control solutions
TERACOM




# TCW210-TH Temperature and humidity data logger

# USER MANUAL

www.teracomsystems.com

# 1. Introduction

The TCW210-TH is a temperature and humidity data logger with an embedded web server. It allows real-time monitoring of temperature, humidity, and dew point data, as well as chart visualization via a standard web browser—no additional software required. The device supports standard M2M protocols like SNMP, Modbus TCP/IP, and HTTP/API for seamless integration into various applications. It is also compatible with popular IoT analytics platforms like ThingSpeak.

The logger supports up to eight temperature or humidity-temperature sensors, which can be connected through either the 1-Wire interface (commonly used in home automation) or the more durable Modbus RTU over RS-485 interface.

All monitored data can be stored in the device's internal FLASH memory, with records created at pre-set time intervals and/or triggered by alarm conditions. The memory capacity ensures storage for at least 36 days of data with minute-by-minute logging. Log files can be periodically uploaded to a dedicated server using HTTP/HTTPS POST, and stored data can be viewed on four graph pages.

Additionally, the device can periodically upload an XML/JSON file containing all monitored parameters to a server via HTTP/HTTPS POST. For each monitored parameter, the device can send email alerts and SNMP traps to up to five recipients. Alarm notifications can also be sent via HTTP/HTTPS POST using XML/JSON status files.

# 2. Features

- Data logger with a capacity for up to 70,000 records
- Supports 1-Wire and Modbus RTU sensors
- HTTP API command support for easy integration
- Periodic HTTP/HTTPS POST of XML/JSON status files for client-server applications
- SNMP v.2 support
- SNMP traps for alarm alerts to up to 5 recipients
- Modbus TCP/IP support for industrial automation
- SMTP with TLS encryption for secure email notifications
- Support for TLS 1.0, TLS 1.1, and TLS 1.2
- Email notifications for up to 5 recipients for alarm conditions
- ThingSpeak service integration for IoT analytics
- NTP support for time synchronization
- Back-up/Restore functionality for configuration settings
- Dynamic DNS support for easy remote access
- 10/100 Mb Ethernet connectivity with Auto-MDIX
- Configurable ports for HTTP, SNMP, and Modbus TCP/IP
- Password-protected web-based configuration and control
- Extended temperature range for harsh environments
- Wide power supply voltage range for flexible installation
- Support for remote firmware updates for easy device maintenance.

## 3. Applications

The TCW210-TH is ideal for environmental monitoring, building automation, and industrial automation. It excels in monitoring temperature and humidity, whether used as a standalone device through a web browser or as part of larger SCADA (Supervisory Control and Data Acquisition) systems in industrial control.

Some common application examples include:

- Pharmaceutical and food processing and storage facilities
- Clean rooms and laboratories
- HVAC systems
- Greenhouses and farms
- Electronics assembly environments

Its versatility makes it a reliable choice for industries requiring precise environmental monitoring and control

## 4. Specifications

- Physical characteristics

  Dimensions: 130 x 70 x 30 mm
  Weight: 140 g

- Environmental limits

  Operating temperature range: -20 to 55°C
  Storage temperature range: -25 to 60°C
  Operating relative humidity range: 5 to 85% (non-condensing)

- Power supply

  Operating voltage range (including -15/+20% according to IEC 62368-1): 10 to 28 VDC
  Current consumption: 170 mA @ 12 VDC

- 1-Wire and RS485 interface

  Output voltage (+VW): 5.0 ± 0.3 VDC
  Maximum output current (+VW): 0.2 A

- Internal FLASH memory

  Endurance: 100 000 cycles (Every settings change is a memory cycle.)

- Lithium battery

  Type: CR1220

  **Caution:** Replacing the battery with an incorrect type may result in an explosion.

## 5. LED indicators

The TCW210-TH controller includes the following LED indicators to display its status:

- **PWR** (red) – Steady ON in normal operation; blinks in sync with the STS LED in the event of a hardware error;
- **STS** (yellow) – Flashes to indicate that the controller's main program is running;
- **NET** (orange) – Indicates network status; remains ON when a network link is established, and blinks to show network activity.

## 6. Installation and setup

Qualified personnel must install the device. It shouldn't be installed outside directly.

The installation process involves mounting the device, connecting it to an IP network, attaching inputs and outputs, supplying power, and configuring it through a web browser.

### 6.1. Mounting

The TCW210-TH should be installed in a clean, dry location on a non-flammable surface. For installations where high ambient temperatures are expected, proper ventilation is recommended to ensure reliable operation.

To maintain sufficient ventilation and electrical isolation, allow 50 mm of space on all sides of the device, as illustrated in Appendix A. Ensure appropriate spacing between adjacent equipment to avoid overheating and ensure safe operation.

### 6.2. Connection

**Warning! Power off before wiring.**

To ensure proper installation, follow these steps:

- Turn off the power before beginning any wiring;
- Connect all sensors securely to the appropriate terminals;
- Turn on the power after all connections are made.

Ensure that all cables are securely attached. Improper wiring or configuration can lead to permanent damage to the TCW210-TH, the connected equipment, or both. Double-check all connections before powering on to avoid potential hazards.




| | | | |
|---|---|---|---|
| **Connector 1** | Ethernet - RJ45 | **Connector 4** | Pin1 – not connected (most left) |
| **Connector 2** | Power - 2.1x5.5mm connector, central positive | | Pin2 – not connected |
| **Connector 3** | Pin1 – GND (most left) | | Pin3 – not connected |
| | Pin2 – GND | | Pin4 – 485- |
| | Pin3 – 1-Wire Data | | Pin5 – 485+ |
| | Pin4 – 1-Wire GND | | Pin6 – not connected |
| | Pin5 – +VDD | | Pin7 – +VDD |
| | Pin6 – +VDD (most right) | | Pin8 – GND |

### 6.2.1. Power supply

TCW210-TH must be powered by the adapter SYS1308(N)-2412-W2E or equivalent, suitable for overvoltage category II and certified for safety compliance. The power supply device should be able to withstand short circuits and secondary circuit overloads. Ensure the equipment is easily accessible for disconnecting from the power supply during use.

### 6.2.2. 1-Wire interface

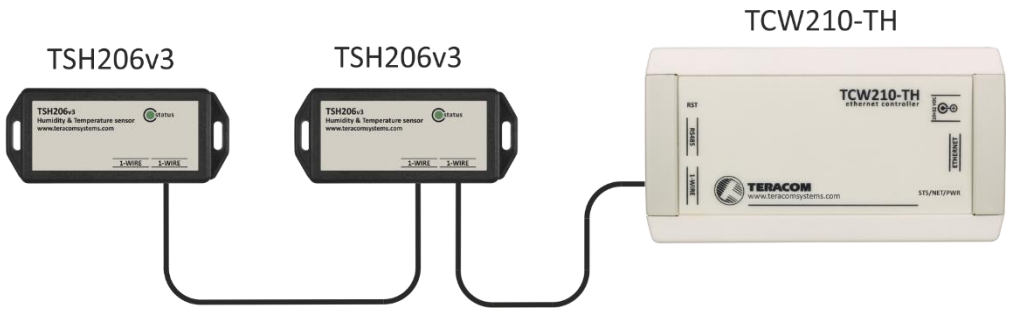
The TCW210-TH supports connecting up to 8 sensors across both interfaces (1-Wire and RS-485 combined).

The device is compatible with 1-Wire temperature and humidity-temperature sensors, allowing for easy integration and monitoring of environmental conditions.

1-Wire, a registered trademark of Analog devices Inc., is designed for connecting multiple sensors over short wiring distances. It is not suitable for long distances or environments with significant electromagnetic interference (EMC). For best practices, refer to Guidelines for Reliable Long-Line 1-Wire Networks.

The sensors typically have three wires: positive voltage (+VDD), ground (GND), and bidirectional data (Data). The specific wire colors for each sensor are provided in the sensor's user manual.

For reliable operation with multiple sensors, it is strongly recommended to use a "daisy-chained" (linear) topology:




Connections to the 1-Wire interface can be made using a standard RJ-12 connector, ensuring simple and secure integration of sensors.

Several factors influence the maximum allowable length of the wires, including the type of cable used, the number of sensors, ambient electromagnetic noise, and the network topology of the sensor connections.

It is strongly recommended to use only UTP (Unshielded Twisted Pair) or FTP (Foiled Twisted Pair) cables, with a total cable length limited to 30 meters. While longer distances may still function, error-free operation cannot be guaranteed beyond the recommended length.

For optimal performance, Teracom guarantees proper operation only when using Teracom 1-Wire sensors.

### 6.2.3. RS-485 interface

The TCW210-TH supports up to 8 sensors across both 1-Wire and RS-485 interfaces combined.

This device is compatible with Modbus RTU (over RS-485) temperature and humidity-temperature sensors. RS-485 is an industry-standard for serial communication, suitable for long distances and environments with significant electrical noise, such as industrial settings.

Modbus RTU protocol specifies that address of the device should be between 1 and 247. The user should take care of appropriate address settings.

For multi-sensors systems "daisy-chained" (linear topology) should be used:




Connections are made using UTP/FTP cables with RJ-45 connectors, following the Ethernet wiring standard ANSI/TIA/EIA T568B. The wiring pinout is as follows:

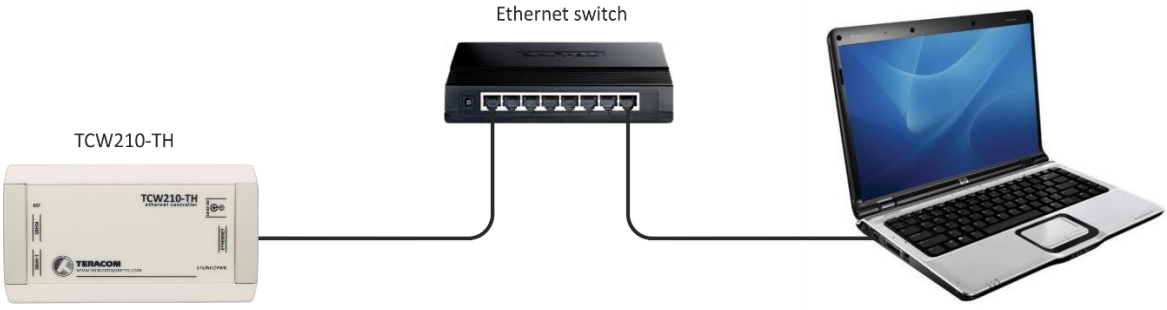
| Pin | Color | Pin | Color |
|---|---|---|---|
| 1 | Orange/White | 5 | Blue/White Tracer |
| 2 | Orange | 6 | Green |
| 3 | Green/White | 7 | Brown/White Tracer |
| 4 | Blue | 8 | Brown |

Proper termination of the RS-485 bus is critical. The last sensor in the chain must have a 120-ohm terminator installed on its unused RJ-45 socket. This terminator is included with the module. Although RS-485 can work over longer distances, we recommend keeping the total cable length within 30 meters for optimal performance.

**Attention!** Ensure the correct termination of the bus to prevent communication errors.

### 6.2.4. Network connection

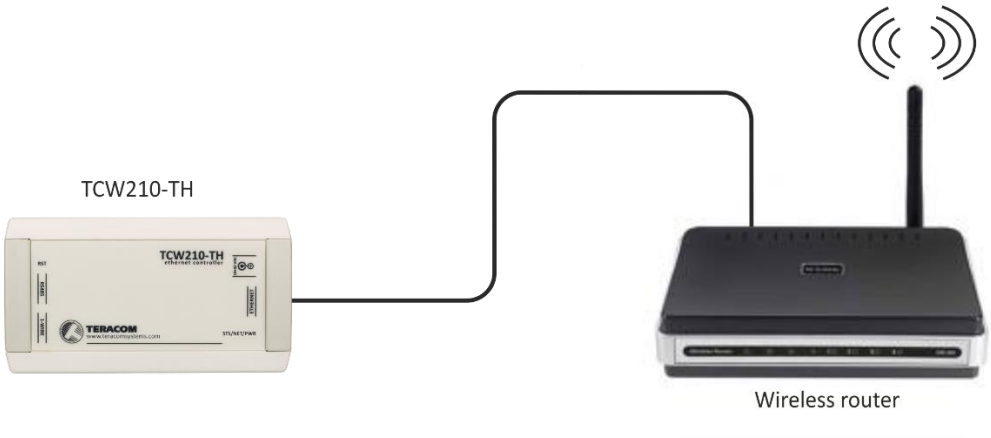
The Ethernet port of TCW210-TH should be connected to 10/100 Base-T Ethernet hub, switch or router.




