, check above of supported functions
- "Register/Coil" - start register or coil address.
- "Amount of registers/Coils" - indicating how many registers to read from start Register/Coil.
- "Reg index" - indicating in which internal registers to store answer data. Registers from 52000 to 52999 is reserved for Modbus devices data. "Reg index" indicates the data index in this area. For example "Reg index" = 0, means data storing from 52000 register. "Reg index" = 10 - data storing from 52010 register
- "Priority" - Request send priority. Higher number means higher priority.
- "Error priority" - always use 0.
- "Set" button - write configuration to device
- "Get" button - read configuration from device



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

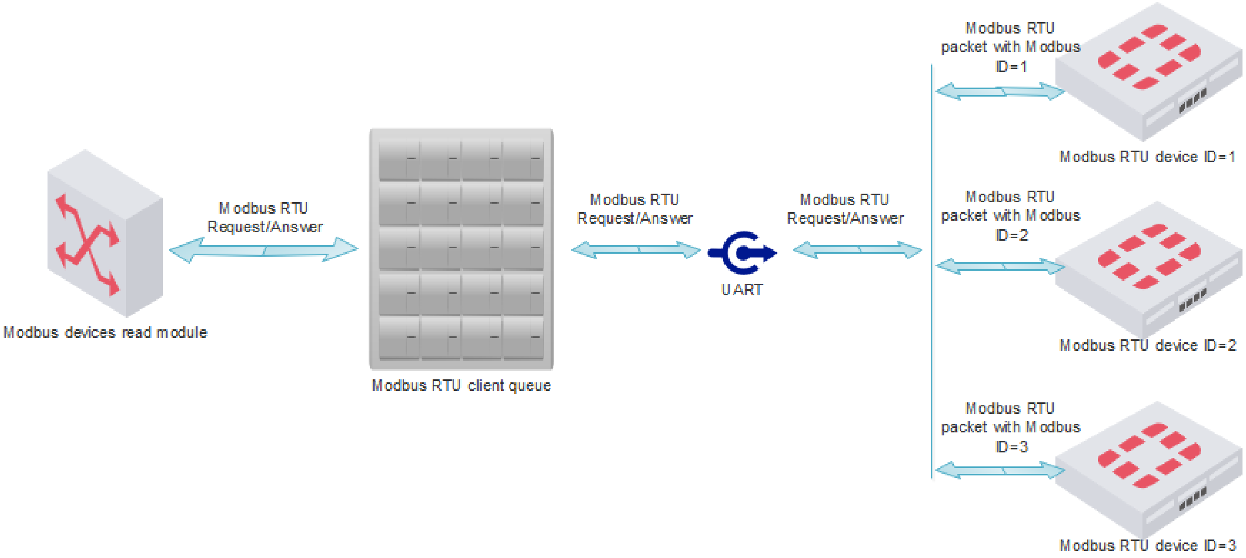
	Address	Function	Register/ Coil	Amount of registers/ Coils	MPC reg. index	Priority	Error priority
1	1	3	10	10	0	0	0
2	2	3	20	20	10	0	0
3	3	4	30	30	30	0	0

