SPC-31
User Manual

03/2025

Table of contents

1. Abbreviations 3
2. Preface 4
2.1 Symbols 4
2.2 Connecting to device 4
2.2.1 USB connection 4
2.2.2 Bluetooth connection 6
2.2.3 Configuration tool software 7
3. Program modules 9
3.1 Archives 9
3.1.1 Events archive 9
3.1.2 Diagnostic archive 10
3.1.3 User-defined archive 11
3.2 TCP modules 13
3.2.1 FTP client 13
3.3 MQTT client 15
3.3.1 MQTT client configuration 15
4. Time settings 17
5. 4G/3G/GPRS configuration 18
6. Wired interfaces 19
6.1 Serial port 19
6.1.1 UART configuration 19
6.1.2 Modbus RTU client 20
6.1.3 Modbus RTU devices 22
6.2 Sensor inputs 24
6.2.1 Analog sensors 25
6.2.2 I2C Pressure Sensors 26
6.3 Pulse inputs 27
7. Manufacturer's warranty 29

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.
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.
Status – Device status indicating LED.
Alarm mode – In the alarm status state, the controller initiates an event notification for the user-selected discrete input mode (Alarm mode: unconnected, connected, or both events).
Central computer – A server or computer to which data can be sent.

2. Preface

2.1 Symbols

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

Symbol	Explanation
CE	With the **CE** marking on a product the manufacturer ensures that the product conforms with the essential requirements of the applicable **EC** directives.
RoHS	Directive on the restriction of the use of certain hazardous substances in electrical and electronic equipment 2002/95/EC. Commonly referred to as the Restriction of Hazardous Substances Directive or **RoHS**)
	Waste Electrical and Electronic Equipment Directive

2.2 Connecting to device

The USB port is used for local configuration of the device. It is also possible to configure the device via Bluetooth or 4G/3G modem. All configuration is done using the Modbus protocol and the device configuration tool software, which can be downloaded from the manufacturer's website.

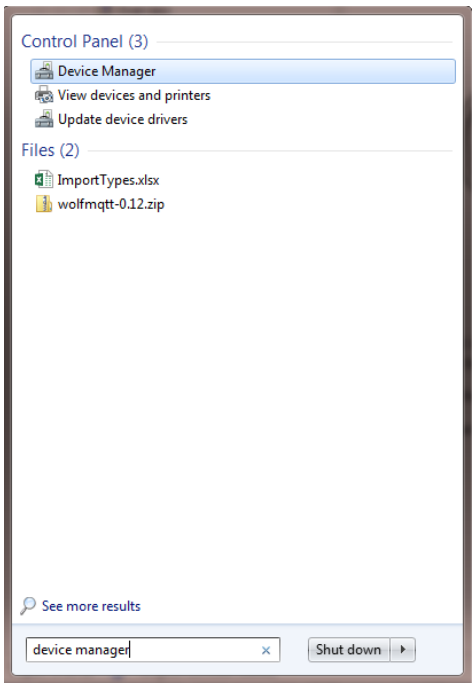
2.2.1 USB connection

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

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

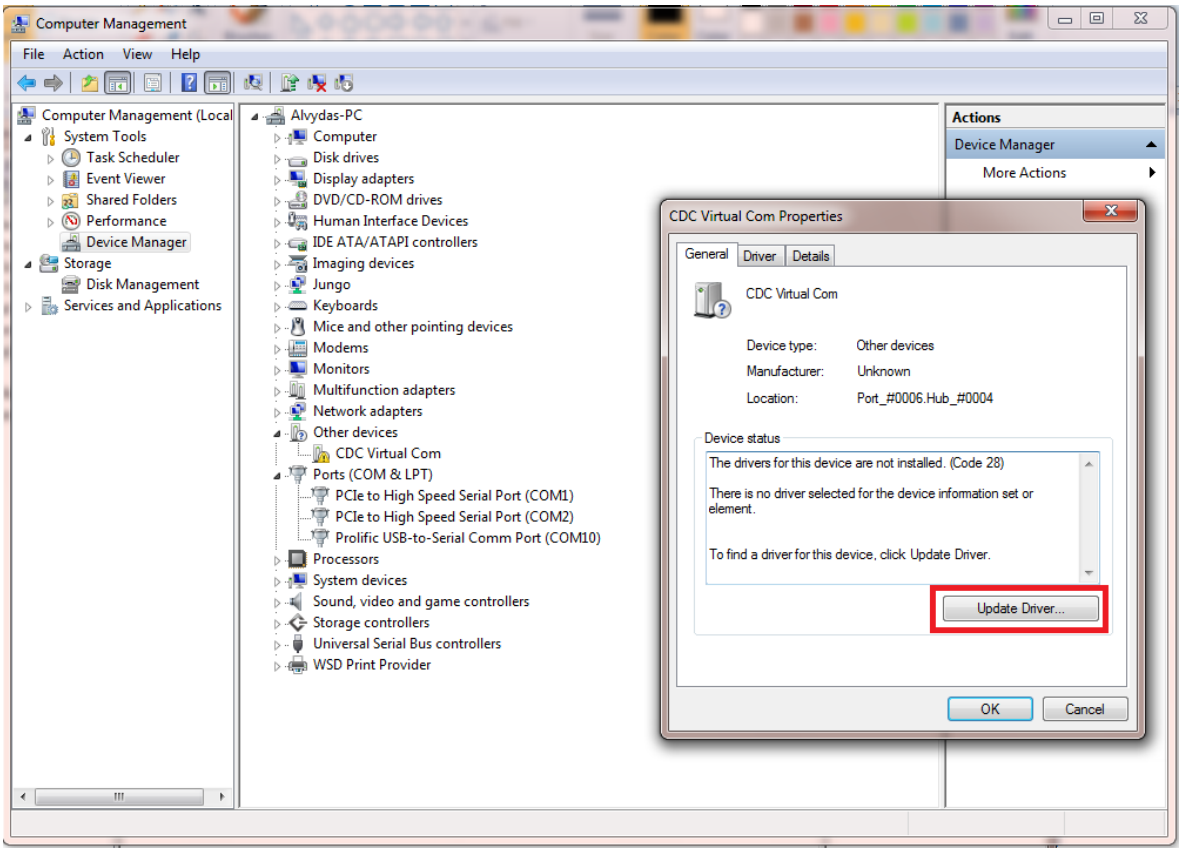
If the USB drivers are not installed automatically, you need to install them manually. Follow these steps:

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



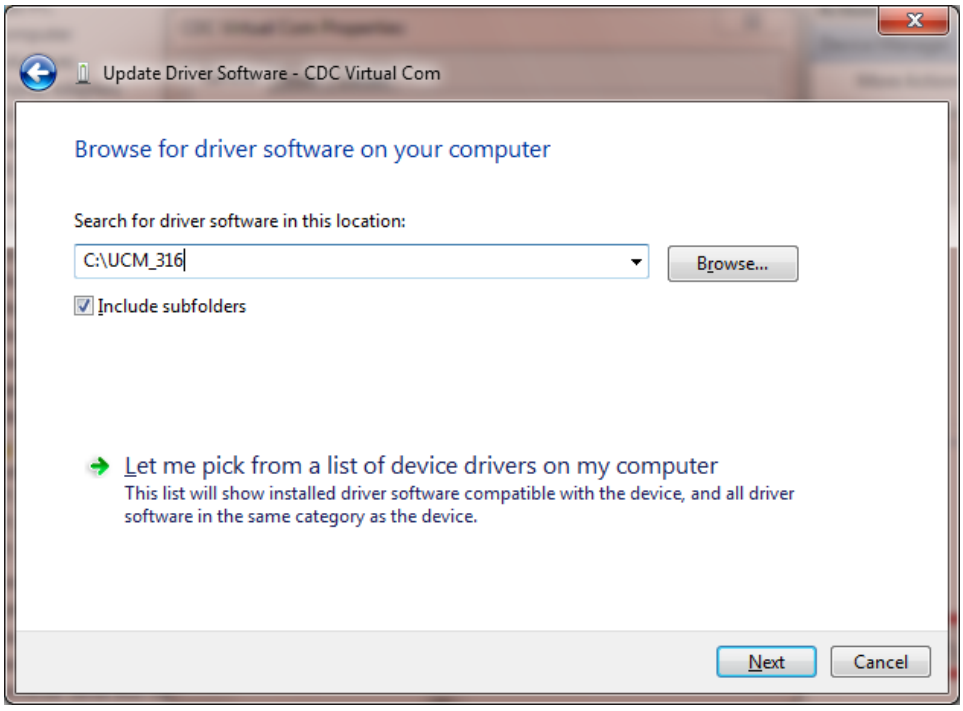
2. Double-click the device category and then double-click the device you want.