For configuration purposes, the TCW210-TH can be directly connected to the Ethernet port of a computer. The device supports Auto-MDIX, so either a standard "straight-through" cable or a "crossover" cable can be used.



TCW210-TH can also be integrated into a wireless network by connecting it through a wireless router.

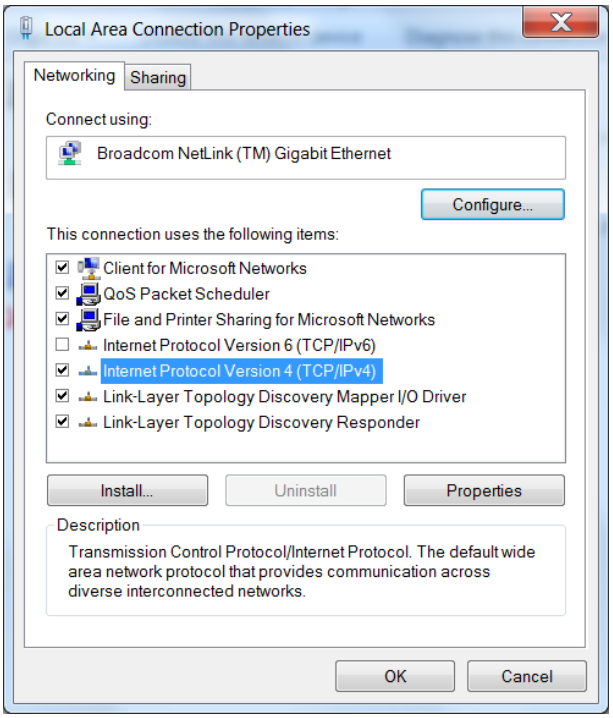



## 6.3. Communication setup

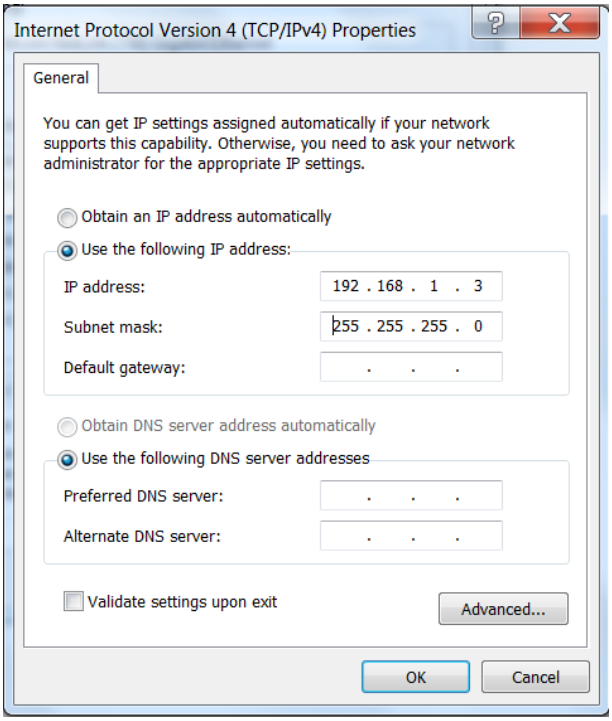
The TCW210-TH is shipped with the following default network settings:

- IP Address: 192.168.1.2
- Subnet Mask: 255.255.255.0
- Default Gateway: 192.168.1.1

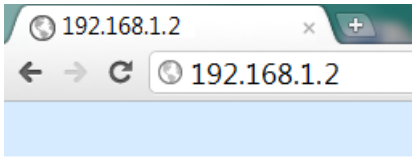
To establish communication with the TCW210-TH, you can assign a temporary IP address to your computer. For computers running Windows OS, this can be done through the “Local Area Connection Properties”:



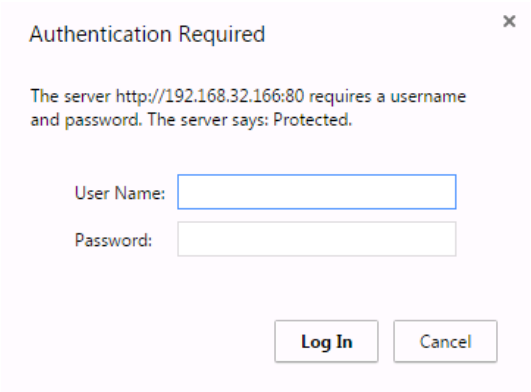
The temporary IP address should be on the same network; for example, you can use 192.168.1.3:




To access the web interface, enter http://192.168.1.2 into your browser's address bar:




If the network settings are configured correctly, a login pop-up window will appear:




The default authorization credentials are as follows: username: admin and password: admin.

It is highly recommended to change both the username and password to prevent unauthorized access.

All TCWxxx controllers connected to the LAN can be easily located using the free tool “TCW Discoverer.” This tool is available for both Windows and Mac operating systems and can be downloaded from www.teracomsystems.com.

## 7. Web interface

The web interface enables configuration, monitoring, and control of the TCW210-TH. All pages are UTF-8 encoded. Please note that the device supports HTTP only; HTTPS is not available.

## 7.1. Monitoring

The Monitoring menu provides access to all parameters in two formats: data view (Monitoring -> Data) and graphical view (Monitoring -> Graph).

### 7.1.1. Data

**1-Wire sensors**

| Pos | Description | Temperature | Humidity | Dew point | ID |
|---|---|---|---|---|---|
| 1 | S1:TST1xx | 22.438°C | ------ | ------ | [2867895F07000058] |
| 2 | S2 | ------ | ------ | ------ | [0000000000000000] |
| 3 | S3 | ------ | ------ | ------ | [0000000000000000] |
| 4 | S4 | ------ | ------ | ------ | [0000000000000000] |

**Modbus sensors**

| Pos | Description | Temperature | Humidity | Dew point | Address |
|---|---|---|---|---|---|
| 5 | S5 | 22.439°C | 31.444%RH | 4.694°C | 1 |
| 6 | S6 | ------ | ------ | ------ | 0 |
| 7 | S7 | ------ | ------ | ------ | 0 |
| 8 | S8 | ------ | ------ | ------ | 0 |

The current state of the TCW210-TH can be viewed on the monitoring page. This page is divided into two sections: one for 1-Wire sensors and one for Modbus RTU sensors.

The TCW210-TH supports a maximum of eight sensors, which can be connected to both interfaces in a ratio that can be adjusted in the "Sensors ratio setup" section found on the Setup->Sensors page. By default, four Modbus RTU sensors are configured.

In the 1-Wire sensors section, all detected sensors are displayed. These sensors must first be set up in the "1-Wire sensors setup" section on the Setup->Sensors page. For Teracom 1-Wire temperature sensors, the readings are shown under the "Temperature" column. For dual sensors, such as the TSH2xx series, both the "Temperature" and "Humidity" columns are utilized, and the Dew point is automatically calculated.

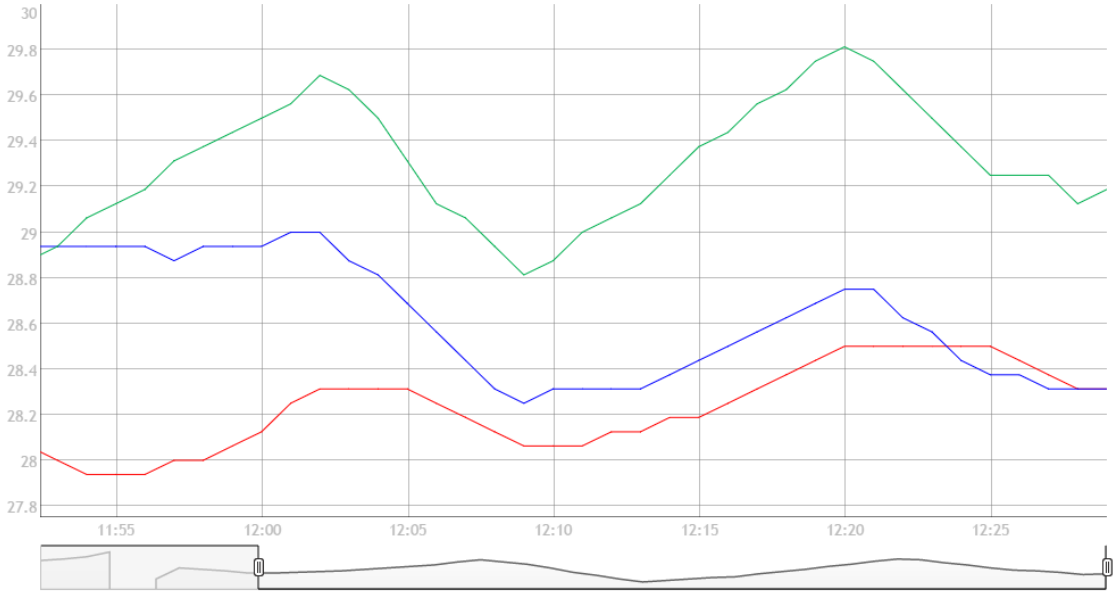
The Modbus RTU sensors section displays all connected Modbus sensors, which should be configured on the Setup->Sensors page.

For every sensor, the monitoring page provides a description, value, and ID information. Descriptions can be up to 15 characters long, and default descriptions can be modified on the Setup->Conditions page. Dual sensors will show both parameters (temperature and humidity), and the Dew point parameter is calculated automatically for them.

Additionally, the page can be set to automatically refresh at intervals between 0 and 253 seconds, where a setting of 0 disables automatic refreshing. This refresh rate can be configured in the Setup->System->Monitoring page automatic refresh section, with a default value of 1 second

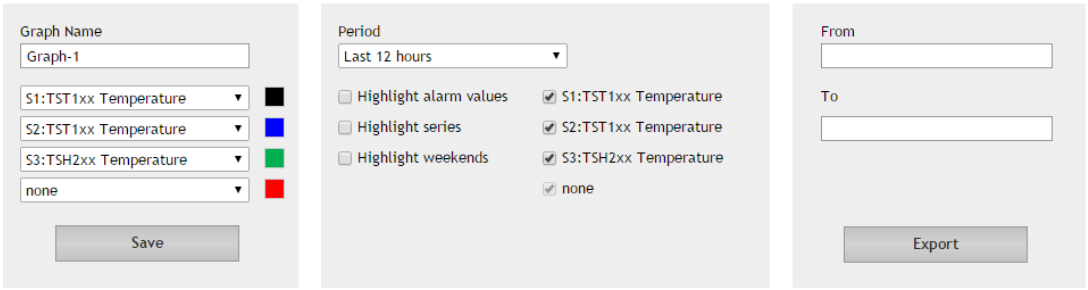
### 7.1.2. Graphs

Each graph page can display up to four parameters, with the option for two different dimensions.




Users can assign different colors to each parameter for better visualization. Additionally, several checkboxes are available for modifying the display settings.

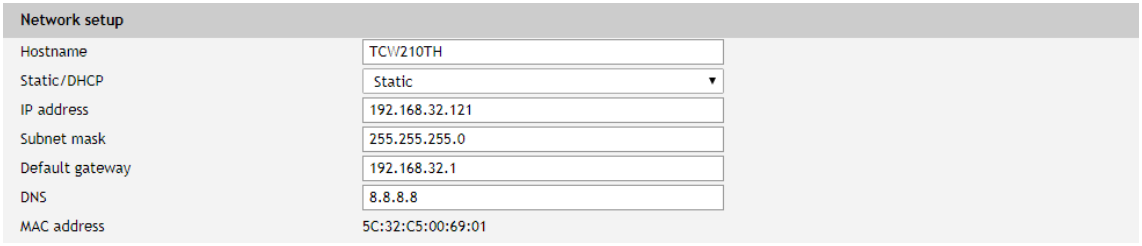
The page also allows for the export of monitored parameters.




## 7.2. Setup

### 7.2.1. Network

Network parameters are configured in this section.




The controller supports both static and dynamic IP addresses. It is advisable to change the default IP address of the controller immediately after the first power-on to prevent potential IP address conflicts when multiple devices are connected to the same network.

Each time a new device is connected to the network, it may be necessary to clear the ARP cache. This can be done by entering arp -d in the command prompt on the computer.