Req: 5629 Answ 5625 TOut:0 Except: 4

Example

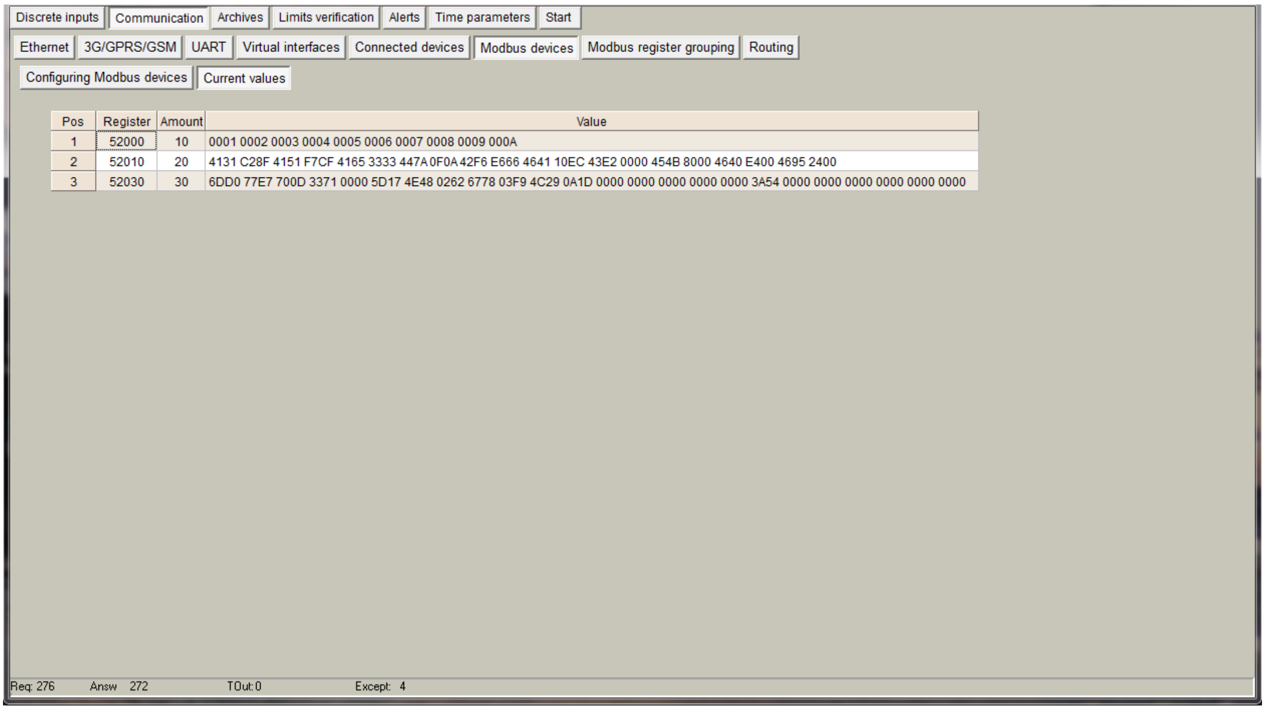
We have 3 Modbus RTU devices connected to one of UART and we need to read some data from them. Let’s look at configuration above. We have 3 requests:

- 1 request - read 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 52000...52009 registers. ((52000+"Reg index")...(52000+"Reg index" + "Amount of registers/Coils" - 1))
- 2 request - read Modbus RTU device with ID=2, function = 3 (Read Multiple Holding Registers), start register = 20 and number of register to read = 20. Device values will be stored in internal 52010...52029 registers.
- 3 request - read Modbus RTU device with ID=3, function = 4 (Read Input Registers), start register = 30 and number of register to read = 30. Device values will be stored in internal 52030...52059 registers.



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

Current values can be checked in "Communication/Modbus Devices/Current values" tab. Every request data is shown in separate line and values is 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

Modbus TCP devices

Overview

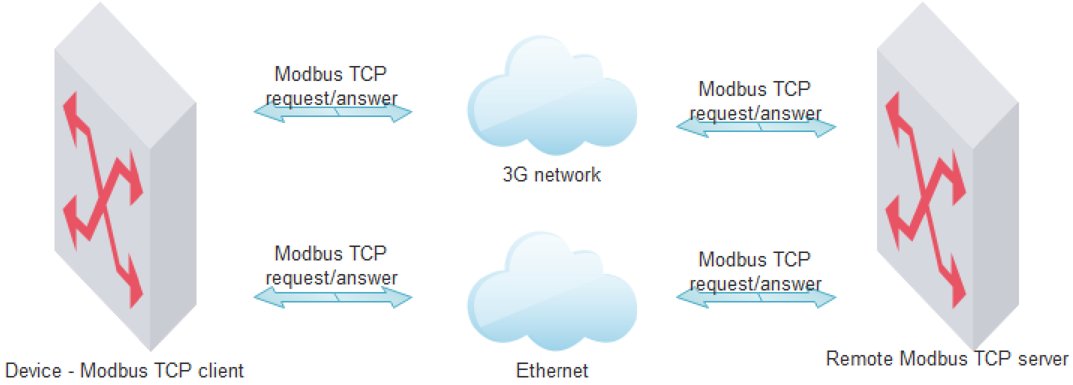
Device is able to read any standard Modbus TCP server which can be reached over 3G/GPRS or LAN network.

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 ID from 1 to 255.