3. Click *Update Driver* and follow the instructions.



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

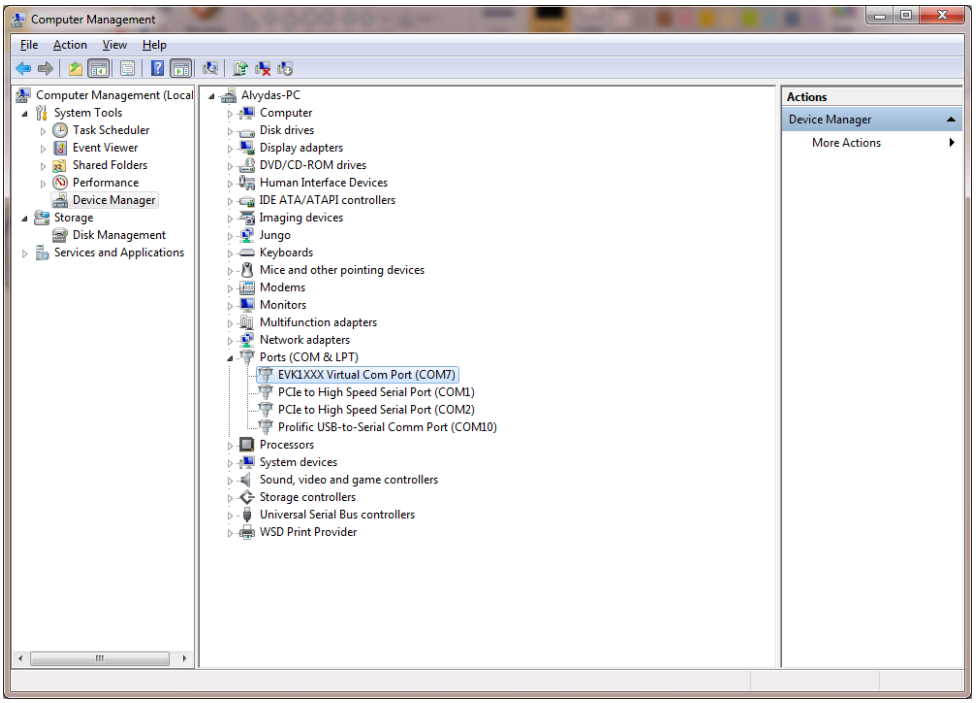
5. Click *Next*.



6. Wait while Windows installs the driver. If you see the message, "Windows can't verify the publisher," select

Install this driver software anyway.

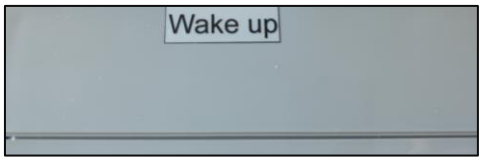
7. After installation, you will see something like "EVK1XXX Virtual Com Port" and a COM port number. Use this COM port to connect with the configuration tool.



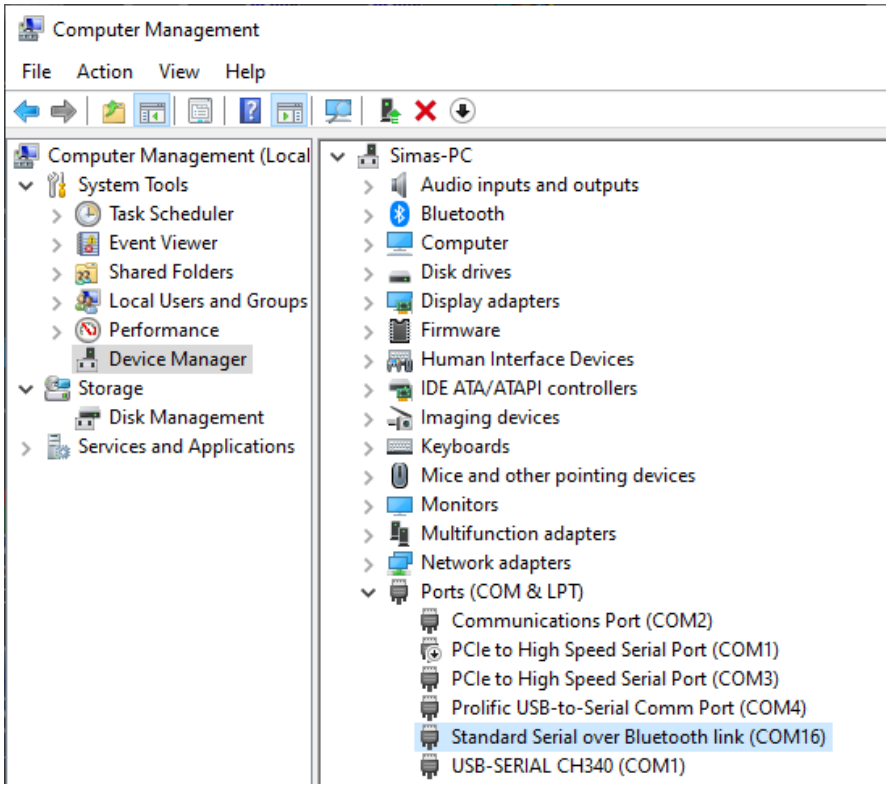
On some Windows versions (Windows 8, Windows 10), you may need to disable third-party driver signature checking before installing the device driver. Please check online for instructions on how to do this.

2.2.2 Bluetooth connection

1. Place a magnet near the "WAKE UP" area of the data logger. This powers on the device and activates the Bluetooth module



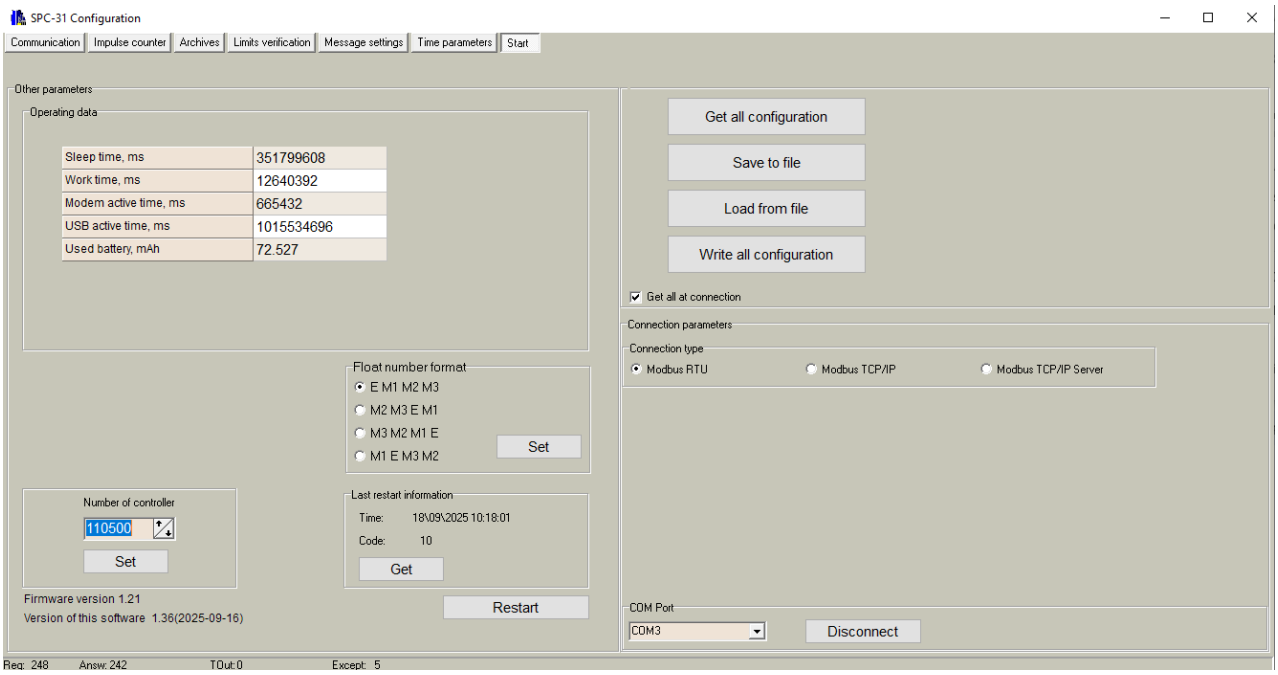
2. On a Windows PC, navigate to “Bluetooth and other devices” and add the newly detected device (commonly identified as BT836, though this may vary by OS version).
3. Once paired, a new COM port is automatically created.
4. The COM port can then be used with the configuration tool.



2.2.3 Configuration tool software

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

- *Get all configuration* button: Reads all configuration from the device.
- *Save to file* button: Saves all configuration to a file, so it can be loaded back to the device later.
- *Load from file* button: Loads saved configuration from a file.
- *Write all configuration*: Writes the loaded configuration to the device.
- *Get all at connection* checkbox: Reads all configuration when connecting over a TCP/IP connection.
- Connection type *Modbus RTU*: Connects to the device over USB or serial port.
- Connection type *Modbus TCP/IP*: Connects to the device over a TCP/IP connection.
- *Number of controller* section: Device serial number.
- *Operating data*: Time in ms spent by the different components. Used battery since last restart
- *Float number format*: Floating-point data byte order (E - exponent, M1, M2, M3 - Mantissa).
- *Last restart information*: Time of the last reset and reset code. The reset code values are:
    1. No TCP packet over GPRS in the configured time.
    2. GPRS task stops working.
    3. Not enough heap memory.
    4. Firmware update reset.
    5. Modbus reset.
    6. Unable to connect to GPRS.
    7. External pin reset.
    8. Watchdog reset.
    9. Brownout reset.
    10. Power-up reset.
    11. No TCP packet over ETHERNET in the configured time.
    12. All TCP sockets are used (if defined UIP_RESET_ALL_CONN_USED).
    13. Periodic reset.


