

RAY-3 documentation

Table of contents

Introduction	4
Abbreviations and explanations	4
Preface	5
Symbols	5
Safety instructions	5
About RAY-3	6
Specification	7
Connecting to device	9
Program modules	14
Archives	14
Overview	14
Events archive	15
Diagnostic archive	16
User defined archive	18
TCP modules	20
FTP server	20
FTP client	21
Csv file creator	21
FTP client configuration	23
MQTT client	24
TCP/IP connection table	28
Routing TCP/IP - serial (request/answer)	29
Routing TCP/IP - serial (transparent)	31
Router 3G/GPRS<->Ethernet	33
Gateway Modbus TCP<->Modbus RTU	35
Hardware	39
System	40
microSD	41
Time settings	41
Ethernet configuration	42
3G/GPRS configuration	44
Serial ports	47
Analog inputs	50
Analog outputs	51
Discrete inputs	53
Discrete outputs	56
Virtual interfaces	59
Virtual COM clients	59
Modbus RTU clients	62
Modbus RTU servers	64
Meters	65
Mbus meters	65
Setting up defaults values	70

Bootloaders	72
Manufacture bootloader	72
Atmel chip ISP	74
Firmware update	76

Introduction

Created with the Personal Edition of HelpNDoc: [Create cross-platform Qt Help files](#)

Abbreviations and explanations

Xn – is a number of socket. This information are provided for manufacturer s purpose and used in data schemas and connection diagrams.

GSM – Global Standart for Mobile Communications. This interfaces is prepared for remote connections and data bidirectional data transfer over Global Standart 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 interfaces 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 an 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 are used as serial interface number.

GND – ground wire contact

RS232 - the traditional name for a series of standards for serial binary single-ended data and control signals connecting between a DTE (Data Terminal Equipment) and a DCE (Data Circuit-terminating Equipment). It is commonly used in computer serial ports. The standard defines the electrical characteristics and timing of signals, the meaning of signals, and the physical size and pin out of connectors. RS232 interfaces are prepared for connection of ppheripheral devices (example energy meters, controllers, machines and etc.).

TD – contact for transfer data wire of RS232 socket

RD – contact for read data wire of RS232 socket

DTR – contact for Data Transmit Ready wire of RS232 socket

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 ppheripheral 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 ppheripheral 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 wire, making it very cost effective.

MBUS+ – contact for M-Bus positive wire

MBUS- – contact fot 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.

Data – contact for data wire

Req – contact for request wire

CL+ – contact for current loop positive wire

CL- – contact for current loop negative wire

Status – device status indicating LED

Uoutput – status of power for external device indicating LED

TX/RX – data transfer/receive indicating LED

TXD – data transferring LED indicator

RXD – data receiving LED indicator

100Mbps – 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.

Created with the Personal Edition of HelpNDoc: [Easy EBook and documentation generator](#)

Preface

Created with the Personal Edition of HelpNDoc: [Benefits of a Help Authoring Tool](#)

Symbols

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

Symbol	Explanation
--------	-------------



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



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

Created with the Personal Edition of HelpNDoc: [Full-featured Documentation generator](#)

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 circuite 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 pheripheral 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;
3. External Back-Up battery's for powering of controller.



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.

About RAY-3

OVERVIEW

Module RAY-3 is created for measuring of analog (voltage, current, resistance) parameters, for tracking the status of discrete inputs, formation of analog and discrete output signals, archiving and sending data to local operator over RS232/RS485 and/or USB and to remote operator over GSM/GPRS/EDGE or UMTS/HSPA+ and Ethernet.

DEFAULT FEATURES

- 6 analog and 4 discrete inputs, 2 voltage and 2 relay outputs;
- Data sending over 2G/3G and/or Ethernet using Modbus TCP/IP protocol;
- Data log (up to 8MB flash) with real time stamp;

- Alarm limits tracing of analog and discrete inputs, logging in event archive, warning about alarms over GPRS, UMTS/HSPA+, Ethernet and by sending SMS for one or few users;
- Internal battery ensure at least 1 hour operation, after power fail;
- Alarms status of analog and discrete channels fixing;
- Data from local device reading over RS232/RS485 interfaces;
- Supported wide range of power supplier (from 9 to 36V) and wide range of operating temperatures (from -25 oC to +60 oC);
- Firmware updating and configuration over any of installed interfaces.

CUSTOMIZING DEVICE

The exceptional feature of this device - a flexible hardware and software configuration, it depends on customer needs. You can choose the desired interfaces and functionality.

Manufacturers code:

701.010.	A	0 - none 1 - RS485 2 - RS232
	B	0 - none 2 - RS232
	C	0 - none 4 - Mbus interface
	D	0 - none 1 - Ethernet
	E	0 - none 3 - 3G modem 4 - 3G e-GPS 5 - 3G for America
	F	0 - none 1 - Current 0/4 -20mA analog input 2 - Voltage -5 -45V analog input 3 - Voltage -10 -+10V analog input 4 - Thermoresistor (Pt1000) analog IN 5 - Thermoresistor (Pt1000) analog IN 6 - Voltage 0 -+10V or Pt100, jumper switchable analog IN 7 - Resistance 0 - 10kΩ analog IN
	G	0 - none 1 - Current 0/4 -20mA analog input 2 - Voltage -5 -45V analog input 3 - Voltage -10 -+10V analog input 4 - Thermoresistor (Pt1000) analog IN 5 - Thermoresistor (Pt1000) analog IN 6 - Voltage 0 -+10V or Pt100, jumper switchable analog IN 7 - Resistance 0 - 10kΩ analog IN
	H	0 - none 1 - Current 0/4 -20mA analog input 2 - Voltage -5 -45V analog input 3 - Voltage -10 -+10V analog input 4 - Thermoresistor (Pt1000) analog IN 5 - Thermoresistor (Pt1000) analog IN 6 - Voltage 0 -+10V or Pt100, jumper switchable analog IN 7 - Resistance 0 - 10kΩ analog IN
	I	0 - none 1 - Current 0/4 -20mA analog input 2 - Voltage -5 -45V analog input 3 - Voltage -10 -+10V analog input 4 - Thermoresistor (Pt1000) analog IN 5 - Thermoresistor (Pt1000) analog IN 7 - Resistance 0 - 10kΩ analog IN
	J	0 - none 1 - Current 0/4 -20mA analog input 2 - Voltage -5 -45V analog input 3 - Voltage -10 -+10V analog input 4 - Thermoresistor (Pt1000) analog IN 5 - Thermoresistor (Pt1000) analog IN 7 - Resistance 0 - 10kΩ analog IN
	K	0 - none 1 - Current 0/4 -20mA analog input 2 - Voltage -5 -45V analog input 3 - Voltage -10 -+10V analog input 4 - Thermoresistor (Pt1000) analog IN 5 - Thermoresistor (Pt1000) analog IN 7 - Resistance 0 - 10kΩ analog IN
	L	0 - none 1 - 1 voltage 0 -+5V analog output 2 - 2 voltage 0 -+5V analog output
	M	0 - none 4 - 4 discrete inputs
	N	0 - none 1 - 1 discrete relay output 2 - 2 discrete relay outputs
	O	0 - none 1 - Backup battery
	P	0 - none 1 - microSD card

Created with the Personal Edition of HelpNDoc: [Easy CHM and documentation editor](#)

Specification

First interface	
RS485	distance up to 1,2km, max 32 transivers, speed up to 19.2 Kbits/s
RS232	distance up to 15m, speed up to 19,2Kbit/s
Second interface	
RS232	distance up to 15m, speed up to 19,2Kbit/s
Third interface	
Mbus	up to 25 devices
Fourth interface	
Ethernet	twisted pair, 10/100 Mbps, distance up to 100m
Fifth interface	
UMTS/HSPA+	2 band, depending on market 850/1900MHz, 900/2100MHz or 800(850)/2100MHz
GSM/GPRS	4 band 850/900/1800/1900 MHz

Discrete IN	4	sink contact
Discrete OUT	2	relay 3A
Analogi IN	6	resistance, voltage or current, reading 10 times per second
Analog OUT	2	voltage 0-10V, load up to 5mA

Protocols

	Modbus RTU Modbus TCP/IP IP ICMP UDP TCP DHCP PPP ARP Sntp IEC60870-5-104:2000 DynDNS FTP server FTP client DNS client
--	--

General

Power	9-36 VDC
Internal battery	3,7V 750 mAh

Regulatory approvals

Safety	EN 60950-1:2006 EN 60950-1:2006/A1:2010 EN60950-1:2006/A11:2009
--------	---

Specification

CPU	CORTEX M4
SD card support	micro SD card up to 8GB
Memory	archive storage 1-8 MB, independent data storage without power about 5 years

LED indication

Power	+
Charging of internal battery	+

Status of discrete input, for each port	+
Serial ports read/write for each port	+
GSM/GPRS modem status	+
Ethernet status	+
Programing and updating	
Remote	3G, Ethernet (RJ45)
Locally	USB, RS232, RS485
Physical characteristics	
Dimmensions	147x128x50 mm
Weight	400 g
Mounting type	on DIN32 rail
Safety class	IP20
Climate conditions	
Operating temperature	-25..+60 °C
Storage temperature	-40..+60 °C
Humidity range	5-95%, non-condensing
Other fuetures	
Real time clock	+
24 months warranty period	+

Created with the Personal Edition of HelpNDoc: [Free HTML Help documentation generator](#)

Connecting to device

Overview

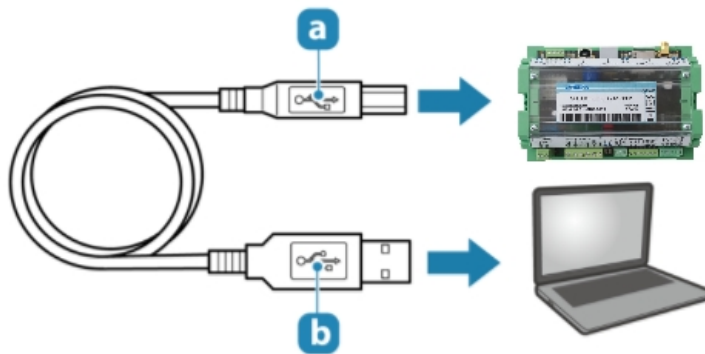
USB port is used for local configure of device. Also it is possible configure device via Ethernet, 3G modem or any of UARTS if them are used as modbus slaves. All configuration is made using modbus protocol and using device configuration tool software which can be downloaded from manufacture website.

Connecting over USB

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

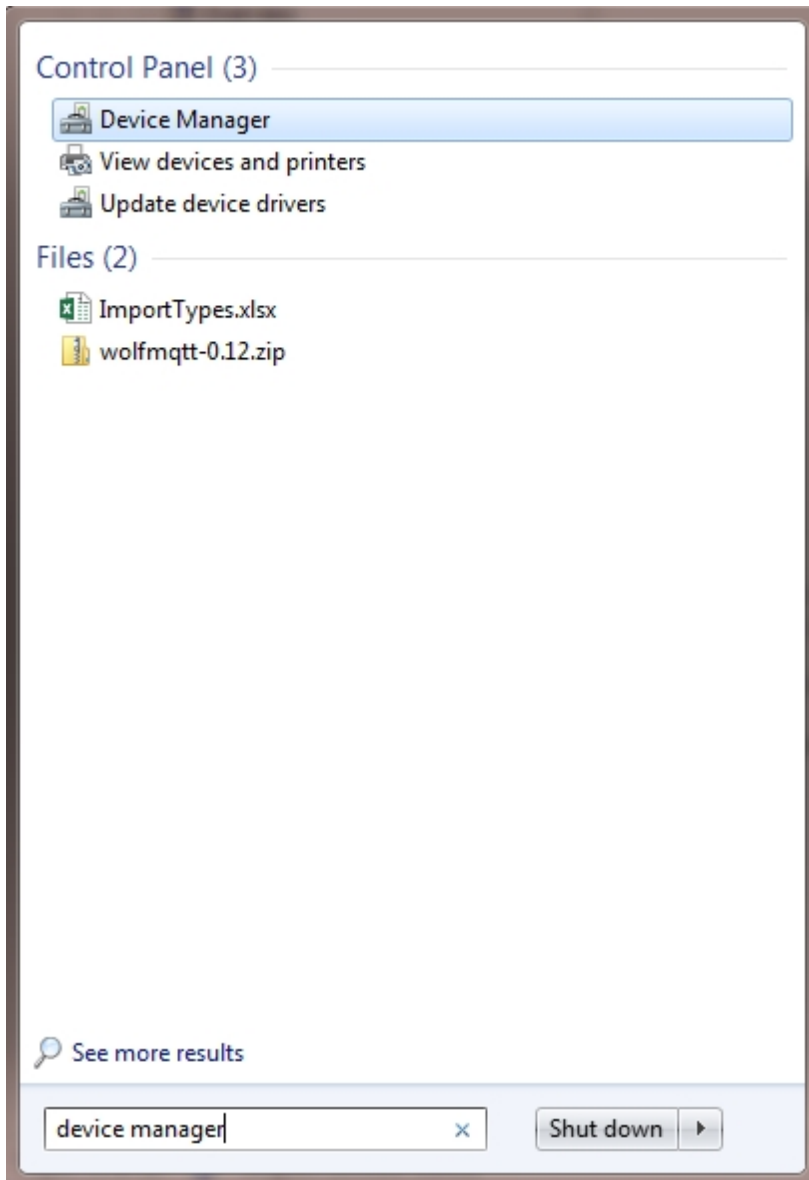
- a) To device USB Type-B

- b) To computer USB Type-A



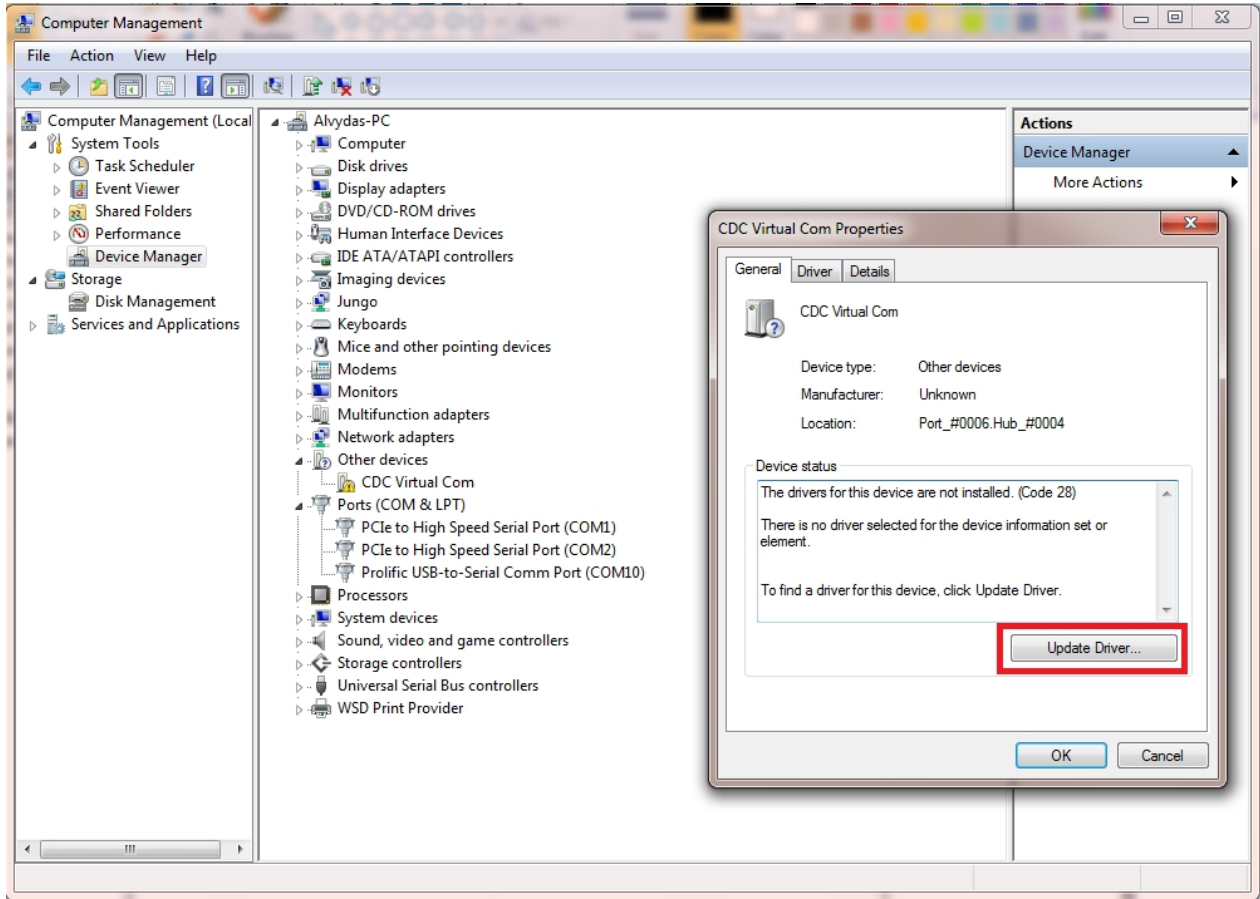
If USB drivers not installed automatically you need to install them manually. Follow this steps:

1. In the search box, type into and then click **Device Manager**.



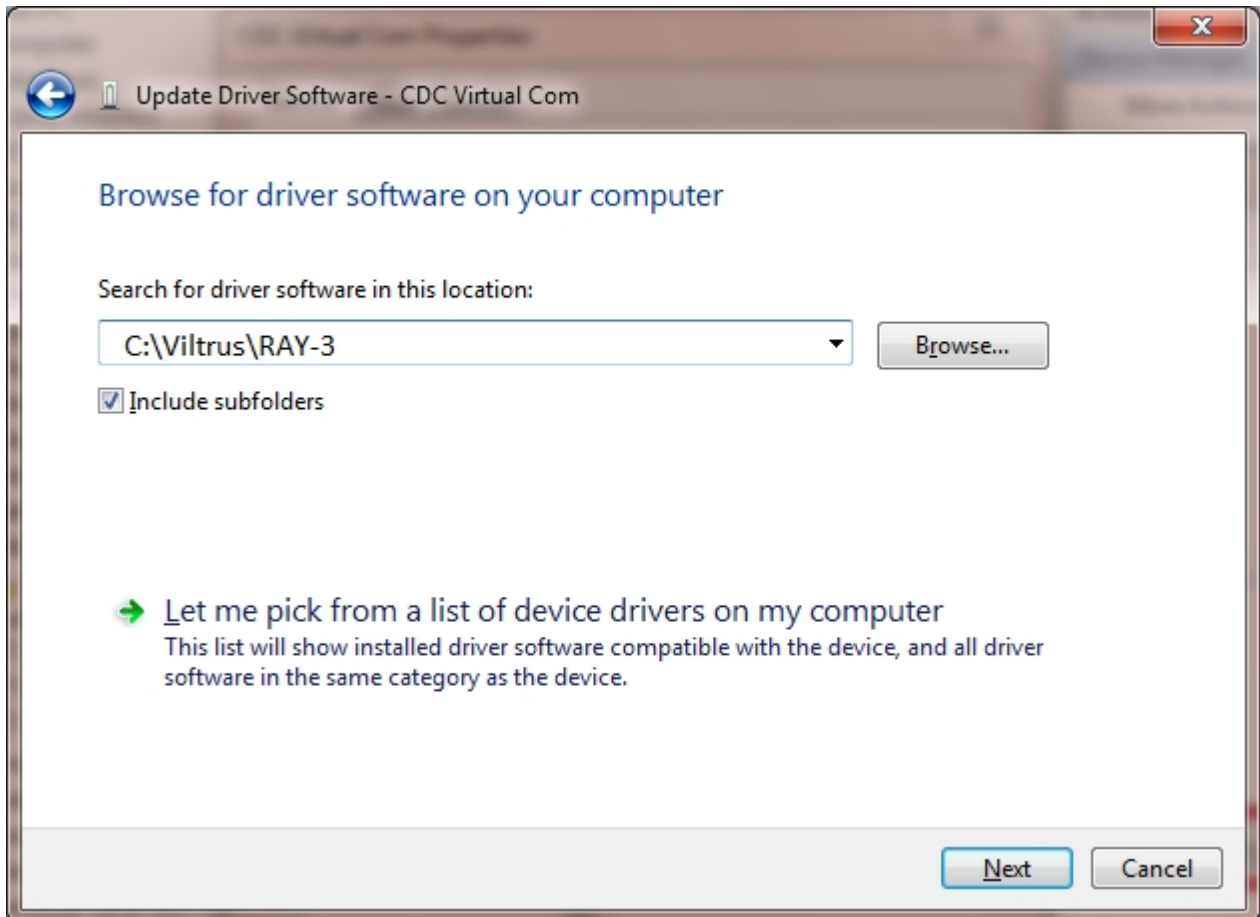
2. Double-click the device category, and then double-click the device you want.