Modbus TCP clients generates Modbus TCP requests to read remote Modbus TCP server and sends them to 3G/GPRS or LAN network. Received data is stored in internal register area (Registers from 52000 to 52999 is reserved for Modbus devices data)


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

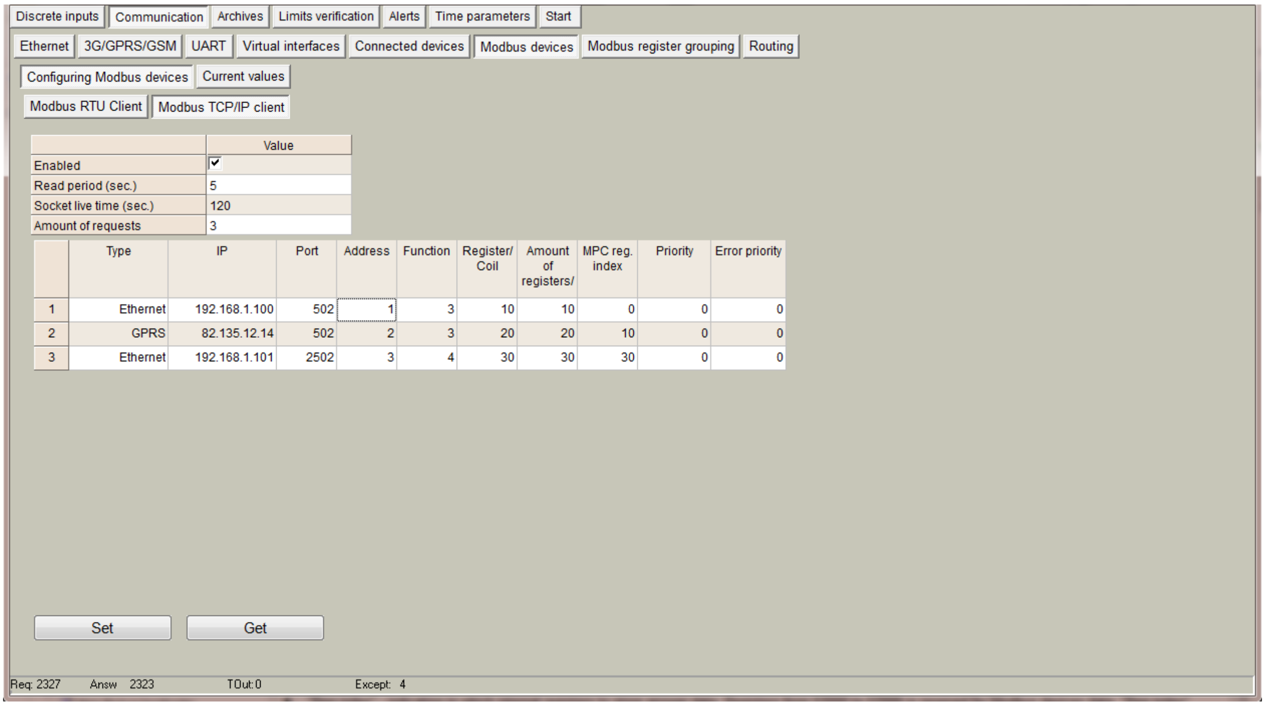
Modbus TCP client configuration

Modbus TCP client configuration is made in "Communication/Modbus Devices/Configuring Modbus devices/Modbus TCP/IP client" tab.

- "Enabled" check box - enables/disables Modbus TCP client
- "Read period (sec)" - read period time in seconds. Values can be from 2 to 3600 s
- "Socket lifetime (sec)" - timeout before closing socket if no data is transferred.
- "Amount of requests" - number of configured requests. Can be configured up to 20 requests

Modbus TCP requests table:

- "Type" - TCP connection channel - Ethernet or 3G/GPRS
- "IP" - remote Modbus TCP server IP address
- "Port" - remote Modbus TCP server TCP port. Standard Modbus TCP port is 502.
- "Address" - Modbus device ID
- "Function" - Modbus function for current request, check above of supported functions
- "Register/Coil" - start register or coil address.
- "Amount of registers/Coils" - indicating how many registers to read from start Register/Coil.
- "Reg index" - indicating in which internal registers to store answer data. Registers from 52000 to 52999 is reserved for Modbus devices data. "Reg index" indicates the data index in this area. For example "Reg index" = 0, means data storing from 52000 register. "Reg index" = 10 - data storing from 52010 register
- "Priority" - Request send priority. Higher number means higher priority.
- "Error priority" - always use 0.
- "Set" button - write configuration to device
- "Get" button - read configuration from 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