SPC-31 Configuration
Communication
Impulse counter
Archives
Limits verification
Message settings
Time parameters
Start
Other parameters
Operating data
Sleep time, ms
351799608
Work time, ms
12640392
Modem active time, ms
665432
USB active time, ms
1015534696
Used battery, mAh
72.527
Float number format
E M1 M2 M3
M2 M3 E M1
M3 M2 M1 E
M1 E M3 M2
Set
Number of controller
110500
Set
Last restart information
Time: 18\09\2025 10:18:01
Code: 10
Get
Restart
Firmware version 1.21
Version of this software 1.36(2025-09-16)
Get all configuration
Save to file
Load from file
Write all configuration
Get all at connection
Connection parameters
Connection type
Modbus RTU
Modbus TCP/IP
Modbus TCP/IP Server
COM Port
COM3
Disconnect
Req: 248
Answ: 242
TOut:0
Except: 5

3. Program modules

3.1 Archives

The device has several types of archives:

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

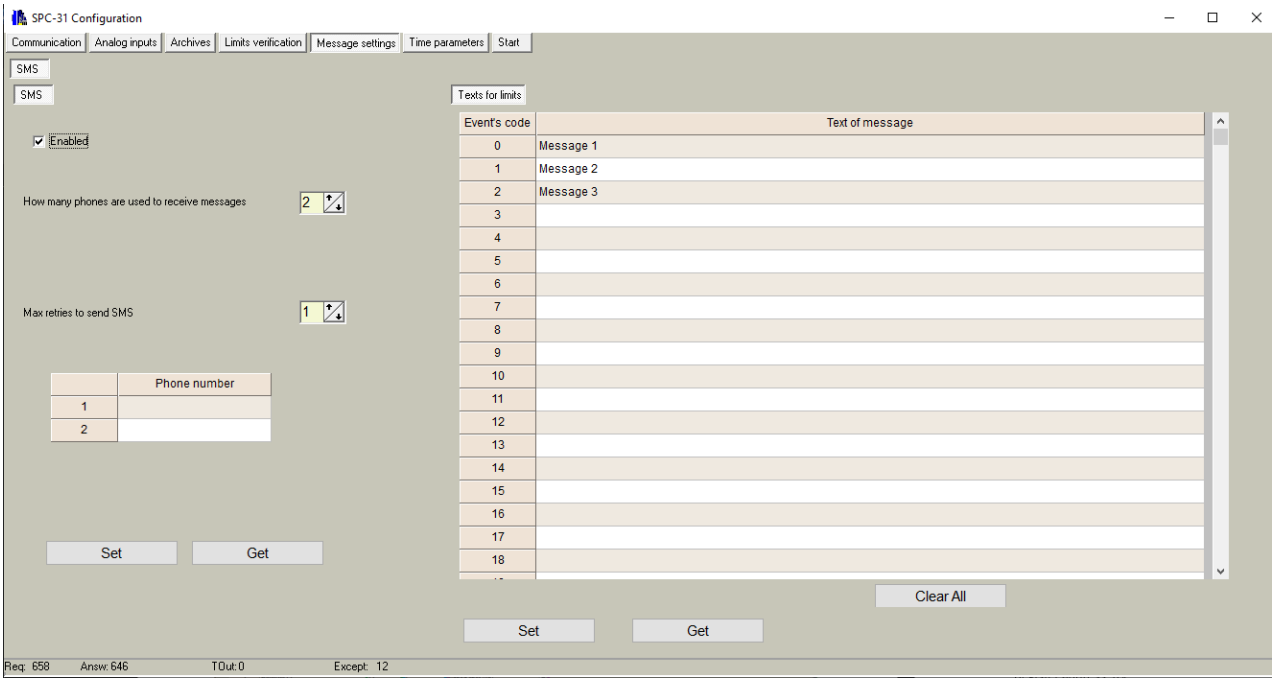
All archives are saved to the internal data flash memory. The maximum archive record count depends on the memory size and available space. If a Micro SD card is present, archives are duplicated to the SD card (a new file is created every day with a date stamp).

3.1.1 Events archive

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

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

The event archive is used to generate SMS, MQTT messages. For every event ID, you can configure the event message text. The event message text can be configured in *"Message settings"*



3.1.2 Diagnostic archive

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

Diagnostic archive record structure:

Variable name	Purpose / Value	Type of value
Time	Record time. If event type=7 then new set time	Long int (32 bits)
Event type	1 - RESET event 2 - Firmware update event 3 – Archive counter change event 4 – Automatic time correction 5 – Time change over MODBUS 6 – Change of internal parameters 7 – Time correction	Long int (32 bits)
Event value (integer)	If event type: 1. Reason of last reset, values: 1. No TCP packet over GPRS in configured time 2. GPRS task stops working 3. Not enough heap memory 4. Firmware update reset 5. Modbus reset 6. Unable connect to GPRS 7. External pin reset 8. Watchdog reset 9. Brownout reset 10. Power up reset 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

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.

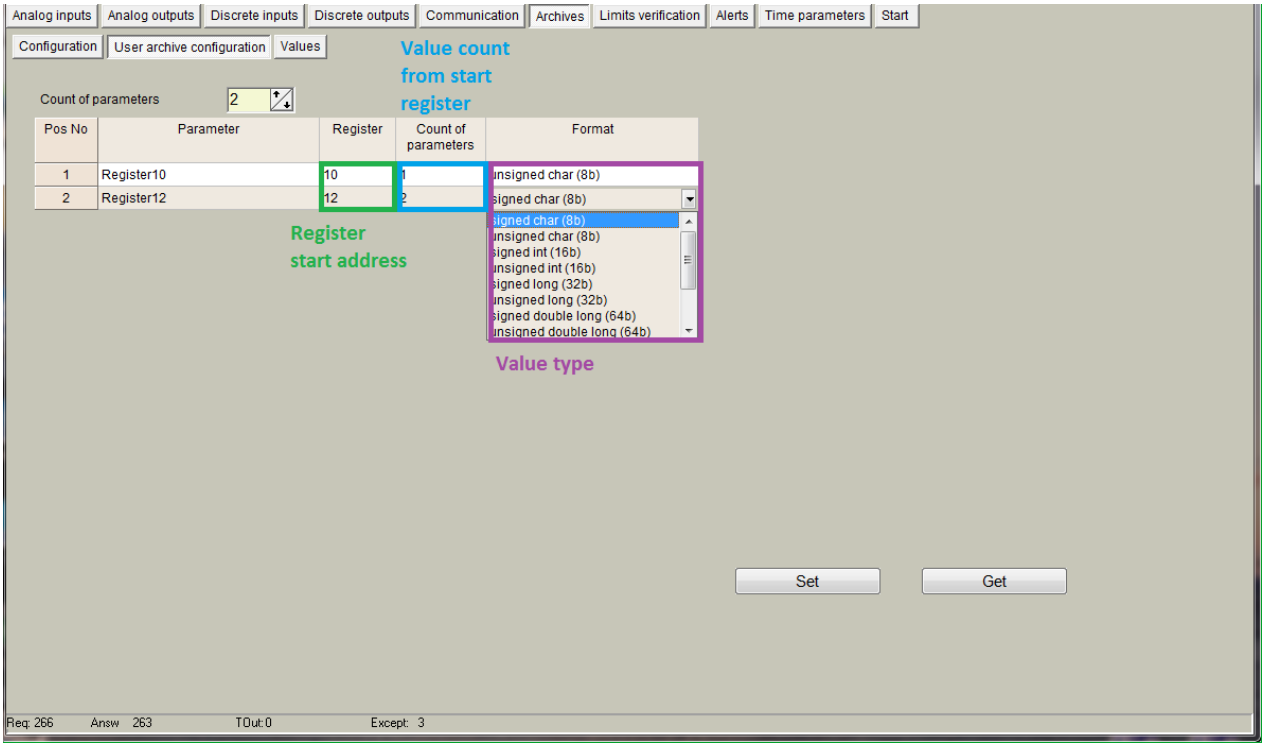
The archive period is in minutes and can range from 1 minute to 1440 minutes (24 hours). You can configure this in the configuration tool under the "Archives/Configuration" section.

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.
4. In the **Format** column, configure the data value type for the current line. Value types can include:

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



Reading user archive over Modbus file system

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

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

Diagnostic archive record structure:

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

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

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

To read the 5th oldest record:

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

3.2 TCP modules

3.2.1 FTP 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.

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>;
```

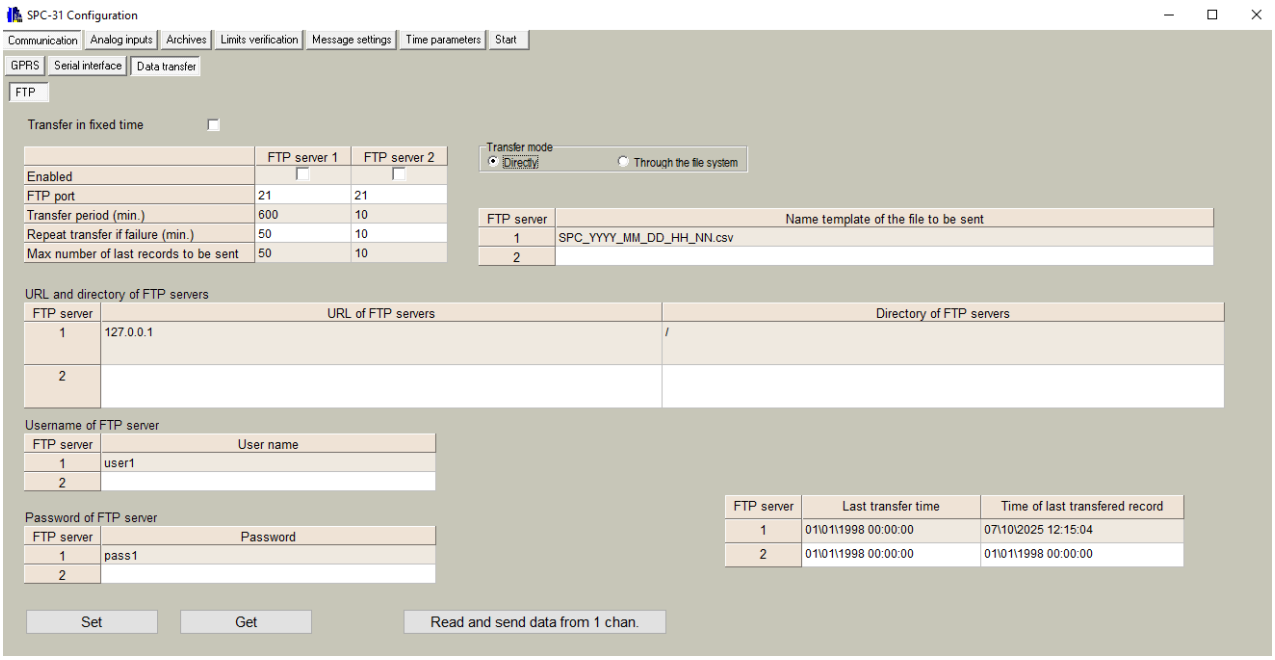
- **<Value Header>** – Configured header from the *"Archives/User archive configuration"* tab.
- **<Value data>** – Configured parameter value taken from the user archive.
- **<Value dimension>** – Configured dimension from the *"Archives/User archive configuration"* tab.
- **<Record Date/Time>** – Stored archive record time.

FTP client configuration
A CSV file can be sent to up to 2 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.

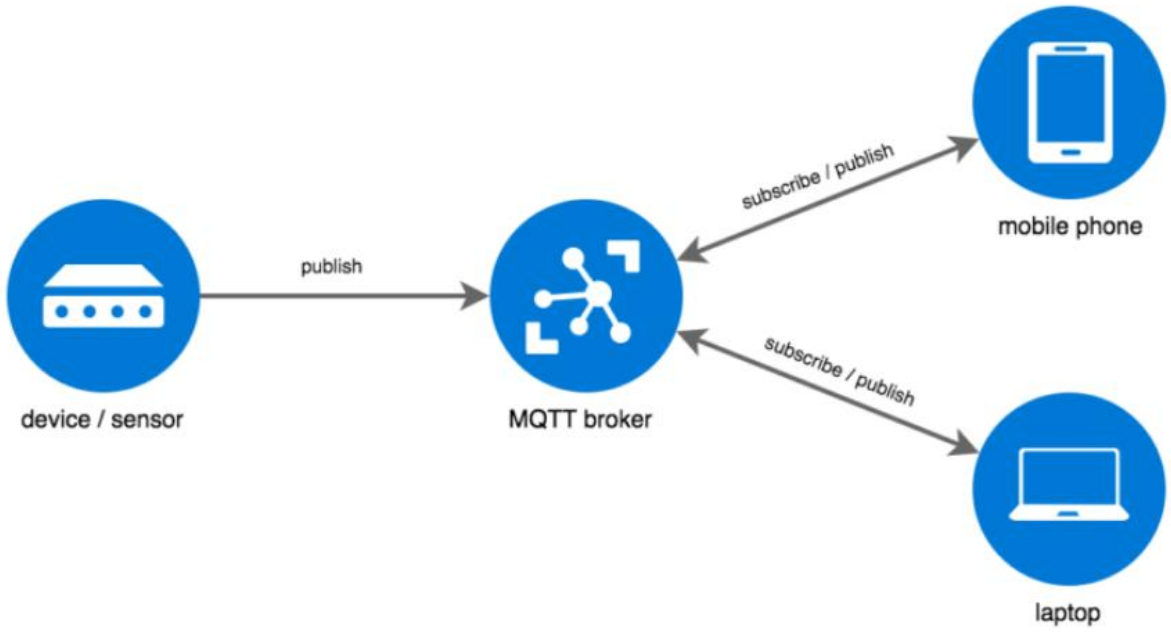
- **"Passwords"** section:
  - "Password" column: The password for the appropriate FTP server.
- **"File Send Status"** section:
  - **"Status"** column: The current status of the FTP client. After the file is sent to the server, the status changes to "File transmitted". There are several other statuses 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.



3.3 MQTT client

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

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



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

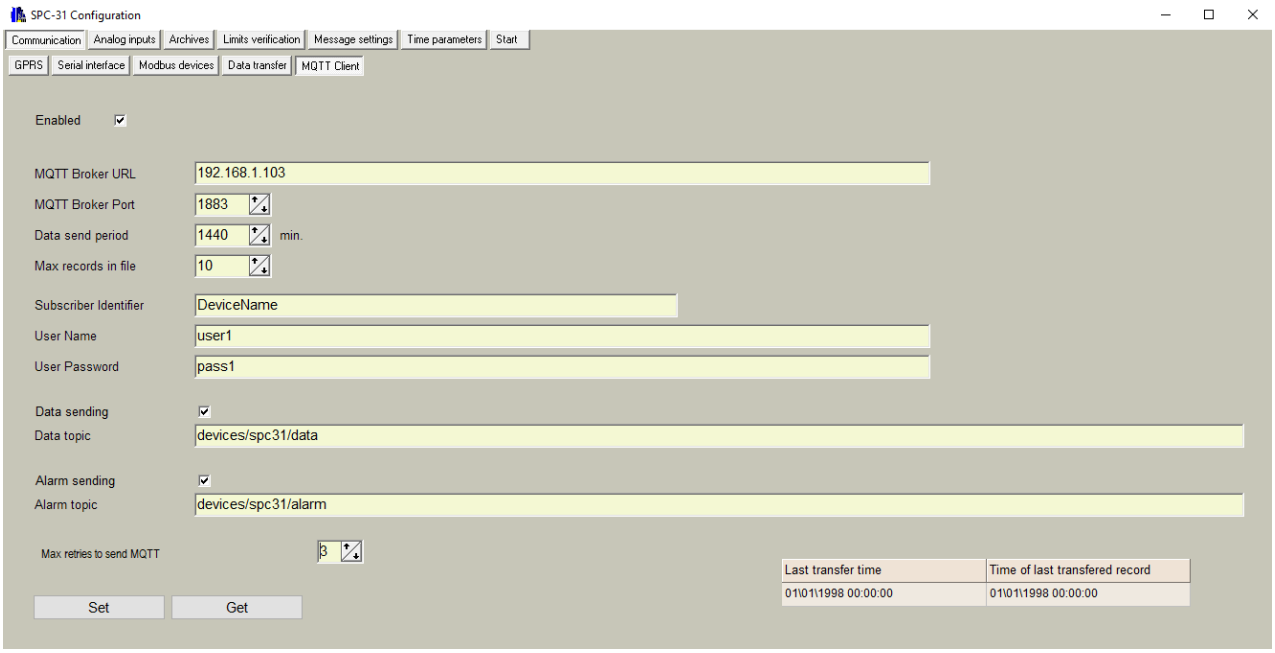
3.3.1 MQTT client configuration

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

- **"Enabled"** - Activates the MQTT client functionality.
- **"MQTT Broker URL"** - URL or IP address of the external MQTT broker
- **"MQTT Broker Port"** - Port used by the broker, typically 1883 for non-secure connections.
- **"Data send period"** - Defines how frequently data will be sent to the MQTT broker, in minutes. Minimum value is 1. Default is 1440 (once per day).
- **"Max records in file"** - Maximum number of records stored in a single transmission file. Used for batching messages.
- **"Subscriber Identifier"** - Optional identifier for the MQTT client. Often used when multiple clients share the same broker.
- **"User Name"** - Username for authenticating with the MQTT broker (if required).
- **"User Password"** - Password for authenticating with the MQTT broker.
- **"Data sending"** - Enable to allow data transmission via MQTT.
- **"Data topic"** - MQTT topic to which the data messages are published. For example, *devices/spc31/data*
- **"Alarm sending"** - Enable to allow alarm events to be sent via MQTT.
- **"Alarm topic"** - MQTT topic to which alarm messages are published. For example,

devices/spc31/alarms.

