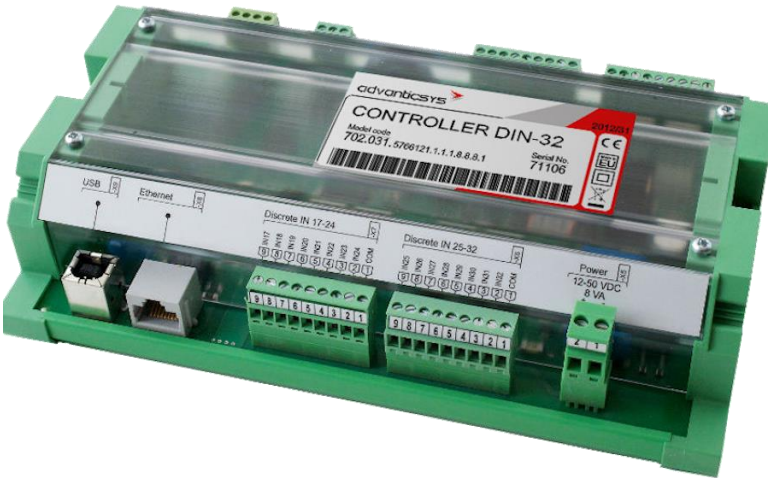
# DIN-32
# User Manual




10/2024

## Table of contents

# 1. Abbreviations

**Xn** – A number representing a socket. This information is provided for the manufacturer's purposes and is used in data schemas and connection diagrams.
**GSM** – Global Standard for Mobile Communications. This interface is designed for remote connections and bidirectional data transfer over the Global Standard Mobile network.
**GPRS** – A packet-oriented mobile data service on the 2G and 4G/3G cellular communication systems' global system for mobile communications (GSM).
**Ethernet** – A family of computer networking technologies for local area networks (LANs), commercially introduced in 1980. Standardized in IEEE 802.3, Ethernet has largely replaced competing wired LAN technologies. This interface is used for connecting to a LAN (Local Area Network).
**IP address** – An Internet Protocol (IP) address is a numerical label assigned to devices participating in a network that uses the Internet Protocol for communication between its nodes.
**TCP/IP** – Transmission Control Protocol, used for communication between computers, serves as the standard for transmitting data over networks and as the basis for standard Internet protocols.
**MAC address** – Media Access Control address, a unique identifier assigned to most network adapters.
**UART** – A Universal Asynchronous Receiver/Transmitter is a type of "asynchronous receiver/transmitter," a part of computer hardware that translates data between parallel and serial forms. UARTs are commonly used in conjunction with communication standards such as EIA RS-232, RS-422, or RS-485. Records (UARTx) on top of the enclosure are also used as the serial interface number.
**GND** – Ground wire contact.
**RS232** – The traditional name for a series of standards for serial binary single-ended data and control signals connecting a DTE (Data Terminal Equipment) and a DCE (Data Circuit-terminating Equipment). It is commonly used in computer serial ports. The standard defines the electrical characteristics and timing of signals, the meaning of signals, and the physical size and pin-out of connectors. RS232 interfaces are prepared for the connection of peripheral devices (e.g., energy meters, controllers, machines, etc.).
**TD** – Contact for the transfer data wire of the RS232 socket.
**RD** – Contact for the read data wire of the RS232 socket.
**DTR** – Contact for the Data Transmit Ready wire of the RS232 socket.
**RS485** – A standard defining the electrical characteristics of drivers and receivers for use in balanced digital multipoint systems. Published by the ANSI Telecommunications Industry Association/Electronic Industries Alliance (TIA/EIA), digital communications networks implementing the EIA-485 standard can be used effectively over long distances and in electrically noisy environments. Multiple receivers may be connected to such a network in a linear, multi-drop configuration. RS485 interfaces are prepared for the connection of peripheral devices (e.g., energy meters, controllers, machines, etc.).
**A+** – Contact for the positive wire of the RS485 socket.
**B-** – Contact for the negative wire of the RS485 socket.
**USB** – Universal Serial Bus is an industry standard that defines the cables, connectors, and protocols used for connection, communication, and power supply between computers and electronic devices. The USB Type-B socket is prepared for connection to a PC (Personal Computer). The USB Type-A socket is prepared for connection to peripheral devices (e.g., memory sticks, etc.).
**Socket** – An endpoint of a bidirectional inter-process communication flow across an Internet Protocol-based computer network, such as the Internet.
**TX/RX** – Data transfer/receive indicating LED.
**Central computer** – A server or computer to which data can be sent.

# 2. Preface

## 2.1 Symbols

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

| Symbol | Explanation |
|---|---|
|  | 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 |
|  | Directive on the restriction of the use of certain hazardous substances in electrical and electronic equipment 2002/95/EC. Commonly referred to as the Restriction of Hazardous Substances Directive or **RoHS**) |
|  | Waste Electrical and Electronic Equipment Directive |

## 2.2 Safety instructions

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

**Before connecting to the power supply, ensure that:**

1. The device is not damaged (no cracks, melted, broken, or exposed areas).
2. The device is used with the correct cables of appropriate thickness.
3. The device and antenna are installed indoors.
4. The device is intended for supply from a Limited Power Source (LPS) with a current rating of over-current protective devices not greater than 2A.
5. The highest transients on the DC secondary circuit of the LPS, derived from the AC main supply, shall be less than 71V peak.
6. The associated equipment (AE), such as the PC and PSU (LPS), shall comply with the requirements of Standard EN 60950-1.
7. The device is dry.
8. Ambient temperature and humidity are within the normal range.
9. Other types of devices (e.g., counters, etc.) are connected correctly according to the manufacturer's regulations.
10. The end of stranded conductors shall not be consolidated by soft soldering and must be terminated properly.
11. The device, PC, and other peripheral devices must be strictly connected through a double-pole breaker (with a current break less than 5A and a space between breaker contacts greater than 3mm). The pole breaker must be part of the building's wiring and located in an accessible place with proper markings.

**Don‘t use:**

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

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

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

## *2.3 Connecting to device*

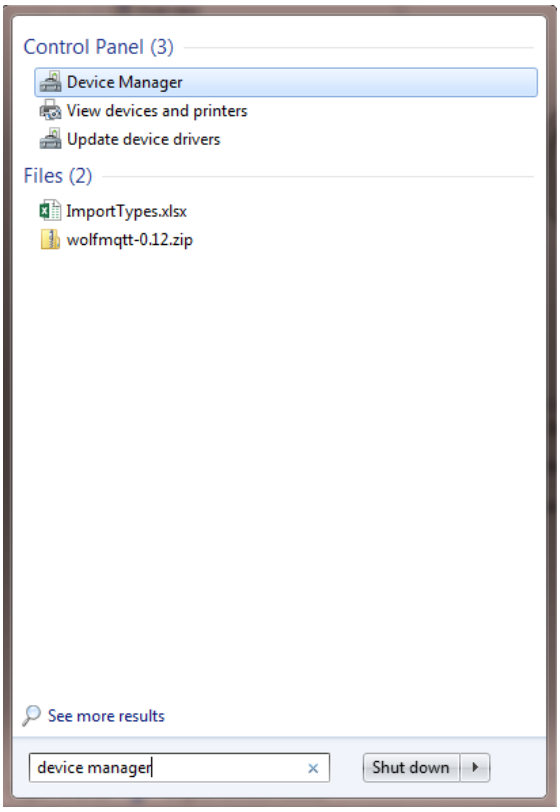
The USB port is used for local configuration of the device. It is also possible to configure the device via Ethernet. All configuration is done using the Modbus protocol and the device configuration tool software, which can be downloaded from the manufacturer’s website.

Use a USB Type-A to Type-B cable to connect the device to a computer:

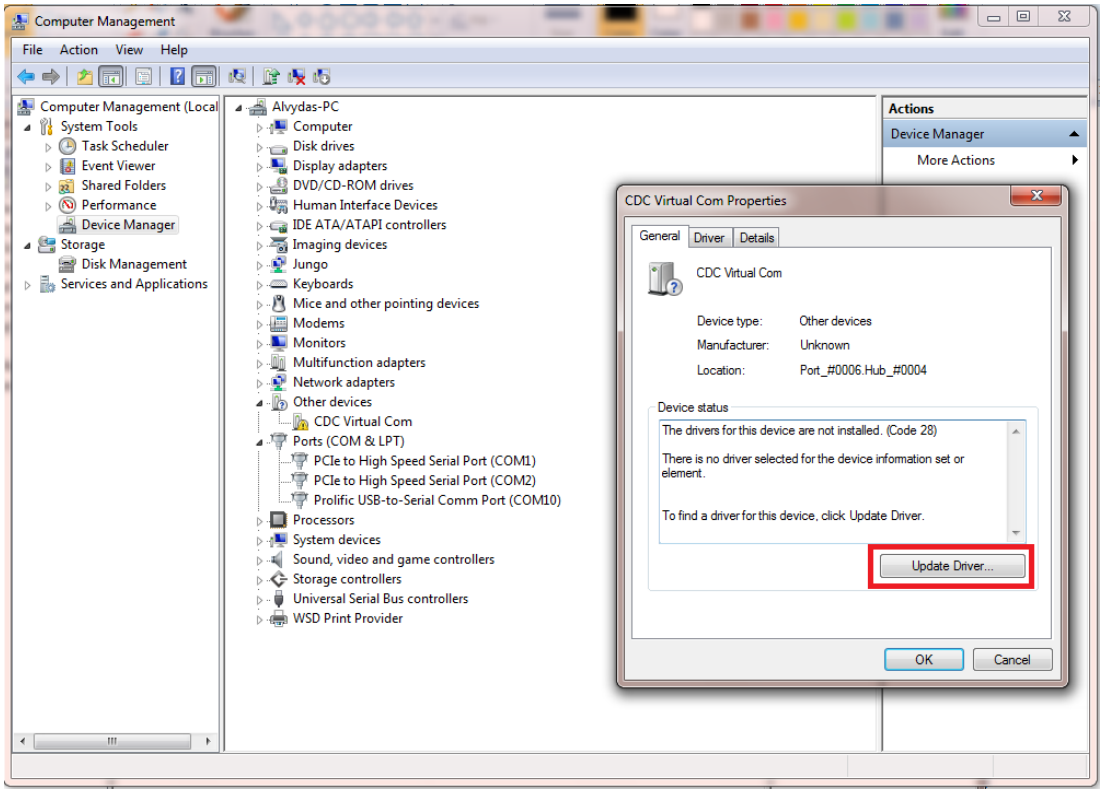
- a) To device: USB Type-B
- b) To computer: USB Type-A

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

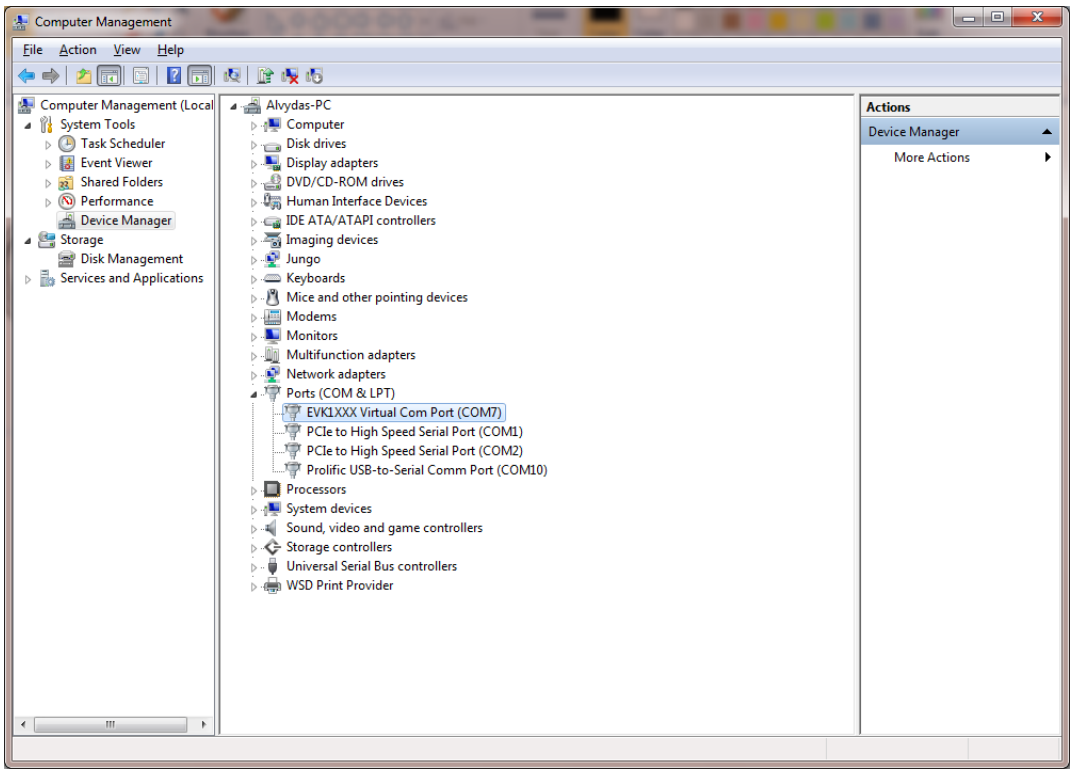
1. In the search box, type *Device Manager* and then click on it.




2. Double-click the device category, and then double-click the device you want.
3. Click *Update Driver* and follow the instructions.




4. Select *Browse my computer for driver software*, click *Browse*, and select the configuration software folder.
5. Click *Next*.
6. Wait while Windows installs the driver. If you see the 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 a COM port number. Use this COM port to connect with the configuration tool.

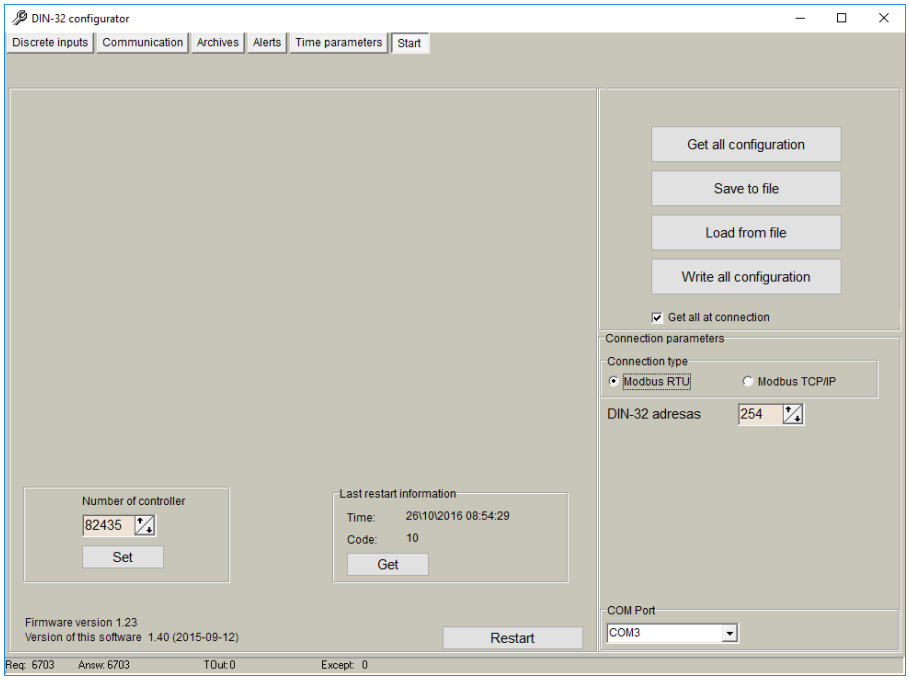


**On some Windows versions (Windows 8, Windows 10), you may need to disable third-party driver signature checking before installing the device driver. Please check online for instructions on how to do this.**

**Configuration tool software**

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

- *Get all configuration* button: Reads all configuration from the device.
- *Save to file* button: Saves all configuration to a file, so it can be loaded back to the device later.
- *Load from file* button: Loads saved configuration from a file.
- *Write all configuration*: Writes the loaded configuration to the device.
- *Get all at connection* checkbox: Reads all configuration when connecting over a TCP/IP connection.
- Connection type *Modbus RTU*: Connects to the device over USB or serial port.
    - DIN-32 address (254)
    - COM port number assigned by your PC to the USB port
- Connection type *Modbus TCP/IP*: Connects to the device over a TCP/IP connection.
- *Number of device* section: Device serial number.
- *Last restart information*: Time of the last reset and reset code. The reset code values are:
    1. No TCP packet over GPRS in the configured time.
    2. GPRS task stops working.
    3. Not enough heap memory.
    4. Firmware update reset.
    5. Modbus reset.
    6. Unable to connect to GPRS.
    7. External pin reset.
    8. Watchdog reset.
    9. Brownout reset.
    10. Power-up reset.
    11. No TCP packet over ETHERNET in the configured time.
    12. ETHERNET task stops working.
    13. All TCP sockets are used (if defined UIP_RESET_ALL_CONN_USED).
    14. Periodic reset.



# 3. Program modules

## *3.1 Archives*

The device has several types of archives:

- **Events archive**: All events will be saved here (discrete input alarms, limits verification).
- **Discrete inputs archive**: All inputs status changes will be saved here.
- **User-defined archive**: This is a user-configured archive, allowing the user to add any existing data register to the archive.

All archives are saved to the internal data flash memory. The maximum archive record count depends on the memory size and available space.

### 3.1.1 Events archive

All events are saved in the events archive.

**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<br>Modbus TCP - 255 | 400 .. 499 | 10000 | 1000 | 4914 |

Event archive record structure:

| Variable name | Purpose / Value | Type of value |
|---|---|---|
| Time | Time of alarm | Long int (32 bits) |
| Alarm identifier | Every identifier is change +1 | Long int (32 bits) |
| Alarm source | The oldest byte value (alarm source):<br>1 – Discrete input alarm<br>Youngest byte value:<br>Channel number; | Int (16 bits) |
| Type of deviation | Value always 0. | Int (16 bits) |
| Deviation value | Discrete input value. | Float |
| | | Total 16 bytes |

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

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

To read the 5th oldest record:

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

### 3.1.2 Discrete inputs archive

The **discrete inputs archive** saves the values and status of all channels.