3. Click Update Driver, and follow the instruction

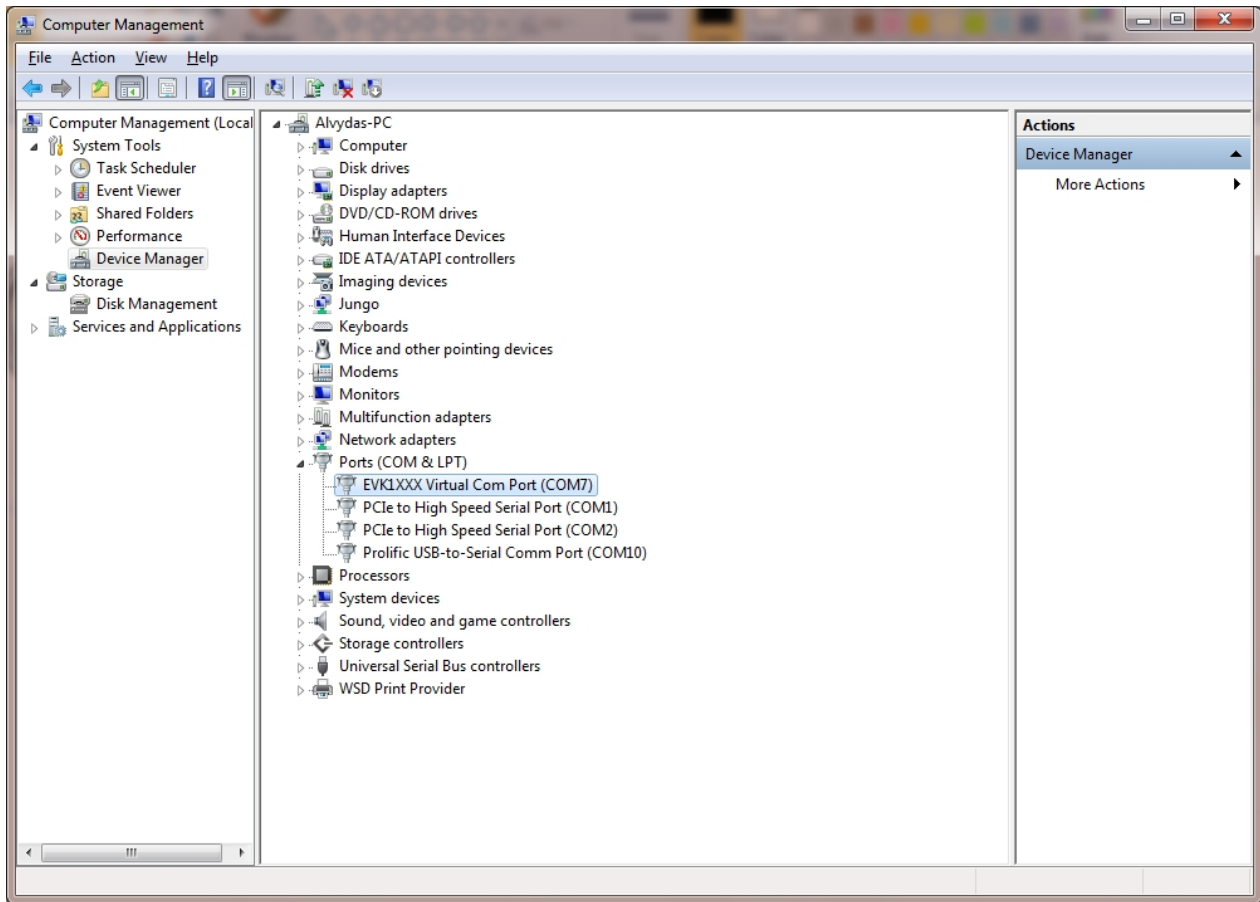


4. Select "Browse my computer for driver software", click "Browse" and select configuration software folder.

5. Click "Next".



6. Wait while Windows installs driver. If you see message "Windows can't verify the publisher" select "Install this driver software anyway".
7. After installation you will see something like "EVK1XXX Virtual Com Port" and com port number. Use this com port for connection with configuration tool.



On some windows version (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

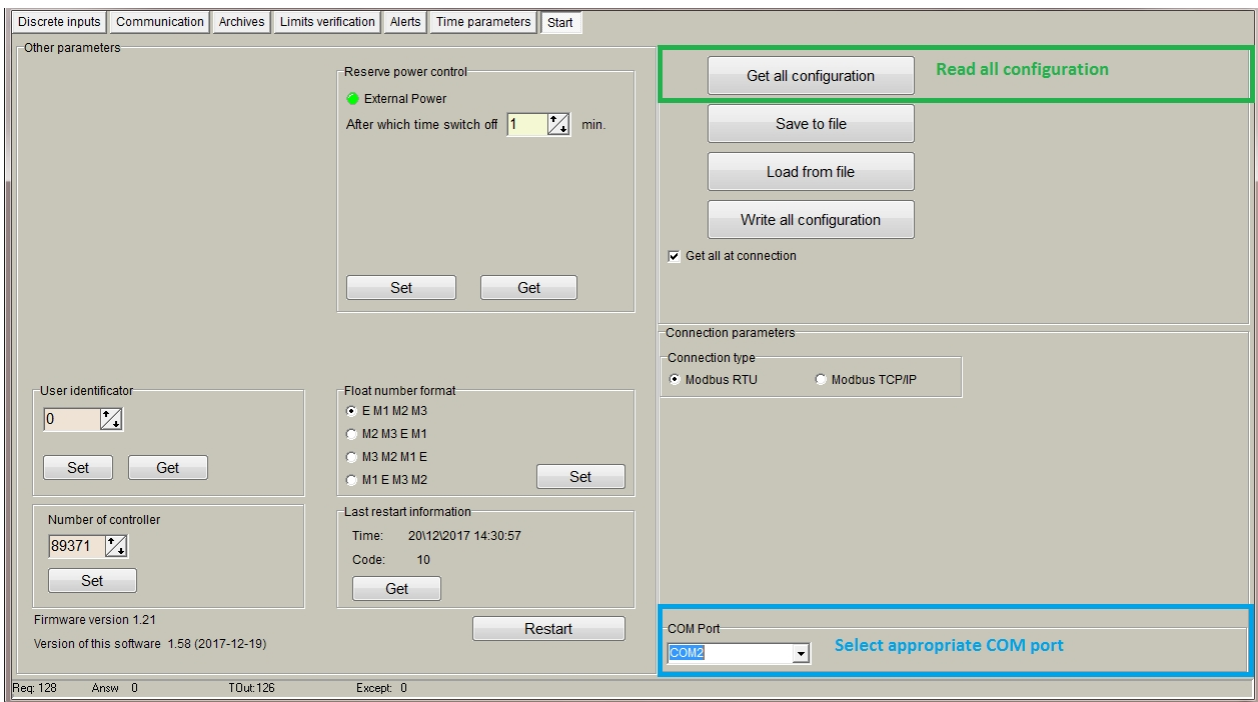
After USB driver is installed run device configuration tool software.

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



Created with the Personal Edition of HelpNDoc: [News and information about help authoring tools and software](#)

Program modules

Created with the Personal Edition of HelpNDoc: [Easy EPub and documentation editor](#)

Archives

Created with the Personal Edition of HelpNDoc: [Free EBook and documentation generator](#)

Overview

Device have 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 is 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).

Created with the Personal Edition of HelpNDoc: [Full-featured EBook editor](#)

Events archive

Overview

All events is 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"

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. If alarm source 4 then value is limits alarm ID.	Int (16 bits)
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 -	900 .. 999	10000	1250	4911

	255				
--	-----	--	--	--	--

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 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	Long int (32 bits)
Event value (float)	If event type: 1 - always 0 3 - new set value(mostly 0, if delete all storage) 4 and 5- always 0 6 - always 0 7 - Time correction value (-30s +30s)	Float (32 bits)
		Total 16 bytes

Records 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)

Created with the Personal Edition of HelpNDoc: [Create help files for the Qt Help Framework](#)

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

The screenshot shows a configuration window with two main sections: 'Storage parameters' and 'Records in archives'.

Storage parameters:

	Period	Delay
User archive (min.)	1	0

Labels below the table: 'Archive period in min' (green) and 'Archive delay in s' (blue).

Records in archives:

	Records	
Alerts	0	Clear
User archive	0	Clear

Label below the table: 'Current records count' (purple).

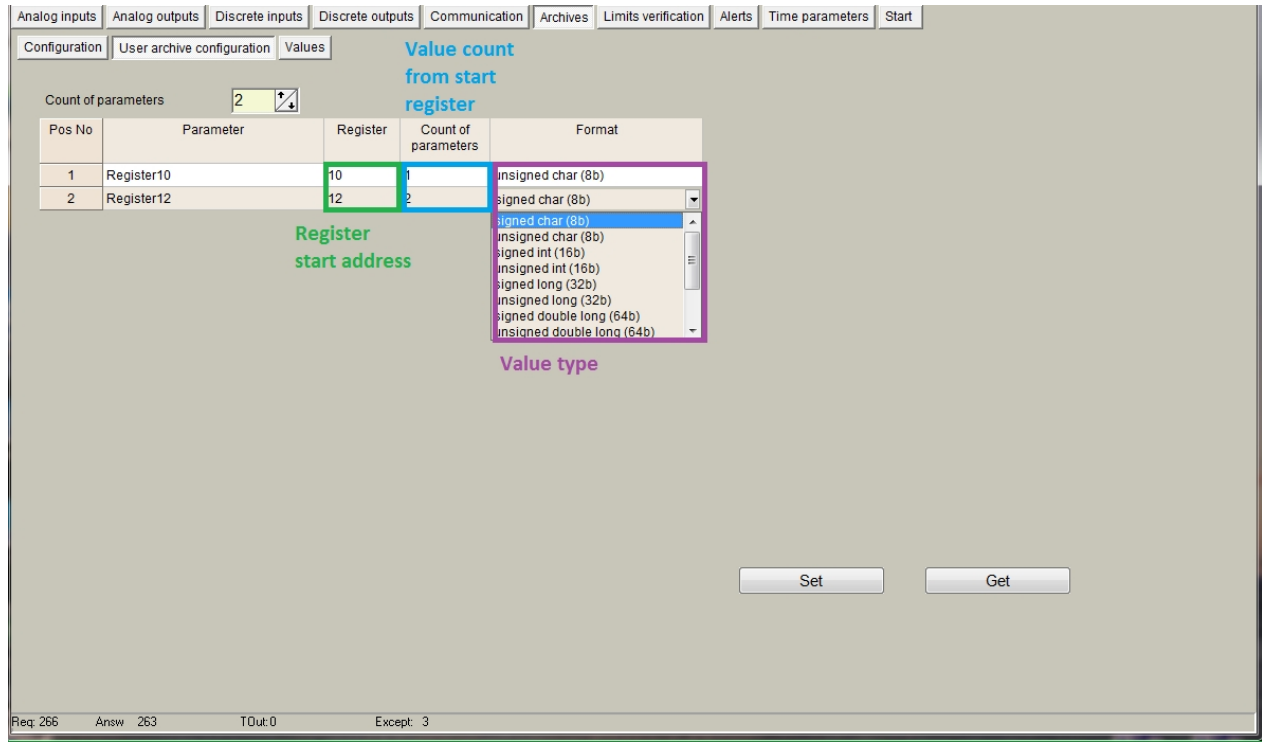
Buttons at the bottom: 'Set' and 'Get'.

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*



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

Reading user archive over Modbus file system

User defined archive can be read using Modbus read file function 20.

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

Diagnostic archive record structure:

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 4929 register or in configuration tool "Archives/User archive configuration" How configure registers check	Long int (32 bits)
		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)

Created with the Personal Edition of HelpNDoc: [Free CHM Help documentation generator](#)

TCP modules

Created with the Personal Edition of HelpNDoc: [Full-featured Documentation generator](#)

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

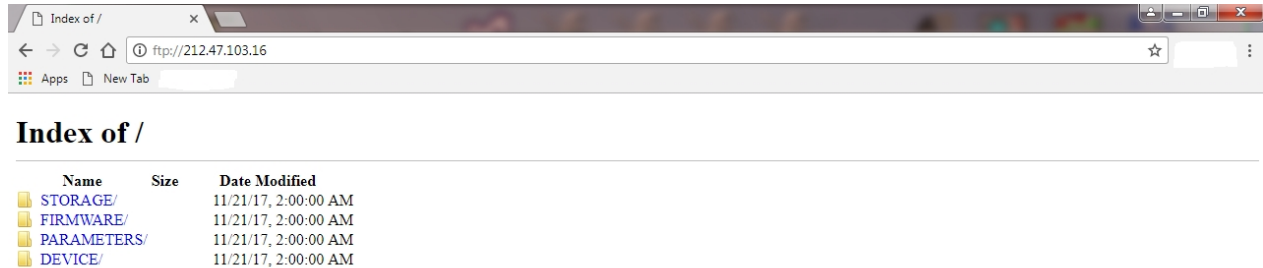
The screenshot shows a configuration interface with several tabs: Analog inputs, Analog outputs, Discrete inputs, Discrete outputs, Communication, Archives, Alerts, Time parameters, and Start. The 'Communication' tab is active, and within it, the 'FTP server' sub-tab is selected. The interface contains two input fields: 'User Name' with the value 'user' and 'User Password' with the value 'pass'. Below these fields are two buttons labeled 'Set' and 'Get'.

Lets 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.

The screenshot shows a Google Chrome browser window with the address bar set to 'ftp://212.47.103.16'. An 'Authentication required' dialog box is displayed in the foreground. The dialog box contains the text 'ftp://212.47.103.16 Your connection to this site is not private'. It has two input fields: 'Username' with the value 'user' and 'Password' with the value '****'. At the bottom of the dialog are two buttons: 'Log in' and 'Cancel'.

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.



Created with the Personal Edition of HelpNDoc: [Free EPub producer](#)

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.

Created with the Personal Edition of HelpNDoc: [Easily create Web Help sites](#)

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.

The screenshot shows a configuration window for Modbus devices. At the top, there are tabs for different device types: Analog inputs, Analog outputs, Discrete outputs, Communication, Archives, Time parameters, and Start. Below these are more specific tabs: 3G/GPRS/GSM, UART, Virtual interfaces, Connected devices, Modbus devices, Data transfer, FTP server, and Routing. Under 'Modbus devices', there are sub-tabs for Common parameters, Modbus TCP/IP, and FTP.

The main area contains a table with the following columns: Parameter, Register/Coil, Amount of parameters or string length, Format, Header, and Dimension. The 'Amount of parameters or string length' column has a dropdown menu set to '3'. The 'Format' column has a dropdown menu open, showing options like signed long (32b), signed long (32b), signed double long (64b), signed double long (64b), float (32b), Double float (64b), Siemens float, and String. The 'Header' column contains 'Value INT16', 'Value INT32', and 'Value FLOAT'. The 'Dimension' column contains 'm3', 'kWh', and 'm3/h'. A checkbox labeled 'Set user archive' is checked.