- **“Max retries to send MQTT”** - Specifies how many times the client will retry sending a message if the MQTT transmission fails.
- **"Set" button** – Writes the configuration to the device.
- **"Get" button** – Reads the configuration from the device.



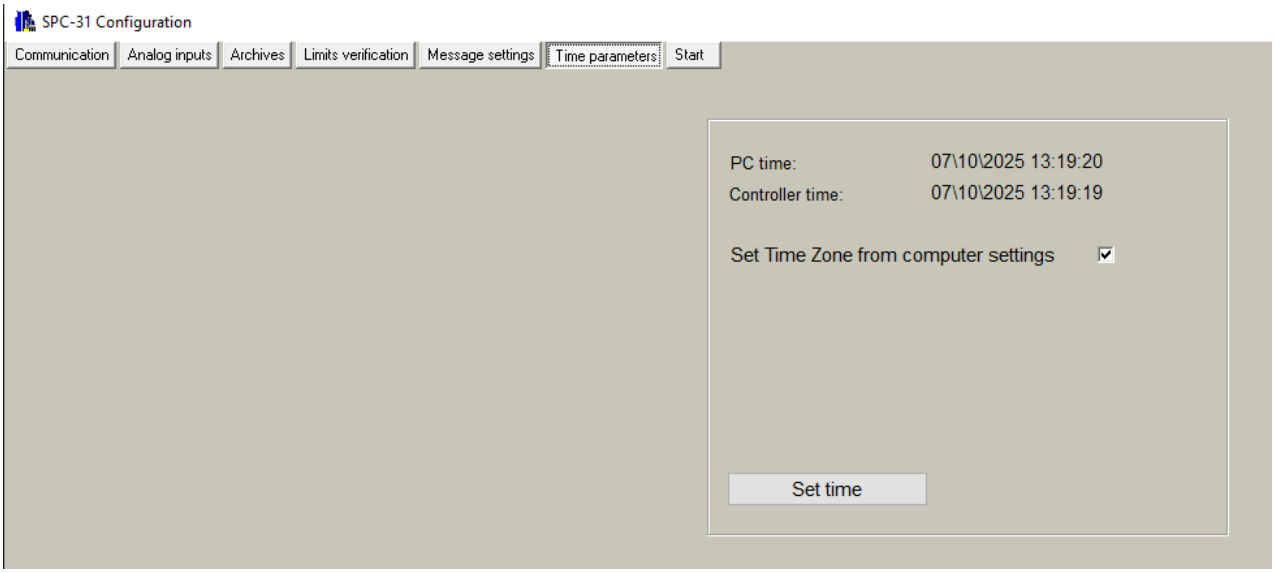
4. Time settings

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.



5. 4G/3G/GPRS configuration

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

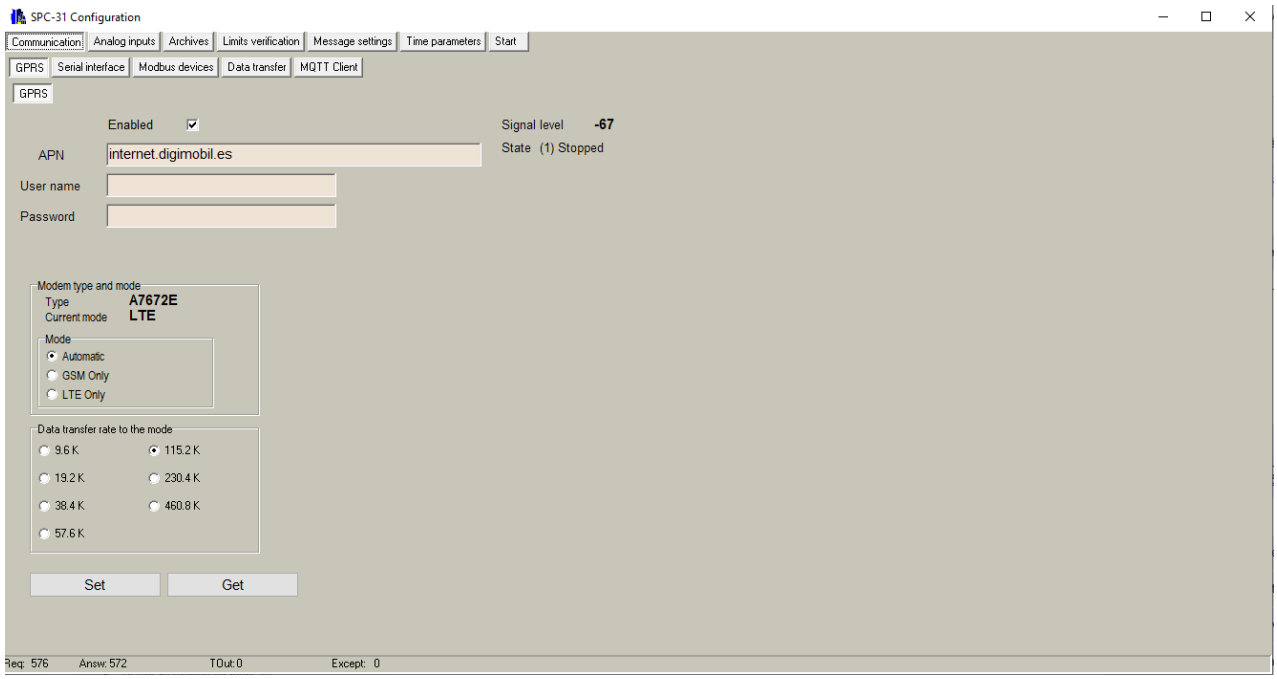
- Data transfer
- Event transfer
- Clock time synchronization
- Device configuration
- Firmware upgrade
- Etc.

Supported Services:

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

4G/3G/GPRS configuration

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



- **Connection Mode Section**:
  - **"Enabled" checkbox** – Enables or disables modem use.
  - **Modem type and mode**:
    - **"LTE Only"** – Connects only in 4G data mode.
    - **"GSM Only"** – Connects only in GSM mode (GSM data calls).
    - **"Automatic"** – Mixed mode, where the device first tries to connect to GPRS. If the connection fails, it stays in GSM mode and after a timeout, tries to connect to GPRS

again.
    - **Signal Level Measurement**:
        - **"Signal level"** – The measured signal level. 51 dBm is the best signal, and 113 dBm is the worst signal.
- **Connection Settings Section**:
    - **"APN"** – The Access Point Name (APN) is the name of a gateway between a GSM, GPRS, 4G/3G, or 4G mobile network and another computer network, frequently the public Internet. A device making a data connection must be configured with an APN to present to the carrier.
    - **"User name"** – The network username.
    - **"Password"** – The network password.
- **"Set" button** – Writes the configuration to the device.
- **"Get" button** – Reads the configuration from the device.

6. Wired interfaces

6.1 Serial port

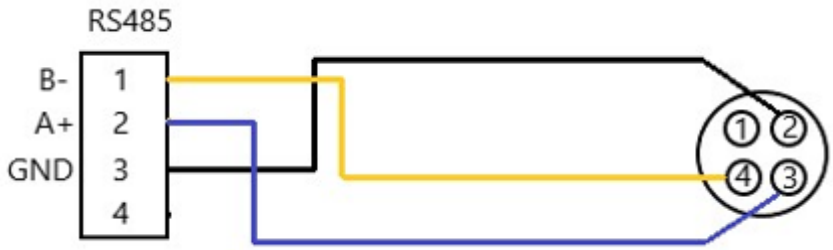
6.1.1 UART configuration

One serial bus connection is available for the connection of RS485 meters, Modbus devices and other devices.

UART characteristics:

Supported baud rates	Supported parity	Supported data bits	Supported stop bits
300 - 38400	Even, Odd, None	7,8	1,2

Wiring diagram



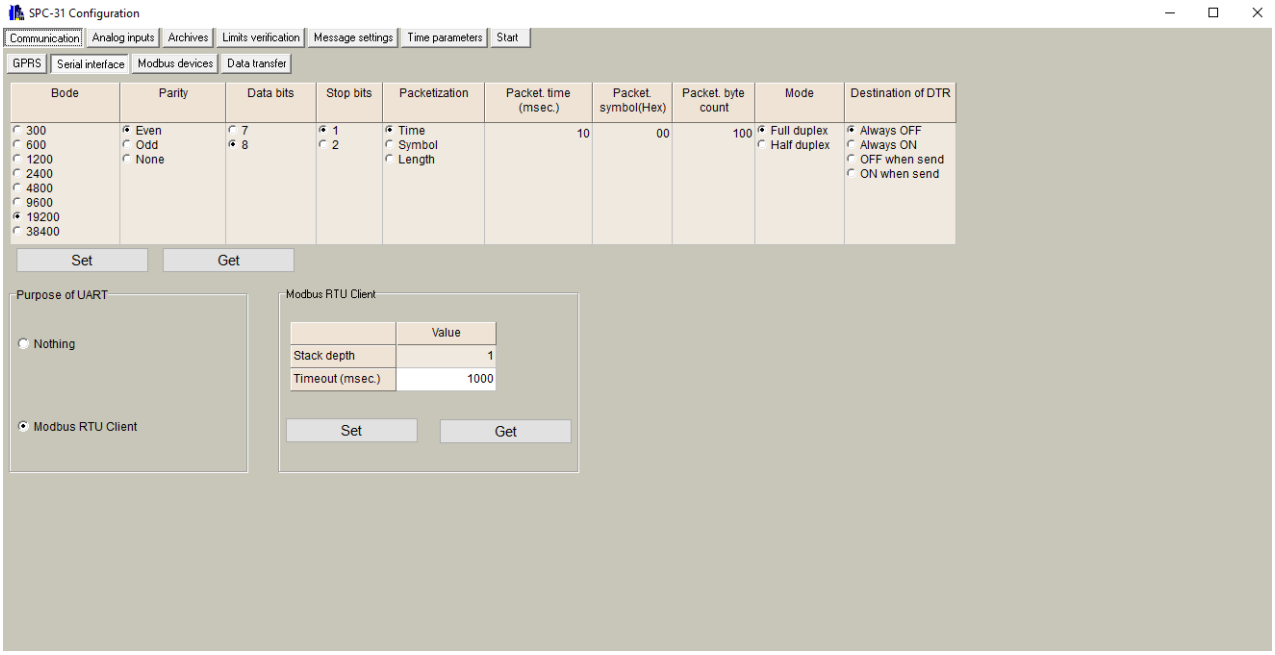
UART settings

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

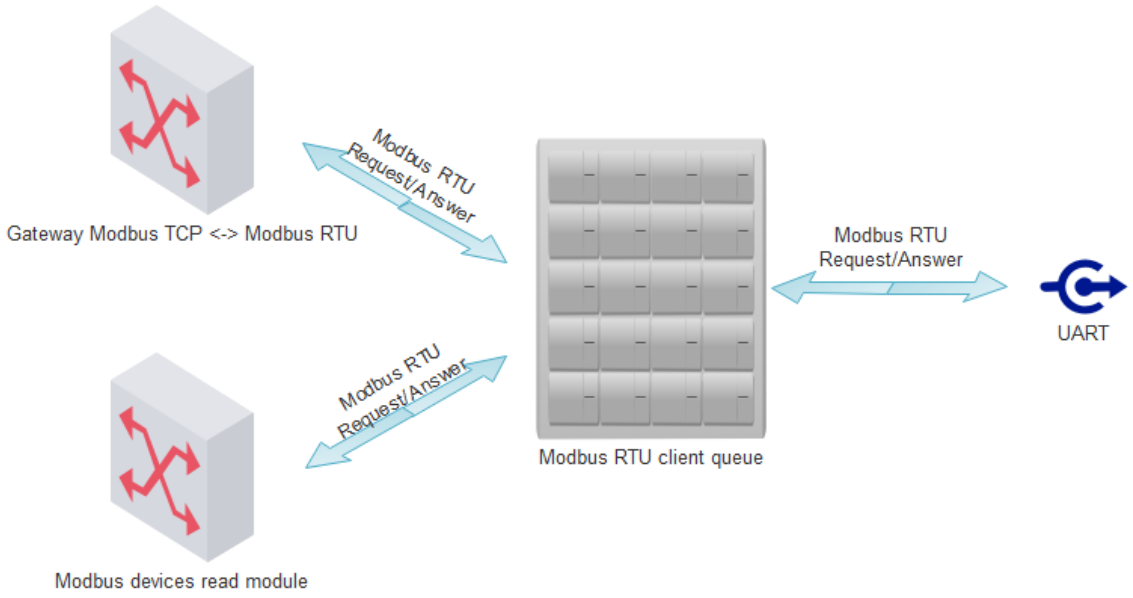
selected.

- **"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.

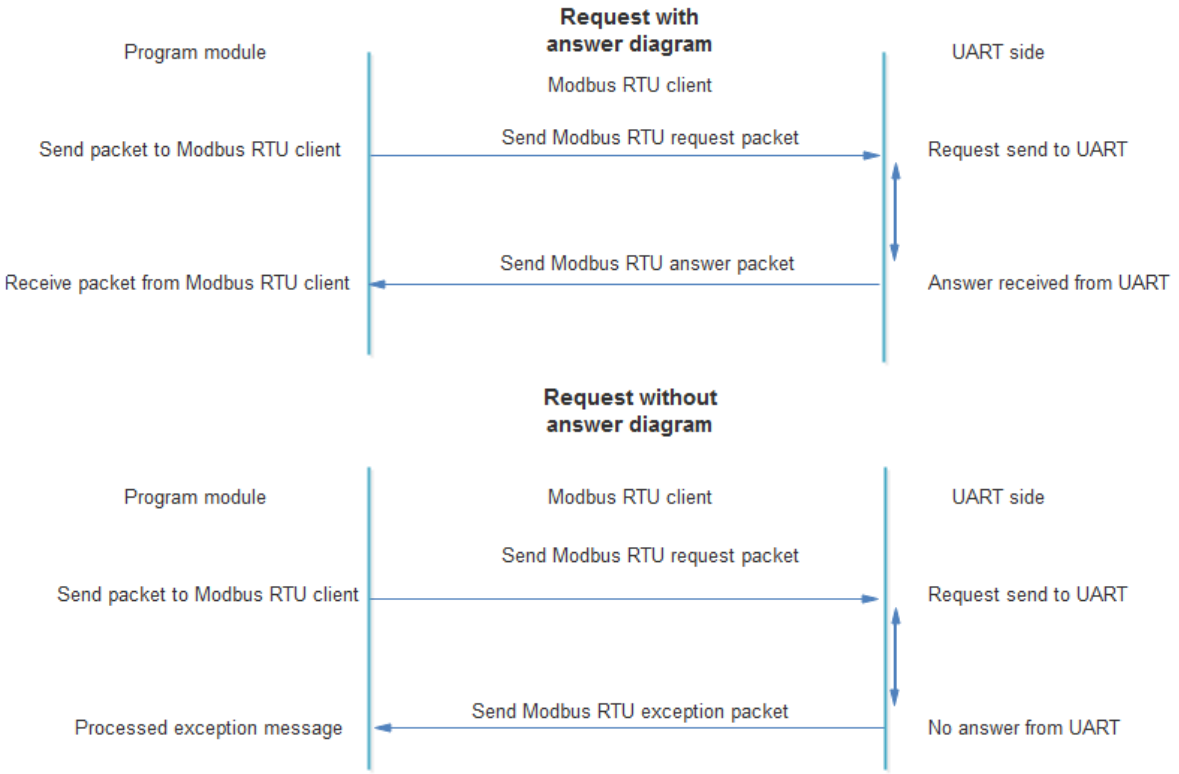


6.1.2 Modbus RTU client

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.