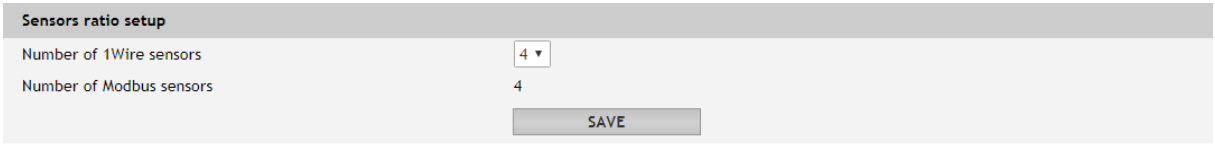
The “Hostname” can be up to 15 characters long and will appear in the search results of the TCW Discoverer tool.

Public DNS servers (such as 8.8.8.8 or 8.8.4.4) can be used instead of the default gateway.

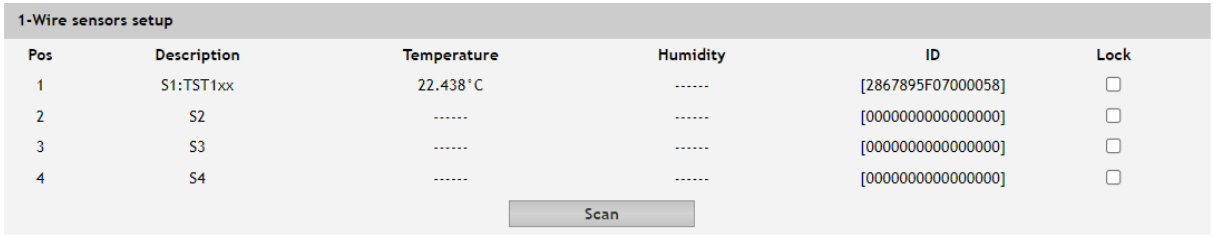
## 7.2.2. Sensors

### 7.2.2.1. Sensors ratio setup

This section allows you to configure the ratio between 1-Wire and Modbus RTU sensors. The default ratio is set to 4:4



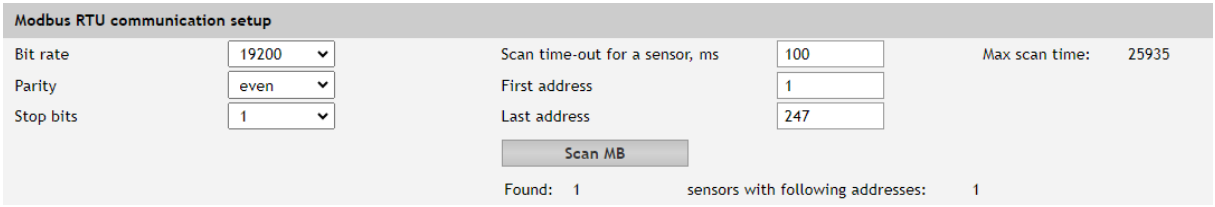

### 7.2.2.2. 1-Wire sensors setup




1-Wire sensors are detected either upon power-on or by clicking the “Scan for New Sensors” button. All detected sensors are displayed in ascending order based on their unique ID numbers.

To lock a 1-Wire sensor in a specific position, each sensor must be added individually. After adding each sensor, perform a new scan and lock the newly detected sensor in its designated position. If all sensors are locked, removing a sensor from the middle will not affect the positions of the remaining sensors after a reset. This feature is particularly useful when the TCW210-TH is integrated into a monitoring and control system managed via SNMP or HTTP API commands.

### 7.2.2.3. Modbus RTU communication setup

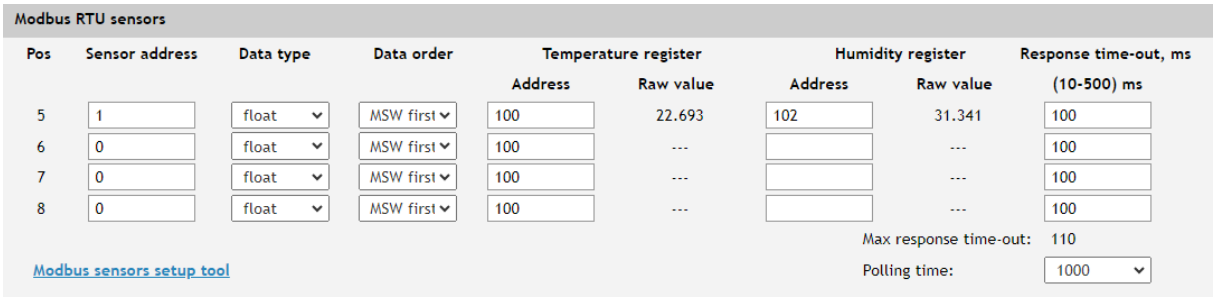



The TCW210-TH supports Modbus RTU communication over the RS-485 interface. All sensors connected to this interface must operate with the same communication settings. By default, the device is configured with the standard Modbus RTU settings: 19200 baud rate, Even parity, and 1 stop bit.

In the right part of this section, there is a tool for scanning the Modbus RTU interface. To optimize scan time, the appropriate address segment should be configured first. After making these adjustments, click the “Save” button at the bottom of the page to apply the changes.

### 7.2.2.4. Modbus RTU sensors

The TCW210-TH is compatible with both Teracom and third-party Modbus RTU sensors.



Before adding these sensors, the user must ensure that each sensor has a unique address, as using two sensors with the same address is not permitted. It is advisable to scan for new sensors before making any changes.

For each sensor, the appropriate register address, data type, and data order must be configured. Be sure to save all changes. If the settings are correct, the corresponding data will be displayed in the “Raw Value” columns.

The TCW210-TH supports Modbus RTU sensors with a response time-out ranging from 10 to 500ms. By default, the response time-out for a new sensor is set to 100ms, but it is recommended to use the minimum response time specified by the sensor's manufacturer.

The total selected response time-out for all sensors contributes to the maximum response time-out for the system. The polling time refers to the interval between two consecutive readings of the same sensor, affecting the system's reaction time. By default, this polling time is set to 1 second.

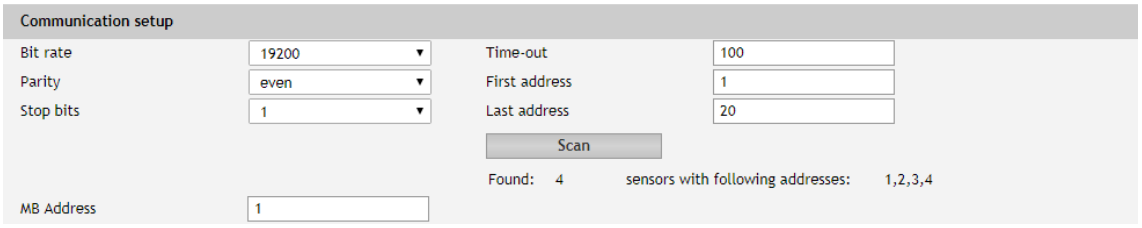
**Important**: The maximum response time-out must not be less than the polling time.

### 7.2.2.5. Sensor setup tool

A link to the Sensor Setup Tool is provided at the bottom of the Modbus RTU Sensors section. This tool can be utilized to modify communication settings for the sensors or simply to read information from a specific register.

#### 7.2.2.5.1. Communication setup

This section is similar to the general Modbus RTU Communication Setup. The only additional field is the Sensor Address. Changes made in this section are not saved and do not alter the general settings of the TCW210-TH.



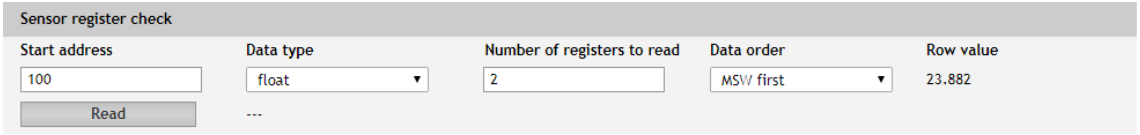

#### 7.2.2.5.2. Sensor communication register setup

This section of the tool is used to check and modify the status of the communication registers for the sensor.



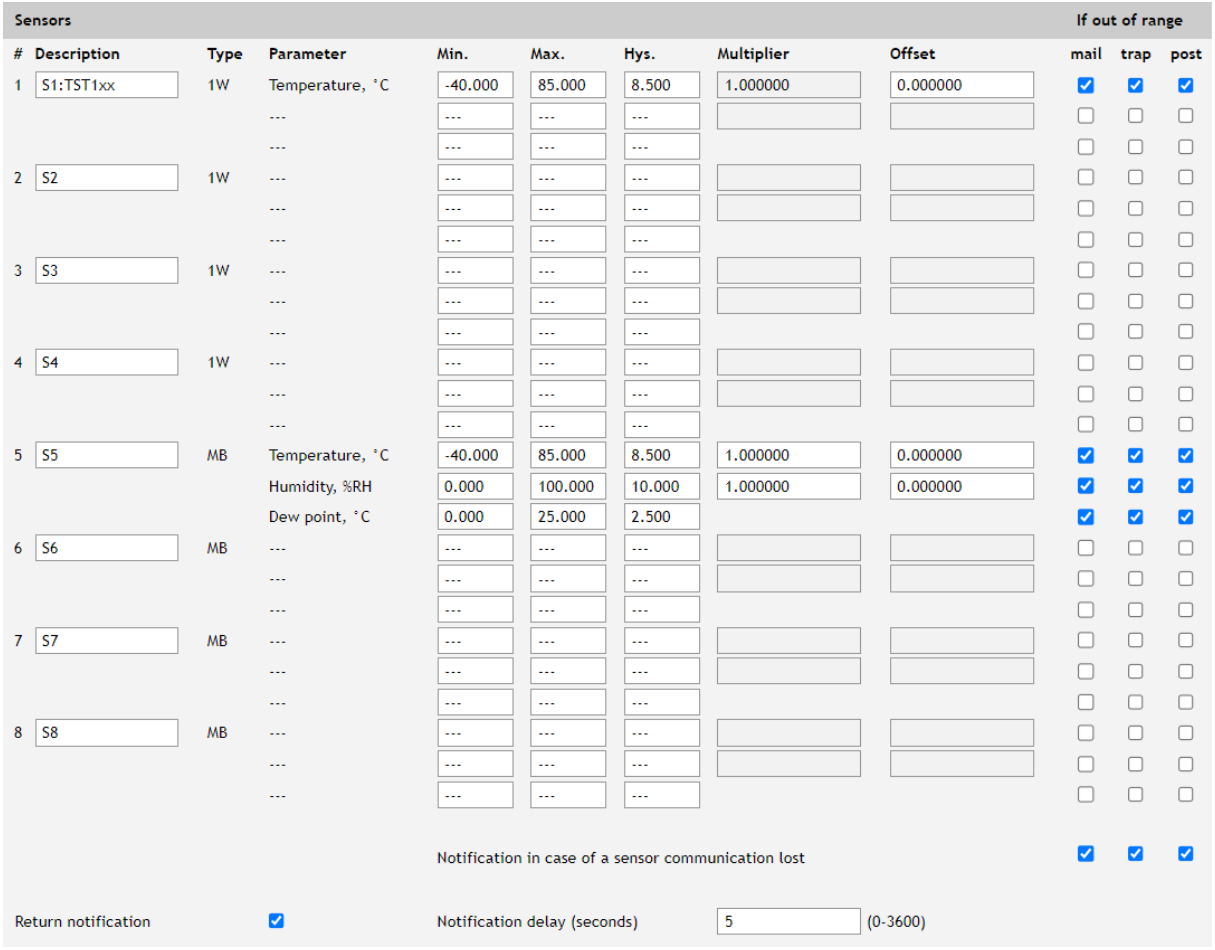

#### 7.2.2.5.3. Sensor register check

This section of the tool is utilized for a comprehensive check of the sensor registers.