**Reading discrete inputs archive over the Modbus file system**
The discrete inputs archive can be read using the Modbus read file function 20.

| Modbus function | Modbus ID | Modbus file address | Max registers in file | Records in file | Current record count register |
|---|---|---|---|---|---|
| 20 - Read File Record | Modbus RTU - 254<br>Modbus TCP - 255 | 100 .. 199 | 9996 | 1666 | 4911 |

Diagnostic archive record structure:

| Variable name | Purpose / Value | Type of value |
|---|---|---|
| Time | Record time.<br>If event type=7 then new set time | Long int (32 bits) |
| Time ms | Time in milliseconds | Int (16 bits) |
| Channel number and status | Most significant shows channel number (0-31).<br>Less significant shows input status (0 - 1) | Int (16 bits) |
| Total counter values | Values of all channels (0-31) | Long int (32 bits) |
| | | Total 12 bytes |

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

File address : 100
Register address: 0
Register count : 8 (16/2, diagnostic archive structure length/2)

To read the 5th oldest record:

File address : 100 + (5/1250)  (record number/record count in file)
Register address: 8*(5-1)
Register count : 8 (16/2, diagnostic archive structure length/2)

### 3.1.3 User-defined archive

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

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

**User archive storage period**
The archive period is synchronized with real time. If the read period is 1 minute, records will be generated every minute (00:00:00, 00:01:00, 00:02:00, etc.). If the read period is 15 minutes, records will be generated every 15 minutes (00:00:00, 00:15:00, 00:30:00, etc.).

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

1. **Count of parameters** – The number of configured lines in the table.
2. In the **Register** column, configure the start register of the value.
3. In the **Count of parameters** column, configure how many data values will be taken from the start register.

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

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

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

Diagnostic archive record structure:

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

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

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

To read the 5th oldest record:

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

## *3.2 TCP modules*

### 3.2.1 FTP client

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

**Csv file creator**
The device creates a CSV report file from user-defined archive values. Every record in the file has its timestamp

(value record time). It can include a "Header" for each value, and a dimension for each value. All data in the file is separated by a ";" symbol, and each record is placed on a new line. A standard file content looks like this:

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

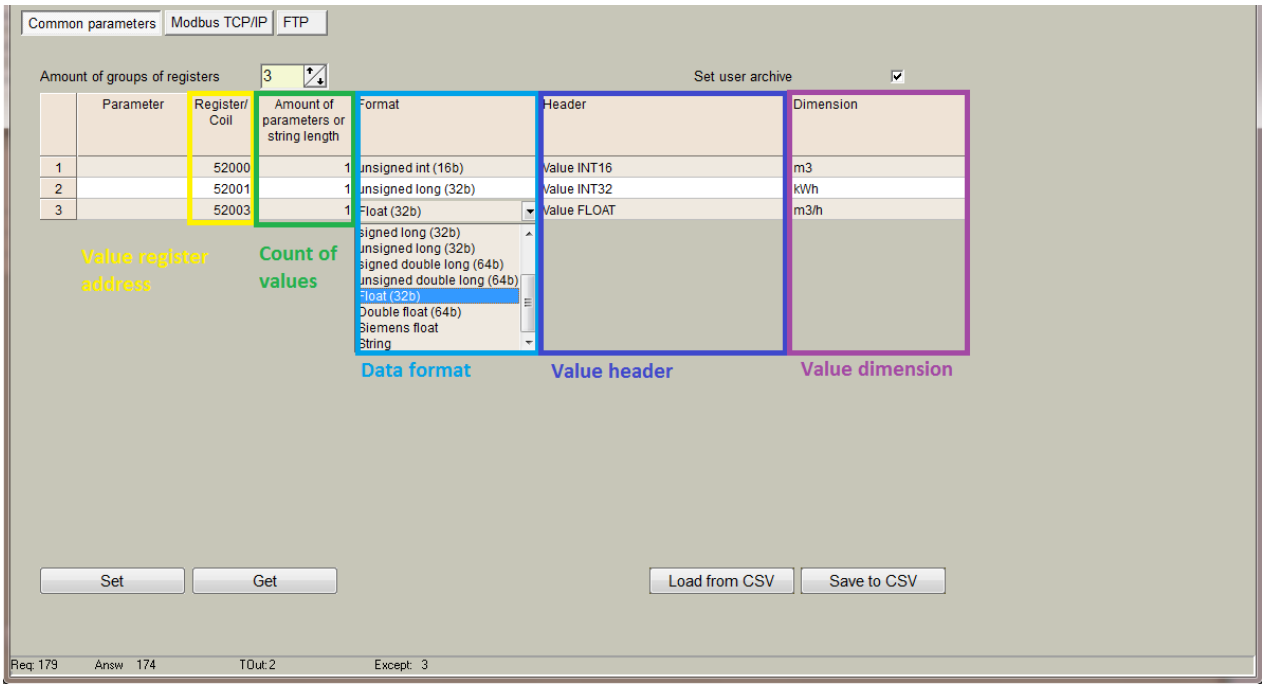
- **<Value Header>** – Configured header from the *"Communication/Data Transfer/Common Parameters"* tab.
- **<Value data>** – Configured parameter value taken from the user archive.
- **<Value dimension>** – Configured dimension from the *"Communication/Data Transfer/Common Parameters"* tab.
- **<Record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

**CSV file creator table**

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

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

- **"Get"** button – Read the configuration from the device.
- **"Load from CSV"** button – Load a saved configuration table from a CSV file.
- **"Save to CSV"** button – Save the configuration table to a CSV file.



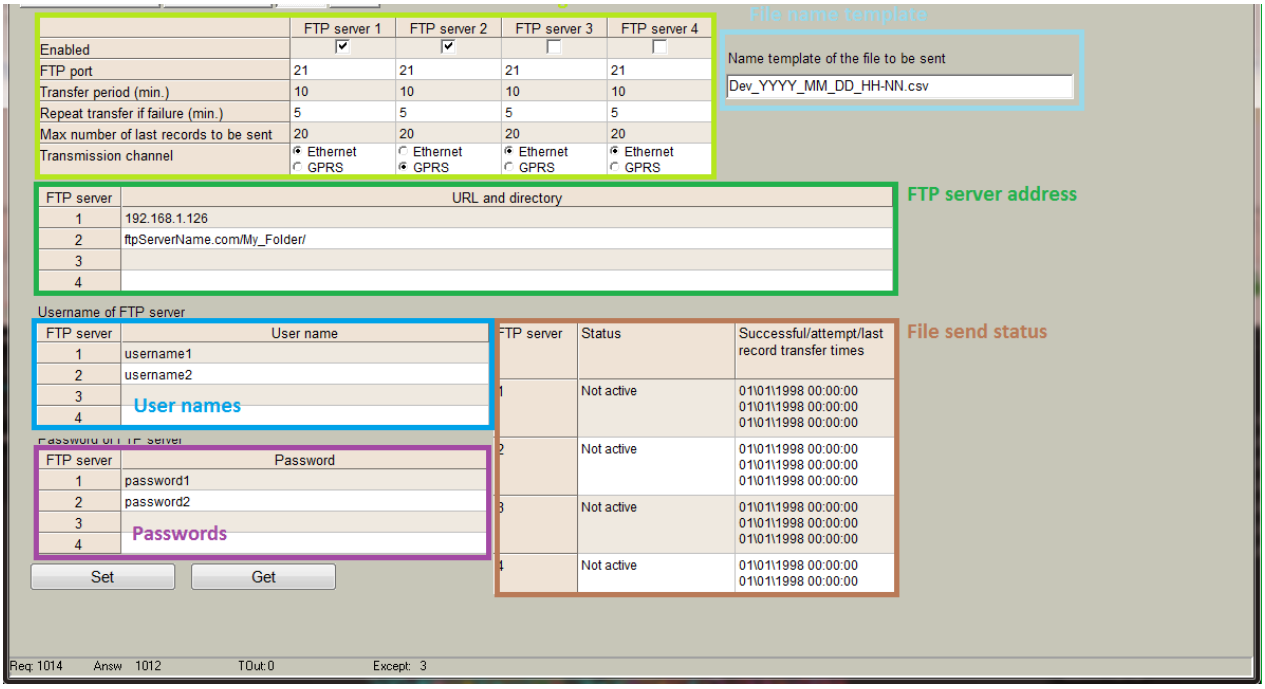

Example:
With this configuration received file will be like this:

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

**FTP client configuration**
A CSV file can be sent to up to 4 FTP servers. Only the FTP protocol is supported using passive mode, user/password authentication, and the CSV file format.
All configuration is done in the *"Communication/Data Transfer/FTP"* tab. Before configuring, you need to have a working FTP server and the following information: its IP address or URL, and the connection username and password. The sections in this tab include:

- "**File Transfer Configuration**" section:
  - **"Enabled"** checkbox: Enables the appropriate FTP server.
  - **"FTP port"**: The appropriate FTP server's TCP port.
  - **"Transfer period (min)"**: File send period. The value is in minutes and can range from 1 minute to 1440 minutes (24 hours).
  - **"Repeat transfer if failure (min)"**: Retries sending the file after the configured time if the file transfer fails. The value is in minutes and can range from 1 minute to 1440 minutes (24 hours). The recommended value is half the transfer period.
  - **"Max number of last records to be sent"**: The maximum number of last records to include in the CSV file. Only new records will be included. If 25 new records are available and the maximum is configured to 20, the file will only contain the 20 newest records, and the last 5 will be lost. If there are 5 new records and the maximum is set to 20, the file will contain 5 new records. Values can range from 1 to 200.
  - **"Transmission channel"**: The device can have 2 transfer channels: Ethernet and GPRS. Select which to use for the appropriate FTP server.