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.

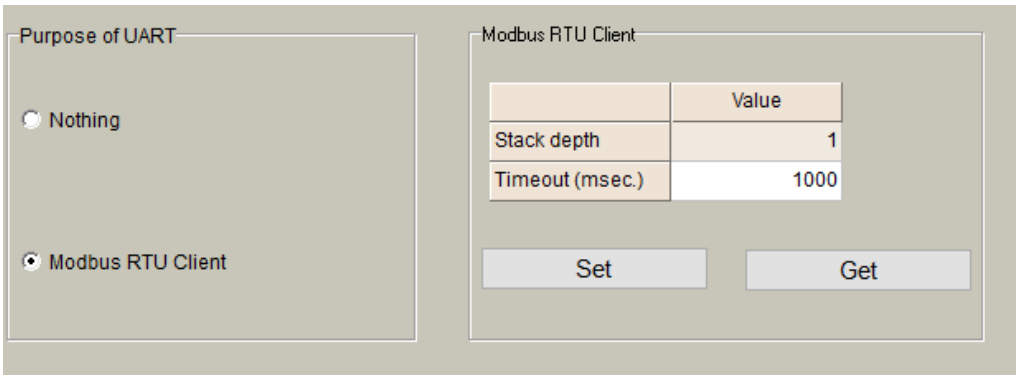


If the response is received after the Modbus RTU client's timeout, the data is lost.

Modbus RTU clients configuration

Modbus RTU client configuration is done in the “Purpose of UART” and “Modbus RTU Client” section within

"Communication/Serial interface" tab.



The device has one Modbus RTU client associated with the physical UART.

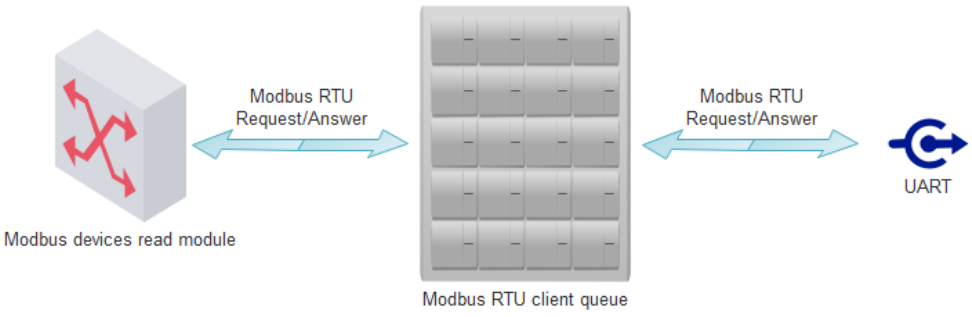
- **"Stack depth"** – The Modbus RTU client queue length (1-10). This parameter determines how many packets can be processed at the same time.
- **"Timeout (msec)"** – The wait time for a response from the UART (in milliseconds, 1-30000).
- **"Set" button** – Writes the configuration to the device.
- **"Get" button** – Reads the configuration from the device.

6.1.3 Modbus RTU devices

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

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

Supported Modbus device IDs range from 1 to 240.
The Modbus devices read module generates Modbus RTU requests and sends them to the appropriate Modbus RTU client. The Modbus RTU client forwards these requests to the configured UART. For details on configuring Modbus requests, refer to the *"Modbus RTU Devices Configuration"* section.

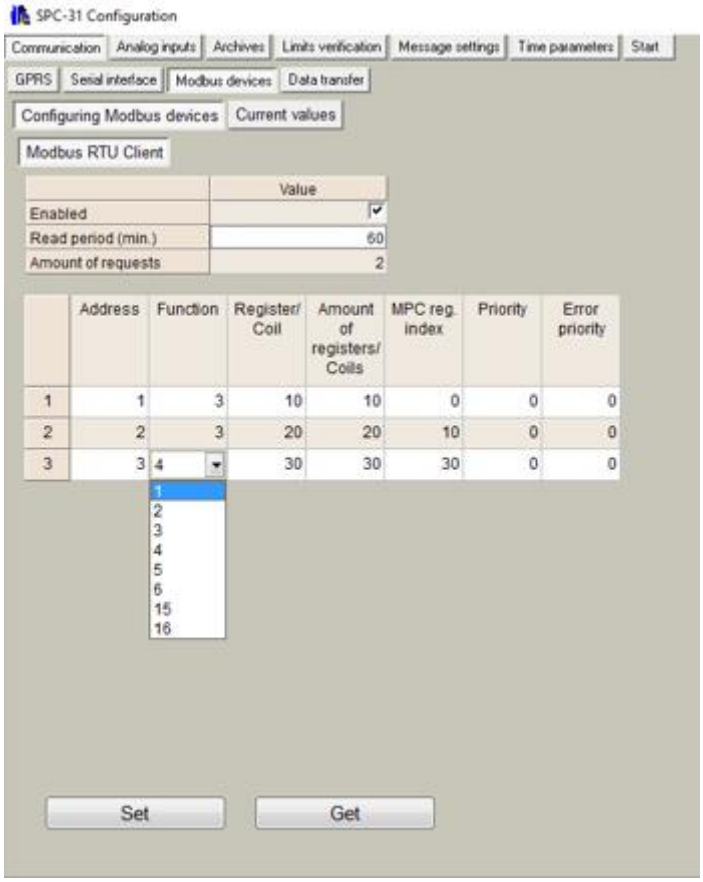


Modbus RTU devices configuration
Modbus RTU devices configuration is done in the *"Communication/Modbus Devices/Configuring Modbus Devices/Modbus RTU Client"* tab. Before configuring your Modbus RTU devices, first configure the Modbus RTU clients, which will link the physical UART with the Modbus RTU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

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

Modbus requests table:

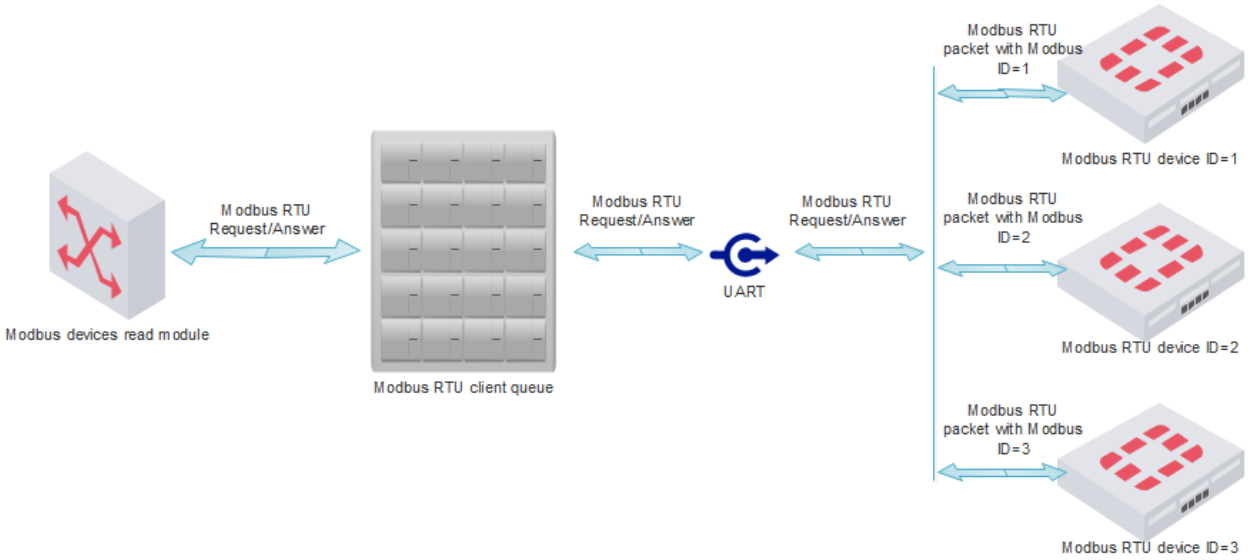
- **"Address"** – Modbus RTU device ID.
- **"Function"** – Modbus function for the current request (see the list of supported functions above).
- **"Register/Coil"** – Start register or coil address.
- **"Amount of registers/Coils"** – Indicates how many registers or coils to read starting from the specified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



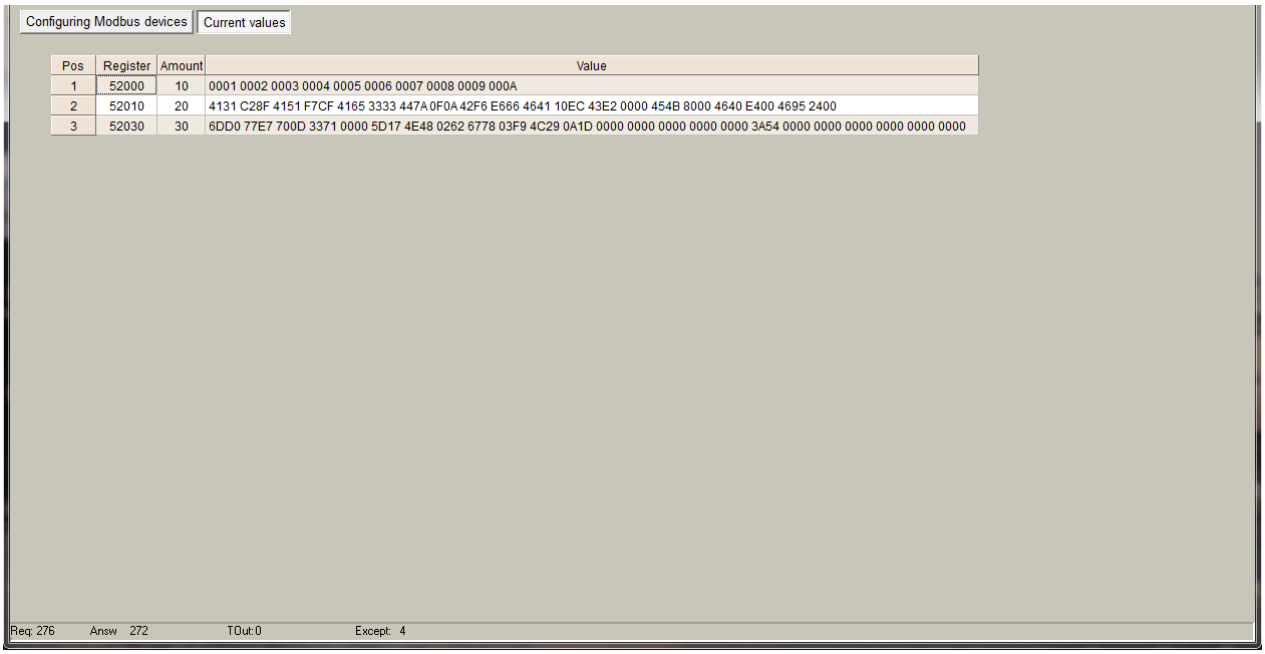
Example

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

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



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



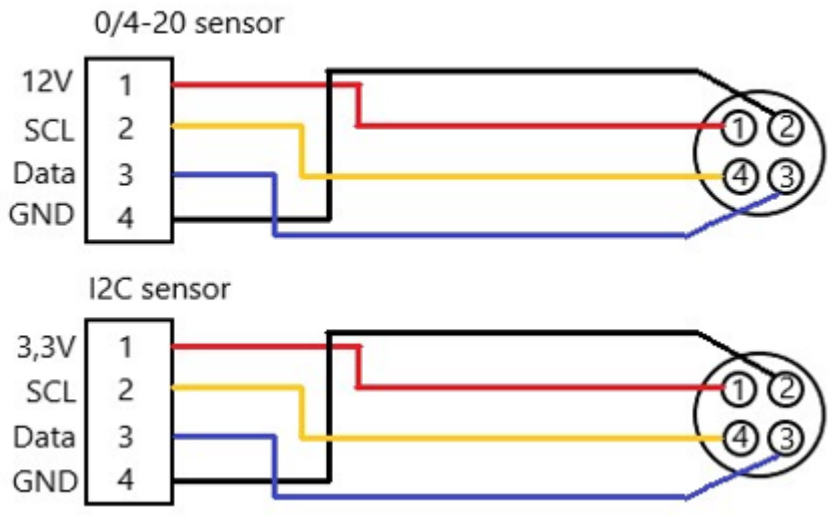
6.2 Sensor inputs

The device has two I2C, voltage, or current analog inputs. Each input can be used as:

- **Current 0/4..20mA** analog input