### 7.2.3. Conditions

This section is dedicated to configuring the trigger and alert conditions for 1-Wire and Modbus RTU sensors.



| Sensors | | | | | | | | If out of range | | |
|---|---|---|---|---|---|---|---|---|---|---|
| # | Description | Type | Parameter | Min. | Max. | Hys. | Multiplier | Offset | mail | trap | post |
| 1 | S1:TST1xx | 1W | Temperature, °C | -40.000 | 85.000 | 8.500 | 1.000000 | 0.000000 | ☑ | ☑ | ☑ |
| | | | --- | --- | --- | --- | | | ☐ | ☐ | ☐ |
| | | | --- | --- | --- | --- | | | ☐ | ☐ | ☐ |
| 2 | S2 | 1W | --- | --- | --- | --- | | | ☐ | ☐ | ☐ |
| | | | --- | --- | --- | --- | | | ☐ | ☐ | ☐ |
| | | | --- | --- | --- | --- | | | ☐ | ☐ | ☐ |
| 3 | S3 | 1W | --- | --- | --- | --- | | | ☐ | ☐ | ☐ |
| | | | --- | --- | --- | --- | | | ☐ | ☐ | ☐ |
| | | | --- | --- | --- | --- | | | ☐ | ☐ | ☐ |
| 4 | S4 | 1W | --- | --- | --- | --- | | | ☐ | ☐ | ☐ |
| | | | --- | --- | --- | --- | | | ☐ | ☐ | ☐ |
| | | | --- | --- | --- | --- | | | ☐ | ☐ | ☐ |
| 5 | S5 | MB | Temperature, °C | -40.000 | 85.000 | 8.500 | 1.000000 | 0.000000 | ☑ | ☑ | ☑ |
| | | | Humidity, %RH | 0.000 | 100.000 | 10.000 | 1.000000 | 0.000000 | ☑ | ☑ | ☑ |
| | | | Dew point, °C | 0.000 | 25.000 | 2.500 | | | ☑ | ☑ | ☑ |
| 6 | S6 | MB | --- | --- | --- | --- | | | ☐ | ☐ | ☐ |
| | | | --- | --- | --- | --- | | | ☐ | ☐ | ☐ |
| | | | --- | --- | --- | --- | | | ☐ | ☐ | ☐ |
| 7 | S7 | MB | --- | --- | --- | --- | | | ☐ | ☐ | ☐ |
| | | | --- | --- | --- | --- | | | ☐ | ☐ | ☐ |
| | | | --- | --- | --- | --- | | | ☐ | ☐ | ☐ |
| 8 | S8 | MB | --- | --- | --- | --- | | | ☐ | ☐ | ☐ |
| | | | --- | --- | --- | --- | | | ☐ | ☐ | ☐ |
| | | | --- | --- | --- | --- | | | ☐ | ☐ | ☐ |
| | | | | Notification in case of a sensor communication lost | | | | | ☑ | ☑ | ☑ |
| Return notification | | ☑ | | Notification delay (seconds) | | | 5 | (0-3600) | | | |

For each sensor, a description of up to 15 characters can be assigned.

The "Offset" field is available for all sensors, allowing for simple adjustments to the displayed value. For all Modbus RTU sensors, a multiplier is enabled, although for Teracom sensors, this should be set to 1.

Each parameter includes fields for trigger conditions, specifically “Min,” “Max,” and “Hys.” The “Min” and “Max” values define the boundaries of the operational range for the monitored parameter. A “Max” trigger condition is triggered when the value exceeds the specified upper limit, while a “Min” trigger condition is activated when the value falls below the defined lower limit. In both cases, the monitored parameter is considered out of range.

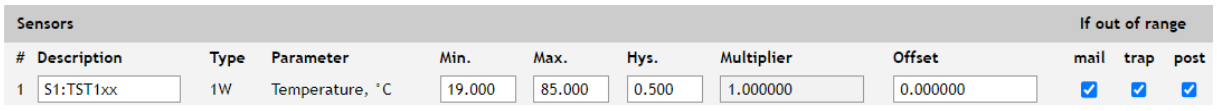
Returning to the acceptable range for the monitored parameter occurs when the value exceeds (Min + Hys) or drops below (Max - Hys). Hysteresis (“Hys”) is incorporated to prevent excessive triggering when the value fluctuates around the trigger points.

Example:

In this scenario, a TCW210-TH, a TST100 sensor, and an appropriate heater are used to manage room temperature. The desired minimum temperature is set at 19°C, while the initial temperature is 17°C. The TST100 sensor is assigned as the first 1-Wire sensor.

Local activation for Relay1 is configured based on Sensor1 with the following parameters:

Min = 19, Max = 85, and Hys = 0.5.

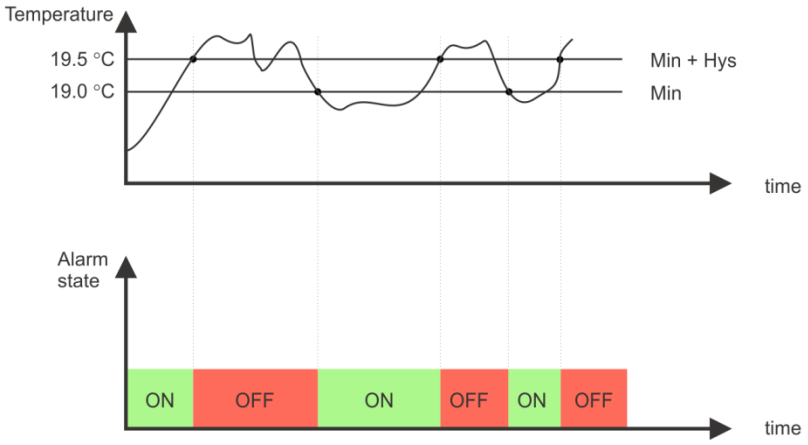


| Sensors | | | | | | | | If out of range | | |
|---|---|---|---|---|---|---|---|---|---|---|
| # | Description | Type | Parameter | Min. | Max. | Hys. | Multiplier | Offset | mail | trap | post |
| 1 | S1:TST1xx | 1W | Temperature, °C | 19.000 | 85.000 | 0.500 | 1.000000 | 0.000000 | ☑ | ☑ | ☑ |

When the controller is powered on, Relay1 is immediately activated because the monitored temperature is below the acceptable range. This action turns on the heater, causing the temperature to rise.

As the temperature increases and reaches 19.5°C (19.0 + 0.5), it falls within the defined range (trigger condition), and Relay1 is deactivated, turning the heater off.

However, as the temperature begins to drop and reaches 19°C, it goes out of range (trigger and alert conditions), which reactivates the relay, turning the heater back on. Additionally, an email notification is sent to alert about the temperature change.



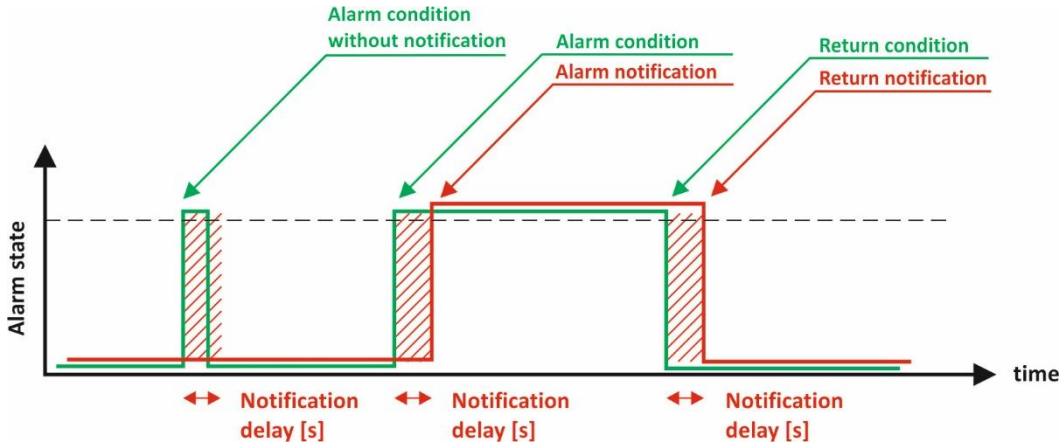

The “Max” value should be set sufficiently high above the desired temperature to prevent unnecessary trigger and alert conditions.

Each sensor or analog input has three independent alert methods available when an alarm condition occurs: email notifications, SNMP traps, and HTTP/HTTPS POSTs (XML file). Each method can be activated via a checkbox.

The “Max” value should be set sufficiently high above the desired temperature to prevent unnecessary trigger and alert conditions. Each sensor or analog input has three independent alert methods available when an alarm condition occurs: email notifications, SNMP traps, and HTTP/HTTPS POSTs (XML file). Each method can be activated via a checkbox.

There is a global checkbox labeled “Return notification” for all sensors and analog inputs. When selected, notifications will also be sent when a parameter returns to within the defined range.

Additionally, a “Notification delay” parameter is available globally for all sensors and analog inputs. This feature acts as a filter for short alarm conditions, helping to reduce false alerts.

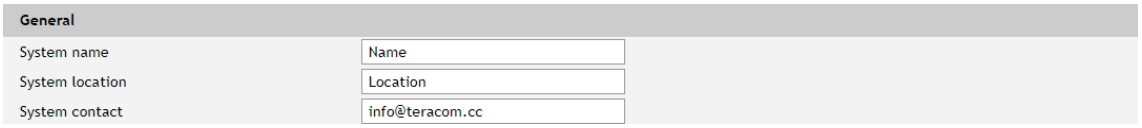


### 7.2.4. System

This page allows you to configure various general settings for the system.

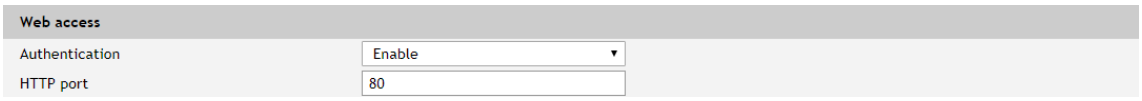
#### 7.2.4.1. General

This section allows you to configure general parameters for device identification.




#### 7.2.4.2. WEB access

This section allows you to deactivate WEB access authentication, which is enabled by default with the credentials of admin/admin. You can also change the HTTP port for WEB access, which can be useful for certain routers that do not support separate outside/inside ports for port forwarding. The default HTTP port is set to 80.




#### 7.2.4.3. HTTP API

This section allows you to activate or deactivate HTTP API access authentication, which is enabled by default.




The authentication details are the same as for WEB access. The controller supports two types of authentications; refer to the explanation for HTTP API below for more information.

#### 7.2.4.4. Monitoring page automatic refresh

The refresh interval for the monitoring page can be configured to range from 0 to 253 seconds. A setting of zero indicates that automatic refresh is disabled.




#### 7.2.4.5. Display

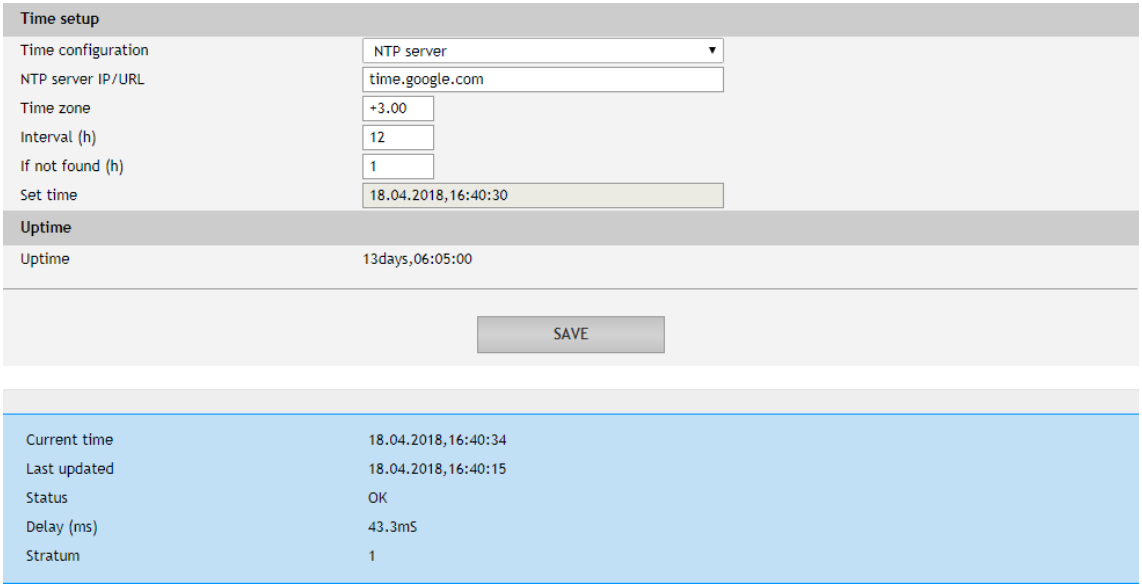
In this section, you can select the working units for the temperature (Celsius or Fahrenheit).

Each section of the Monitoring Page has a checkbox; when checked, the corresponding section will be displayed.




### 7.2.5. Time

The internal real-time clock (RTC) of the controller can be configured either manually or automatically.




For automatic clock synchronization, the controller supports NTP (Network Time Protocol), and all necessary parameters for this synchronization are available in this section.