- "**FTP server address**" section:
  - **"URL and directory" column**: The IP or URL address of the FTP server (up to 127 characters). It can be an IP address like "127.0.0.1" or a URL like "www.myftp.com". Directory listings can also be used, e.g., "www.myftp.com/MyFiles/".
- **"Usernames"** section:
  - "User name" column: The username for the appropriate FTP server.
- **"Passwords"** section:
  - "Password" column: The password for the appropriate FTP server.
- **"File Send Status"** section:
  - **"Status"** column: The current status of the FTP client. After the file is sent to the server, the status changes to "File transmitted". There are several other statuses for process checks: "Connecting to server", "Sending user", "Sending password", "Sending data file", and others.
  - **"Successful/Attempt/Last record transfer times"** column: Shows the times of some operations. "Successful" time is the time of the last successfully completed file send to the FTP server. "Attempt" time is the time of the last file send attempt (whether successful or not). "Last record transfer" time is the time the last record was sent.
- **"File Name Template"** section: The file name can be up to 127 characters long, including the ".csv" extension. The file name can include fixed fields that will be replaced with the date and time. Fixed fields include:
  - YYYY – Year
  - MM – Month
  - DD – Day
  - HH – Hour
  - NN – Minute
  - For example, if the file name template is "Dev_YYYY_MM_DD__HH-NN.csv" and the date is 2017.03.25 at 14:25, the file name will be "Dev_2017_03_25__14-25.csv".
- **"Set"** button: Writes the configuration to the device.
- **"Get"** button: Reads the configuration from the device.



Example:

In the picture above, two FTP servers are configured:

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

# 4. Hardware

## 4.1 Time settings

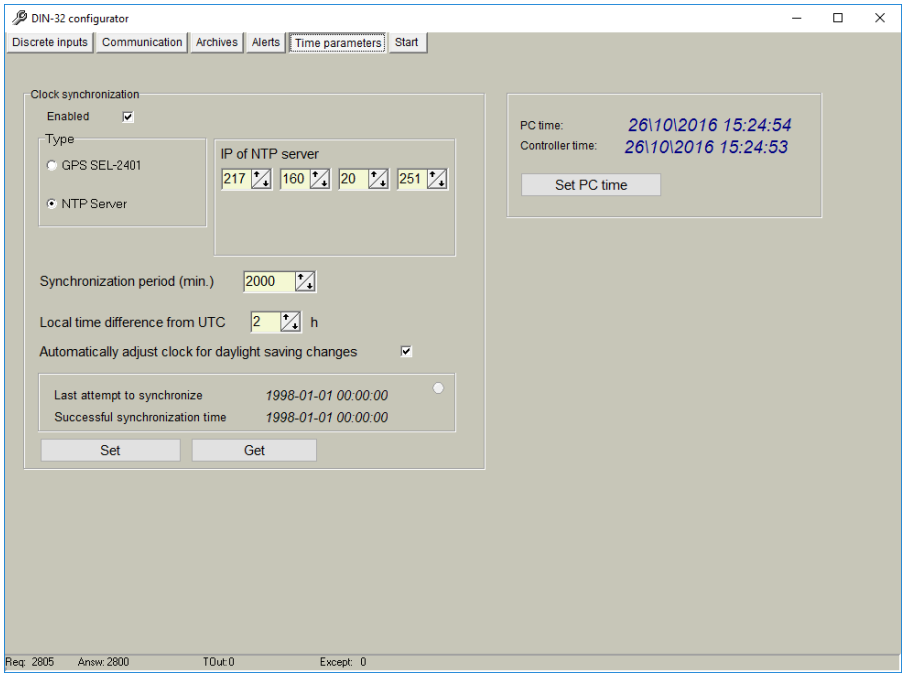
The device has an integrated battery-backed real-time clock (RTC) with a calendar. The RTC works in UTC time, and the user can configure it to return local time based on their location (selecting the time zone and summer/winter time usage).

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

**Setting Time with Configuration Tool**

Time settings can be changed in the *"Time Parameters"* tab. It allows users to set up time synchronization between DIN-32 Real Time Clock (RTC) and external time references. Several parameters can be configured under this tab:

- *"Set PC time"* It synchronizes internal RTC with PC time.
- *"Clock synchronization":* It must be enabled if synchronization between DIN-32 and any Network Time Protocol (NTP) server is requested. This option is particularly accurate since every given period (*"Synchronization period"* parameter), the device connects to an NTP server to get current UTC time. Also, automatic summertime is adjusted.




## 4.2 Ethernet configuration

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

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

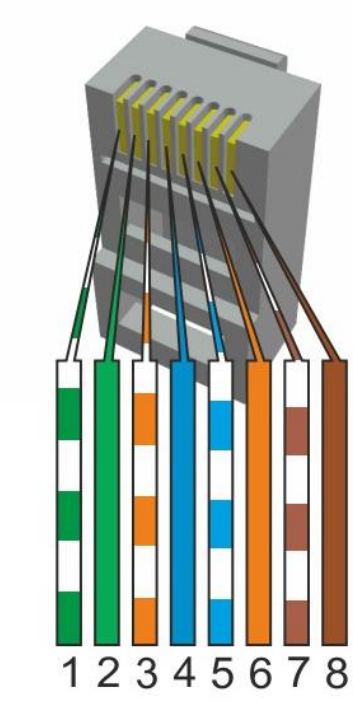
**Supported Services:**

- Modbus TCP/IP client
- FTP client

**Ethernet connection**

Use a standard RJ45 cable to connect the device to an 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- |