At the bottom of the window, there are four buttons: 'Set', 'Get', 'Load from CSV', and 'Save to CSV'. The status bar at the very bottom shows 'Req: 179', 'Answ: 174', 'TOut: 2', and '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;
```

Created with the Personal Edition of HelpNDoc: [Create iPhone web-based documentation](#)

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.

The screenshot shows the 'File transfer configuration' interface. It includes a 'File name template' field with the value 'Dev_YYYY_MM_DD__HH-NN.csv'. Below this are four columns for FTP server configuration (server 1 to 4). Server 1 is configured with Ethernet, IP 192.168.1.126, username 'username1', and password 'password1'. Server 2 is configured with GPRS, URL 'ftpServerName.com/My_Folder/', username 'username1', and password 'password1'. A 'File send status' table shows that all servers are 'Not active' with a timestamp of '0110111998 00:00:00'. At the bottom, there are 'Set' and 'Get' buttons.

FTP server	Enabled	FTP port	Transfer period (min.)	Repeat transfer if failure (min.)	Max number of last records to be sent	Transmission channel	URL and directory	User name	Password	Status	Successful/attempt/last record transfer times
1	<input checked="" type="checkbox"/>	21	10	5	20	<input checked="" type="radio"/> Ethernet	192.168.1.126	username1	password1	Not active	0110111998 00:00:00 0110111998 00:00:00 0110111998 00:00:00
2	<input checked="" type="checkbox"/>	21	10	5	20	<input checked="" type="radio"/> GPRS	ftpServerName.com/My_Folder/	username1	password1	Not active	0110111998 00:00:00 0110111998 00:00:00 0110111998 00:00:00
3	<input type="checkbox"/>	21	10	5	20	<input checked="" type="radio"/> Ethernet				Not active	0110111998 00:00:00 0110111998 00:00:00 0110111998 00:00:00
4	<input type="checkbox"/>	21	10	5	20	<input checked="" type="radio"/> Ethernet				Not active	0110111998 00:00:00 0110111998 00:00:00 0110111998 00:00:00

Example

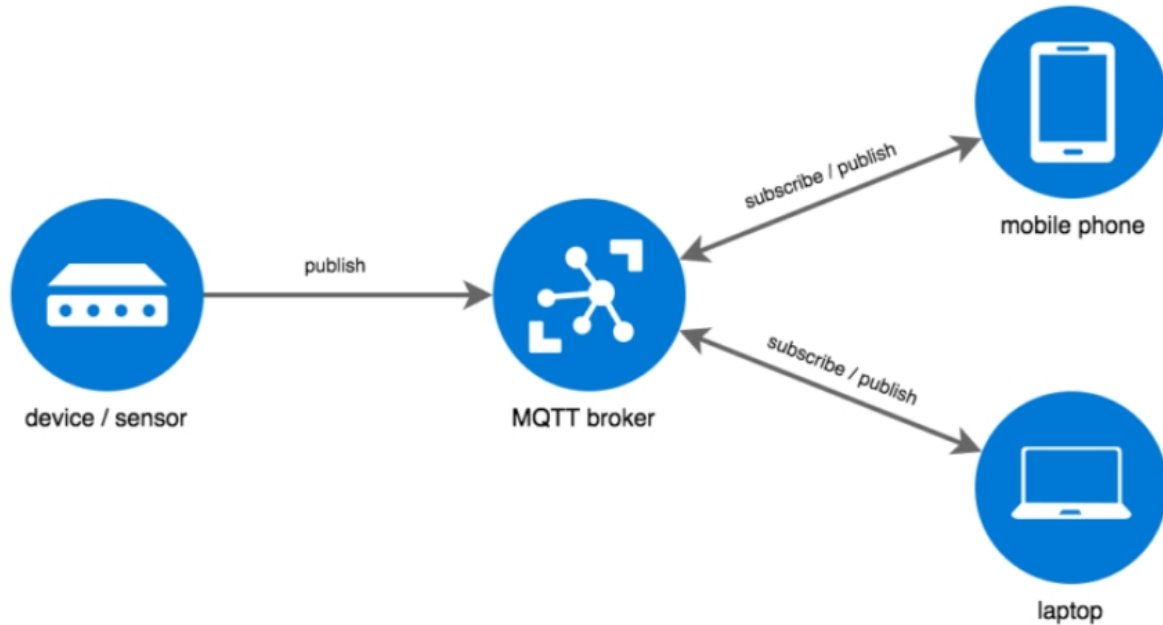
in the picture above is configured 2 FTP servers

- 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 send 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 connectio 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.

The screenshot shows the 'MQTT Subscriber' configuration page in the RAY-3 interface. The page has a navigation bar at the top with tabs for various settings: 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, Modbus register grouping, Data transfer, FTP server, MQTT Subscriber, and Routing. The 'MQTT Subscriber' tab is selected.

The configuration fields are as follows:

- Enabled:**
- Transmission channel:** Radio buttons for Ethernet and 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.

At the bottom of the configuration area, there are two buttons: 'Set' and 'Get'. Below the configuration area, the status bar shows: Req: 380, Answ: 377, TOut: 0, Except: 3.

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.

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 it's 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.

Texts for discrete inputs		Texts for limits	
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			

Events messages

Clear All

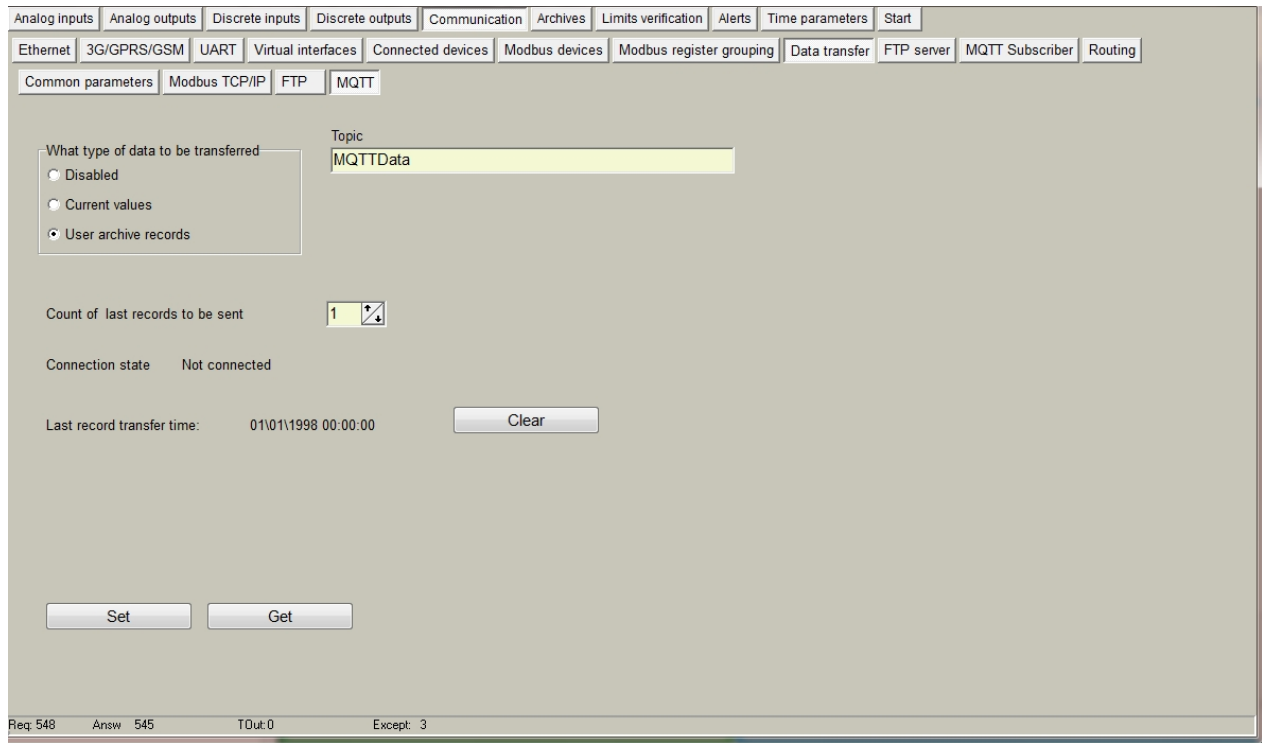
Set Get

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.



Created with the Personal Edition of HelpNDoc: [Easily create Help documents](#)

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

Parameter	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
Incoming port	502	21	1231	502	0	0
Outcoming 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.

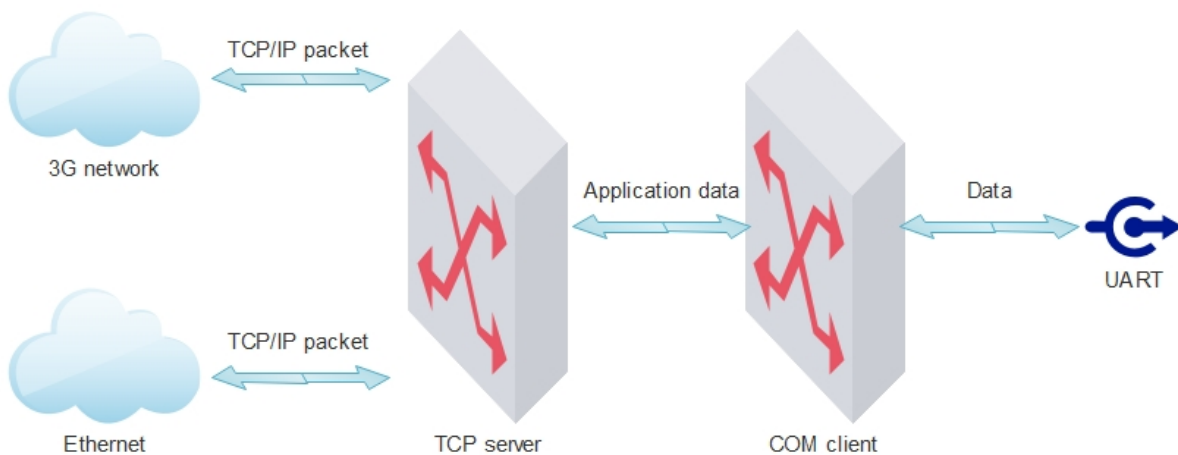
- 1 connection remote host 82.135.139.27 connected to 502 port (Modbus TCP/IP connection). Connection established through 3G/GPRS channel.
- 2 connection remote host 82.135.139.27 connected to 21 port (FTP). Connection established through 3G/GPRS channel.
- 3 connection remote host 82.135.139.27 connected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

Created with the Personal Edition of HelpNDoc: [Qt Help documentation made easy](#)

Routing TCP/IP - serial (request/answer)

Overview

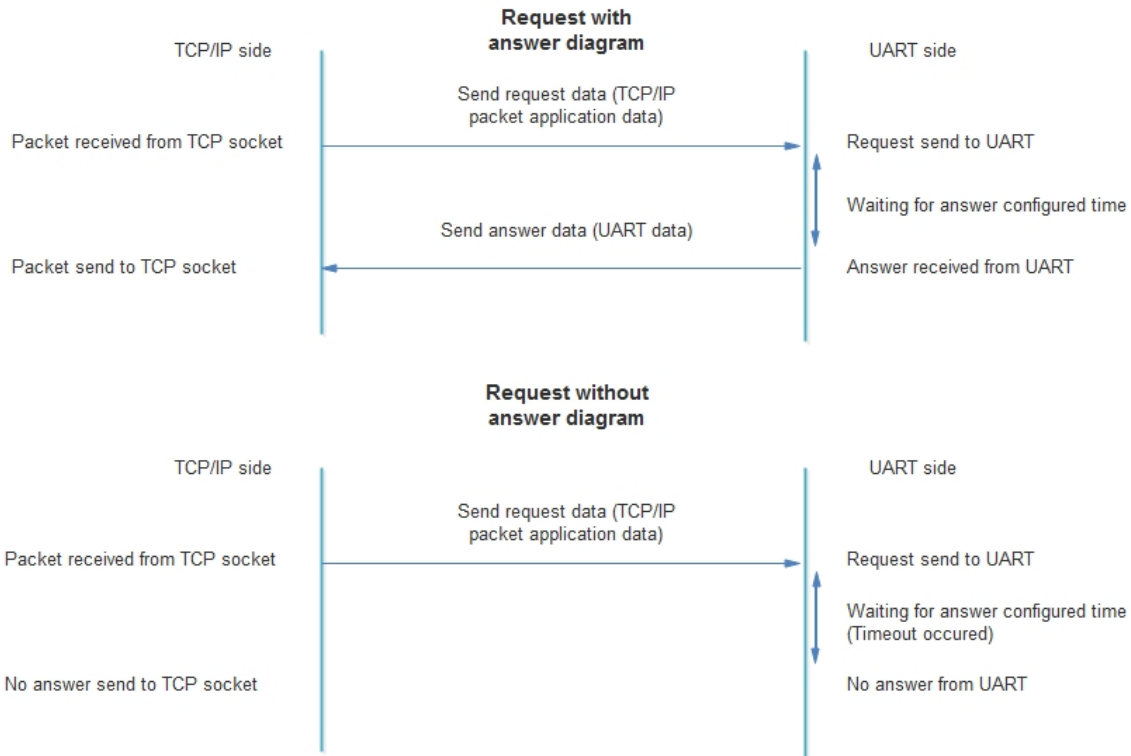
TCP server routes application data (request) from TCP network to serial port and return the serial data (answer) to TCP network. TCP server works through virtual service - COM client so other services can access UART too. TCP server resends application data to UART and back without data modifications.



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.

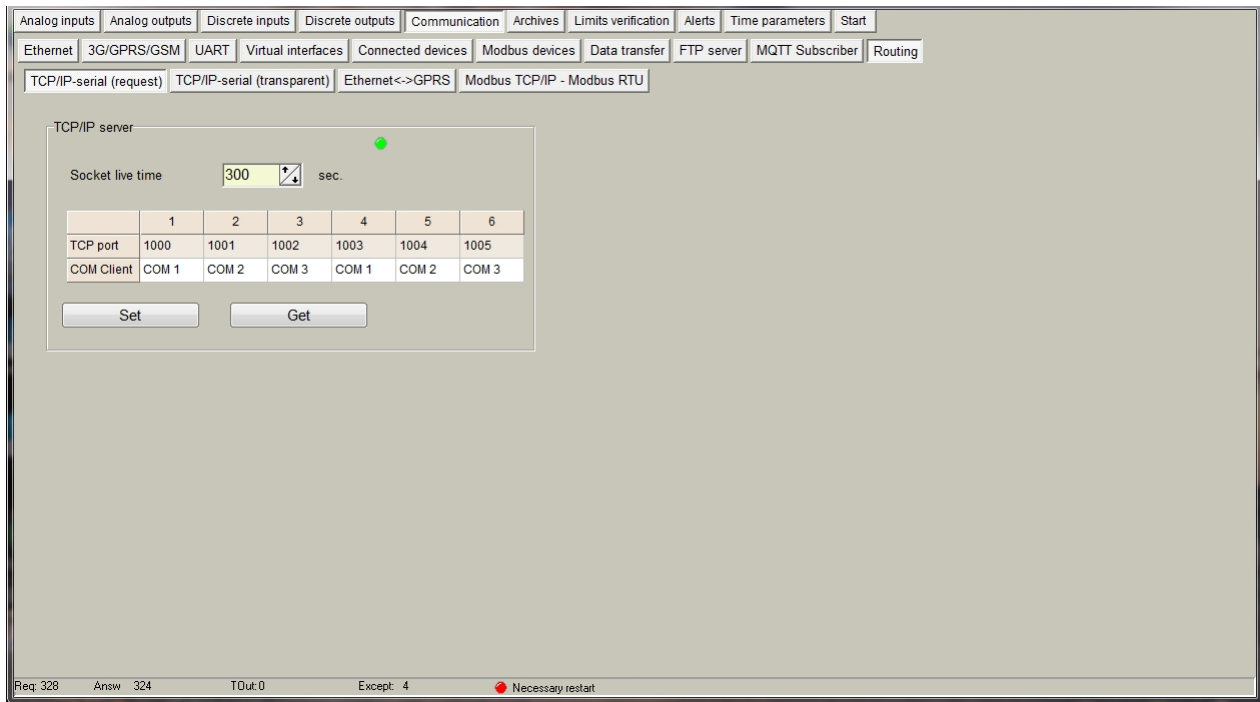


If answer data is received after timeout, it will be lost.

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.

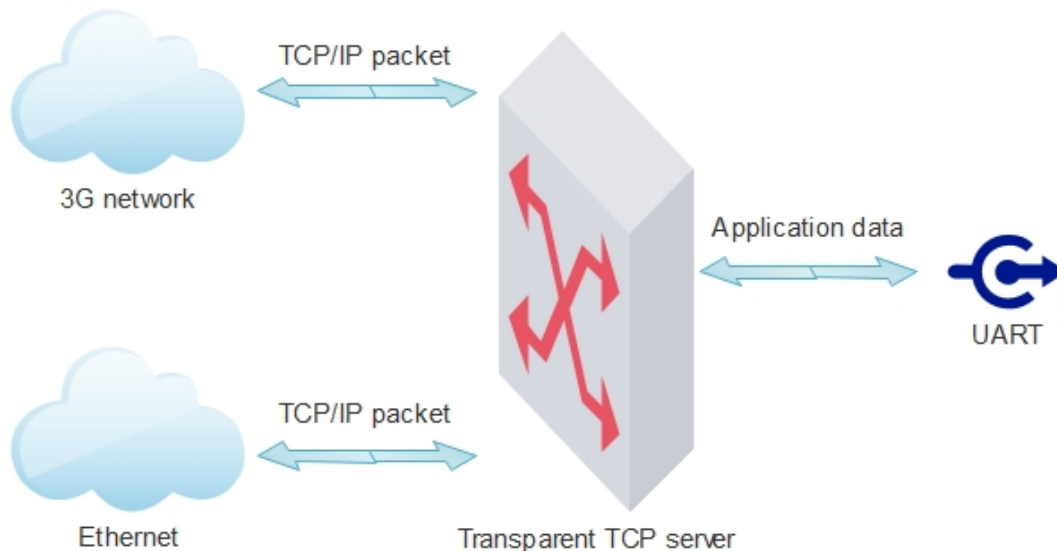


Created with the Personal Edition of HelpNDoc: [Easy EBook and documentation generator](#)

Routing TCP/IP - serial (transparent)

Overview

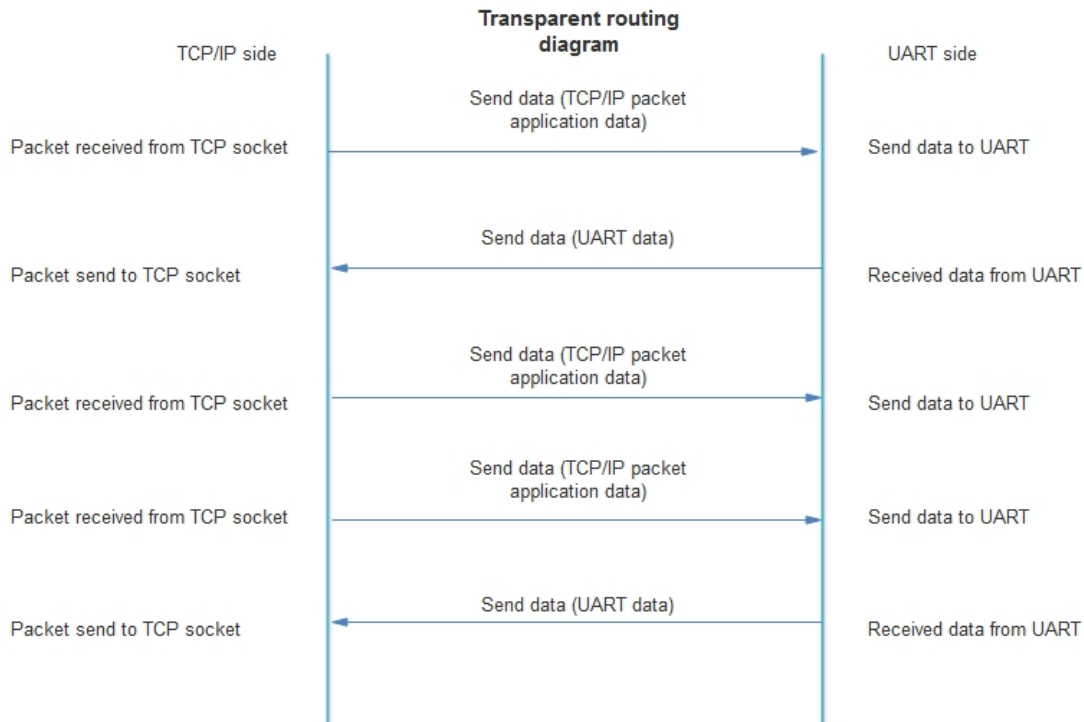
Transparent TCP server routes application data from TCP/IP network to UART and routes the serial data from UART to TCP/IP network. Difference from TCP server - Transparent TCP server works directly with UARTS and all data is transferred in both direction (no answer timeouts).



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 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.

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

TCP/IP-serial (transparent)

	1	2	3	4	5	6
Status	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Enabled	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
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

Set Get

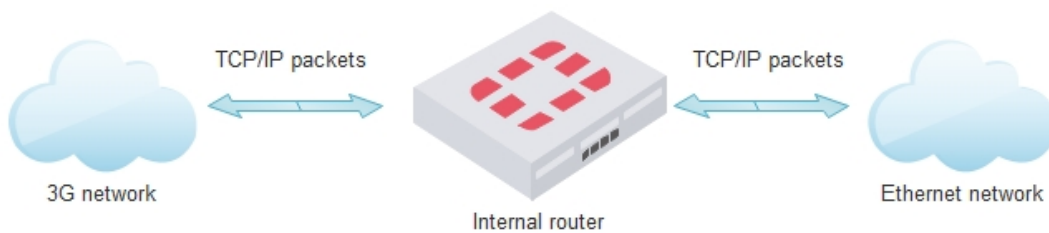
Req: 0 Answ: 0 TOut: 0 Except: 0

Created with the Personal Edition of HelpNDoc: [Free Web Help generator](#)

Router 3G/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

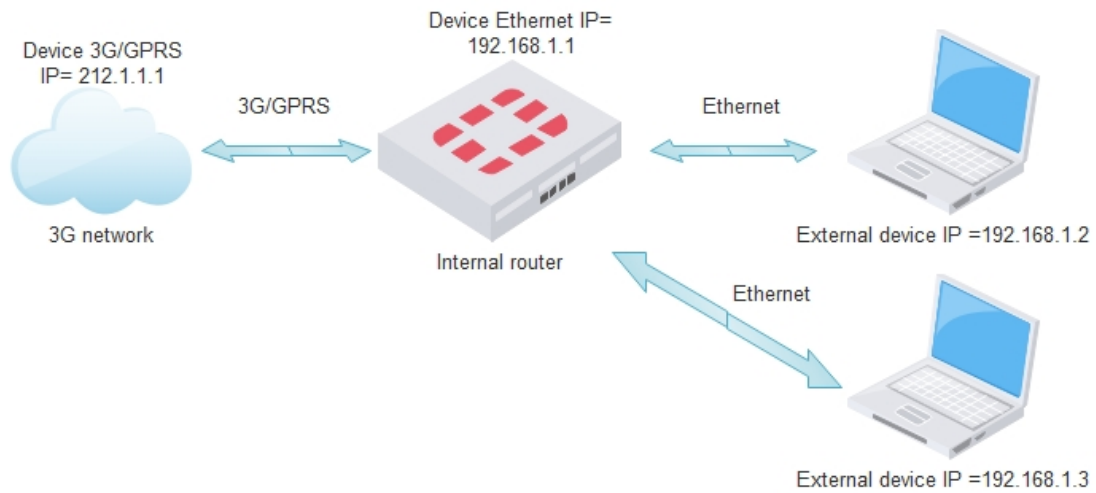
Router enabled:

	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

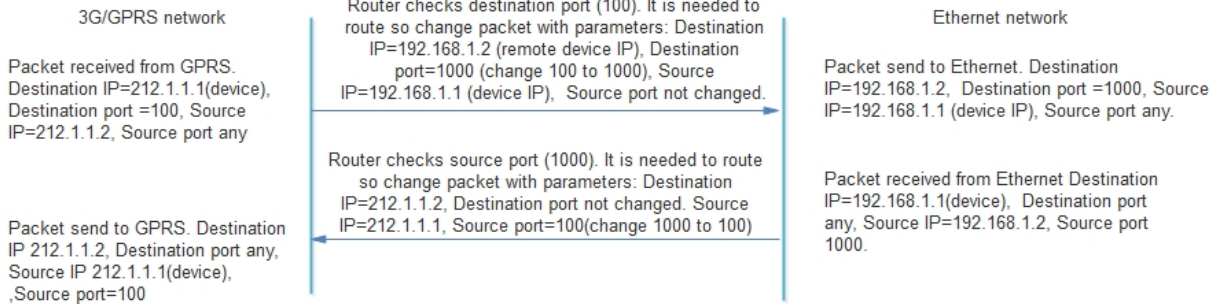
Example

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



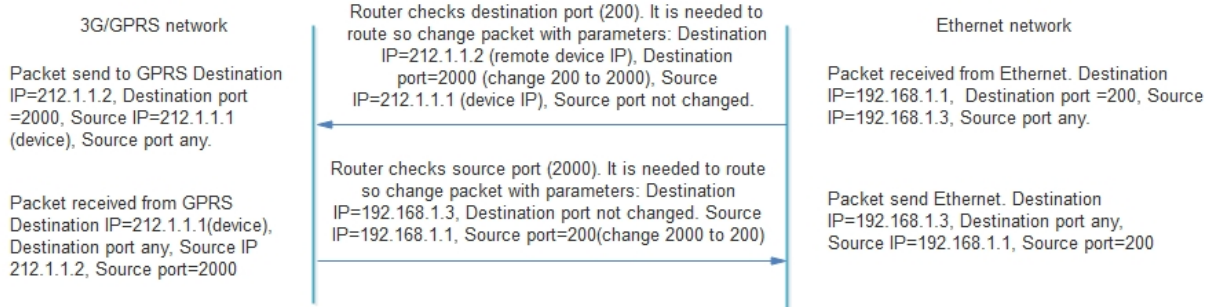
- First line, routing from GPRS to Ethernet, incoming port 100, outgoing port 1000, and outgoing Ethernet IP 192.168.1.2