By default, NTP synchronization is disabled, with the server set to time.google.com, a time zone of +00:00, and an interval of 12 hours.

## 7.3. Services

### 7.3.1. Modbus TCP/IP

TCW210-TH supports Modbus TCP/IP via the Ethernet interface.



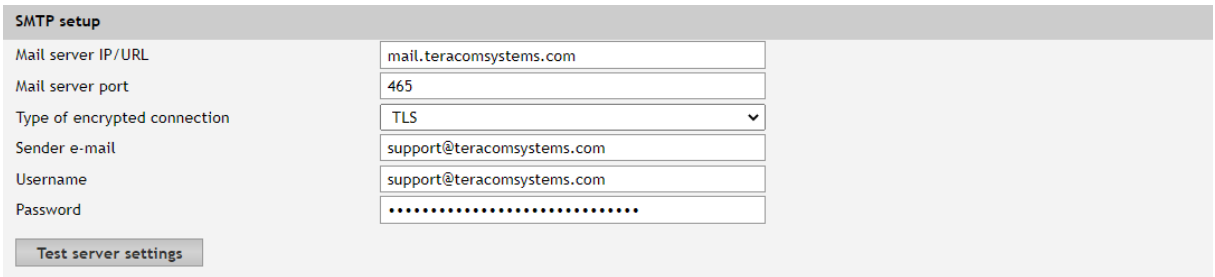

By default, Modbus functionality is disabled. The standard port for this protocol is 502.

A table containing the register addresses can be found in section, titled " Modbus TCP/IP".

### 7.3.2. SMTP

This page allows you to input the valid SMTP settings for email alerts and specify the recipients' addresses.

#### 7.3.2.1. SMTP setup




The mail server address can be configured using either a hostname (e.g., mail.teracomsystems.com) or an IP address. By default, the SMTP port is set to 25 for non-encrypted connections. If the default port does not work, please consult your Internet Service Provider (ISP).

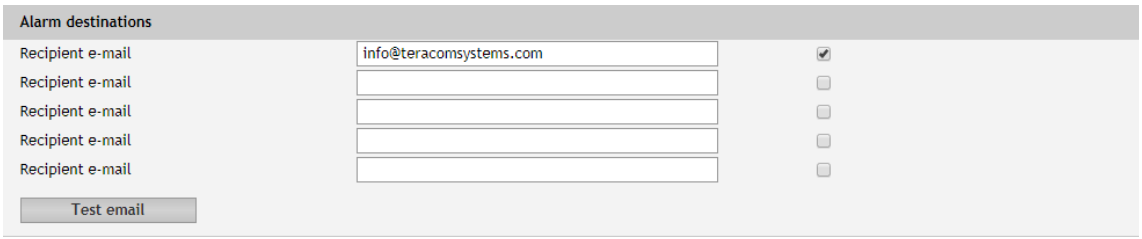
The sender's email, username, and password are standard authentication details, each allowing for up to 128 characters in length.

A button is available to test the server settings, providing feedback on the results. In this test, the sender and recipient email addresses are the same.

For secure communication with mail servers, the Transport Layer Security (TLS) protocol is employed. The TCW210-TH supports TLS versions 1.0, 1.1, and 1.2, using RSA for key exchange, agreement, and authentication. This ensures compatibility with nearly all public servers.

Please note that STARTTLS is not supported.

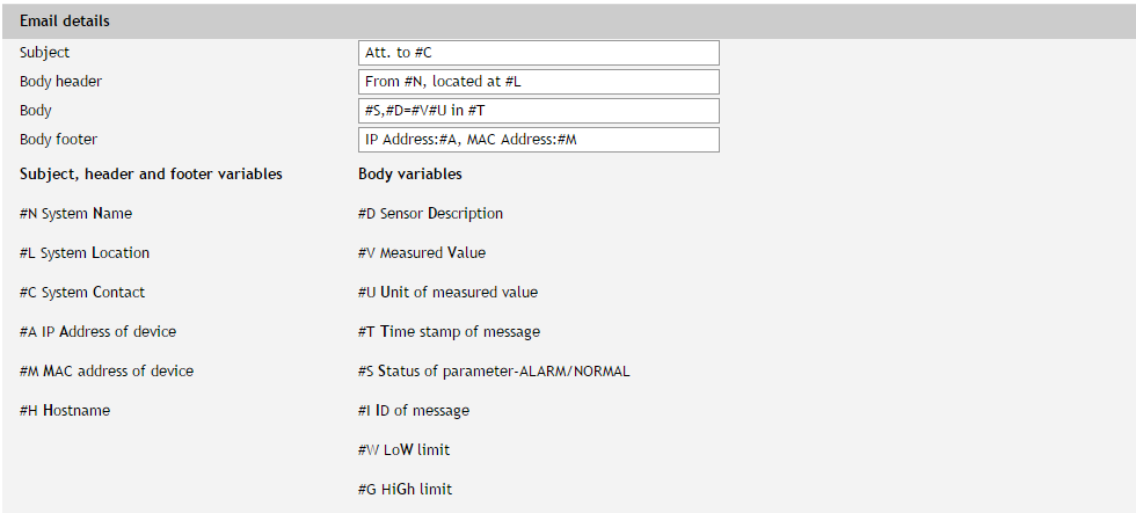
### 7.3.2.2. Alarm destination




You can configure up to five email recipients, each of whom can be activated independently using a checkbox.

### 7.3.2.3. E-mail details

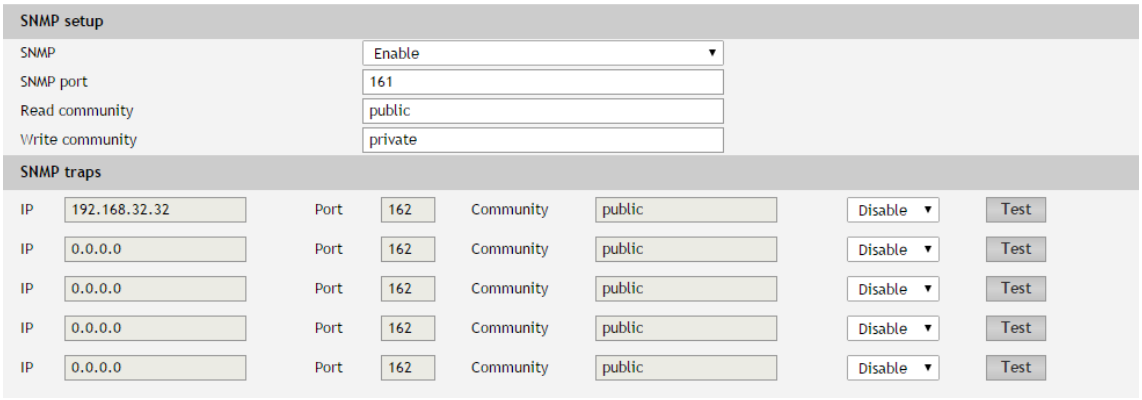
The subject, body header, body, and footer of the email can be customized using a predefined set of keys. All available keys are described on the page.



### 7.3.3. SNMP

The TCW210-TH supports SNMP v.2, allowing the device to integrate into monitoring and control systems using the SNMP protocol.

This section allows you to configure all necessary parameters for the proper operation of SNMP.




By default, SNMP is disabled, with the port set to 161. The read community is set to "public," and the write community is set to "private." In the event of an alarm condition, SNMP traps can be sent to up to five independent recipients, each capable of having different ports and communities. A dedicated button is available for testing the SNMP trap functionality.

SNMP traps can be triggered under the following conditions:

- An event occurs (status change) on Digital Inputs.
- A measured parameter on Analog Inputs falls outside the specified range.
- A measured parameter on the 1-Wire bus exceeds the defined limits.
- A restart condition is initiated.

An SNMP trap will be sent following a reset.

The current MIB file can be downloaded from here.

### 7.3.4. Logger

The TCW210-TH supports data logging for all monitored parameters and relay statuses, allowing historical records to be stored locally and periodically uploaded to a remote server.

The logger operates in three modes: Time, Alarm, and Time & Alarm, providing flexible recording based on time intervals, alarm events, or both.

- In Time mode, records are created at regular intervals defined by the Log Interval setting.
- In Alarm mode, records are generated only when an alarm condition occurs.
- In Time & Alarm mode, both periodic intervals and alarm events trigger new log records.

The log interval is configurable from 10 to 3600 seconds, allowing higher-resolution data collection when shorter intervals are selected.

For synchronized logging, the device supports Logger Record Sync, which aligns log records to a specific time reference. When synchronization is enabled, the logger creates records at exact intervals based on the configured synchronization point rather than simply counting from the moment the device starts. This is useful for applications requiring precise time-based measurements or synchronization with external systems.

The Sync to the minute parameter defines the exact second within each minute when synchronized logging begins. This ensures predictable and consistent timestamps across all recorded data.

Example:

If the current time is 09:12:22, Logger Record Sync is enabled, Sync to the minute is set to 00, and the Log Interval is set to 15 minutes, the logger will create records at 09:15:00, 09:30:00, 09:45:00, 10:00:00, 10:15:00, and so on.

When the device powers up, the first record is created immediately, and subsequent records follow the synchronized schedule.

The TCW220 supports periodic upload of stored log records to a remote HTTP or HTTPS server in CSV format, enabling seamless integration with cloud platforms, monitoring software, and custom backend systems.

For secure communication, HTTPS connections support TLS 1.0, TLS 1.1, and TLS 1.2, providing encrypted data transfer and server authentication.

The upload interval is configurable from 1 to 24 hours, with synchronization to a user-defined daily time reference.

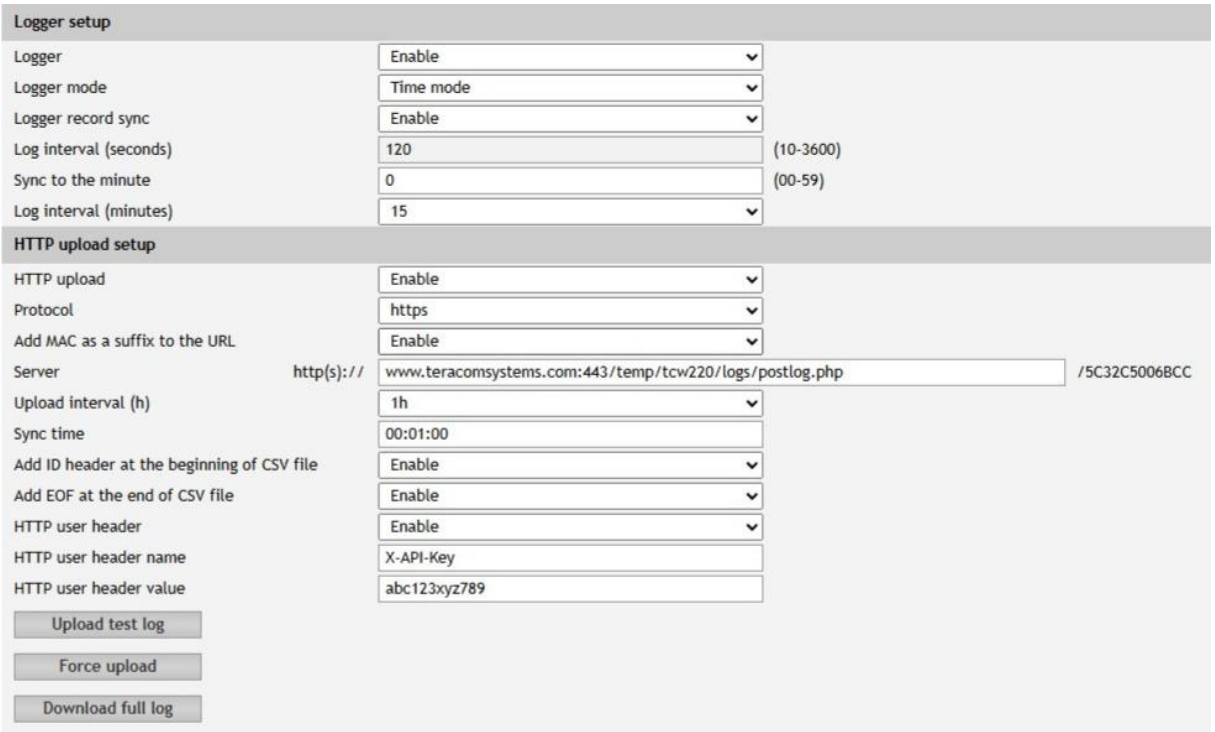
The Sync Time parameter defines the exact time of day used as a reference point for periodic uploads. This ensures uploads occur at predictable intervals aligned to the configured schedule.