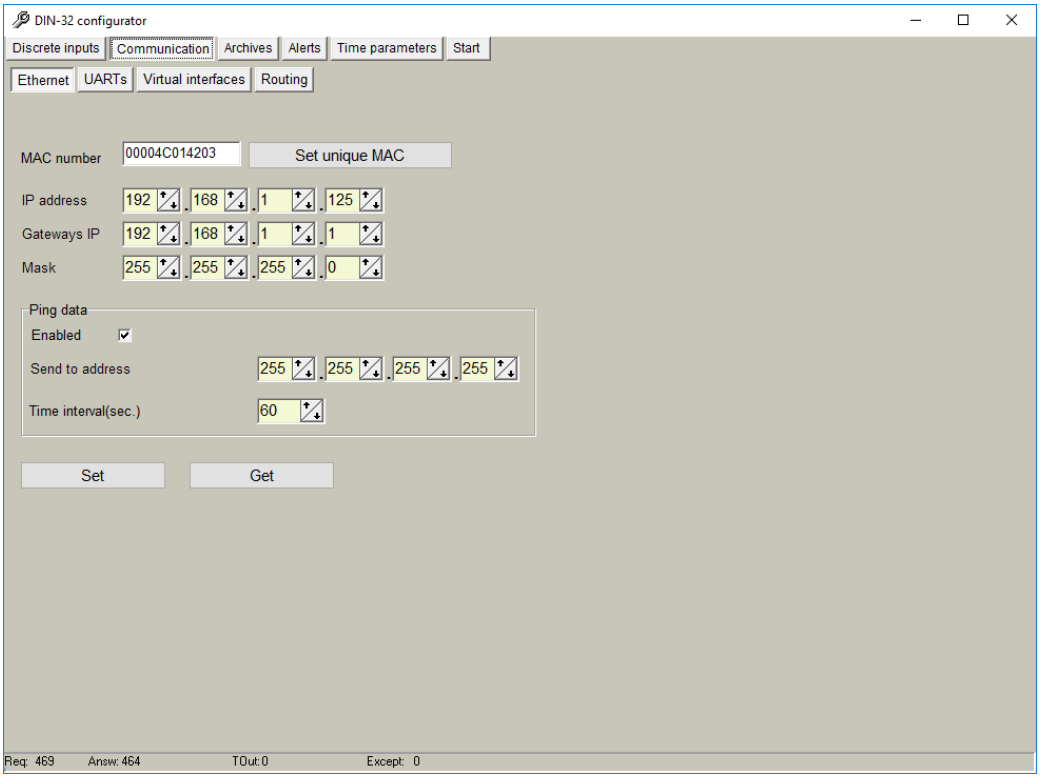
**LED indicators**

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

**Ethernet configuration**

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

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




**Default settings**

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

## *4.3 Serial ports*

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

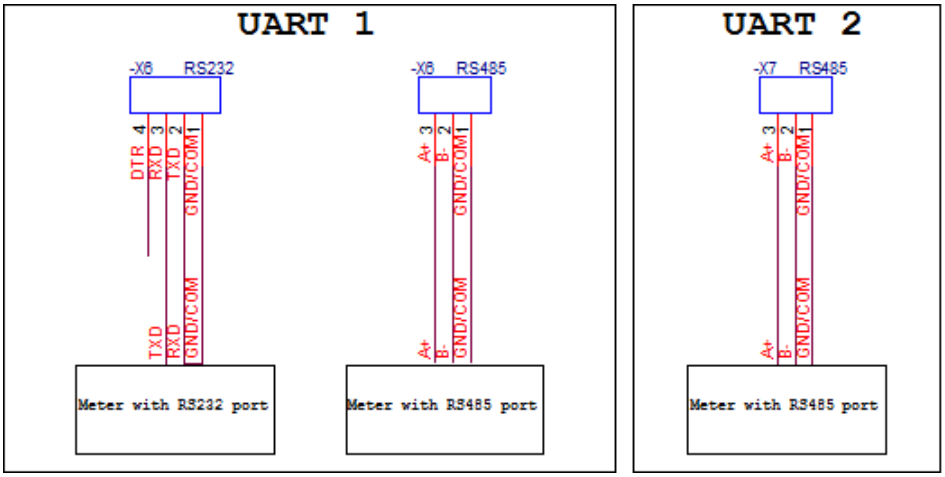
| Port number | Available options | Description |
|---|---|---|
| UART 1 | RS232 or RS485 | Can be used as Modbus slave |
| UART 2 | RS232 | Can be used as Modbus slave |

**UART characteristics:**

| Port number | Supported baud rates | Supported parity | Supported data bits | Supported stop bits |
|---|---|---|---|---|
| UART 1 | 300 - 38400 | Even, Odd, Mark, Space, None | 5,6,7,8 | 1,2 |
| UART 2 | 300 - 38400 | Even, Odd, Mark, Space, None | 5,6,7,8 | 1,2 |

**Wiring diagrams**

Different type meter connection to device:



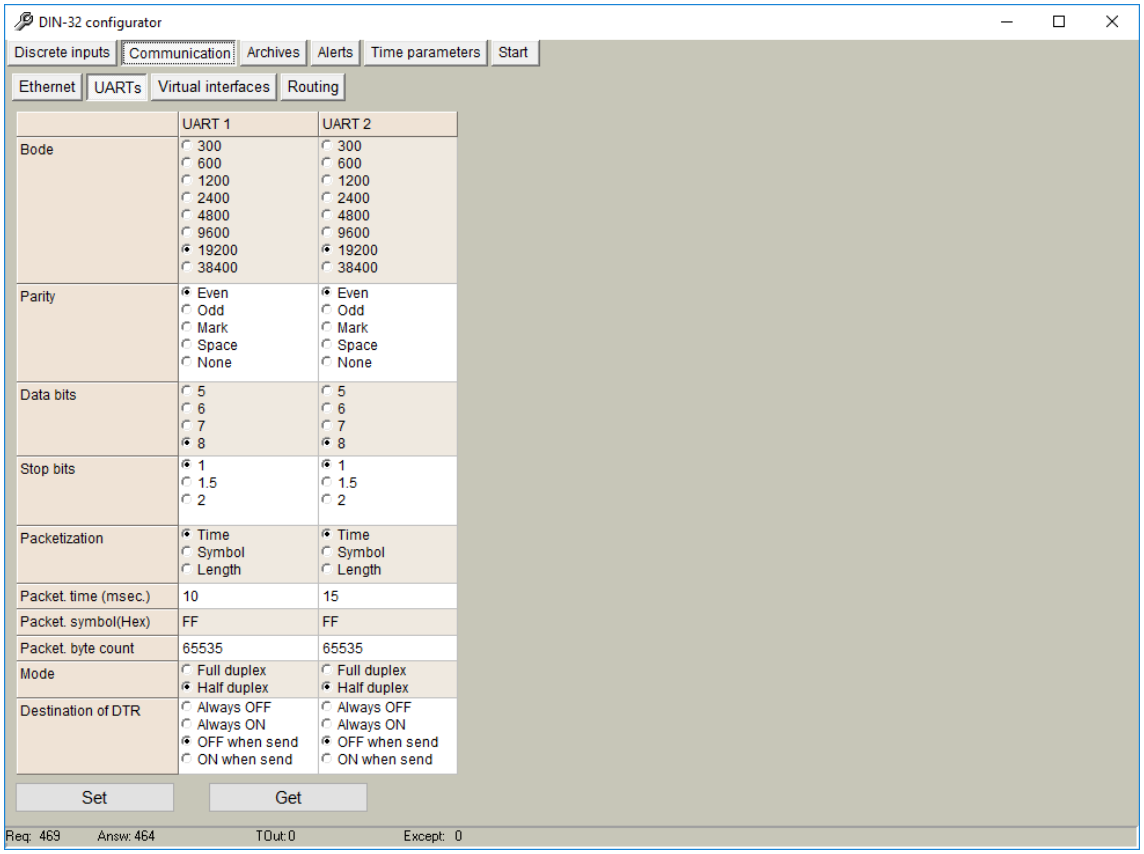

**Led indicators**

| 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 |

**UART settings**

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

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



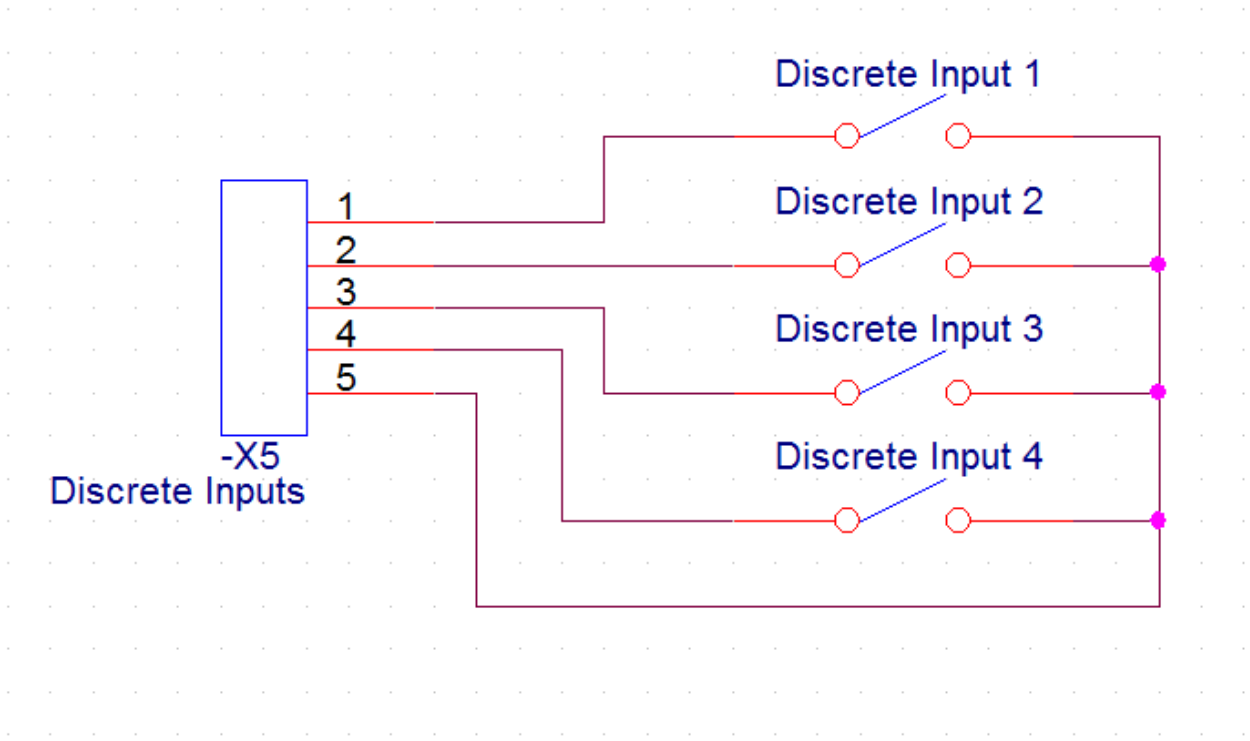

## 4.4 Discrete inputs

The device includes 32 sink contact discrete inputs, all with the same common signal. The device periodically tracks the status of all discrete channels, and when a change occurs on any channel, it stores that change with the real-time value. Additionally, a report can be initiated if user-defined.