Routing diagram



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

Routing diagram

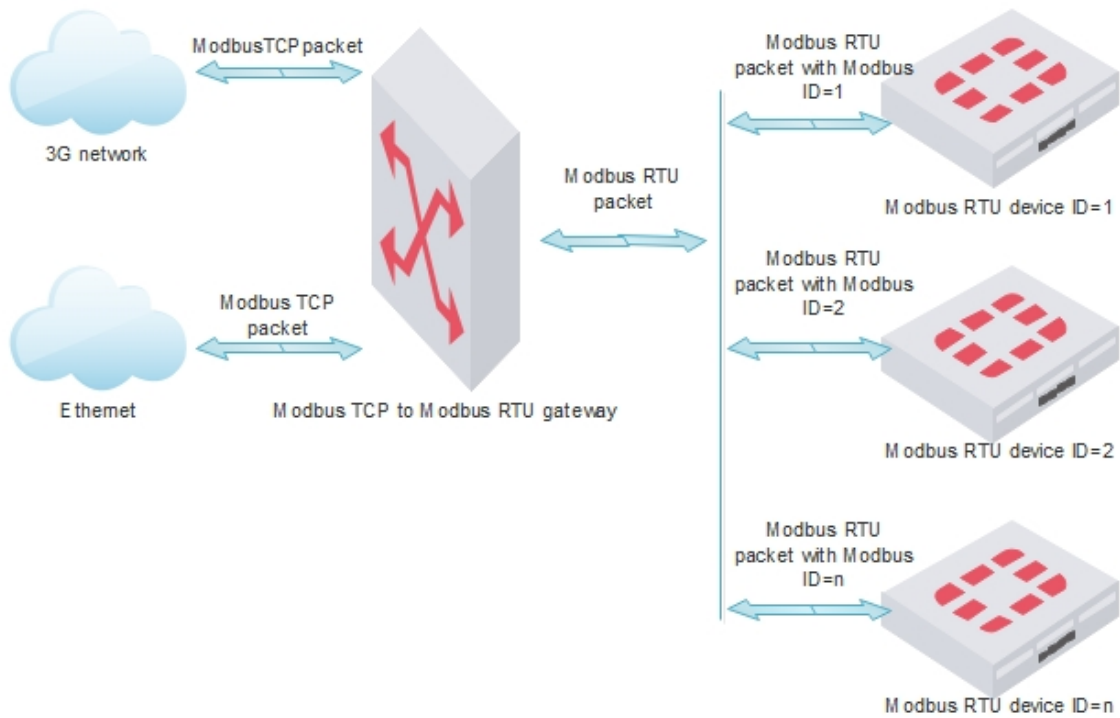


Gateway Modbus TCP<->Modbus RTU

Overview

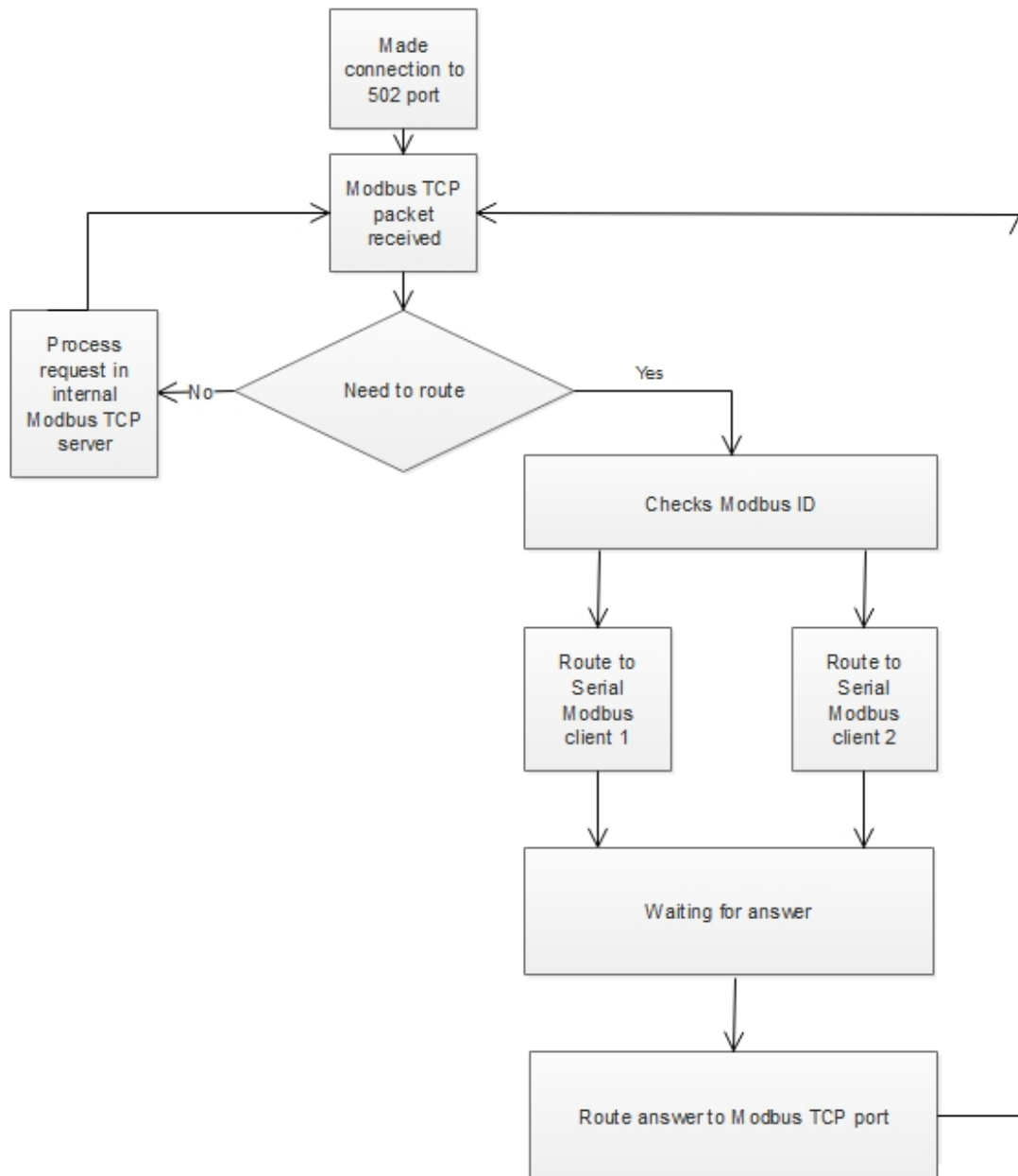
Modbus TCP<->Modbus RTU gateway is used directly to access Modbus RTU device connected to our device using Modbus TCP protocol. If route function is used, received Modbus TCP packet is changed to Modbus RTU packet and send to appropriate Serial Modbus client. In TCP network side device works as Modbus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

Standard network diagram:

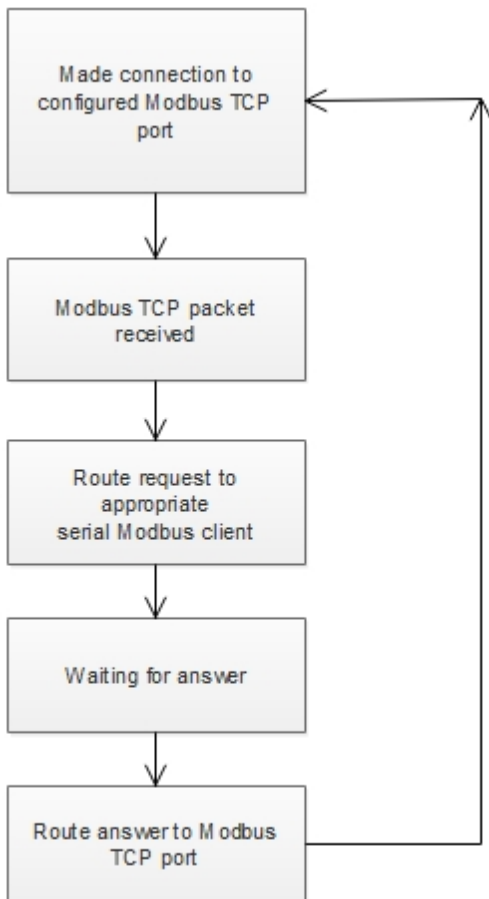


Where are two route modes:

- 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:



- 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:



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

Set Get

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

Set Get

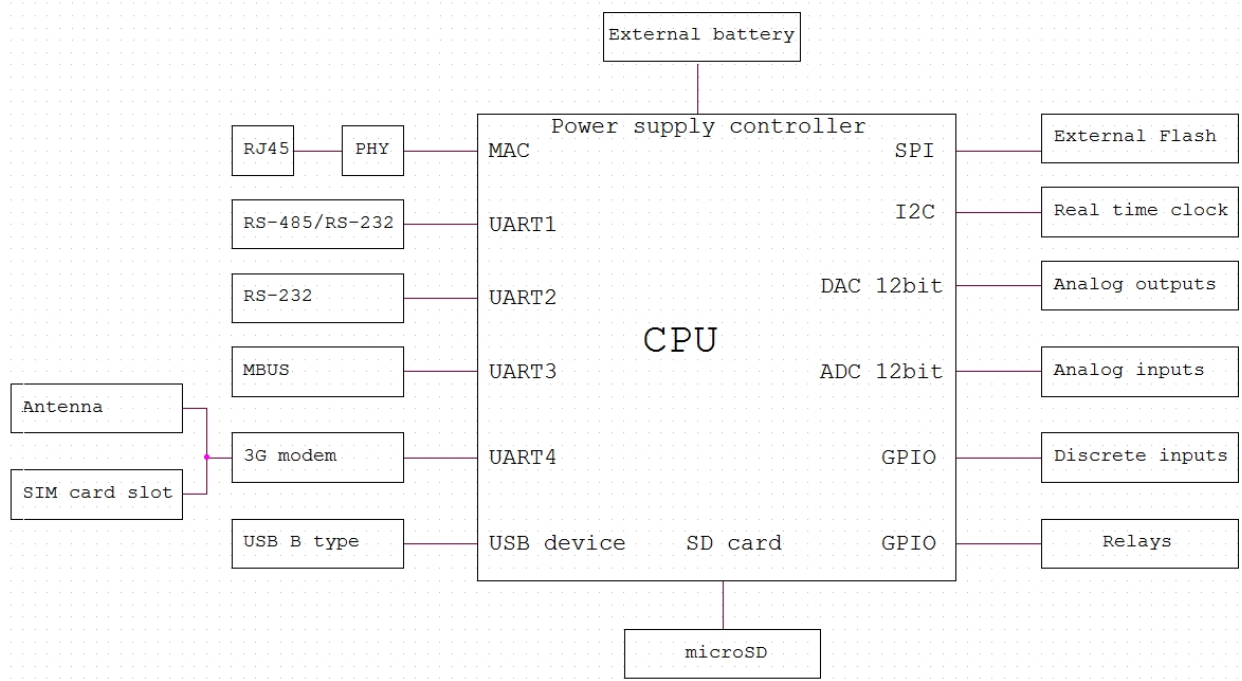
Req: 361 Answ: 357 TOut: 0 Except: 4

System

Specification

Main CPU	ARM Cortex-M4 32 bit with 2 Kbytes Cache running at 120 MHz
CPU Flash	512kB
CPU RAM	128kB
External Flash	8MB
microSD™ Slot	Up to 16 GB
OS	Real time operating system (FreeRTOS)
Clock	Real-time clock with battery backup

CPU board block diagram



Led indicators

Name	Label and type	Color	Function
Cycle	H11, two color	Green	100ms On, 100ms Off - Device is running, bootloader mode
			1000ms On, 1000ms Off - Device is running, normal work mode
			Always On or Always Off - Device is not working
MBUS line	H11, two color	Red	Always On - MBUS line is shorted
			Always Off - MBUS line is working normal

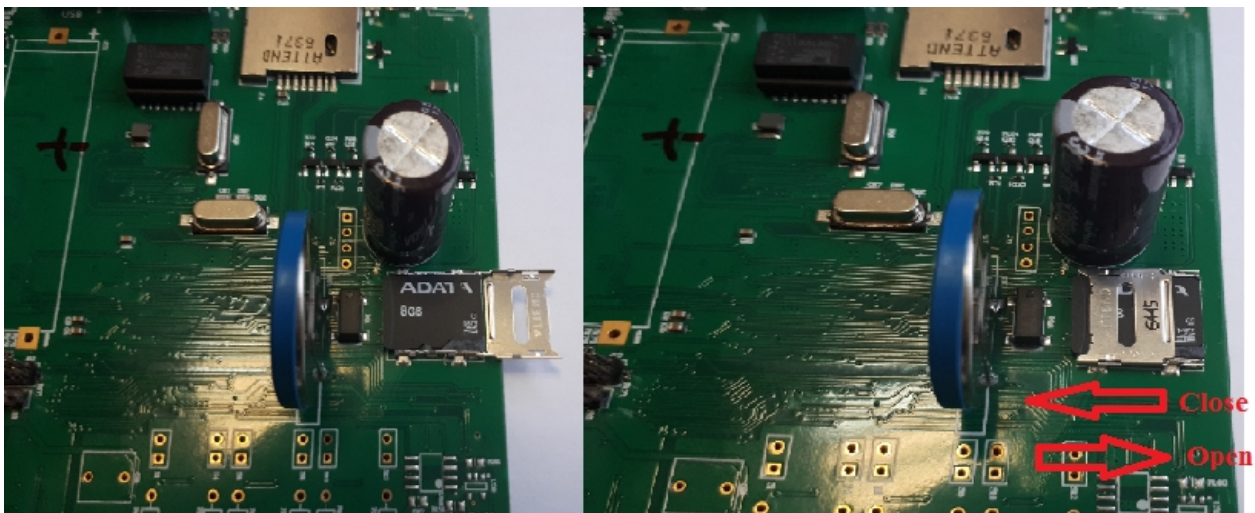
microSD

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 it's 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:

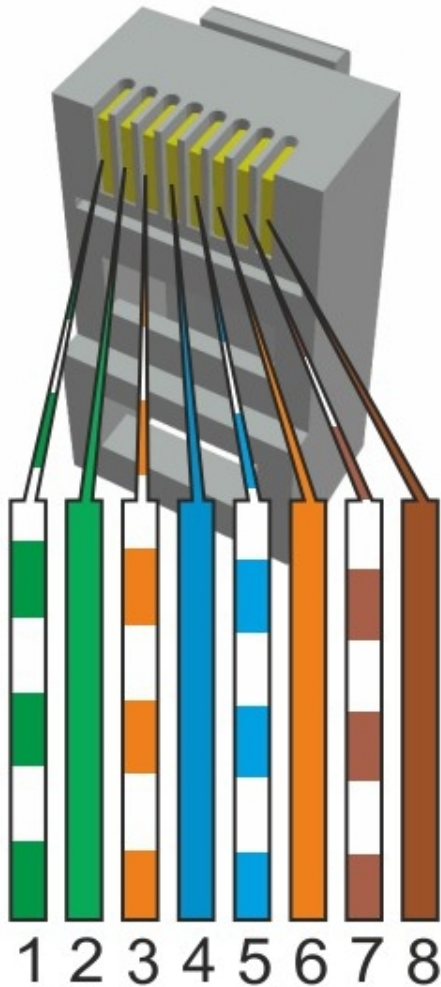
- Modbus TCP/IP server
- Modbus TCP/IP client
- FTP client
- FTP server
- MQTT client
- DNS client
- SNTP
- ICMP
- Request/Answer to UART channel
- Transparent to UART channel
- Router to GPRS/3G network

Ethernet connection and port pin-outs

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



Cable pin-out



Pin	Description	10base-T	100Base-T	1000Base-T
1	Transmit Data+ or BiDirectional	TX+	TX+	BI_DA+
2	Transmit Data- or BiDirectional	TX-	TX-	BI_DA-
3	Receive Data+ or BiDirectional	RX+	RX+	BI_DB+
4	Not connected or BiDirectional	n/c	n/c	BI_DC+
5	Not connected or BiDirectional	n/c	n/c	BI_DC-
6	Receive Data- or BiDirectional	RX-	RX-	BI_DB-
7	Not connected or BiDirectional	n/c	n/c	BI_DD+
8	Not connected or BiDirectional	n/c	n/c	BI_DD-

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 it's network settings.

- "MAC number" - device individual MAC address
- "IP address" - device IP address
- "Gateway IP" - gateway IP address
- "Mask" - network mask

Communication Archives Limits verification Alerts Time parameters Start

Ethernet UART Virtual interfaces Connected devices Modbus devices Modbus register grouping Data transfer FTP server MQTT Subscriber Routing

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

Created with the Personal Edition of HelpNDoc: [Easily create Qt Help files](#)

3G/GPRS configuration

Overview

Device has integrated 3g modem with standard size SIM card. There is 3 types of modems for Europe area SIM5360E ,America area SIM5360A and SIM5360J

Modem specification:

- SIM5360A :
 - Dual-Band UMTS/HSPA+ 850/1900MHz
 - Quad-Band GSM/GPRS/EDGE 850/900/1800/1900MHz
- SIM5360(J)E :
 - Dual-Band UMTS/HSPA+ 900/2100MHz
 - Quad-Band GSM/GPRS/EDGE 850/900/1800/1900MHz
- SIM5360J(D) :
 - Dual-Band UMTS/HSPA+ 800(850)/2100MHz
 - Quad-Band GSM/GPRS/EDGE 850/900/1800/1900MHz
- GPRS multi-slot class 12
- EDGE multi-slot class 12
- Output power
 - UMTS 850/1900: 0.25W
 - UMTS 900/2100: 0.25W

- GSM850/GSM900: 2W
- DCS1800/PCS1900: 1W

3G/GPRS interface is used for:

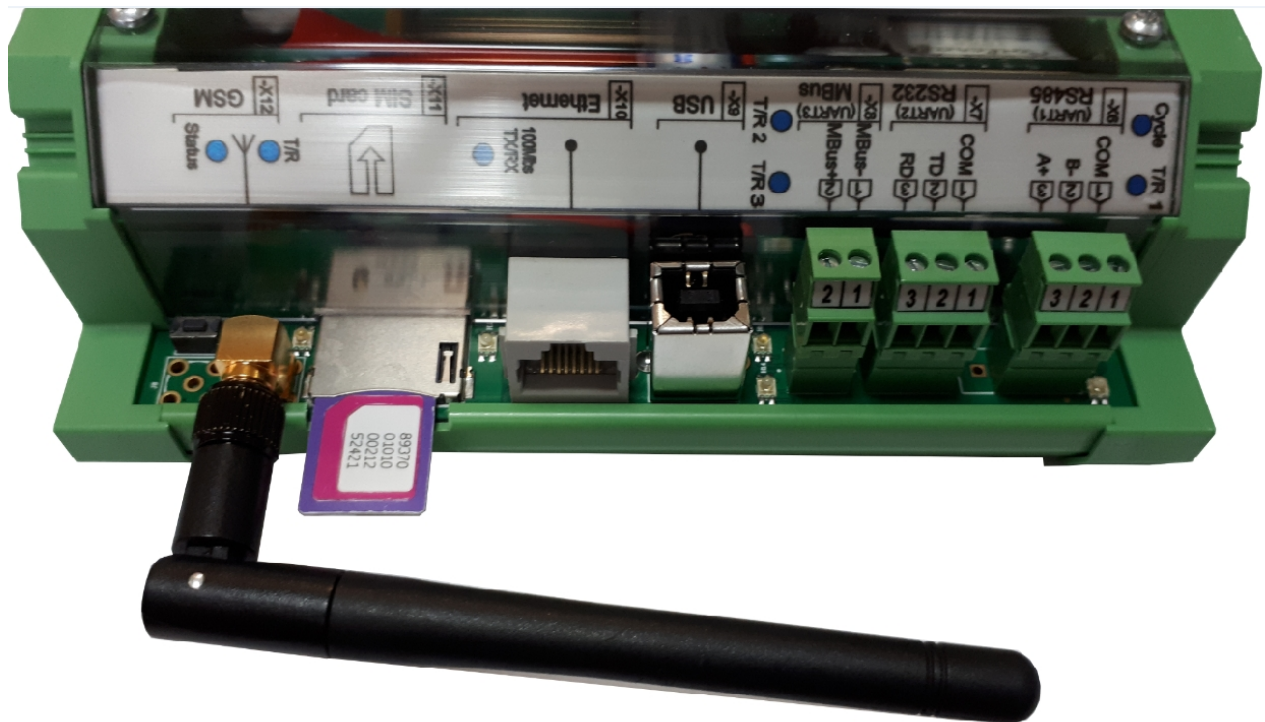
- Data transfer
- Events transfer
- Clock time synchronization
- Device configuration
- Firmware upgrade
- Etc...

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

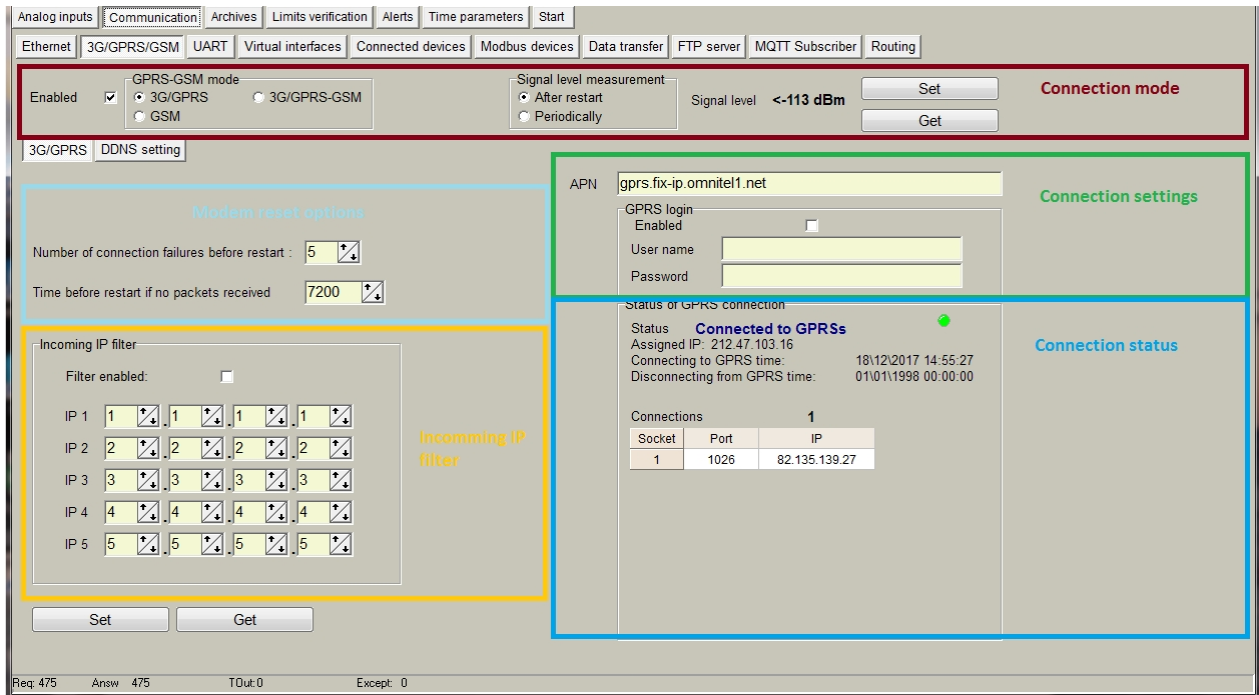
Name	Label and type	Color	Function
Status	H13, one color	Red	Always On - Searching Network/Call Connect 200ms ON, 200ms OFF - Connected to 3G/GPRS network 800ms ON, 800ms OFF - Registered network Off - Power off / Sleep
TX	H9, two color	Red	Blinks - Data is sending to Modem
RX	H9, two color	Green	Blinks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

3G/GPRS configuration

All configuration is made in "Communication/3G/GPRS/GSM" tab. Before configure 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.



Default settings

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

Created with the Personal Edition of HelpNDoc: [Generate Kindle eBooks with ease](#)

Serial ports

Overview

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

Port number	Available options	Description
UART 1	RS232 or RS485	Can be used as: <ul style="list-style-type: none"> ● Modbus slave

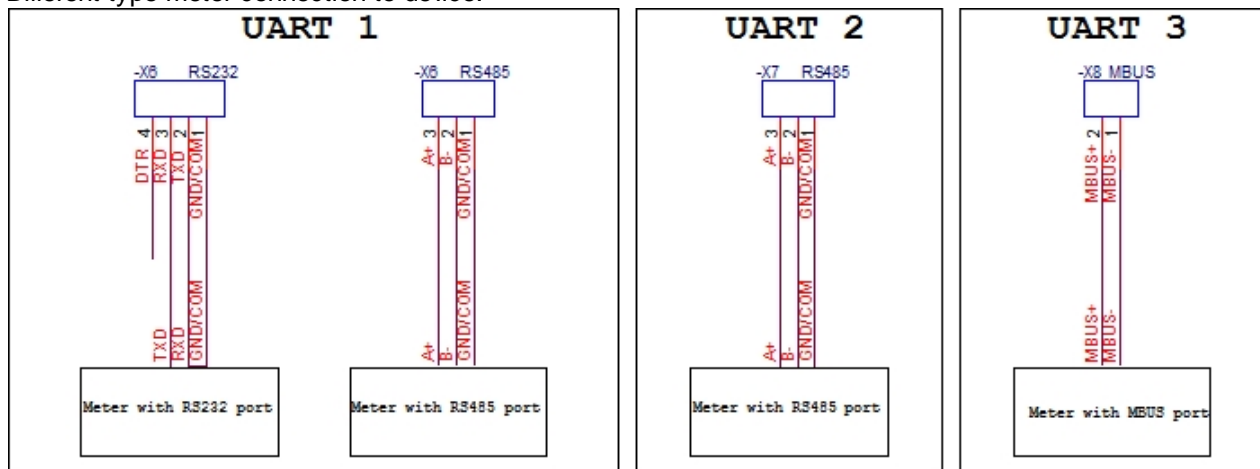
		<ul style="list-style-type: none"> • Modbus master • Mbus meter reading (with RS232/RS485 <-> MBUS converter) • Request/Answer channel • Transparent channel
UART 2	RS232	Can be used as: <ul style="list-style-type: none"> • Modbus slave • Modbus master • Mbus meter reading (with RS232 <-> MBUS converter) • Request/Answer channel • Transparent channel
UART 3	MBUS up to 20 devices	Can be used as: <ul style="list-style-type: none"> • MBUS meter reading • Request/Answer channel • Transparent channel

UART characteristics:

Port number	Supported baud rates	Supported parity	Supported data bits	Supported stop bits
UART 1	300 - 57600	Even, Odd, Mark, Space, None	5,6,7,8	1,2
UART 2	300 - 57600	Even, Odd, Mark, Space, None	5,6,7,8	1,2
UART 3	300 - 19200	Even, Odd, Mark, Space, None	5,6,7,8	1,2

Wiring diagrams

Different type meter connection to device:



Led indicators

Name	Label and type	Color	Function
------	----------------	-------	----------

T/R1	H7, two color	Green	Blinks - Data is receiving on UART1
T/R1	H7, two color	Red	Blinks - Data is sending on UART1
T/R2	H15, two color	Green	Blinks - Data is receiving on UART2