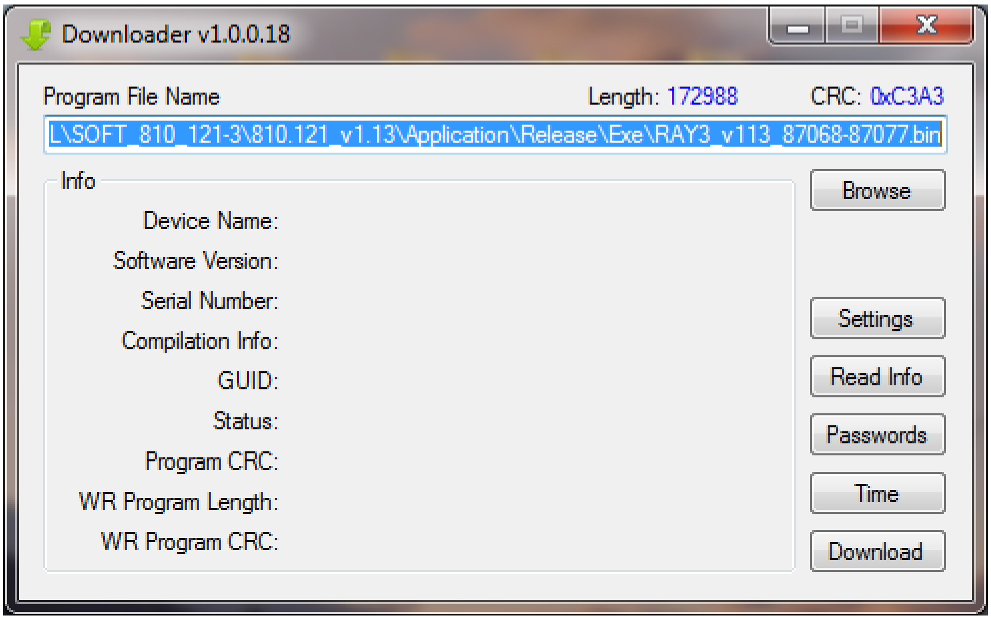
Overview

Overview

Device is able to read some included electricity meters with different protocols
Supported electricity meter list:

Firmware update

1. Run downloader

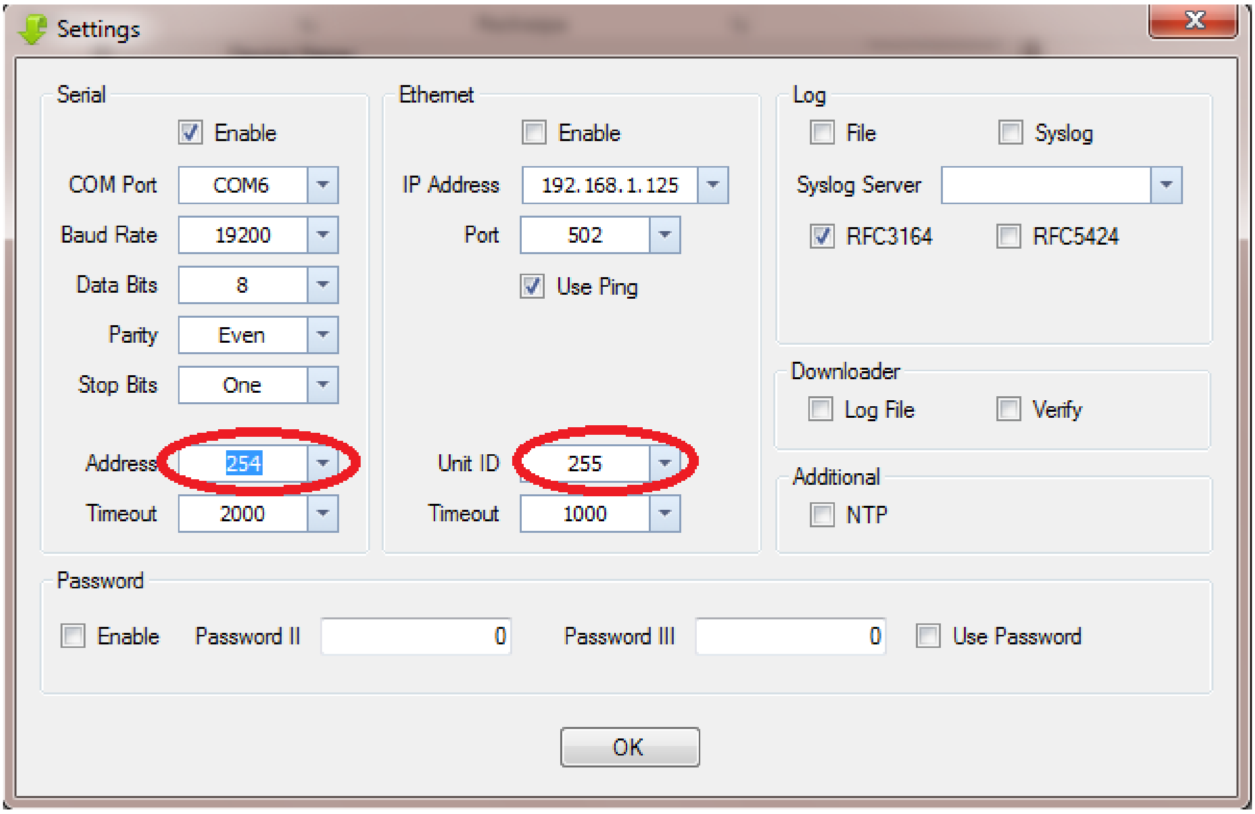


Program File Name Length: 172988 CRC: 0xC3A3

L\SOFT_810_121-3\810.121_v1.13\Application\Release\Exe\RAY3_v113_87068-87077.bin

Device Name:
Software Version:
Serial Number:
Compilation Info:
GUID:
Status:
Program CRC:
WR Program Length:
WR Program CRC:

2. Click “Browse” and select firmware bin file.
3. Click “Settings”, select communication parameters. For “Serial” communication use Modbus address 254, for “Ethernet” – 255



Settings

Serial
☑ Enable
COM Port: COM6
Baud Rate: 19200
Data Bits: 8
Parity: Even
Stop Bits: One
Address: 254
Timeout: 2000

Ethernet
□ Enable
IP Address: 192.168.1.125
Port: 502
☑ Use Ping
Unit ID: 255
Timeout: 1000

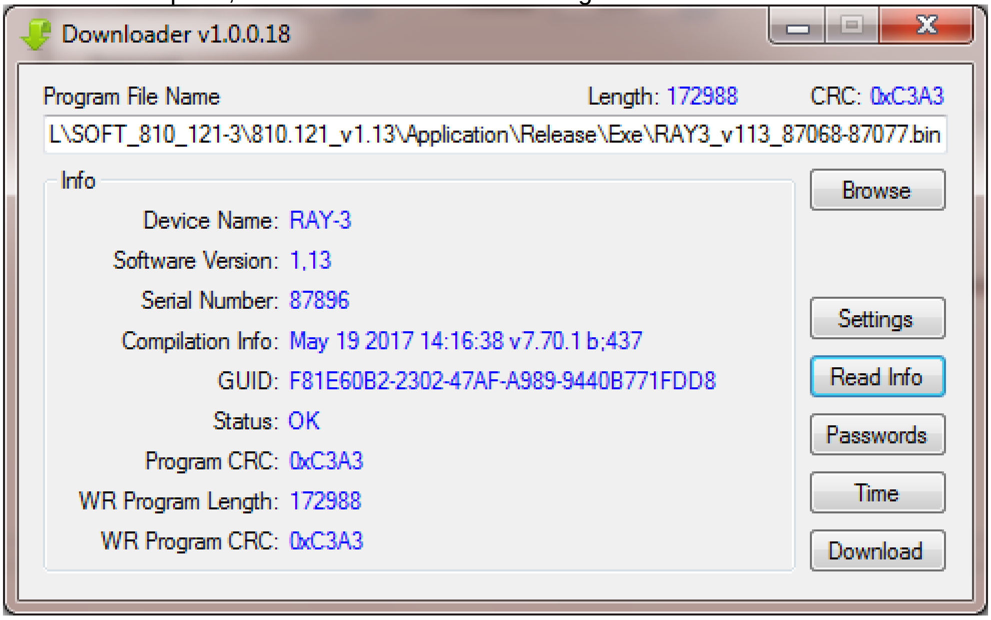
Log
□ File □ Syslog
Syslog Server:
☑ RFC3164 □ RFC5424

Downloader
□ Log File □ Verify

Additional
□ NTP

Password
□ Enable Password II: 0 Password III: 0 □ Use Password

4. Click “Read info”, if communication ok reads device description. If you have problems reading description, review communication settings.