- **Voltage 0..+5V** analog input
- **Voltage 0..+10V** analog input
- **I2C** digital input

The purpose of the input is specified when ordering the device.

Wiring diagram



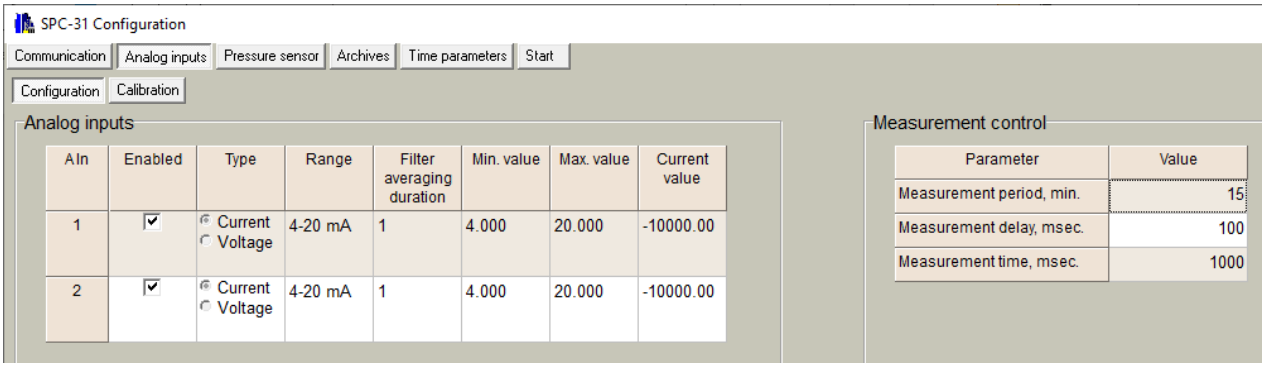
The purpose of the input is selected when you order the device, and different types of sources cannot be connected to the same input.

6.2.1 Analog sensors

Analog sensor inputs configuration is done in the *"Analog inputs"* tab:
Configuration

To configure the analog inputs, go to *"Analog Inputs > Configuration":*

- **Type and Range**: Select the sensor's output range.
- **Min. value / Max. value**: Define the sensor's operational range.
- **Current value**: Displays the current sensor reading, adjusted based on the configured operational range.
- **Measurement period:** Configurable from 1 min to 44,640 min
- **Measurement delay:** 10–1000 ms delay before starting measurement (allows sensor warm-up; check your sensor's response time)
- **Measurement time:** Configurable from 100 to 10,000 ms (only valid when powered via USB)



Analog inputs registers can be read using Modbus read file function 3.

Start register	End register	Register name	Description	Format	Type
16262	16263	Analog Input 1 Value	Current or voltage depending on the configured input type.	Float (32-bit)	Read-only
16264	16265	Analog Input 2 Value	Current or voltage depending on the configured input type.	Float (32-bit)	Read-only
16266	16267	Analog Input 1 Converted Value	Scaled or calibrated value derived from the raw input.	Float (32-bit)	Read-only
16268	16269	Analog Input 2 Converted Value	Scaled or calibrated value derived from the raw input.	Float (32-bit)	Read-only

These registers must be mapped under the user-defined archive parameters to enable transmission to external FTP or MQTT servers.

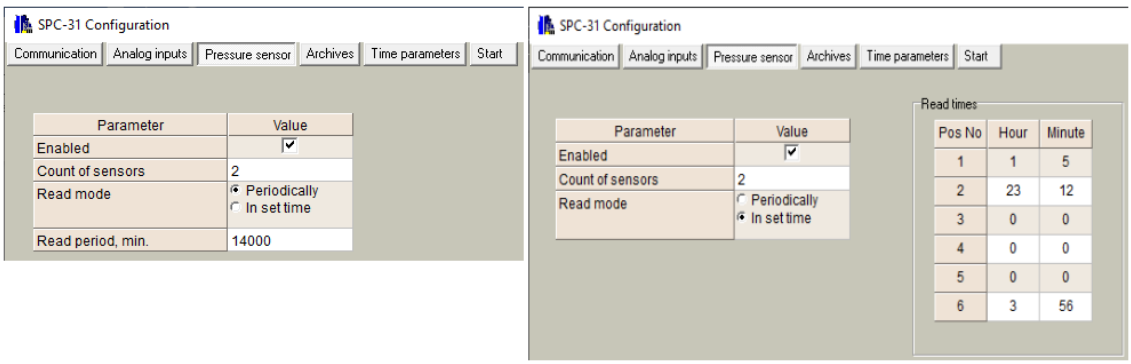
6.2.2 I2C Pressure Sensors

The device supports I2C pressure sensors with different supply voltages (specify voltage when placing the order).

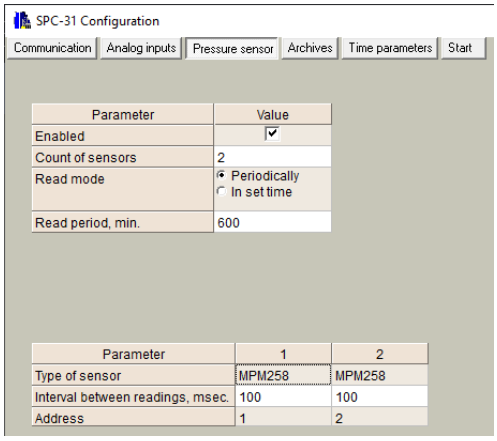
Configuration

To configure the I2C inputs, go to *"Pressure Sensors":*

- **"Enabled" checkbox** – Enables or disables the I2C ports.
- **Count of sensors** – Amount of sensors to be read from the device
- **Read mode:** Reading time can be of two types:
  - **Periodic:** from 1 minute to 43,200 minutes
  - **In set time (Scheduled):** by selecting the exact hour and minute for the sensor to be read

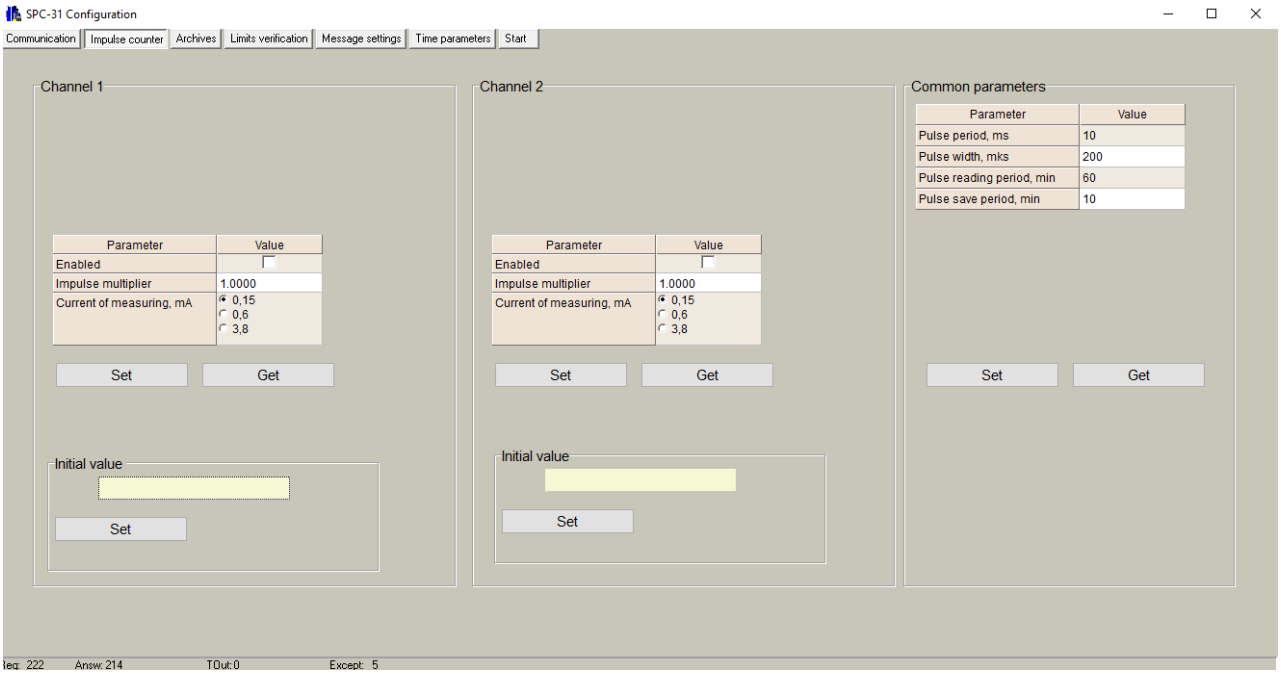


- **Type of sensor:** Select the sensor type from the list.