T/R2	H15, two color	Red	Blinks - Data is sending on UART2
T/R3	H14, two color	Green	Blinks - Data is receiving on UART3
T/R3	H14, two color	Red	Blinks - Data is sending on UART3
MBUS line	H11, two color	Red	Always On - MBUS line is shorted
			Always Off - MBUS line is working normal

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.

Analog inputs				Communication	Archives	Limits verification	Alerts	Time parameters	Star			
Ethernet				3G/GPRS/GSM	UART	Virtual interfaces	Connected devices	Modbus devices	Data transfer	FTP server	MQTT Subscriber	Routing
	UART 1	UART 2	UART 3									
Bode	<input type="radio"/> 300 <input type="radio"/> 600 <input type="radio"/> 1200 <input type="radio"/> 2400 <input type="radio"/> 4800 <input type="radio"/> 9600 <input checked="" type="radio"/> 19200 <input type="radio"/> 38400 <input type="radio"/> 57600	<input type="radio"/> 300 <input type="radio"/> 600 <input type="radio"/> 1200 <input type="radio"/> 2400 <input type="radio"/> 4800 <input type="radio"/> 9600 <input checked="" type="radio"/> 19200 <input type="radio"/> 38400 <input type="radio"/> 57600	<input type="radio"/> 330 <input type="radio"/> 630 <input type="radio"/> 1200 <input checked="" type="radio"/> 2400 <input type="radio"/> 4300 <input type="radio"/> 9500 <input type="radio"/> 19200 <input type="radio"/> 33400 <input type="radio"/> 57600									
Parity	<input checked="" type="radio"/> Even <input type="radio"/> Odd <input type="radio"/> Mark <input type="radio"/> Space <input type="radio"/> None	<input checked="" type="radio"/> Even <input type="radio"/> Odd <input type="radio"/> Mark <input type="radio"/> Space <input type="radio"/> None	<input checked="" type="radio"/> Even <input type="radio"/> Odd <input type="radio"/> Mark <input type="radio"/> Space <input type="radio"/> None									
Data bits	<input type="radio"/> 5 <input type="radio"/> 6 <input type="radio"/> 7 <input checked="" type="radio"/> 8	<input type="radio"/> 5 <input type="radio"/> 6 <input type="radio"/> 7 <input checked="" type="radio"/> 8	<input type="radio"/> 5 <input type="radio"/> 6 <input type="radio"/> 7 <input checked="" type="radio"/> 8									
Stop bits	<input checked="" type="radio"/> 1 <input type="radio"/> 2	<input checked="" type="radio"/> 1 <input type="radio"/> 2	<input checked="" type="radio"/> 1 <input type="radio"/> 2									
Packetization	<input checked="" type="radio"/> Time <input type="radio"/> Symbol <input type="radio"/> Length	<input checked="" type="radio"/> Time <input type="radio"/> Symbol <input type="radio"/> Length	<input checked="" type="radio"/> Time <input type="radio"/> Symbol <input type="radio"/> Length									
Packet time (msec.)	10	10	100									
Packet symbol(Hex)	00	00	00									
Packet byte count	1	100	1									
Mode	<input type="radio"/> Full duplex <input checked="" type="radio"/> Half duplex	<input checked="" type="radio"/> Full duplex <input type="radio"/> Half duplex	<input type="radio"/> Full duplex <input checked="" type="radio"/> Half duplex									
Destination of DTR	<input type="radio"/> Always OFF <input type="radio"/> Always ON <input checked="" type="radio"/> OFF when send <input type="radio"/> ON when send	<input checked="" type="radio"/> Always OFF <input type="radio"/> Always ON <input type="radio"/> OFF when send <input type="radio"/> ON when send	<input type="radio"/> Always OFF <input type="radio"/> Always ON <input checked="" type="radio"/> OFF when send <input type="radio"/> ON when send									
				Set		Get						
Req: 592 Answ: 341 TOut: 247 Except: 4												

Created with the Personal Edition of HelpNDoc: [Free CHM Help documentation generator](#)

Analog inputs

Overview

Six single-ended resistance, voltage or current analog inputs available in device. Each analog input can be used as:

- Current 0/4..20mA analog input
- Voltage 0..+5V analog input
- Voltage 0..+10V analog input
- Thermo resistor (PT100) analog input
- Thermo resistor (PT1000) analog input
- Resistance 0..10k analog input
- Voltage 0..10V or PT100 Jumper switchable analog input

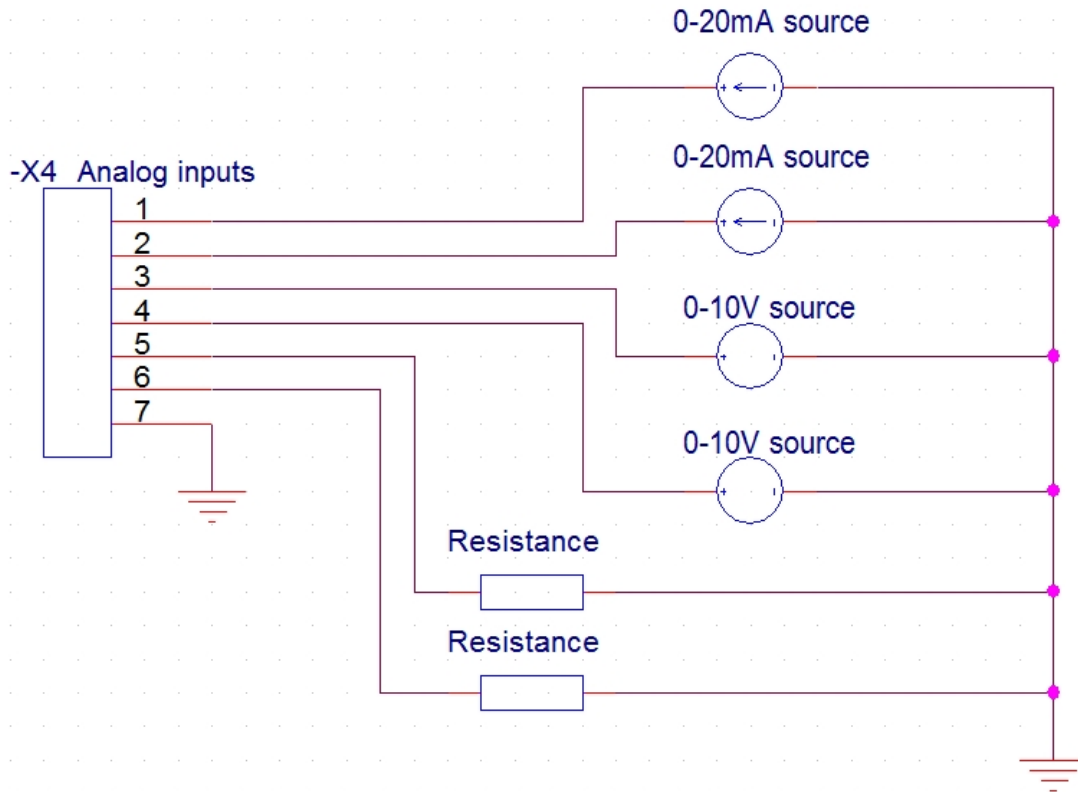
Purpose of analog input is indicated when you order device (check more "[Top label information](#)" topic about device code)

Specification

Description	Value
Inputs per device	Up to 6 single-ended
Input Voltage ranges	0..+5V 0..+10V
Input Current ranges	0..20mA 4..20mA
Input Resistance ranges	PT100 (80-250) PT1000 (850-1950) 0..10k
Accuracy	0.15% of full scale range
Resolution	12 bit Analog to Digital converter

Linearity	+/-1 LSB
Isolation	No isolation
Reading sample rate	10 times per second

Wiring diagram



Wiring diagram above shows how connect 3 different types of analog input sources (current, voltage and resistance). This device code can be like 701.010.xxx.x.x.112345.x.x.x.x.x

- 1-2 inputs are 0..20 mA current input
- 3 input is 0..5 V voltage input
- 4 input is 0..10 V voltage input
- 5 input is PT100 resistance input
- 6 input is PT1000 resistance input



Purpose of analog input is selected when you order device and can't be connected different types of analog sources to the same analog input.

Selectable analog inputs

Can be used Voltage 0..10V or PT100 Jumper switchable analog inputs. So to the same input can be connected Voltage 0..10 V source or PT100 sensor. Which input to use user selects with jumpers. Jumpers is located under the cover, so to access jumpers you need to remove upper cover of device.

Created with the Personal Edition of HelpNDoc: [Easy to use tool to create HTML Help files and Help web sites](#)

Analog outputs

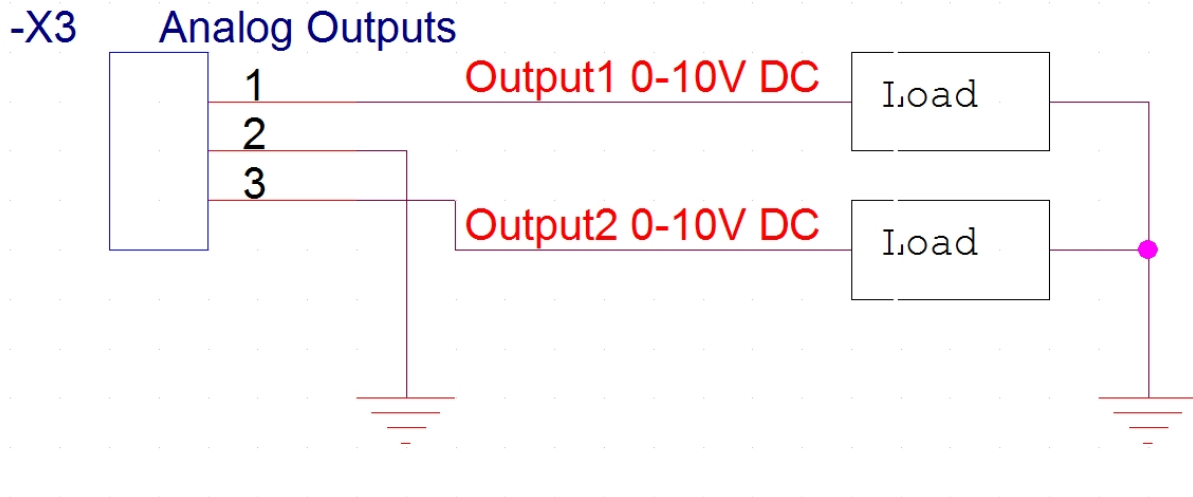
Overview

Two 0-10V analog outputs available in device. Device use 12bit DAC with operational amplifier output (Maximum load 5mA).

Voltage output is configurable and can be:

- 0 to 5 V DC, 5mA
- 0 to 10 V DC, 5mA

Wiring diagram



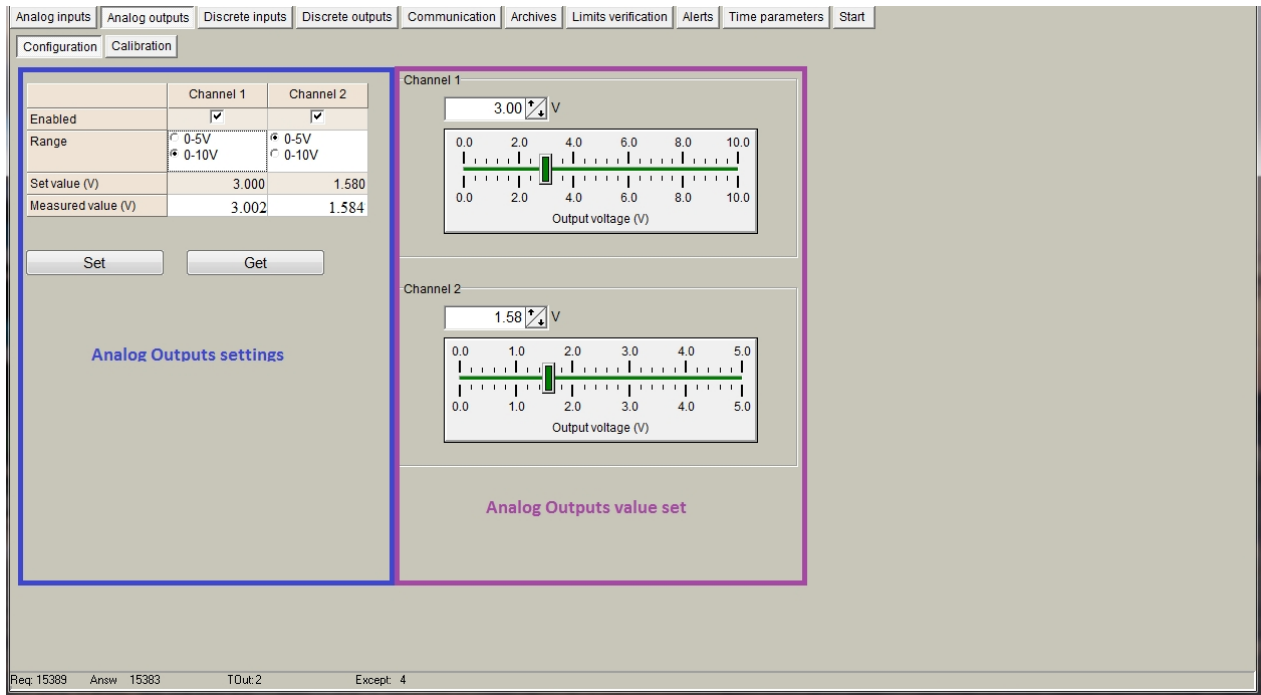
Analog Outputs configuration

Analog Outputs configuration is made in "Analog Outputs/Configuration" tab.

"Analog outputs settings" section:

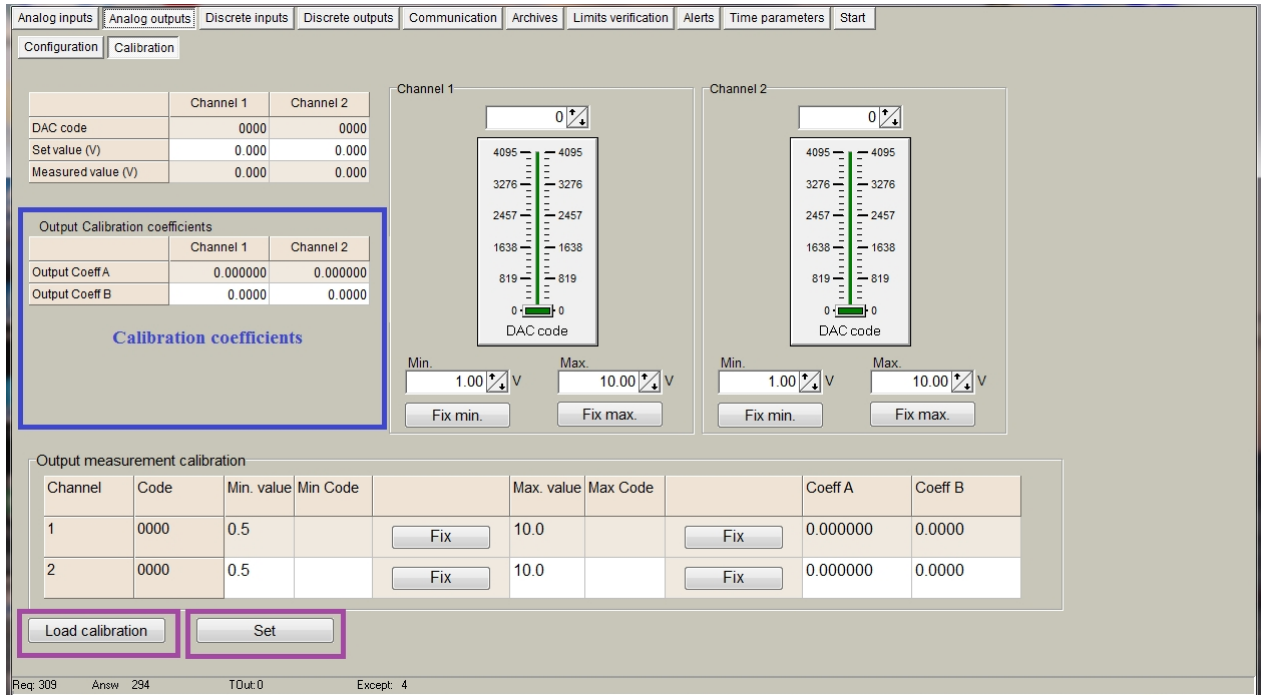
- "Enabled" check box - enable the appropriate analog output.