**Discrete inputs purpose**:

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

**Wiring diagram**




**Discrete Inputs configuration**

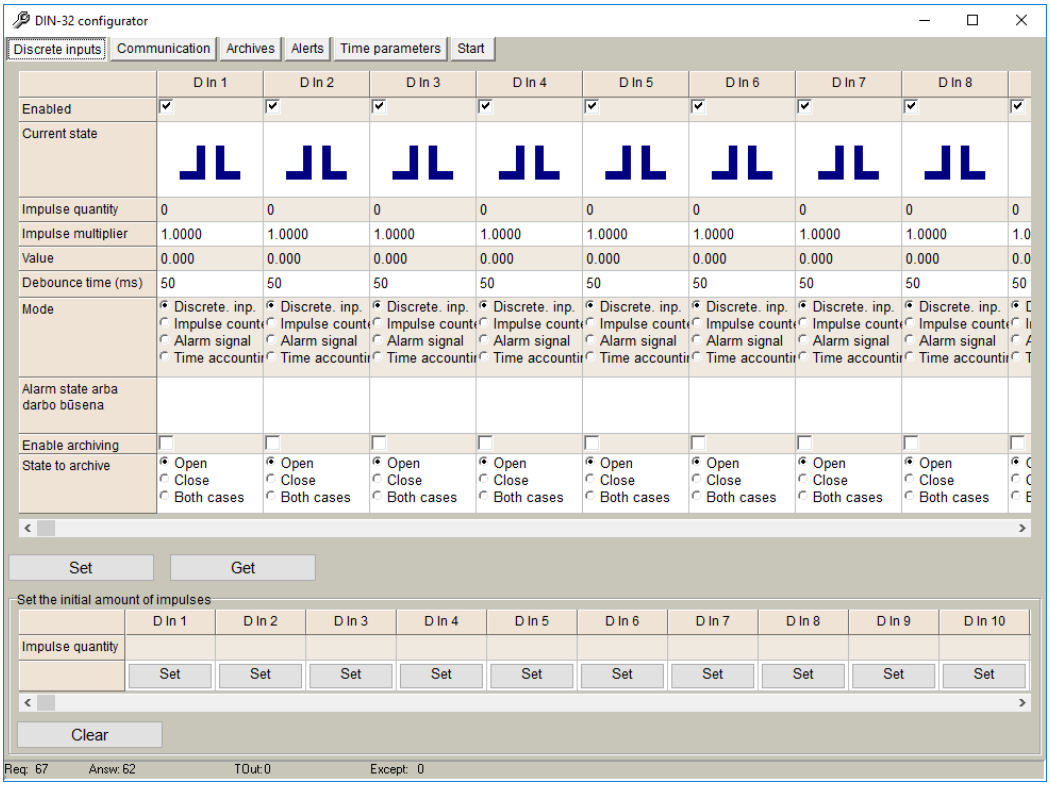
Discrete inputs configuration is done in the *"Discrete Inputs"* tab.

**"Discrete inputs settings" section:**

- **"Enabled" checkbox** – Enables the appropriate discrete input.
- **"Current state"** – Displays the current state of the discrete input. For example, the first discrete input may be shorted (active), while others are not shorted.
- **"Impulse quantity"** – The impulse counter for the appropriate discrete input.
- **"Impulse multiplier"** – The impulse counter multiplier for the appropriate discrete input.
- **"Value"** – The result of multiplying *"Impulse quantity"* by *"Impulse multiplier"*. This is used to convert the impulse count to a physical value.
- **"Debounce time (ms)"** – The filter time for the appropriate discrete input.
- **"Mode"** – The purpose of the appropriate discrete input:
  - **"Discrete inp."** – Standard discrete input.
  - **"Impulse counter"** – Counts discrete input state changes, with the count configured by the *"State to archive"* parameter.
  - **"Alarm signal"** – Discrete input with an alarm function (creates an alarm record in the events archive). The alarm state is configured with the *"Alarm state"* parameter.
- **"Alarm state"** – The alarm state for the appropriate discrete input:
  - **"Open"** – Generates an alarm if the discrete input is not shorted.
  - **"Close"** – Generates an alarm if the discrete input is shorted.
  - **"Both cases"** – Generates an alarm in both states.
- **"Enable archiving"** – Enables the alarm archive for changes in the appropriate discrete input.
- **"State to archive"** – The discrete input state for counting impulses.
- **"Set" button** – Writes the configuration to the device.
- **"Get" button** – Reads the configuration from the device.

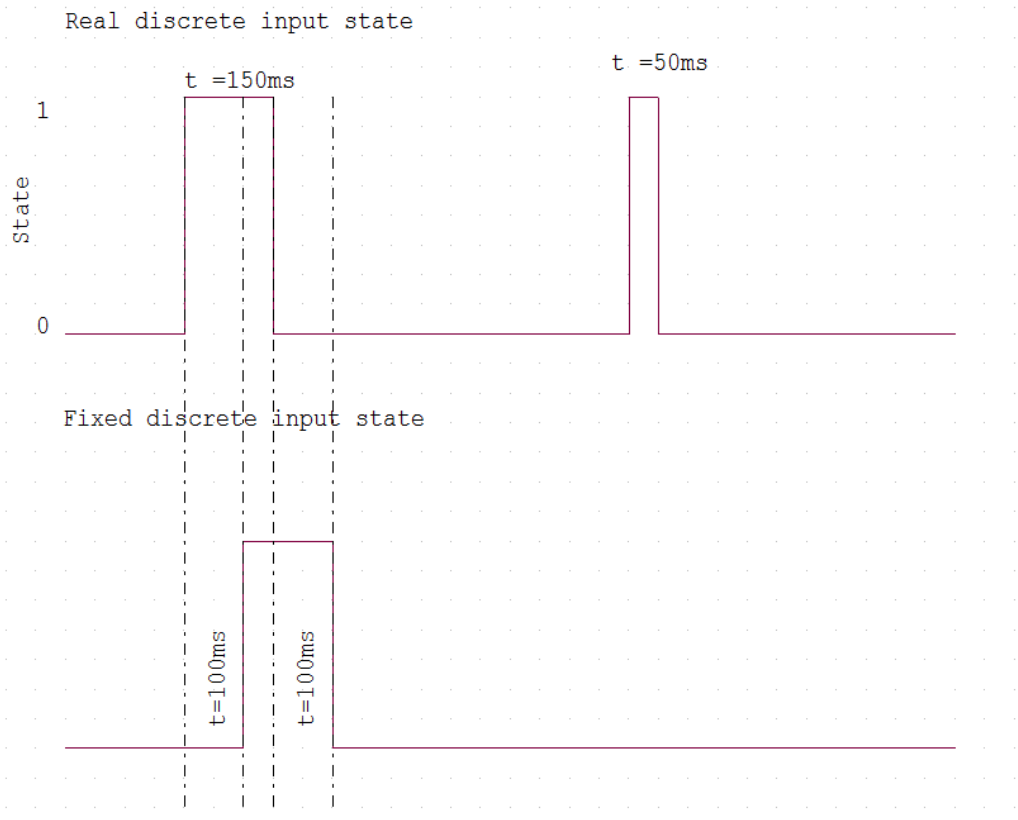
**"Impulse counter value set" section:**

- **"Set" button** – Sets the impulse counter value for the appropriate discrete input.
- **"Clear" button** – Resets all discrete input impulse counter values to 0.




**Example**

In the diagram below, we see two signals. The first is the real signal on the discrete input pins, and the second is the filtered signal. The configured *"Debounce time"* is 100 ms. The first impulse is detected after 100 ms because the *"Debounce time"* is 100 ms. A 50 ms impulse is not detected because its duration is shorter than the *"Debounce time"*.