Example:

If the upload interval is set to 3 hours and the Sync Time is 09:00, uploads will occur at 09:00, 12:00, 15:00, 18:00, 21:00, 00:00, 03:00, and 06:00.

If the current time is 19:31 and periodic upload is enabled, the first upload will occur at 21:00.

To improve compatibility with server-side processing and file organization, the TCW220 provides additional CSV upload options, including automatic MAC address suffixing in the upload URL, optional CSV header row insertion, configurable end-of-file markers, and custom HTTP headers for authenticated or API-based endpoints.




Configuration parameters:

- Logger – Enables or disables the logger functionality.
- Logger mode – Defines when data is recorded:

  Time – Records data at regular intervals.

  Alarm – Records data only when an alarm event occurs.

Time & Alarm – Records data periodically and on alarm events.

- Logger record sync – Enables synchronization of log records to the configured time reference.
- Log interval (seconds) – Defines the interval between log records. Adjustable from 10 to 3600 seconds.
- Sync to the minute – Defines the exact second within each minute used for synchronized logging.
- HTTP upload – Enables or disables periodic HTTP/HTTPS upload of logger records.
- Protocol – Selects the communication protocol for uploads (HTTP/HTTPS).
- Add MAC as a suffix to the URL – Appends the device MAC address to the configured server URL, allowing unique identification of uploaded files from multiple devices.
- Server – Specifies the destination server address (IP address, domain name, and optional port) where CSV log files will be uploaded.
- Upload interval (h) – Defines how often stored records are uploaded to the server.
- Sync time – Defines the daily reference time for upload synchronization.
- Add ID header at the beginning of CSV file – Adds a header row at the beginning of the CSV file containing parameter identifiers for easier automated processing.
- Add EOF at the end of CSV file – Appends an end-of-file marker to improve compatibility with server-side parsers.
- HTTP User Header – Enables or disables sending a custom HTTP header with the upload request.
- HTTP User Header Name – Defines the custom HTTP header field name.
- HTTP User Header Value – Defines the value associated with the custom HTTP header field.
- Upload test log – Sends a test CSV file to verify server connectivity and upload configuration.
- Force upload – Immediately uploads all unsent logger records to the configured server.
- Download full log – Downloads the complete logger file from the device for local storage or analysis.

### 7.3.5. HTTP POST

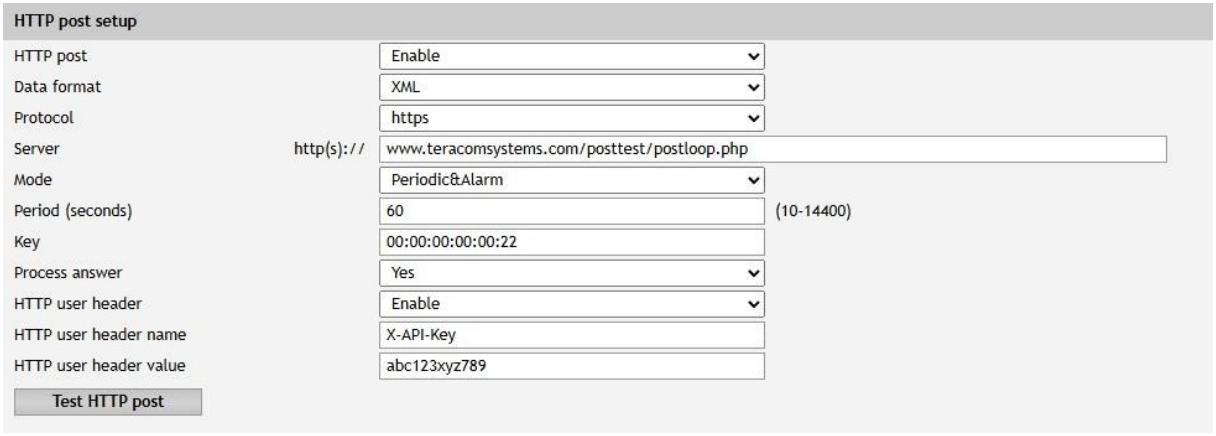
TCW210-TH supports automatic data transmission to a remote server through HTTP or HTTPS POST requests, enabling seamless integration with cloud platforms, SCADA systems, monitoring software, and custom backend applications.

The device can periodically upload operational data, alarm events, or both, in structured XML or JSON format, making it easy to process and integrate with external systems.

For secure communication, HTTPS connections support TLS 1.0, TLS 1.1, and TLS 1.2, providing encrypted data transfer and server authentication.

The transmission interval is configurable from 10 to 14,400 seconds (4 hours), allowing flexible reporting rates depending on application requirements.

The TCW241 supports adding custom HTTP headers to outgoing POST requests, enabling compatibility with modern APIs, cloud services, and authenticated endpoints.




Configuration parameters:

- HTTP POST - Enables or disables the HTTP POST functionality
- Data format - Defines the payload structure sent to the server – XML/JSON
- Server - Specifies the destination server address (IP address, domain name, and optional port) where the data will be posted.
- Mode - Defines when data is transmitted:

Periodic & Alarm – Sends data at regular intervals and immediately when an alarm event occurs.

Periodic only – Sends data only at the configured interval.

Alarm only – Sends data only when an alarm condition is triggered.

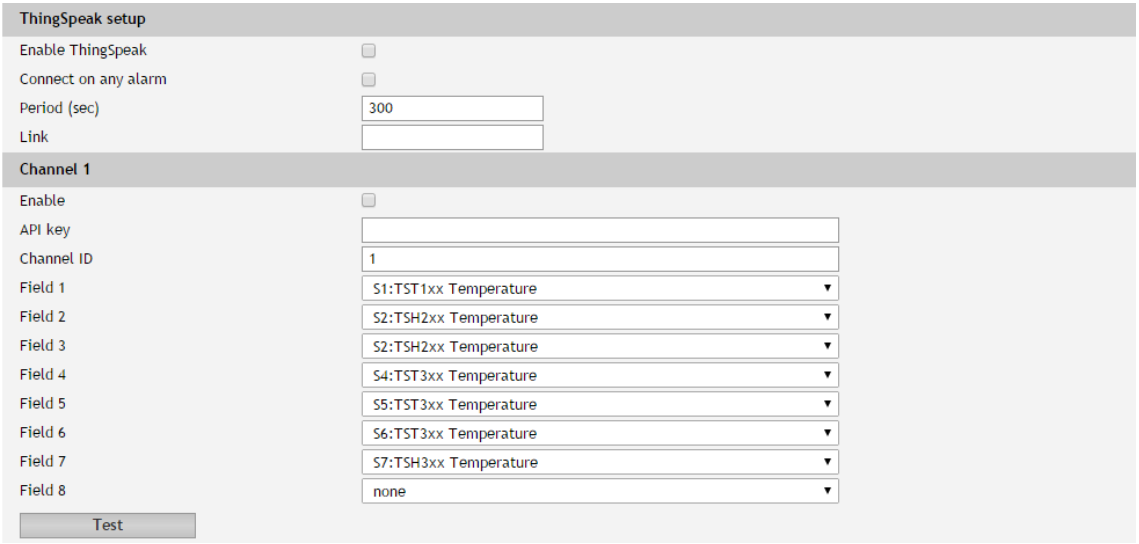
- Period (seconds) - Sets the interval between periodic transmissions. Adjustable from 10 to 14,400 seconds.
- Key - A user-defined identifier included in every XML/JSON payload. This can be used by the receiving server to identify the source device, associate data with a specific installation, or authenticate incoming requests.
- Process answer - When enabled, the TCW241 processes the HTTP response from the remote server. This allows the server to send supported API commands back to the device for remote control or configuration updates (see HTTP API Commands section).
- HTTP User Header - Enables or disables sending a custom HTTP header.
- HTTP User Header Name - Defines the header field name (for example: Authorization, X-API-Key, Content-Type).
- HTTP User Header Value - Defines the value associated with the header field.

### 7.3.6. Cloud

The ThingSpeak server is an open data platform and API designed for the Internet of Things (IoT). It enables users to collect, store, analyze, visualize, and act on data gathered from various sensors.

The central component of ThingSpeak activity is the channel, which includes an API key, channel ID, and eight data fields.

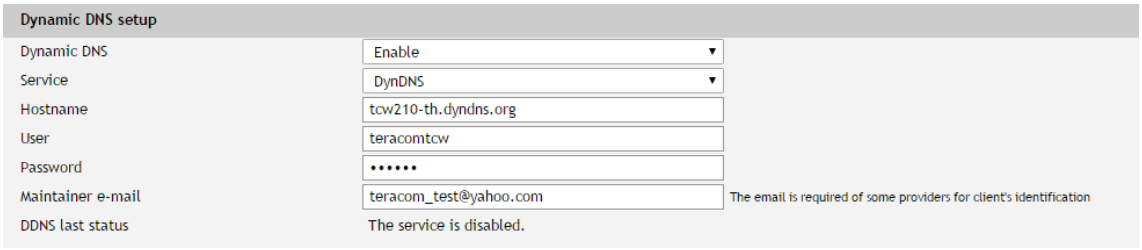
The TCW210-TH features four channel sections: Channel 1, Channel 2, Channel 3, and Channel 4.




### 7.3.7. Dynamic DNS

With dynamic DNS, the TCW210-TH can be accessed from the public Internet without the need for a broadband account with a static IP address.

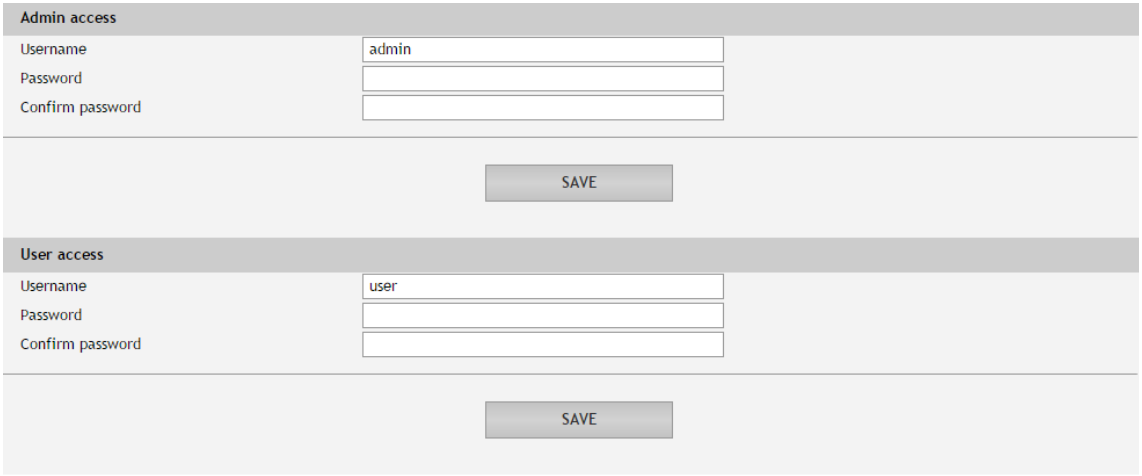
TCW210-TH supports the following DNS services: DynDNS, No-IP, and DNS-O-Matric.




## 7.4. Administration

### 7.4.1. User/Password

The TCW210-TH supports two user roles: "Admin" and "User."



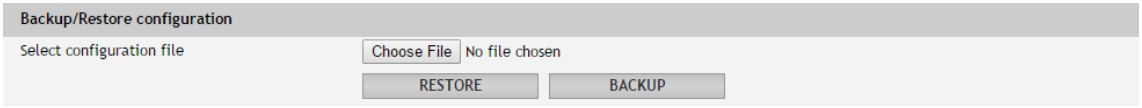

The Admin role has full administrative rights and can modify all settings.

The User role has limited access, allowing only viewing of the Monitoring page and preventing any modifications to settings.

Both usernames and passwords can be up to 31 characters in length.

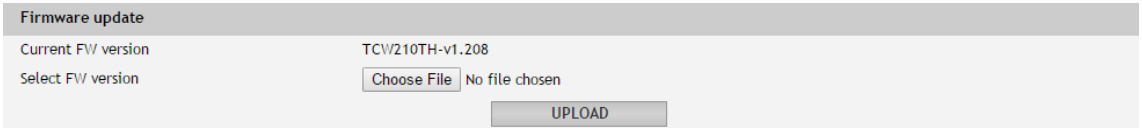
### 7.4.2. Backup/Restore

TCW210-TH allows for the backup and restoration of all user settings. These settings are saved in an XML backup file, which can be utilized for restoring configurations on multiple devices. This feature is particularly useful for applying similar settings across a batch of controllers.