- **Interval between readings:** Time interval between the first and second sensor reading in milliseconds.
- **Address:** I2C address of the sensor.



6.3 *Pulse inputs*

Pulse inputs configuration is done in the *"Impulse counter"* tab. This screen allows you to configure the parameters for two pulse counter channels (Channel 1 and Channel 2), as well as general pulse timing parameters



Channel "X" configuration section

- **"Enabled" checkbox** – Enables or disables the selected channel. Check this box to activate the channel.
- **"Impulse multiplier"** – The impulse counter multiplier for the appropriate discrete input.
  - Example: If 1 pulse = 1 litre, set to 1.0000. If 1 pulse = 0.1 litre, set to 0.1000.
- **Current of Measuring**: Selects the input current for measurement. Choose depending on your sensor's output signal.
  - 0.15 mA: For low-current sensors
  - 0.6 mA: For medium-current sensors
  - 3.8 mA: For high-current sensors
- **Initial Value**: Defines the initial counter value when the system starts or resets. Enter a value (e.g., 0) before clicking Set.
- **"Set" button** – Writes the configuration to the device.
- **"Get" button** – Reads the configuration from the device.

Common Parameters section:

- **Pulse period (**ms): Minimum time interval between pulses (in milliseconds). Default: 10 ms
- **Pulse width** (µs): Duration of each pulse signal (in microseconds). Default: 200 µs
- **Pulse reading period** (min): How often the system reads and updates the pulse count (in minutes). Default: 60 min
- **Pulse save period** (min): How often the system saves pulse data to memory (in minutes).
- **"Set" button** – Writes the configuration to the device.
- **"Get" button** – Reads the configuration from the device.

Pulse inputs registers can be read using Modbus read file function 3.

Start register	End register	Register name	Format	Type
0	3	Number of pulses counted in channel 1	UInt64	Read-only
4	7	Number of pulses counted in channel 2	UInt64	Read-only
16	19	Value in double format in channel 1	Double	Read-only
20	23	Value in double format in channel 2	Double	Read-only
32	33	Value in float format in channel 1	Float	Read-only
34	35	Value in float format in channel 2	Float	Read-only

These registers must be mapped under the user-defined archive parameters to enable transmission to external FTP or MQTT servers.

7. Manufacturer's warranty

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

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

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

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