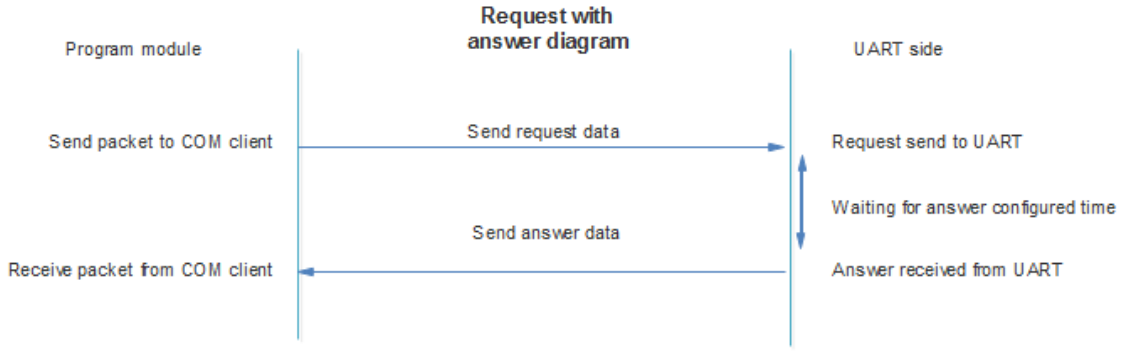
# 5. Virtual interfaces

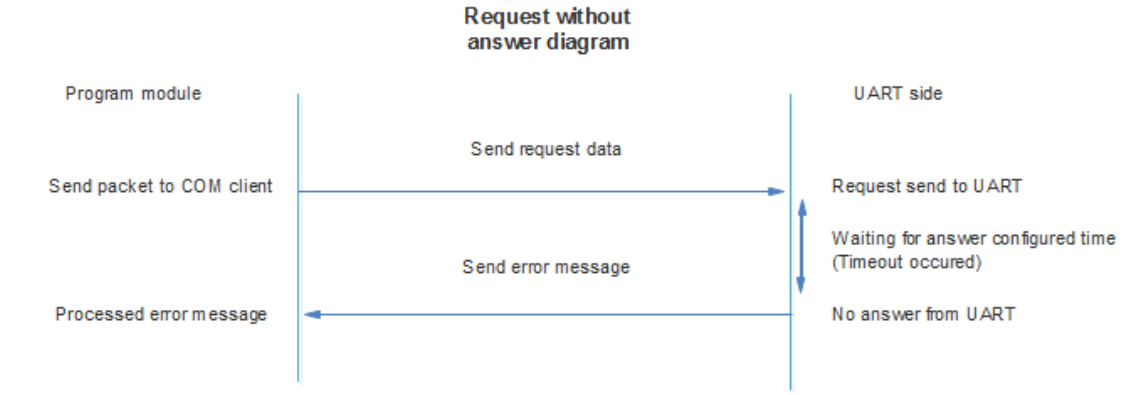
## 5.1 Virtual COM clients

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

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

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

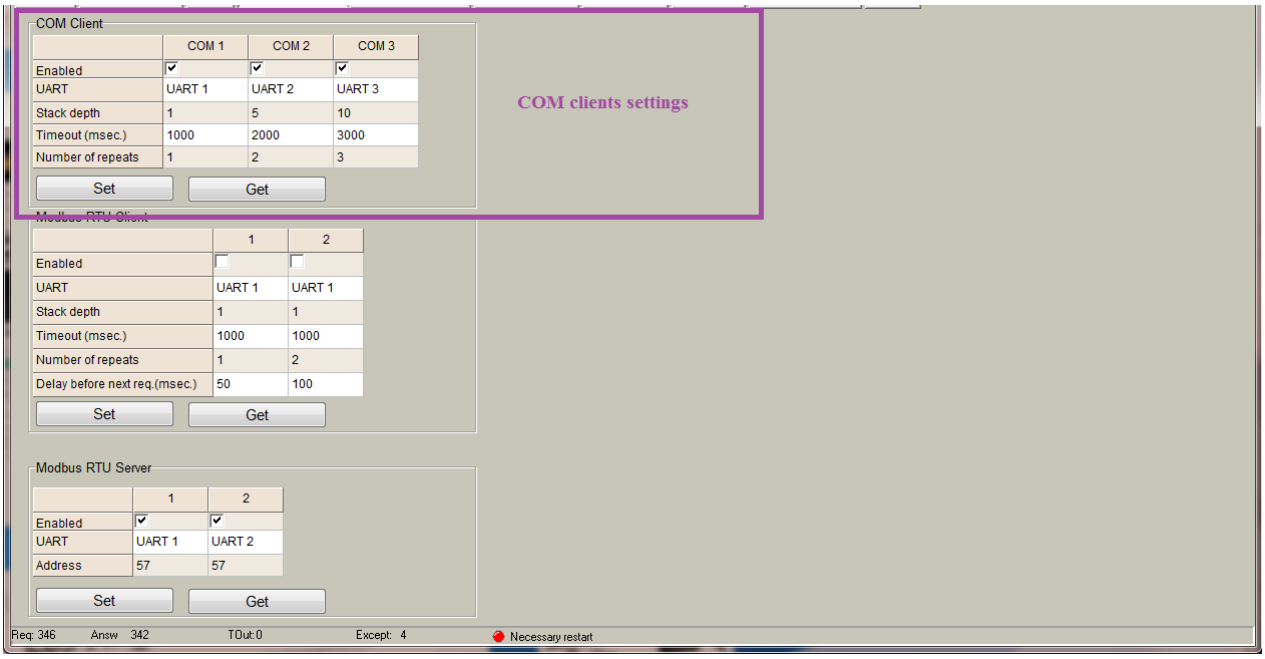



**Virtual COM clients configuration**
Virtual COM clients configuration is done in the *"Communication/Virtual Interfaces"* tab. The device can have up to 3 COM clients associated with different physical UARTs.

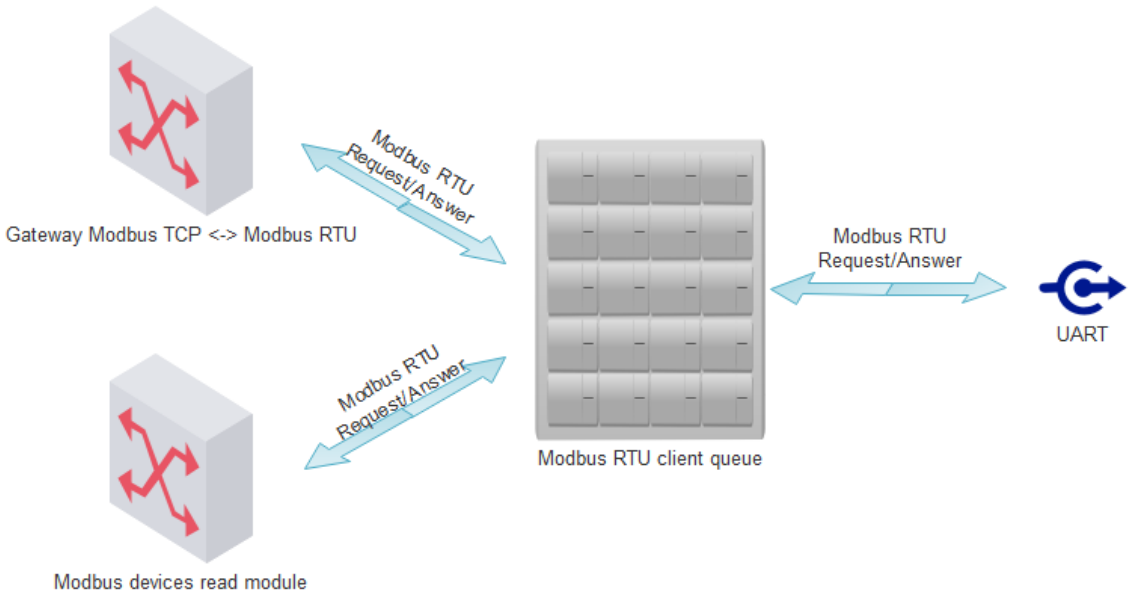
- **"Enabled" checkbox** – Enables/disables the appropriate COM client.
- **"UART"** – The physical UART associated with the appropriate COM client (UART1, UART2, UART3). The same UART cannot be used in other virtual interfaces (e.g., another COM client, Modbus RTU client, or Modbus RTU server). In the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

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




## 5.2 *Modbus RTU clients*

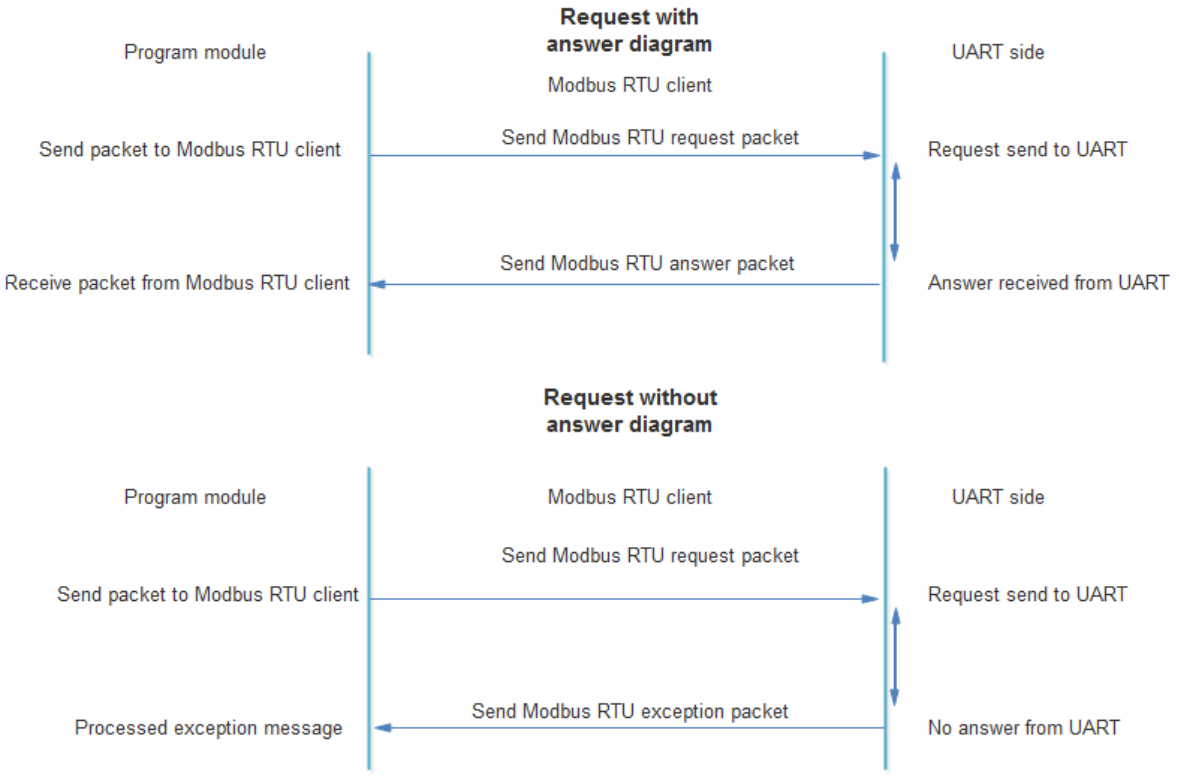
A Modbus RTU client is a virtual interface used to associate Modbus RTU devices connected to a physical UART with internal Modbus modules. It allows more than one module to access the UART simultaneously to send Modbus requests and receive responses.