### 7.4.3. FW update

The TCW210-TH can be updated through the web interface.




To update the device, follow these steps:

- Go to www.teracomsystems.com and download the latest firmware;
- From Administration->FW update select downloaded .cod file and click “Upload” button;
- Once the firmware update is complete, the Login page will appear.

Important: Do not turn off the power supply during the update process, as this can damage the device.

## 7.5. Logout

The TCW210-TH supports multi-session access; however, it is recommended to log out after completing your tasks for security and best practices.

# 8. Protocols and API

## 8.1. SNMP

The Simple Network Management Protocol (SNMP) is a standard internet protocol used for managing devices on IP networks. In typical SNMP implementations, one or more administrative computers, referred to as managers, monitor and control devices on a Local Area Network (LAN). Each managed device continuously runs a software component known as an agent, which reports information via SNMP to the manager.

The TCW210-TH can be configured and monitored using SNMP. This can be achieved with any SNMP v.2 compatible program. The parameters that can be modified are organized by function in the tables below. To obtain a valid Object Identifier (OID) number, replace the “x” symbol with “1.3.6.1.4.1.38783.” To save the changes, set configurationSaved (OID x.2.3.5.0) to "1".

**product**

| OID | Name | Access | Description | Syntax |
|---|---|---|---|---|
| x.4.1.1.0 | name | read-only | Device name | DisplayString |
| x.4.1.2.0 | version | read-only | Firmware version | DisplayString |
| x.4.1.3.0 | date | read-only | Release date | DateAndTime |

**setup -> network**

| OID | Name | Access | Description | Syntax |
|---|---|---|---|---|
| x.4.2.1.1.0 | deviceID | read-only | Device ID (default MAC address) | MacAddress |
| x.4.2.1.2.0 | hostName | read-only | Hostname | DisplayString |
| x.4.2.1.3.0 | deviceIP | read-only | Device IP address | IpAddress |

**setup -> io -> sensorsSetup -> sensor1setup**

| OID | Name | Access | Description | Syntax |
|---|---|---|---|---|
| x.4.2.2.1.1.1.0 | s1description | read-write | Sensor 1 description | DisplayString |

| | | | | |
|---|---|---|---|---|
| x.4.2.2.1.1.2.1.0 | s11MAXInt | read-write | S11 maximum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.1.2.2.0 | s11MINInt | read-write | S11 minimum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.1.2.3.0 | s11HYSTInt | read-write | S11 hysteresis value x1000 in Integer format | Integer32 |
| x.4.2.2.1.1.2.4.0 | s11MULTInt | read-write | S11 multiplier value x1000 in Integer format | Integer32 |
| x.4.2.2.1.1.2.5.0 | s11OFFSETInt | read-write | S11 offset value x1000 in Integer format | Integer32 |
| x.4.2.2.1.1.3.1.0 | s12MAXInt | read-write | S12 maximum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.1.3.2.0 | s12MINInt | read-write | S12 minimum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.1.3.3.0 | s12HYSTInt | read-write | S12 hysteresis value x1000 in Integer format | Integer32 |
| x.4.2.2.1.1.3.4.0 | s12MULTInt | read-write | S12 multiplier value x1000 in Integer format | Integer32 |
| x.4.2.2.1.1.3.5.0 | s12OFFSETInt | read-write | S12 offset value x1000 in Integer format | Integer32 |
| x.4.2.2.1.1.4.1.0 | s13MAXInt | read-write | S13 maximum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.1.4.2.0 | s13MINInt | read-write | S13 minimum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.1.4.3.0 | s13HYSTInt | read-write | S13 hysteresis value x1000 in Integer format | Integer32 |

### setup -> io -> sensorsSetup -> sensor2setup

| OID | Name | Access | Description | Syntax |
|---|---|---|---|---|
| x.4.2.2.1.2.1.0 | s2description | read-write | Sensor2 description | DisplayString |
| x.4.2.2.1.2.2.1.0 | s21MAXInt | read-write | s21 maximum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.2.2.2.0 | s21MINInt | read-write | S21 minimum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.2.2.3.0 | s21HYSTInt | read-write | S21 hysteresis value x1000 in Integer format | Integer32 |
| x.4.2.2.1.2.2.4.0 | s21MULTInt | read-write | S21 multiplier value x1000 in Integer format | Integer32 |
| x.4.2.2.1.2.2.5.0 | s21OFFSETInt | read-write | S21 offset value x1000 in Integer format | Integer32 |
| x.4.2.2.1.2.3.1.0 | s22MAXInt | read-write | S22 maximum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.2.3.2.0 | s22MINInt | read-write | S22 minimum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.2.3.3.0 | s22HYSTInt | read-write | S22 hysteresis value x1000 in Integer format | Integer32 |
| x.4.2.2.1.2.3.4.0 | S22MULTInt | read-write | S22 multiplier value x1000 in Integer format | Integer32 |
| x.4.2.2.1.2.3.5.0 | s22OFFSETInt | read-write | S22 offset value x1000 in Integer format | Integer32 |
| x.4.2.2.1.2.4.1.0 | s23MAXInt | read-write | S23 maximum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.2.4.2.0 | s23MINInt | read-write | S23 minimum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.2.4.3.0 | s23HYSTInt | read-write | S23 hysteresis value x1000 in Integer format | Integer32 |

### setup -> io -> sensorsSetup -> sensor3setup

| OID | Name | Access | Description | Syntax |
|---|---|---|---|---|
| x.4.2.2.1.3.1.0 | s3description | read-write | Sensor 3 description | DisplayString |
| x.4.2.2.1.3.2.1.0 | s31MAXInt | read-write | S31 maximum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.3.2.2.0 | s31MINInt | read-write | S31 minimum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.3.2.3.0 | s31HYSTInt | read-write | S31 hysteresis value x1000 in Integer format | Integer32 |
| x.4.2.2.1.3.2.4.0 | s31MULTInt | read-write | S31 multiplier value x1000 in Integer format | Integer32 |
| x.4.2.2.1.3.2.5.0 | s31OFFSETInt | read-write | S31 offset value x1000 in Integer format | Integer32 |
| x.4.2.2.1.3.3.1.0 | s32MAXInt | read-write | S32 maximum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.3.3.2.0 | s32MINInt | read-write | S32 minimum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.3.3.3.0 | s32HYSTInt | read-write | S32 hysteresis value x1000 in Integer format | Integer32 |
| x.4.2.2.1.3.3.4.0 | s32MULTInt | read-write | S32 multiplier value x1000 in Integer format | Integer32 |
| x.4.2.2.1.3.3.5.0 | s32OFFSETInt | read-write | S32 offset value x1000 in Integer format | Integer32 |
| x.4.2.2.1.3.4.1.0 | s33MAXInt | read-write | S33 maximum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.3.4.2.0 | s33MINInt | read-write | S33 minimum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.3.4.3.0 | s33HYSTInt | read-write | S33 hysteresis value x1000 in Integer format | Integer32 |

### setup -> io -> sensorsSetup -> sensor4setup

| OID | Name | Access | Description | Syntax |
|---|---|---|---|---|
| x.4.2.2.1.4.1.0 | s4description | read-write | Sensor 4 description | DisplayString |
| x.4.2.2.1.4.2.1.0 | s41MAXInt | read-write | S41 maximum value x1000 in Integer format | Integer32 |

| | | | | |
|---|---|---|---|---|
| x.4.2.2.1.4.2.2.0 | s41MINInt | read-write | S41 minimum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.4.2.3.0 | s41HYSTInt | read-write | S41 hysteresis value x1000 in Integer format | Integer32 |
| x.4.2.2.1.4.2.4.0 | s41MULTInt | read-write | S41 multiplier value x1000 in Integer format | Integer32 |
| x.4.2.2.1.4.2.5.0 | s41OFFSETInt | read-write | S41 offset value x1000 in Integer format | Integer32 |
| x.4.2.2.1.4.3.1.0 | s42MAXInt | read-write | S42 maximum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.4.3.2.0 | s42MINInt | read-write | S42 minimum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.4.3.3.0 | s42HYSTInt | read-write | S42 hysteresis value x1000 in Integer format | Integer32 |
| x.4.2.2.1.4.3.4.0 | s42MULTInt | read-write | S42 multiplier value x1000 in Integer format | Integer32 |
| x.4.2.2.1.4.3.5.0 | s42OFFSETInt | read-write | S42 offset value x1000 in Integer format | Integer32 |
| x.4.2.2.1.4.4.1.0 | s43MAXInt | read-write | S43 maximum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.4.4.2.0 | s43MINInt | read-write | S43 minimum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.4.4.3.0 | s43HYSTInt | read-write | S43 hysteresis value x1000 in Integer format | Integer32 |

### setup -> io -> sensorsSetup -> sensor5setup

| OID | Name | Access | Description | Syntax |
|---|---|---|---|---|
| x.4.2.2.1.5.1.0 | s5description | read-write | Sensor 5 description | DisplayString |
| x.4.2.2.1.5.2.1.0 | s51MAXInt | read-write | S51 maximum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.5.2.2.0 | s51MINInt | read-write | S51 minimum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.5.2.3.0 | s51HYSTInt | read-write | S51 hysteresis value x1000 in Integer format | Integer32 |
| x.4.2.2.1.5.2.4.0 | s51MULTInt | read-write | S51 multiplier value x1000 in Integer format | Integer32 |
| x.4.2.2.1.5.2.5.0 | s51OFFSETInt | read-write | S51 offset value x1000 in Integer format | Integer32 |
| x.4.2.2.1.5.3.1.0 | s52MAXInt | read-write | S52 maximum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.5.3.2.0 | s52MINInt | read-write | S52 minimum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.5.3.3.0 | s52HYSTInt | read-write | S52 hysteresis value x1000 in Integer format | Integer32 |
| x.4.2.2.1.5.3.4.0 | s52MULTInt | read-write | S52 multiplier value x1000 in Integer format | Integer32 |
| x.4.2.2.1.5.3.5.0 | s52OFFSETInt | read-write | S52 offset value x1000 in Integer format | Integer32 |
| x.4.2.2.1.5.4.1.0 | s53MAXInt | read-write | S53 maximum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.5.4.2.0 | s53MINInt | read-write | S53 minimum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.5.4.3.0 | s53HYSTInt | read-write | S53 hysteresis value x1000 in Integer format | Integer32 |

### setup -> io -> sensorsSetup -> sensor6setup

| OID | Name | Access | Description | Syntax |
|---|---|---|---|---|
| x.4.2.2.1.6.1.0 | s6description | read-write | Sensor 6 description | DisplayString |
| x.4.2.2.1.6.2.1.0 | s61MAXInt | read-write | S61 maximum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.6.2.2.0 | s61MINInt | read-write | S61 minimum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.6.2.3.0 | s61HYSTInt | read-write | S61 hysteresis value x1000 in Integer format | Integer32 |
| x.4.2.2.1.6.2.4.0 | s61MULTInt | read-write | S61 multiplier value x1000 in Integer format | Integer32 |
| x.4.2.2.1.6.2.5.0 | s61OFFSETInt | read-write | S61 offset value x1000 in Integer format | Integer32 |
| x.4.2.2.1.6.3.1.0 | s62MAXInt | read-write | S62 maximum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.6.3.2.0 | s62MINInt | read-write | S62 minimum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.6.3.3.0 | s62HYSTInt | read-write | S62 hysteresis value x1000 in Integer format | Integer32 |
| x.4.2.2.1.6.3.4.0 | s62MULTInt | read-write | S62 multiplier value x1000 in Integer format | Integer32 |
| x.4.2.2.1.6.3.5.0 | s62OFFSETInt | read-write | S62 offset value x1000 in Integer format | Integer32 |
| x.4.2.2.1.6.4.1.0 | s63MAXInt | read-write | S63 maximum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.6.4.2.0 | s63MINInt | read-write | S63 minimum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.6.4.3.0 | s63HYSTInt | read-write | S63 hysteresis value x1000 in Integer format | Integer32 |