- "Range" selection - select output voltage range for the appropriate analog output.
- "Set value (V)" - user defined value.
- "Measured value (V)" - measured value at output.

In "Analog outputs value setting" section user can manually set value for each output . Use number box to enter value in V, or use slider to select output value.



Calibration

Manufacture makes all needed calibration of analog outputs. If for some reasons calibration values are lost ask manufacture for calibration file. Each device has its own calibration file and in file name present device serial number (filename kal_XXXXX.ini where "XXXXX" is serial number). Copy file "kal_XXXXX.ini" to configuration tool software folder. Run configuration tool and on "Analog Outputs/Calibration" tab click "Load calibration" button. If calibration file not present you will get warning message "File not find..." otherwise calibration coefficient will be loaded to calibration coefficients table. Press "Set" button to write calibration to device.



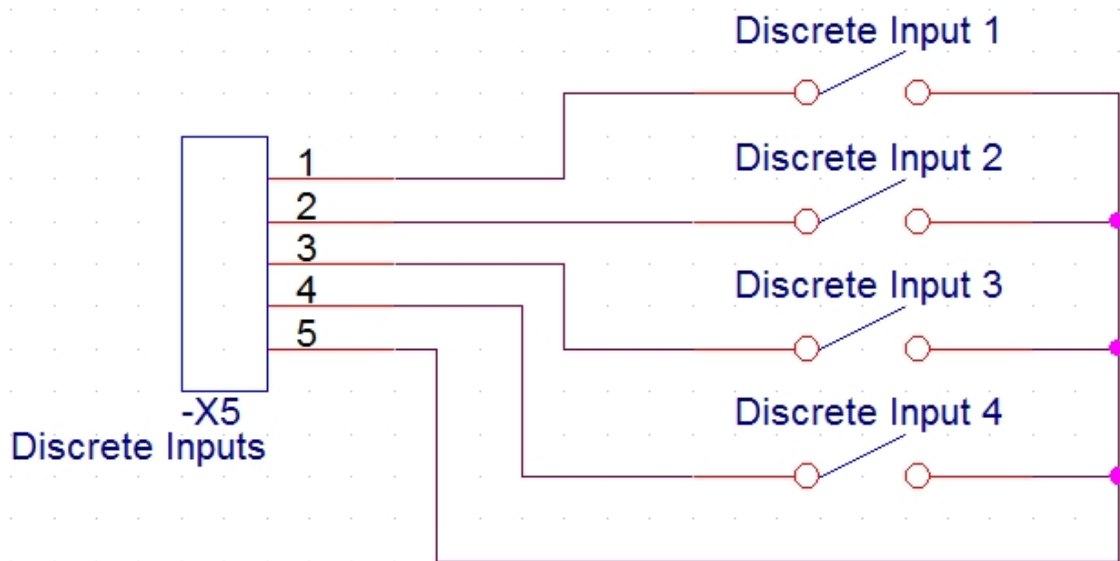
Overview

Four sink contact discrete inputs available in device. All inputs with the same common signal. Controller periodically tracks all discrete channels status and during change on each channel it stores that change with real time value. Also (if user defined) can be initiated report.

Discrete inputs purpose:

- Tracking of discrete signal status.
- Filtering from discrete signal fluctuations.
- Discrete signal change storage.
- "Alarm" status (events) fixation.
- Impulse counting.

Wiring diagram



Discrete Inputs configuration

Discrete inputs configuration is made in "Discrete Inputs" tab.

"Discrete inputs settings" section:

- "Enabled" check box - enables the appropriate discrete input
- "Current state" - shows discrete input current state. Example first discrete input is shorted (active), other - not shorted.
- "Impulse quantity" - the appropriate discrete input impulse counter.
- "Impulse multiplier" - the appropriate discrete input impulse counter multiplier.
- "Value" - multiplied "Impulse quantity"x"Impulse multiplier" value. This is used to convert impulse value to physical value.
- "De bounce time (ms)" - the appropriate discrete input filter time.
- "Mode" - the appropriate discrete input purpose"
 - "Discrete inp." - standard discrete input
 - "Impulse counter" - counts discrete input state changes, which state count configure with "State to archive" parameter.

- "Alarm signal" - discrete input with alarm function (creates alarm record in events archive). Alarm state configures with "Alarm state" parameter.
- "Alarm state" - the appropriate discrete input alarm state.
 - "Open" - generates alarm if discrete input not shorted.
 - "Close" - generates alarm if discrete input shorted.
 - "Both cases" - generates alarm on both states.
- "Enable archiving" - enables alarm archive of the appropriate discrete input changes.
- "State to archive" - discrete input state on which counts impulses
- "Set" button - write configuration to device.
- "Get" button - read configuration from device.

"Impulse counter value set" section:

- "Set" button - set the appropriate discrete input impulse counter value.
- "Clear" button - set all discrete inputs impulse counters values to 0.

The screenshot displays the configuration interface for discrete inputs. It is divided into two main sections: "Discrete inputs settings" and "Impulse counter value set".

Discrete inputs settings: This section is a table with columns for D In 1, D In 2, D In 3, and D In 4. The rows include:

	D In 1	D In 2	D In 3	D In 4
Enabled	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Current state				
Impulse quantity	0	0	0	0
Impulse multiplier	1.0000	1.0000	1.0000	1.0000
Value	0.000	0.000	0.000	0.000
Debounce time (ms)	100	100	100	100
Mode	<input checked="" type="radio"/> Discrete. inp. <input type="radio"/> Impulse counter <input type="radio"/> Alarm signal	<input checked="" type="radio"/> Discrete. inp. <input type="radio"/> Impulse counter <input type="radio"/> Alarm signal	<input checked="" type="radio"/> Discrete. inp. <input type="radio"/> Impulse counter <input type="radio"/> Alarm signal	<input checked="" type="radio"/> Discrete. inp. <input type="radio"/> Impulse counter <input type="radio"/> Alarm signal
Alarm state	<input checked="" type="radio"/> Open <input type="radio"/> Close <input type="radio"/> Both cases	<input checked="" type="radio"/> Open <input type="radio"/> Close <input type="radio"/> Both cases	<input checked="" type="radio"/> Open <input type="radio"/> Close <input type="radio"/> Both cases	<input checked="" type="radio"/> Open <input type="radio"/> Close <input type="radio"/> Both cases
Enable archiving	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
State to archive	<input checked="" type="radio"/> Open <input type="radio"/> Close <input type="radio"/> Both cases	<input checked="" type="radio"/> Open <input type="radio"/> Close <input type="radio"/> Both cases	<input checked="" type="radio"/> Open <input type="radio"/> Close <input type="radio"/> Both cases	<input checked="" type="radio"/> Open <input type="radio"/> Close <input type="radio"/> Both cases

Below the table are "Set" and "Get" buttons.

Impulse counter value set: This section is titled "Set the initial amount of impulses" and "Impulse counter value set". It contains a table for setting impulse quantities:

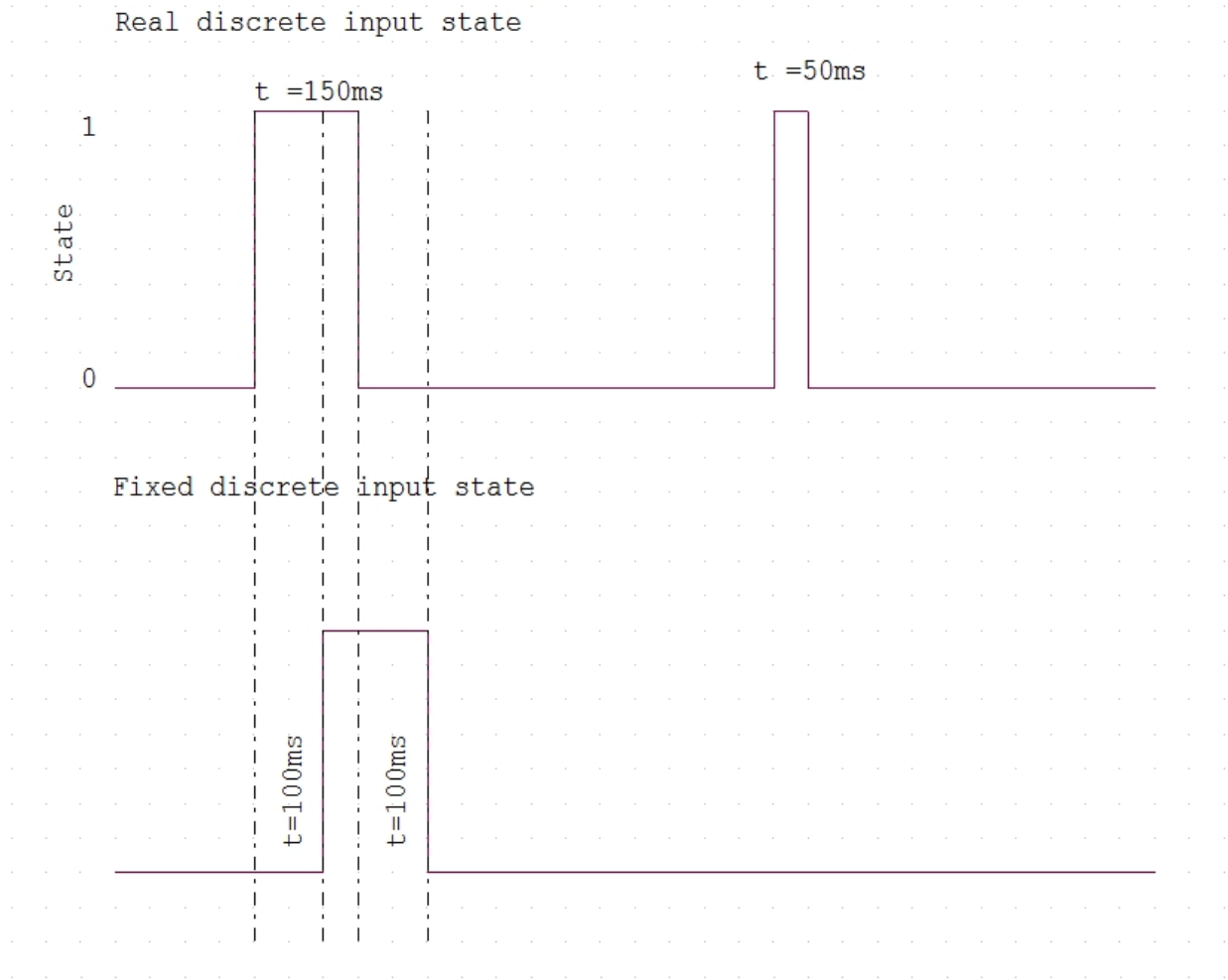
	D In 1	D In 2	D In 3	D In 4
Impuls quantity				
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	<input type="button" value="Set"/>	<input type="button" value="Set"/>	<input type="button" value="Set"/>	<input type="button" value="Set"/>

A "Clear" button is located to the right of this table.

At the bottom of the interface, there is a status bar with the text: "Req: 88 Answ: 88 TOut: 0 Except: 0".

Example

On picture below we see 2 signals, first is real signal on discrete input pins and second is filtered signal. Configured "Debounce time" is 100 ms. First impulse is fixed after 100 ms because "Debounce time" is 100ms. 50ms impulse is not fixed because its duration is smaller than "Debounce time"