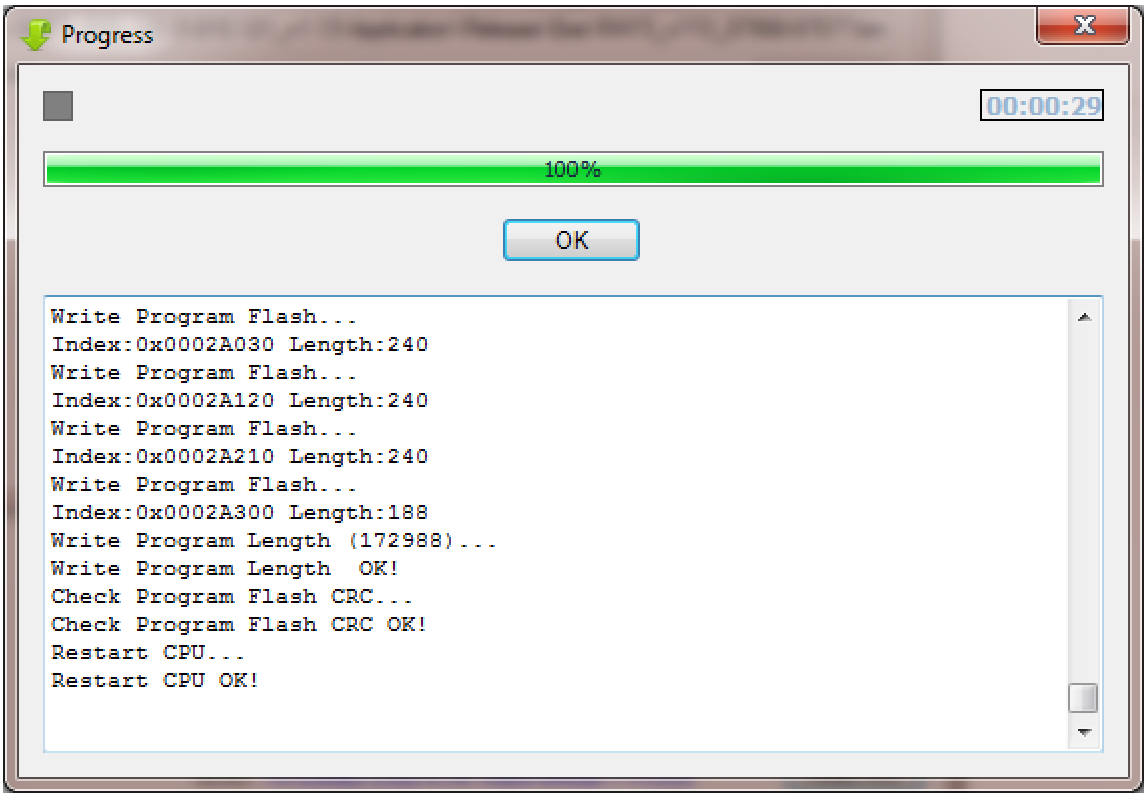


Downloader v1.0.0.18

Program File Name Length: 172988 CRC: 0xC3A3
L\SOFT_810_121-3\810.121_v1.13\Application\Release\Exe\RAY3_v113_87068-87077.bin

Info
Device Name: RAY-3
Software Version: 1,13
Serial Number: 87896
Compilation Info: May 19 2017 14:16:38 v7.70.1 b;437
GUID: F81E60B2-2302-47AF-A989-9440B771FDD8
Status: OK
Program CRC: 0xC3A3
WR Program Length: 172988
WR Program CRC: 0xC3A3

5. Click “Download” to start program device. After download ends close program window, and click “Read info” to check firmware update was successful.



```
Write Program Flash...
Index:0x0002A030 Length:240
Write Program Flash...
Index:0x0002A120 Length:240
Write Program Flash...
Index:0x0002A210 Length:240
Write Program Flash...
Index:0x0002A300 Length:188
Write Program Length (172988)...
Write Program Length  OK!
Check Program Flash CRC...
Check Program Flash CRC OK!
Restart CPU...
Restart CPU OK!
```