Modbus RTU clients are used with the following modules:

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

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

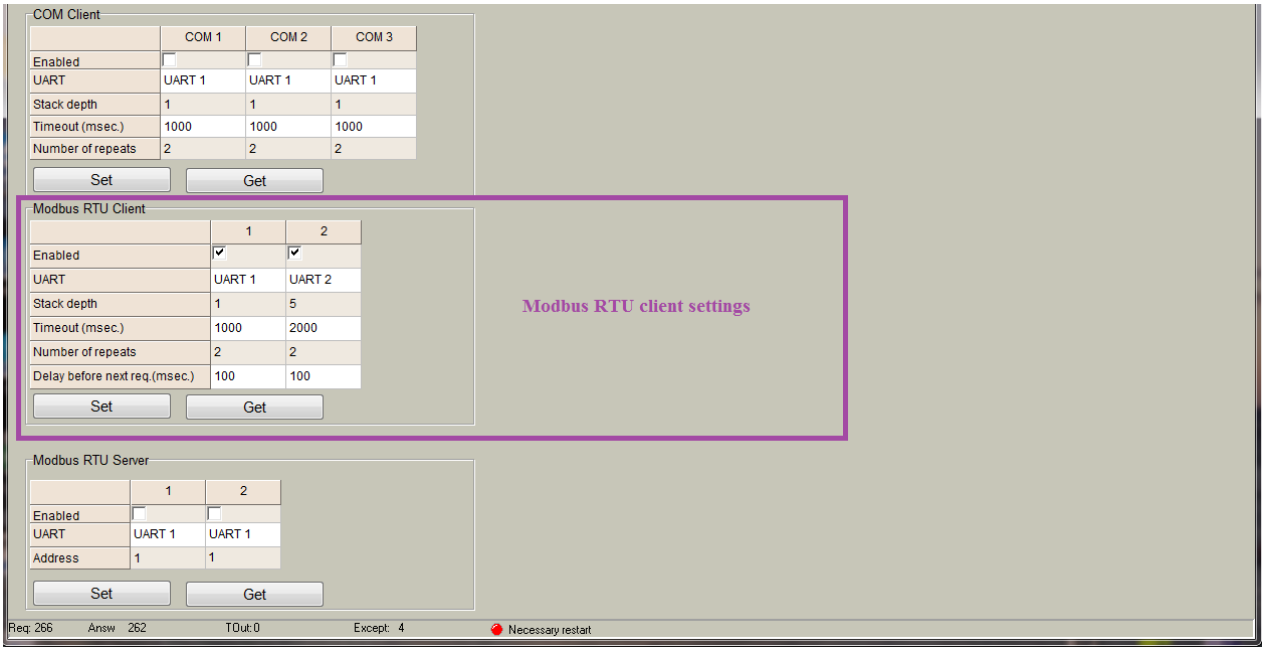



**If the response is received after the Modbus RTU client's timeout, the data is lost.**

**Modbus RTU clients configuration**

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

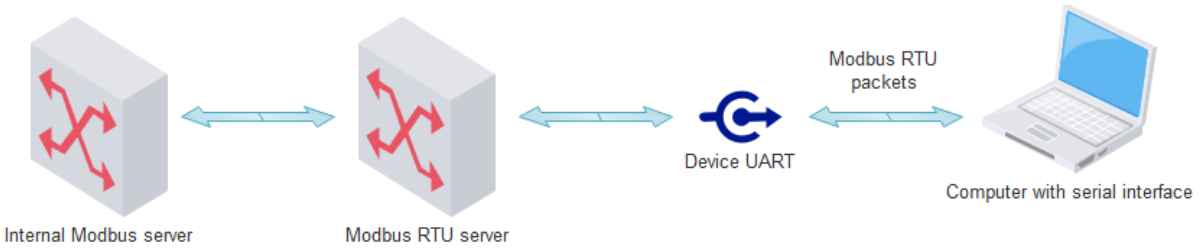
- **"Enabled" checkbox** – Enables/disables the appropriate Modbus RTU client.
- **"UART"** – The physical UART associated with the appropriate Modbus RTU client (UART1, UART2, UART3). The same UART cannot be used in another virtual interface (e.g., another COM client, Modbus RTU client, or Modbus RTU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




## 5.3 Modbus RTU servers

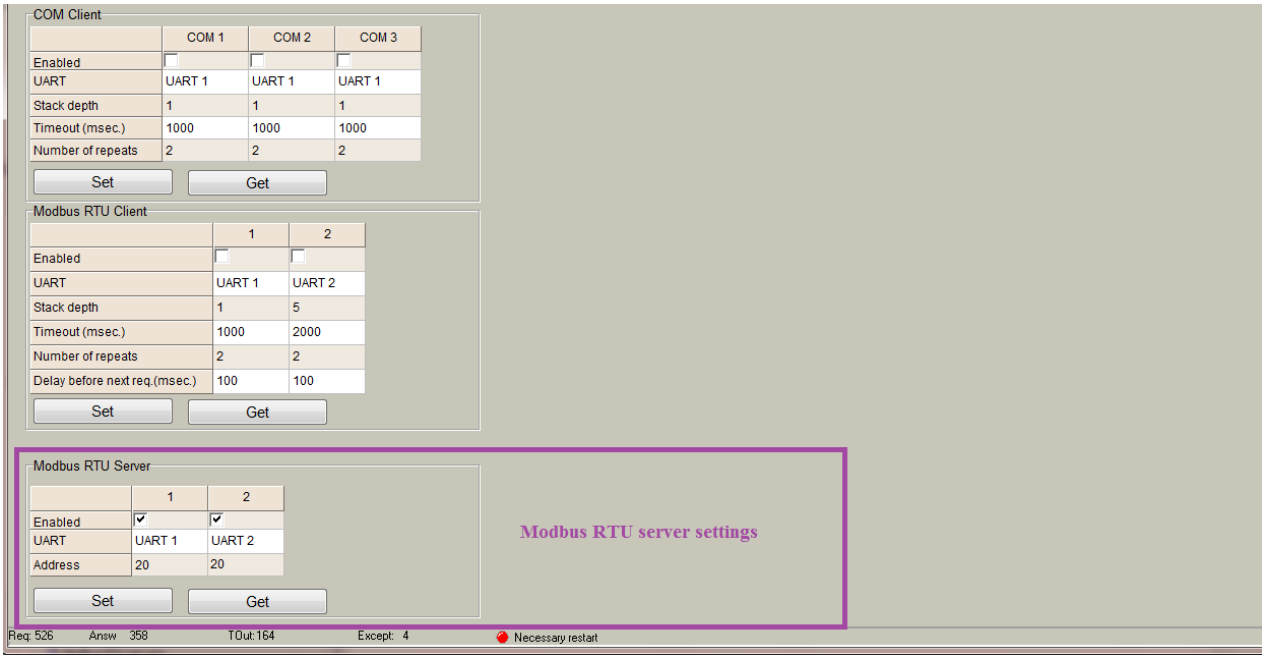
A Modbus RTU server is a virtual interface between the internal Modbus server and physical UARTs. It allows users to read the device's internal Modbus register area using a serial connection.
Modbus RTU servers can be used for:

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




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

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


COM Client
COM 1
COM 2
COM 3
Enabled
UART
UART 1
UART 1
UART 1
Stack depth
1
1
1
Timeout (msec.)
1000
1000
1000
Number of repeats
2
2
2
Set
Get
Modbus RTU Client
1
2
Enabled
UART
UART 1
UART 2
Stack depth
1
5
Timeout (msec.)
1000
2000
Number of repeats
2
2
Delay before next req.(msec.)
100
100
Set
Get
Modbus RTU Server
1
2
Enabled
UART
UART 1
UART 2
Address
20
20
Set
Get
Modbus RTU server settings
Req: 526
Answ 358
TOut:164
Except: 4
Necessary restart

# 6. Manufacturer's warranty

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

This warranty supersedes all other warranties, whether expressed or implied, including implied warranties of merchantability or fitness for a particular purpose, whether arising by law, custom, or conduct. The remedies provided under this warranty are exclusive and replace any other rights or remedies. ADVANTIC SISTEMAS Y SERVICIOS S.L. shall not, under any circumstances, be held liable for any consequential or incidental damages. If you believe your product is defective and still under warranty, please contact ADVANTICSYS at info@advanticsys.com or by phone at +34 914221023. After confirmation from our support team that the product is defective, we will issue a Return Merchandise Authorization (RMA) number and arrange for the replacement of your product.

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

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