Created with the Personal Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

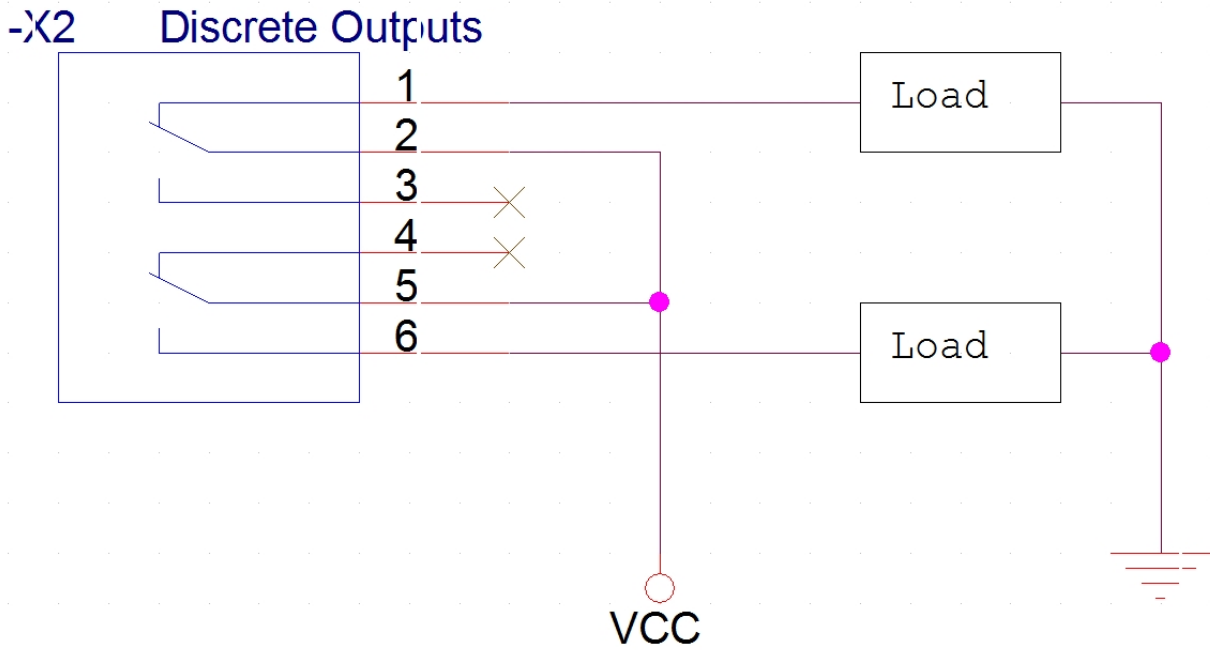
Discrete outputs

Overview

Two relay outputs available in device. Every output has two connections NC and NO and can drive up to 3A on 250VAC. Every output save its position after power OFF/ON and can be used driven directly outputs (controlled setting its values via configuration tool) or can be driven with week sheduller.

Wiring diagram

First "Load" will be powered when DI1 = 0, second when DI2 = 1.



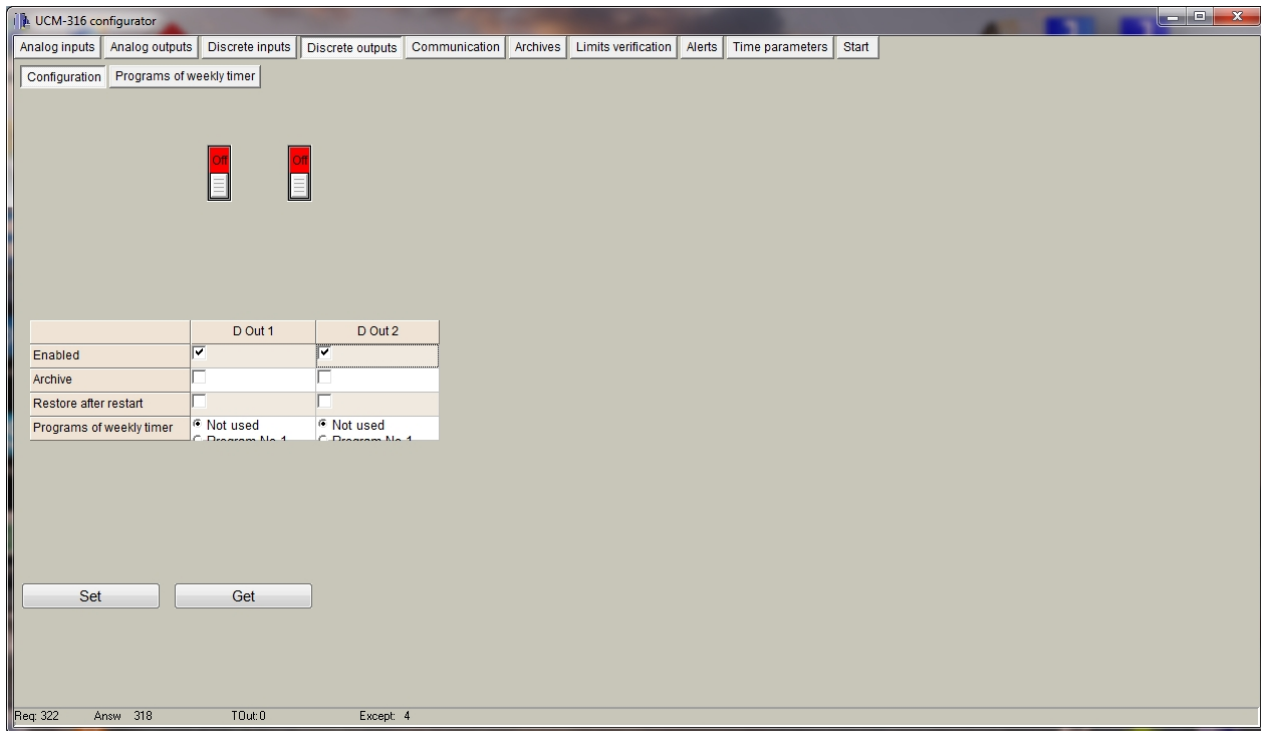
Led indicators

Name	Label and type	Color	Function
Relay 1	H2, one color	Red	On - Relay is in On state Off - Relay is in Off state
Relay 2	H1, one color	Red	On - Relay is in On state Off - Relay is in Off state

Discrete Outputs configuration

Discrete inputs configuration is made in "Discrete Outputs/Configuration" tab.

- "Enabled" check box - enable the appropriate discrete output.
- "Archive" check box - option not used in this device.
- "Restore after restart" check box - enable the restore of discrete output state after restart. If disabled, discrete output state is set to "Off" after restart.
- "Programs of weekly timer" - select week sheduler for the appropriate discrete output. Check "Week sheduler" section for more information.
 - "Not used" - sheduler is disabled. Discrete output state changes only by setting it manually.
 - "Program No 1" - Discrete output state changes depending on configured sheduler (1)
 - "Program No 2" - Discrete output state changes depending on configured sheduler (2)
- "Set" button - write configuration to device.
- "Get" button - read configuration from device.



Week sheduler

Every discrete output can be associated with appropriate sheduler program, so its output state will be changed automatically depending on sheduler configured time points and states. It use weekly sheduler and can be programmed up to 64 points for every program. Device checks sheduler state every 1 minute and puts output to configured state.

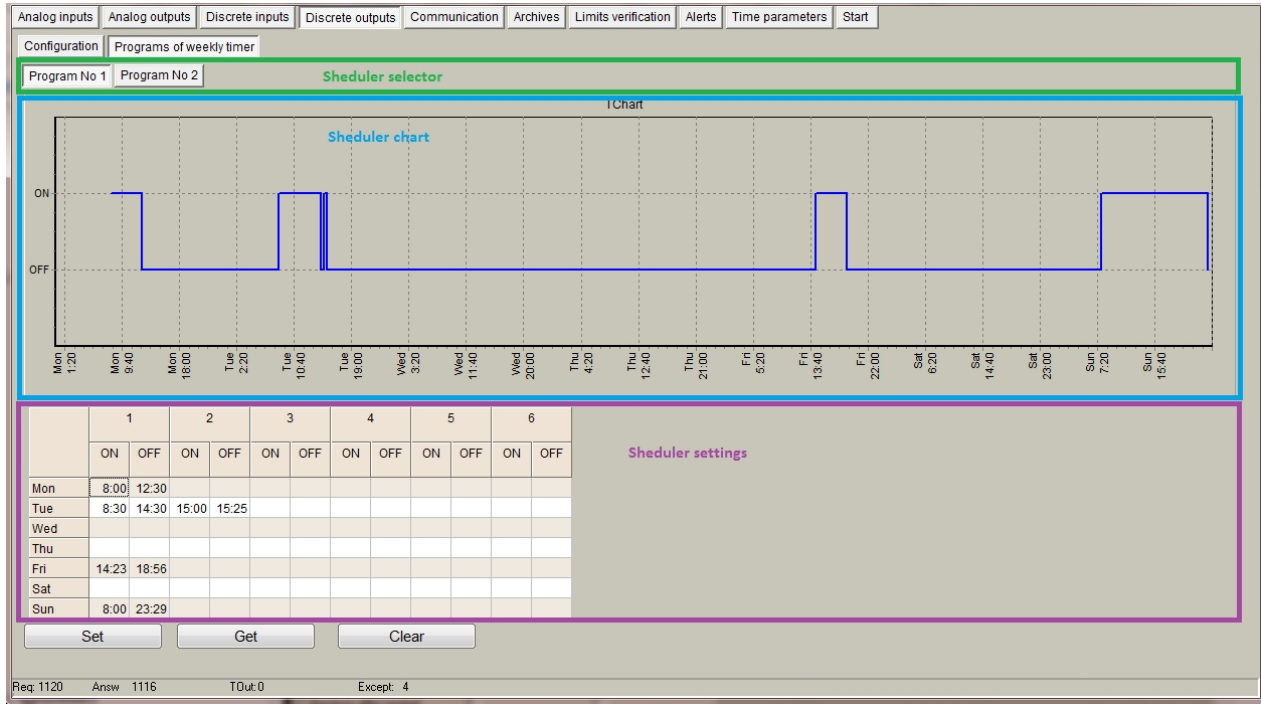
"Sheduler selector" section:

- "Program No 1" - configure sheduler 1.
- "Program No 2" - configure sheduler 2.

"Sheduler chart" section shows configured time points in graph view.

"Sheduler settings" section allow configure time points and output states. For every week day it is possible configure up to 6 time points for On and Off states, but maximum time points for all week can be up to 64.

- "Set" button - write configuration to device.
- "Get" button - read configuration from device.
- "Clear" button - clear all time points.



Example

In picture above we see configured 10 time points for all week days. On Monday 8:00 output state will be set to On and leaves until 12:30 (set output state to Off). On Tuesday 8:30 state will be set to On and so on. From Sunday 23:29 till Monday 8:00 state will be Off.



If user manually change state of output, sheduller program return it to configured state on next check (Device checks sheduller state every 1 minute and puts output to configured state).

Created with the Personal Edition of HelpNDoc: [Easily create CHM Help documents](#)

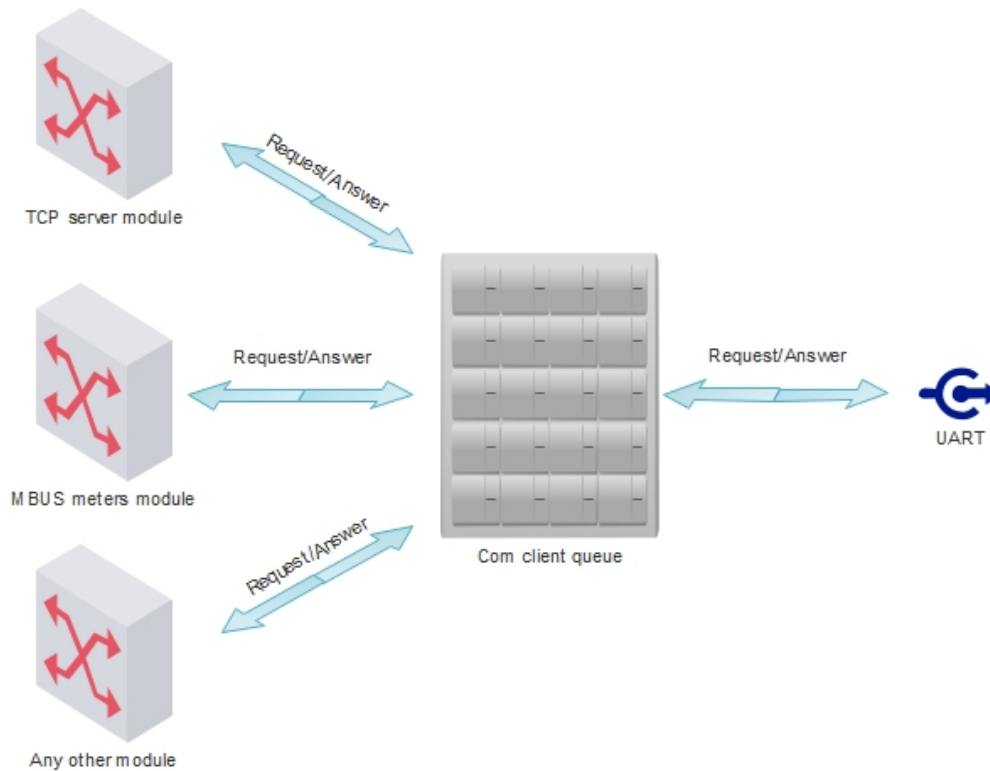
Virtual interfaces

Created with the Personal Edition of HelpNDoc: [Easily create PDF Help documents](#)

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.



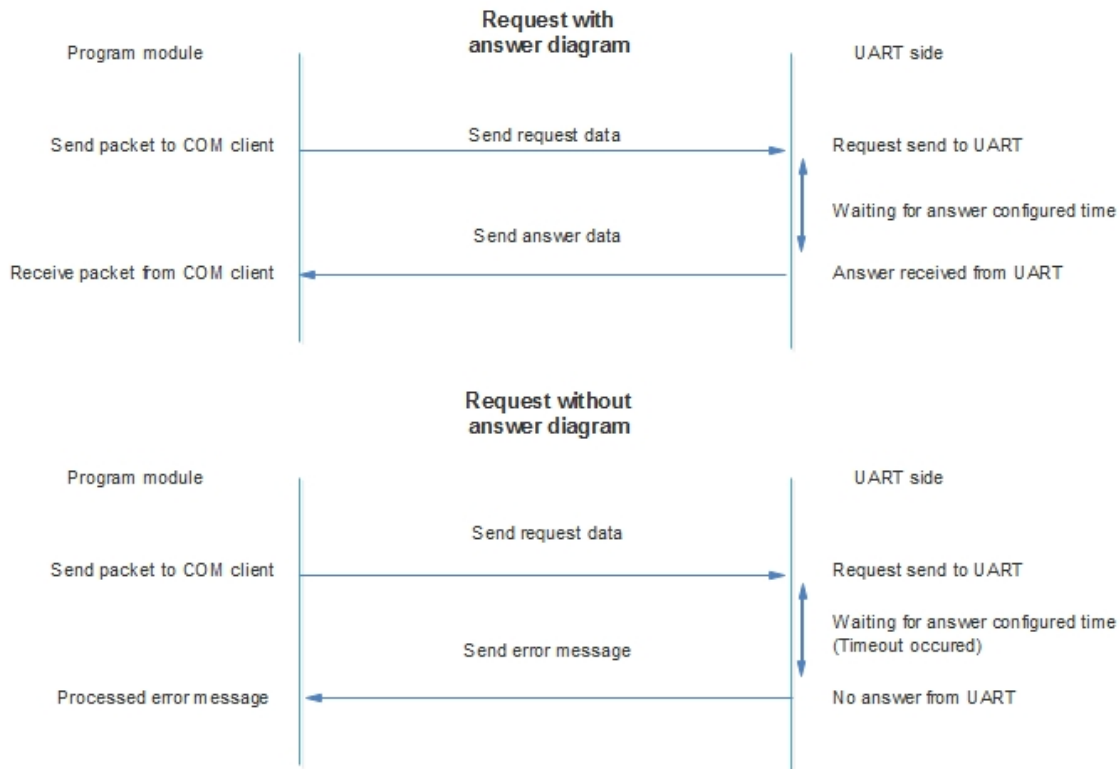
COM client is used with this modules:

- 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).



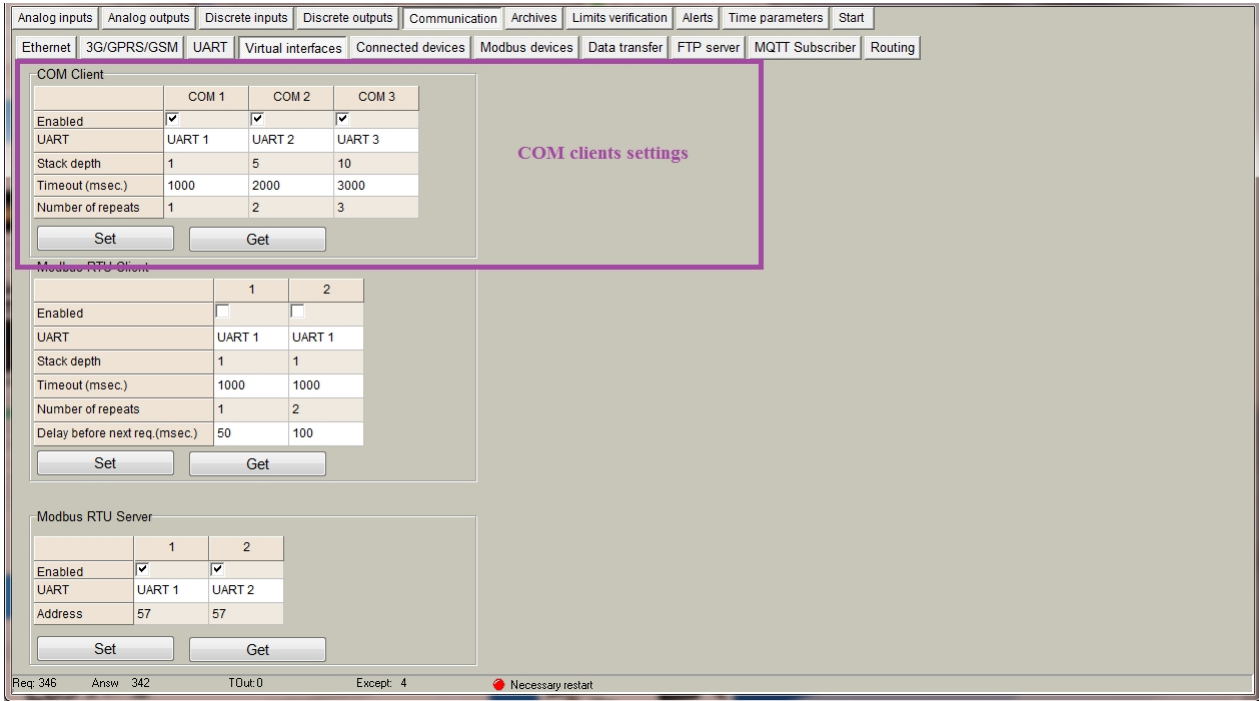
If answer is received after COM client timeout, this data is lost.



Virtual COM clients 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

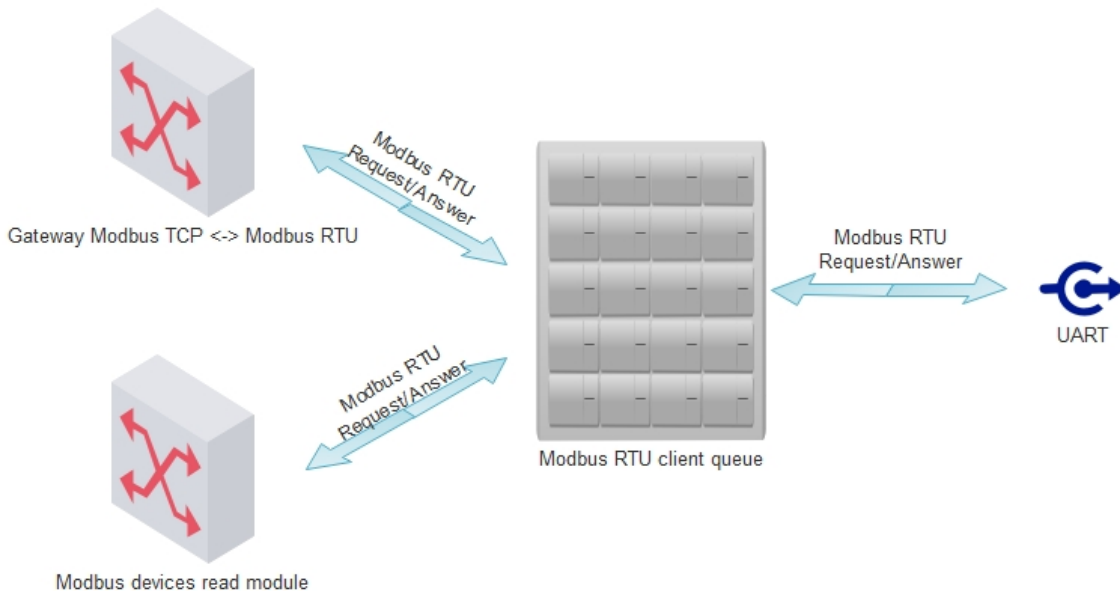


Created with the Personal Edition of HelpNDoc: [Easily create Help documents](#)

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.

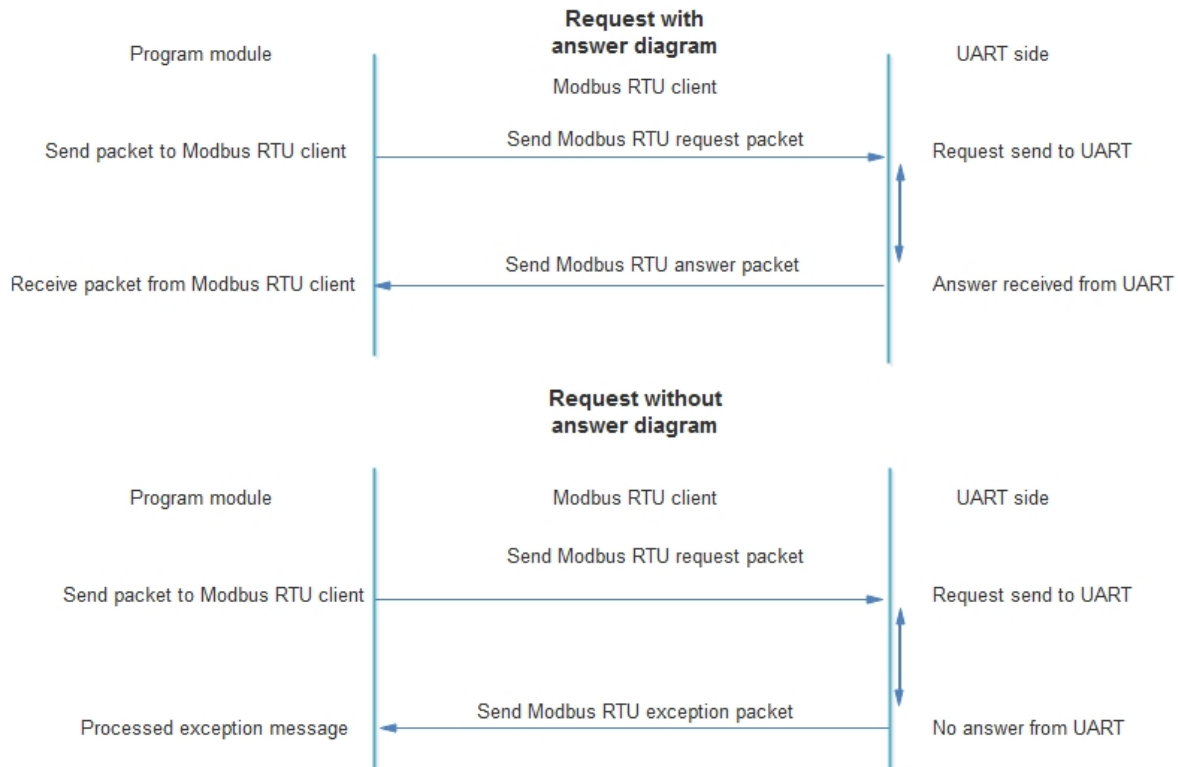


Modbus RTU client is used with this modules:

- 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.

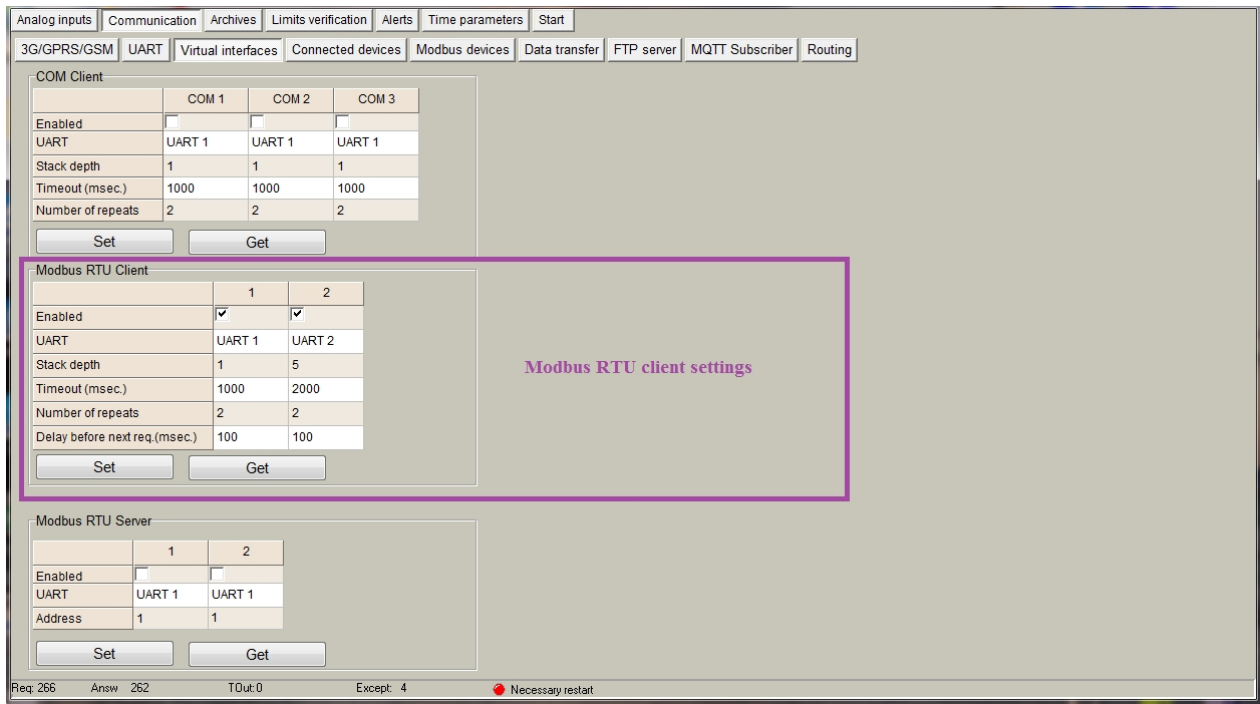


If answer is received after Modbus RTU client timeout, this data is lost.

Modbus RTU clients 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



Created with the Personal Edition of HelpNDoc: [Produce online help for Qt applications](#)

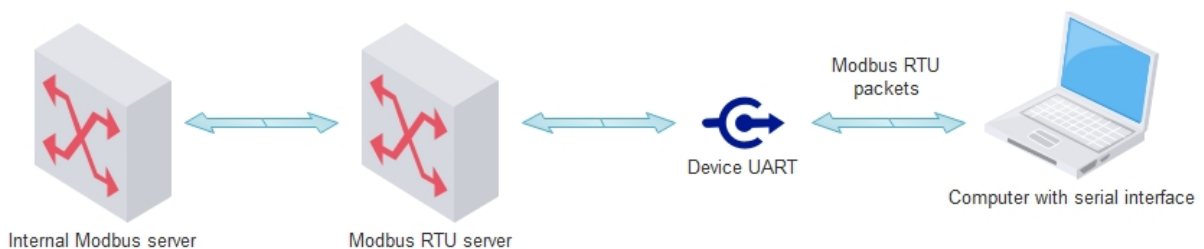
Modbus RTU servers

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).

COM Client			
	COM 1	COM 2	COM 3
Enabled	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UART	UART 1	UART 1	UART 1
Stack depth	1	1	1
Timeout (msec.)	1000	1000	1000
Number of repeats	2	2	2
<input type="button" value="Set"/> <input type="button" value="Get"/>			

Modbus RTU Client		
	1	2
Enabled	<input type="checkbox"/>	<input type="checkbox"/>
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
<input type="button" value="Set"/> <input type="button" value="Get"/>		

Modbus RTU Server		
	1	2
Enabled	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
UART	UART 1	UART 2
Address	20	20
<input type="button" value="Set"/> <input type="button" value="Get"/>		

Modbus RTU server settings

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

Created with the Personal Edition of HelpNDoc: [Produce electronic books easily](#)

Meters

Created with the Personal Edition of HelpNDoc: [Create help files for the Qt Help Framework](#)

Mbus 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...

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

Ethernet | **UART** | Virtual interfaces | Connected devices | Modbus devices | Modbus register grouping | Data transfer | FTP server | MQTT Subscriber

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

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

Ethernet | **UART** | Virtual interfaces | Connected devices | Modbus devices | Modbus register grouping | Data transfer | FTP server | MQTT Subscriber

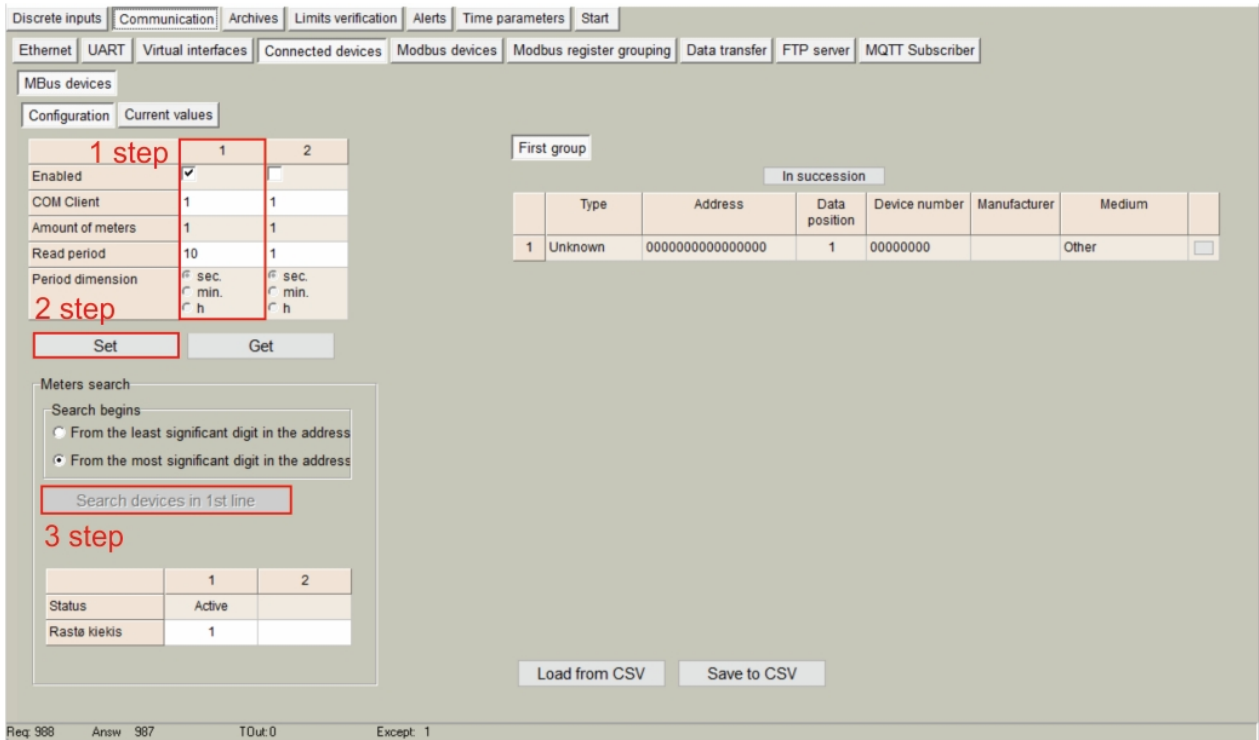
COM Client

	COM 1	COM 2	COM 3
Enabled	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
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) this 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.

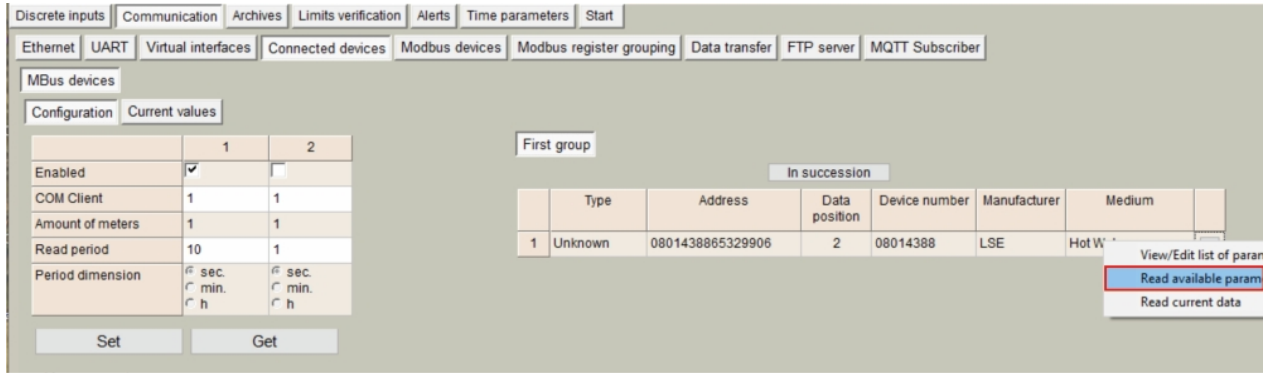
NOTE: If you have connected a lot meters, their search can take a while.



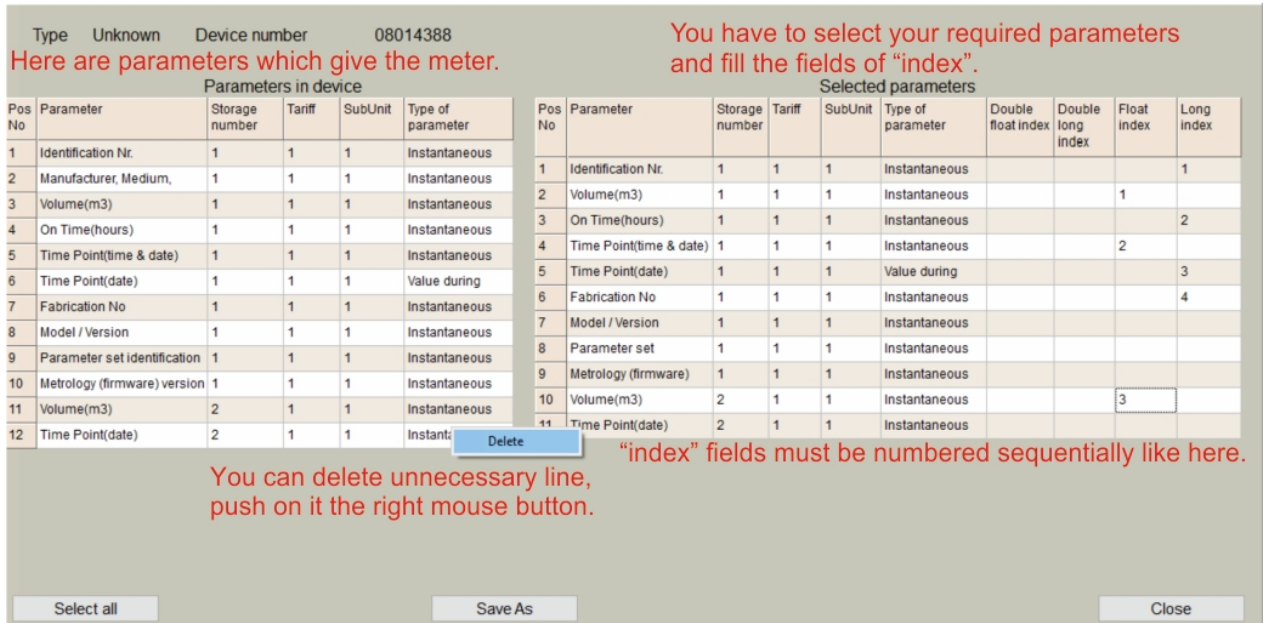
When the search is end you will see this window. There you will see new found M-Bus devices. You have to include newly found devices and send it to controller. How to do it, you can see at the picture below.



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.



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 needed 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 the list with the mouse right button.



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

Type Unknown Device number 08014388

Parameters in device						Selected parameters									
Pos No	Parameter	Storage number	Tariff	SubUnit	Type of parameter	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	Identification Nr.	1	1	1	Instantaneous				1
2	Manufacturer, Medium,	1	1	1	Instantaneous	2	Volume(m3)	1	1	1	Instantaneous			1	
3	Volume(m3)	1	1	1	Instantaneous	3	On Time(hours)	1	1	1	Instantaneous				2
4	On Time(hours)	1	1	1	Instantaneous	4	Time Point(time & date)	1	1	1	Instantaneous			2	
5	Time Point(time & date)	1	1	1	Instantaneous	5	Time Point(date)	1	1	1	Value during				3
6	Time Point(date)	1	1	1	Value during	6	Fabrication No	1	1	1	Instantaneous				4
7	Fabrication No	1	1	1	Instantaneous	7	Volume(m3)	2	1	1	Instantaneous			3	
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										

New type of Mbus device

Name

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.

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

Ethernet | UART | Virtual interfaces | Connected devices | Modbus devices | Modbus register grouping | Data transfer | FTP server | MQTT Subscriber

MBus devices

Configuration | Current values

	1	2
Enabled	<input checked="" type="checkbox"/>	<input type="checkbox"/>
COM Client	1	1
Amount of meters	1	1
Read period	10	1
Period dimension	<input type="radio"/> sec. <input type="radio"/> min. <input type="radio"/> h	<input type="radio"/> sec. <input type="radio"/> min. <input type="radio"/> h

First group

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

Unknown
Weser
LSE

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

If your settings are correct, you 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.

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

Ethernet | UART | Virtual interfaces | Connected devices | Modbus devices | Modbus register grouping | Data transfer | FTP server | MQTT Subscriber

MBus devices

Configuration | Current values

Device/Parameter	Last read time	Storage number	Tariff	SubUnit	Type of parameter	Double float	Double long	Float	Long
2 - LSE (08014388)	0110112000 03:28:43								
Identification Nr.		1	1	1	Instantaneou				8014388
Volume(m3)		1	1	1	Instantaneou			0.135	
On Time(hours)		1	1	1	Instantaneou				75371
Time Point(time & date)		1	1	1	Instantaneou			1486149120.0	
Time Point(date)		1	1	1	Value during				943920000
Fabrication No		1	1	1	Instantaneou				8014388
Volume(m3)		2	1	1	Instantaneou			0.135	

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

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

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

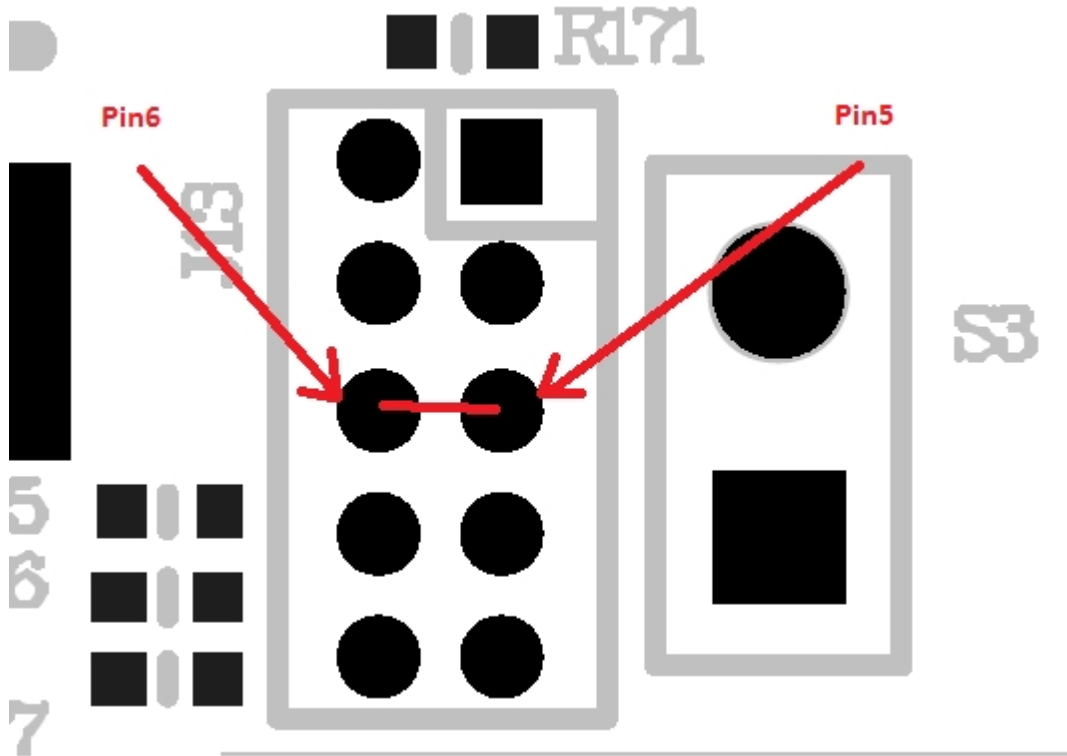
Req: 1239 Answ: 1238 TOut: 0 Except: 1

Created with the Personal Edition of HelpNDoc: [Easy CHM and documentation editor](#)

Setting up defaults values

How to set default values for RAY-3. (works from 1.18 firmware version)

1. Disconnect power and remove upper case of device.
2. Check J13 connector on the board. Small 10 pin connector near the battery and relays.



4. Run configuration tool and check configuration.

Ethernet default settings:

IP 192.168.1.125

GW 192.168.1.254

Mask 255.255.255.0

UART1 sets as modbus RTU server with UART settings 19200,8,even,1. Modbus address=1.

Created with the Personal Edition of HelpNDoc: [Free iPhone documentation generator](#)

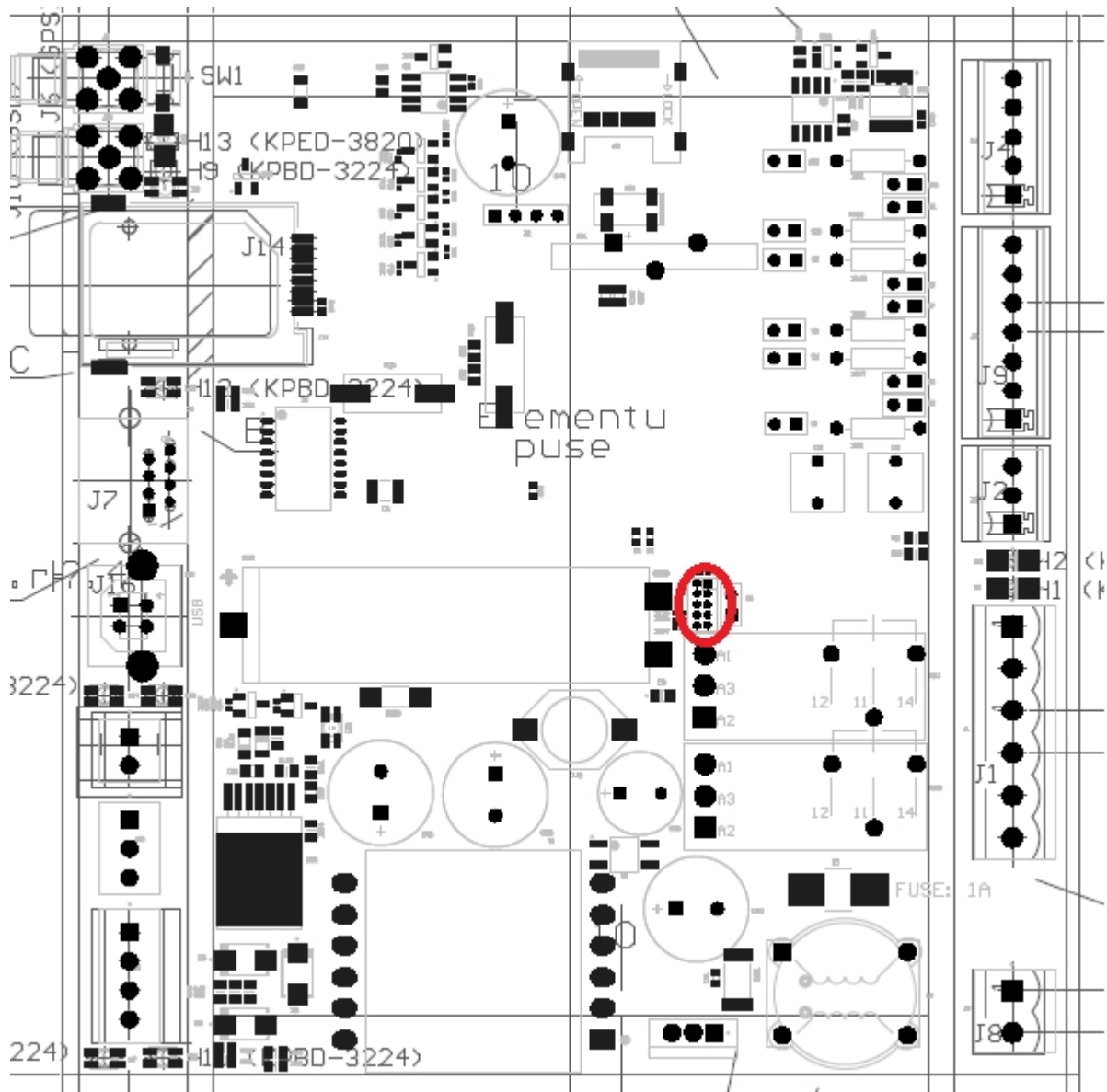
Bootloaders

Created with the Personal Edition of HelpNDoc: [Free Qt Help documentation generator](#)

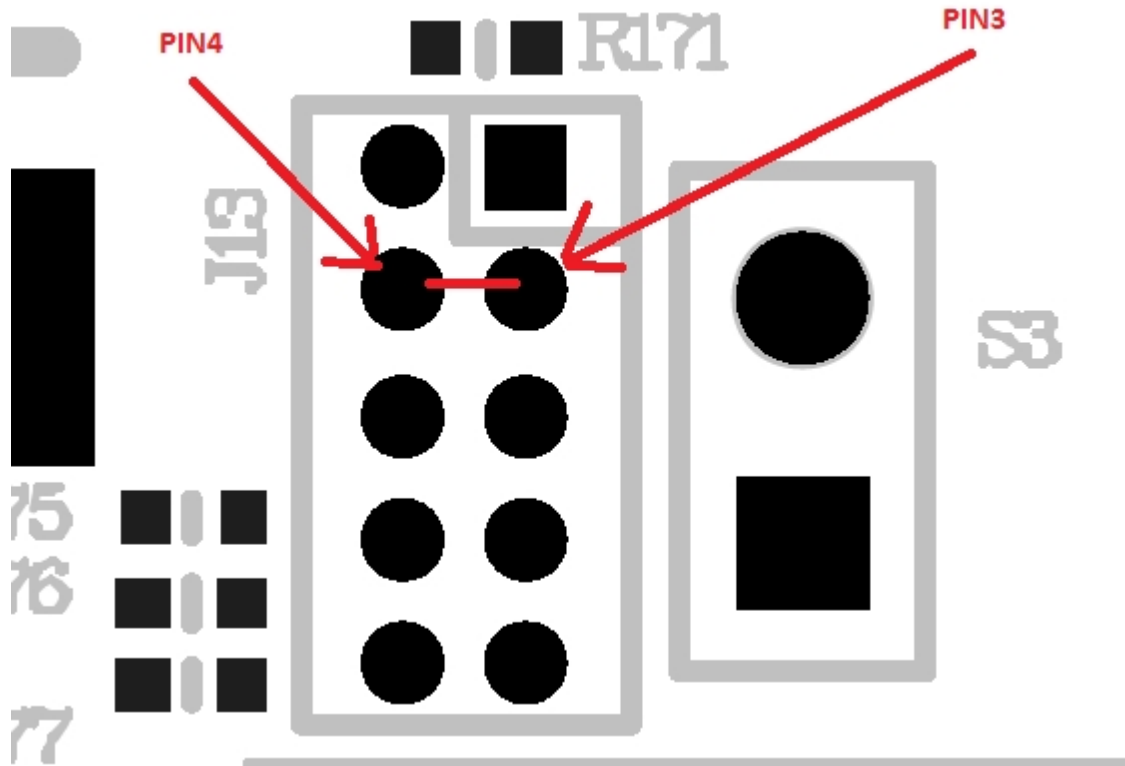
Manufacture bootloader

How to set RAY-3 into manufacture bootloader mode.

1. Disconnect power and remove upper case of device.
2. Check J13 connector on the board. Small 10 pin connector near the battery and relays.



- Short pin3 and pin4 of J13 and connect power supply to device. Leave pin3 and pin4 connected until status led starts blinking 4 times in second (it will take 2-3 seconds), this means device is in bootloader mode. In working mode status led blinks once a second.



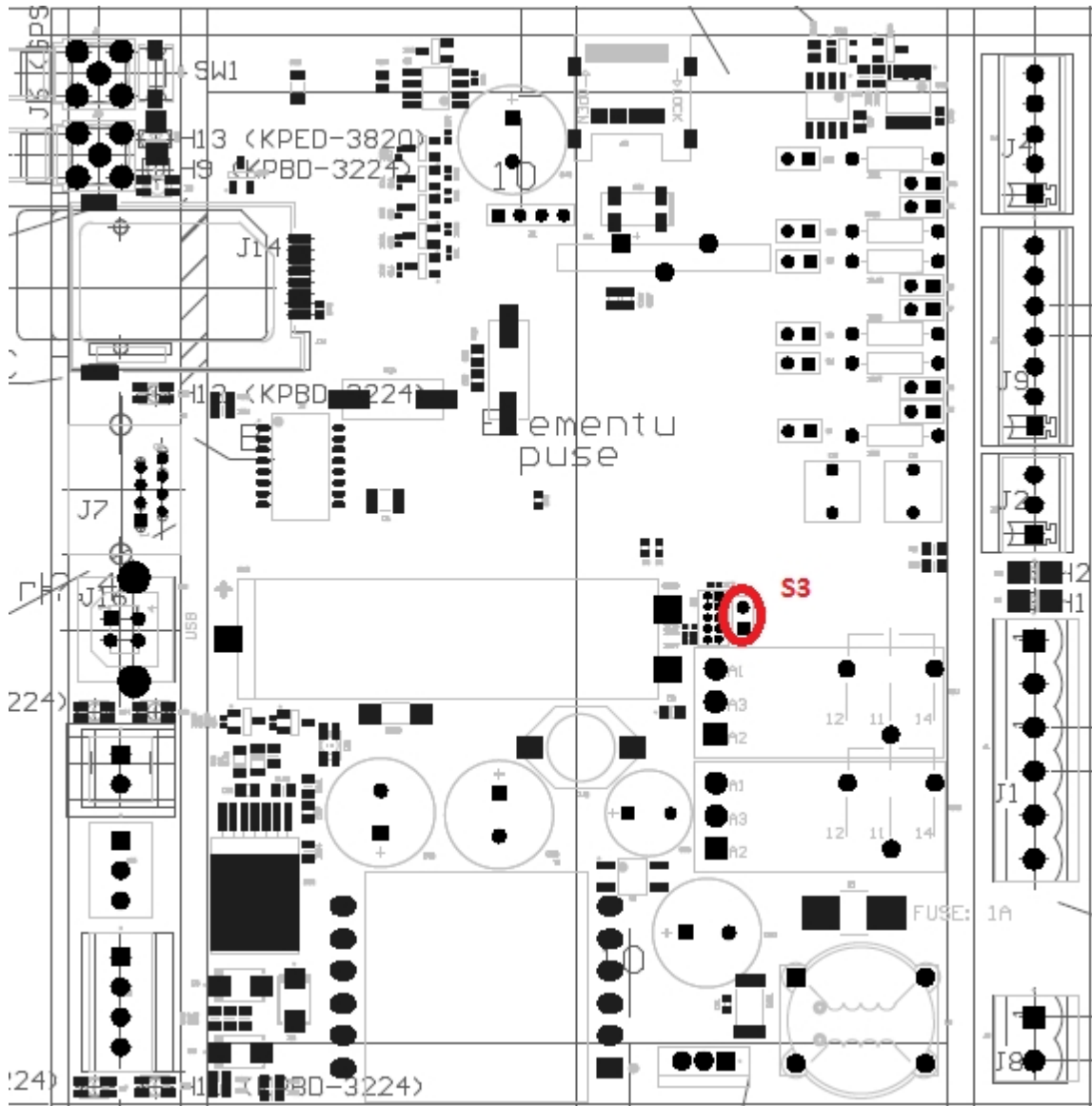
4. Now it is possible to connect USB and start update firmware using “Downloader”. Check Firmware update section.

Created with the Personal Edition of HelpNDoc: [What is a Help Authoring tool?](#)

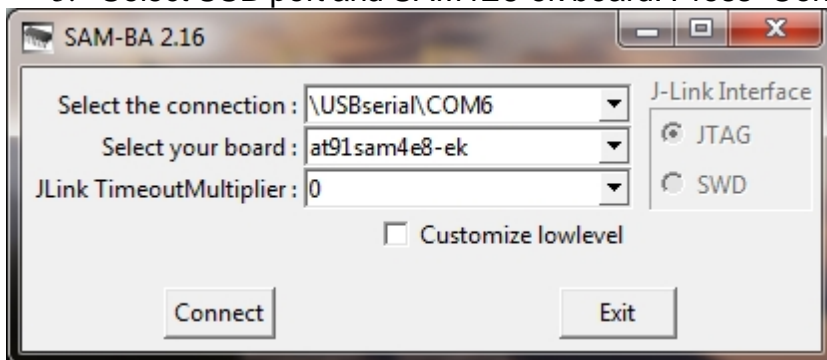
Atmel chip ISP

**How to set RAY-3 into factory bootloader mode (SAMBA).
(Be carefully, this will erase all memory of processor and you will need to program manufacture bootloader first)**

1. Disconnect power and remove upper case of device.
2. Check S3 connector on the board. 2 pin connector near battery and relay.
3. Short S3. Power up device, wait 1s, disconnect the power, disconnect S3.
4. Power up device, connect USB and run Atmel Samba ISP program. You can download Samba from atmel website <http://www.atmel.com/tools/atmelsam-bain-systemprogrammer.aspx>



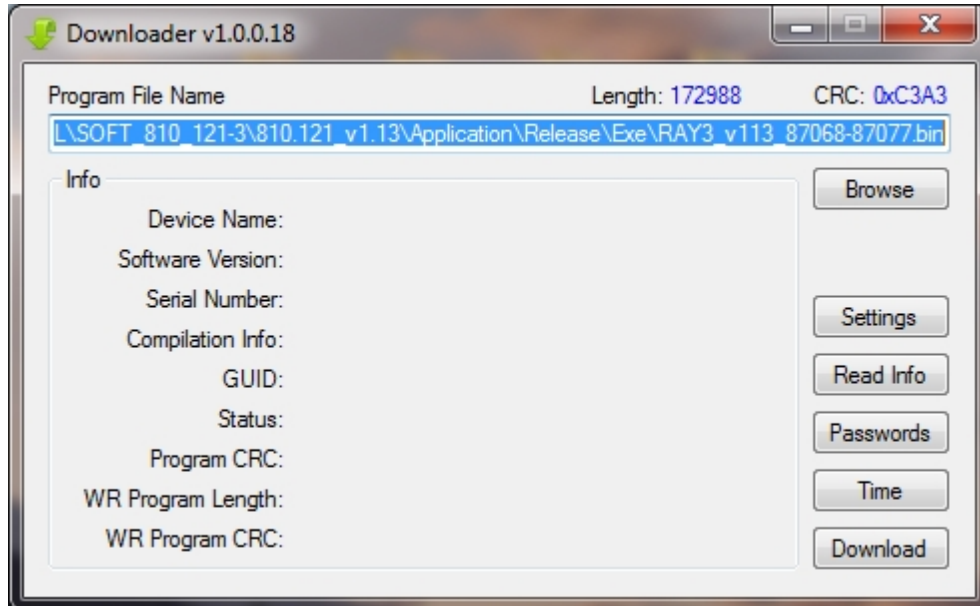
5. Select USB port and SAM4E8-ek board. Press “Connect”



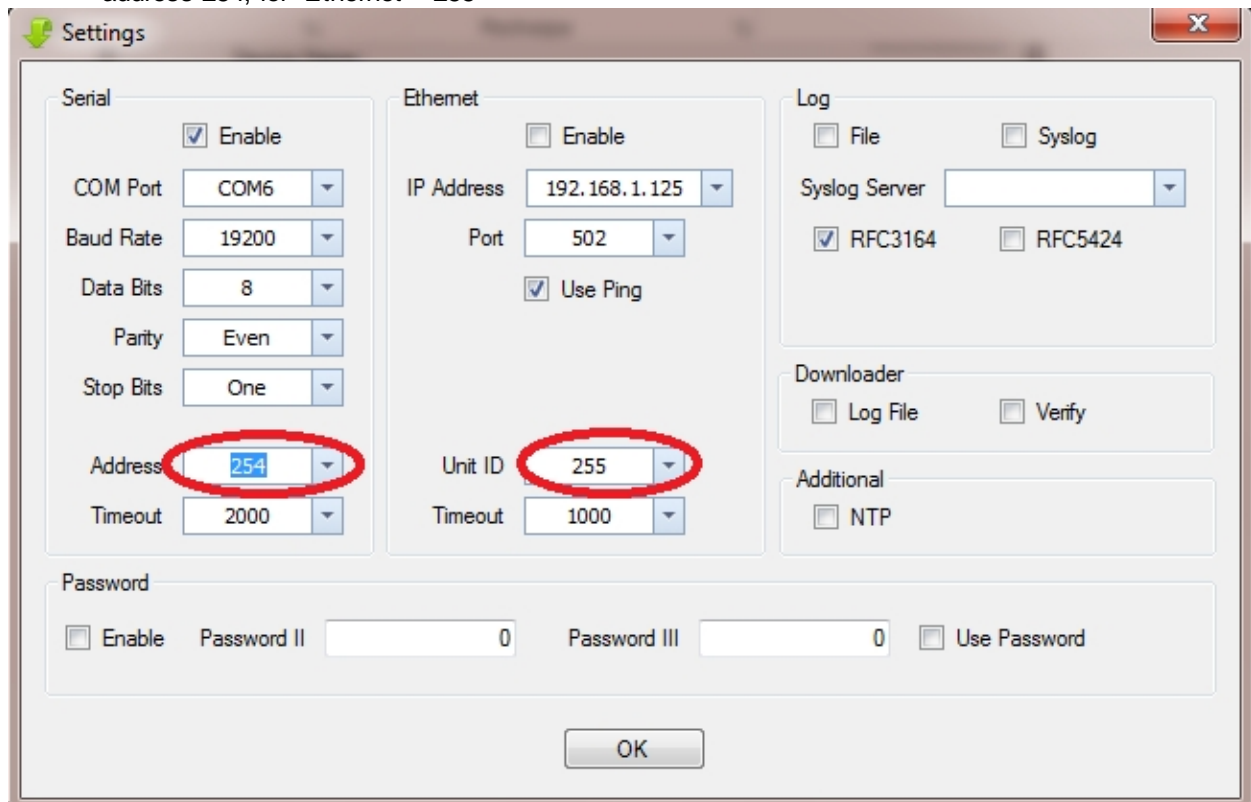
6. Select manufacture bootloader file in “Send File Name” and press “Send file”. If you get any popup press “Yes” or “OK”.
7. On scripts dropdown select “Boot from flash” and press “Execute”.
8. Close Samba program and reconnect device power
9. Now device is in manufacture bootloader mode. Upload firmware using “Downloader” software. Check "Firmware update" section.

Firmware update

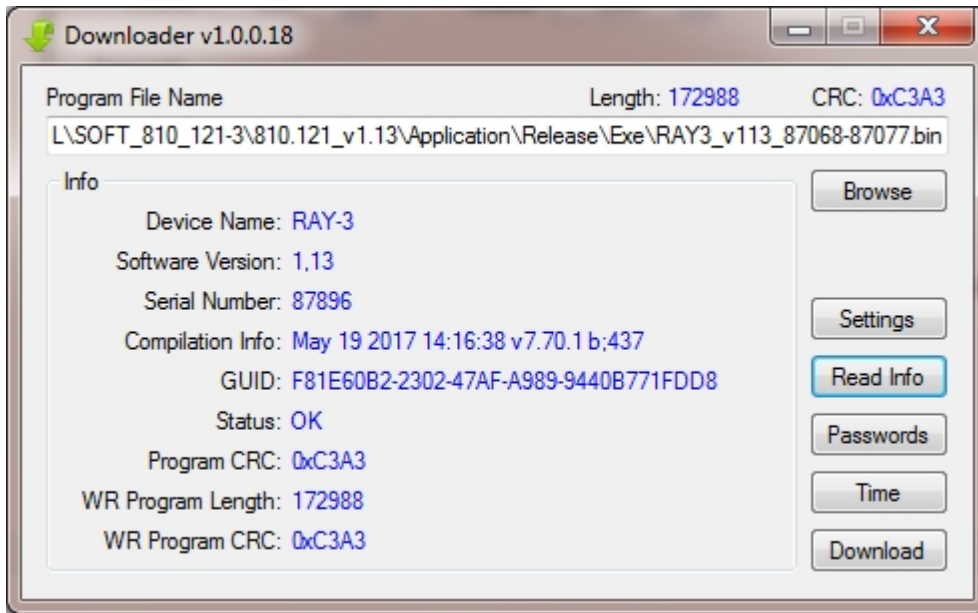
1. Run downloader



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



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



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