### setup -> io -> sensorsSetup -> sensor7setup

| OID | Name | Access | Description | Syntax |
|---|---|---|---|---|
| x.4.2.2.1.7.1.0 | s7description | read-write | Sensor 7 description | DisplayString |

| | | | | |
|---|---|---|---|---|
| x.4.2.2.1.7.2.1.0 | s71MAXInt | read-write | S71 maximum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.7.2.2.0 | s71MINInt | read-write | S71 minimum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.7.2.3.0 | s71HYSTInt | read-write | S71 hysteresis value x1000 in Integer format | Integer32 |
| x.4.2.2.1.7.2.4.0 | s71MULTInt | read-write | S71 multiplier value x1000 in Integer format | Integer32 |
| x.4.2.2.1.7.2.5.0 | s71OFFSETInt | read-write | S71 offset value x1000 in Integer format | Integer32 |
| x.4.2.2.1.7.3.1.0 | s72MAXInt | read-write | S72 maximum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.7.3.2.0 | s72MINInt | read-write | S72 minimum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.7.3.3.0 | s72HYSTInt | read-write | S72 hysteresis value x1000 in Integer format | Integer32 |
| x.4.2.2.1.7.3.4.0 | s72MULTInt | read-write | S72 multiplier value x1000 in Integer format | Integer32 |
| x.4.2.2.1.7.3.5.0 | s72OFFSETInt | read-write | S72 offset value x1000 in Integer format | Integer32 |
| x.4.2.2.1.7.4.1.0 | s73MAXInt | read-write | S73 maximum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.7.4.2.0 | s73MINInt | read-write | S73 minimum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.7.4.3.0 | s73HYSTInt | read-write | S73 hysteresis value x1000 in Integer format | Integer32 |

### setup -> io -> sensorsSetup -> sensor8setup

| OID | Name | Access | Description | Syntax |
|---|---|---|---|---|
| x.4.2.2.1.8.1.0 | s8description | read-write | Sensor 8 description | DisplayString |
| x.4.2.2.1.8.2.1.0 | s81MAXx10Int | read-write | S81 maximum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.8.2.2.0 | s81MINx10Int | read-write | S81 minimum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.8.2.3.0 | s81HYSTx10Int | read-write | S81 hysteresis value x1000 in Integer format | Integer32 |
| x.4.2.2.1.8.2.4.0 | s81MULTInt | read-write | S81 multiplier value x1000 in Integer format | Integer32 |
| x.4.2.2.1.8.2.5.0 | s81OFFSETInt | read-write | S81 offset value x1000 in Integer format | Integer32 |
| x.4.2.2.1.8.3.1.0 | s82MAXx10Int | read-write | S82 maximum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.8.3.2.0 | s82MINx10Int | read-write | S82 minimum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.8.3.3.0 | s82HYSTx10Int | read-write | S82 hysteresis value x1000 in Integer format | Integer32 |
| x.4.2.2.1.8.3.4.0 | s82MULTInt | read-write | S82 multiplier value x1000 in Integer format | Integer32 |
| x.4.2.2.1.8.3.5.0 | s82OFFSETInt | read-write | S82 offset value x1000 in Integer format | Integer32 |
| x.4.2.2.1.8.4.1.0 | s83MAXx10Int | read-write | S83 maximum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.8.4.2.0 | s83MINx10Int | read-write | S83 minimum value x1000 in Integer format | Integer32 |
| x.4.2.2.1.8.4.3.0 | s83HYSTx10Int | read-write | S83 hysteresis value x1000 in Integer format | Integer32 |

### monitorNcontrol -> sensors -> sensor1

| OID | Name | Access | Description | Syntax |
|---|---|---|---|---|
| x.4.3.1.1.1.0 | s11Int | read-only | S11 value x1000 in Integer format | Integer32 |
| x.4.3.1.1.2.0 | s12Int | read-only | S12 value x1000 in Integer format | Integer32 |
| x.4.3.1.1.3.0 | s13Int | read-only | S13 value x1000 in Integer format | Integer32 |
| x.4.3.1.1.4.0 | s1ID | read-only | S1 ID value | OCTET STRING (SIZE (16)) |
| x.4.3.1.1.5.1.0 | s11Al | read-only | S11 alarm status | INTEGER {normal(0),alarm(1)} |
| x.4.3.1.1.5.2.0 | s12Al | read-only | S12 alarm status | INTEGER {normal(0),alarm(1)} |
| x.4.3.1.1.5.3.0 | s13Al | read-only | S13 alarm status | INTEGER {normal(0),alarm(1)} |
| x.4.3.1.1.6.0 | s11RawInt | read-only | S11 raw value x1000 in Integer format | Integer32 |
| x.4.3.1.1.7.0 | s12RawInt | read-only | S12 raw value x1000 in Integer format | Integer32 |

### monitorNcontrol -> sensors -> sensor2

| OID | Name | Access | Description | Syntax |
|---|---|---|---|---|
| x.4.3.1.2.1.0 | s21Int | read-only | S21 value x1000 in Integer format | Integer32 |
| x.4.3.1.2.2.0 | s22Int | read-only | S22 value x1000 in Integer format | Integer32 |
| x.4.3.1.2.3.0 | s23Int | read-only | S23 value x1000 in Integer format | Integer32 |
| x.4.3.1.2.4.0 | s2ID | read-only | S2 ID value | OCTET STRING (SIZE (16)) |
| x.4.3.1.2.5.1.0 | s21Al | read-only | S21 alarm status | INTEGER |

| | | | | |
|---|---|---|---|---|
| | | | | {normal(0),alarm(1)} |
| x.4.3.1.2.5.2.0 | s22Al | read-only | S22 alarm status | INTEGER {normal(0),alarm(1)} |
| x.4.3.1.2.5.3.0 | s23Al | read-only | S23 alarm status | INTEGER {normal(0),alarm(1)} |
| x.4.3.1.2.6.0 | s21RawInt | read-only | S21 raw value x1000 in Integer format | Integer32 |
| x.4.3.1.2.7.0 | s22RawInt | read-only | S22 raw value x1000 in Integer format | Integer32 |

### monitorNcontrol -> sensors -> sensor3

| OID | Name | Access | Description | Syntax |
|---|---|---|---|---|
| x.4.3.1.3.1.0 | s31Int | read-only | S31 value x1000 in Integer format | Integer32 |
| x.4.3.1.3.2.0 | s32Int | read-only | S32 value x1000 in Integer format | Integer32 |
| x.4.3.1.3.3.0 | s3ID | read-only | S3 ID value | OCTET STRING (SIZE (16)) |
| x.4.3.1.3.5.1.0 | s31Al | read-only | S31 alarm status | INTEGER {normal(0),alarm(1)} |
| x.4.3.1.3.5.2.0 | s32Al | read-only | S32 alarm status | INTEGER {normal(0),alarm(1)} |
| x.4.3.1.3.5.3.0 | s33Al | read-only | S33 alarm status | INTEGER {normal(0),alarm(1)} |
| x.4.3.1.3.6.0 | s31RawInt | read-only | S31 raw value x1000 in Integer format | Integer32 |
| x.4.3.1.3.7.0 | s32RawInt | read-only | S32 raw value x1000 in Integer format | Integer32 |

### monitorNcontrol -> sensors -> sensor4

| OID | Name | Access | Description | Syntax |
|---|---|---|---|---|
| x.4.3.1.4.1.0 | s41Int | read-only | S41 value x1000 in Integer format | Integer32 |
| x.4.3.1.4.2.0 | s42Int | read-only | S42 value x1000 in Integer format | Integer32 |
| x.4.3.1.4.3.0 | s43Int | read-only | S43 value x1000 in Integer format | Integer32 |
| x.4.3.1.4.3.0 | s4ID | read-only | S4 ID value | OCTET STRING (SIZE (16)) |
| x.4.3.1.4.5.1.0 | s41Al | read-only | S41 alarm status | INTEGER {normal(0),alarm(1)} |
| x.4.3.1.4.5.2.0 | s42Al | read-only | S42 alarm status | INTEGER {normal(0),alarm(1)} |
| x.4.3.1.4.5.3.0 | s43Al | read-only | S43 alarm status | INTEGER {normal(0),alarm(1)} |
| x.4.3.1.4.6.0 | s41RawInt | read-only | S41 raw value x1000 in Integer format | Integer32 |
| x.4.3.1.4.7.0 | s42RawInt | read-only | S42 raw value x1000 in Integer format | Integer32 |

### monitorNcontrol -> sensors -> sensor5

| OID | Name | Access | Description | Syntax |
|---|---|---|---|---|
| x.4.3.1.5.1.0 | s51Int | read-only | S51 value x1000 in Integer format | Integer32 |
| x.4.3.1.5.2.0 | s52Int | read-only | S52 value x1000 in Integer format | Integer32 |
| x.4.3.1.5.3.0 | s53Int | read-only | S53 value x1000 in Integer format | Integer32 |
| x.4.3.1.5.4.0 | s5ID | read-only | S5 ID value | OCTET STRING (SIZE (16)) |
| x.4.3.1.5.5.1.0 | s51Al | read-only | S51 alarm status | INTEGER {normal(0),alarm(1)} |
| x.4.3.1.5.5.2.0 | s52Al | read-only | S52 alarm status | INTEGER {normal(0),alarm(1)} |
| x.4.3.1.5.5.3.0 | s53Al | read-only | S53 alarm status | INTEGER {normal(0),alarm(1)} |
| x.4.3.1.5.6.0 | s51RawInt | read-only | S51 raw value x1000 in Integer format | Integer32 |
| x.4.3.1.5.7.0 | s52RawInt | read-only | S52 raw value x1000 in Integer format | Integer32 |

### monitorNcontrol -> sensors -> sensor6

| OID | Name | Access | Description | Syntax |
|---|---|---|---|---|
| x.4.3.1.6.1.0 | s61Int | read-only | S61 value x1000 in Integer format | Integer32 |
| x.4.3.1.6.2.0 | s62Int | read-only | S62 value x1000 in Integer format | Integer32 |
| x.4.3.1.6.3.0 | s63Int | read-only | S63 value x1000 in Integer format | Integer32 |
| x.4.3.1.6.4.0 | s6ID | read-only | S6 ID value | OCTET STRING (SIZE (16)) |
| x.4.3.1.6.5.1.0 | s61Al | read-only | S61 alarm status | INTEGER {normal(0),alarm(1)} |
| x.4.3.1.6.5.2.0 | s62Al | read-only | S62 alarm status | INTEGER {normal(0),alarm(1)} |
| x.4.3.1.6.5.3.0 | s63Al | read-only | S63 alarm status | INTEGER |

| | | | | |
|---|---|---|---|---|
| | | | | {normal(0),alarm(1)} |
| x.4.3.1.6.6.0 | s61RawInt | read-only | S61 raw value x1000 in Integer format | Integer32 |
| x.4.3.1.6.7.0 | s62RawInt | read-only | S62 raw value x1000 in Integer format | Integer32 |

**monitorNcontrol -> sensors -> sensor7**

| OID | Name | Access | Description | Syntax |
|---|---|---|---|---|
| x.4.3.1.7.1.0 | s71Int | read-only | S71 value x1000 in Integer format | Integer32 |
| x.4.3.1.7.2.0 | s72Int | read-only | S72 value x1000 in Integer format | Integer32 |
| x.4.3.1.7.3.0 | s73Int | read-only | S73 value x1000 in Integer format | Integer32 |
| x.4.3.1.7.4.0 | s7ID | read-only | S7 ID value | OCTET STRING (SIZE (16)) |
| x.4.3.1.7.5.1.0 | s71Al | read-only | S71 alarm status | INTEGER {normal(0),alarm(1)} |
| x.4.3.1.7.5.2.0 | s72Al | read-only | S72 alarm status | INTEGER {normal(0),alarm(1)} |
| x.4.3.1.7.5.3.0 | s73Al | read-only | S73 alarm status | INTEGER {normal(0),alarm(1)} |
| x.4.3.1.7.6.0 | s71RawInt | read-only | S71 raw value x1000 in Integer format | Integer32 |
| x.4.3.1.7.7.0 | s72RawInt | read-only | S72 raw value x1000 in Integer format | Integer32 |

**monitorNcontrol -> sensors -> sensor8**

| OID | Name | Access | Description | Syntax |
|---|---|---|---|---|
| x.4.3.1.8.1.0 | s81Int | read-only | S81 value x1000 in Integer format | Integer32 |
| x.4.3.1.8.2.0 | s82Int | read-only | S82 value x1000 in Integer format | Integer32 |
| x.4.3.1.8.3.0 | s83Int | read-only | S83 value x1000 in Integer format | Integer32 |
| x.4.3.1.8.4.0 | s8ID | read-only | S8 ID value | OCTET STRING (SIZE (16)) |
| x.4.3.1.8.5.1.0 | s81Al | read-only | S81 alarm status | INTEGER {normal(0),alarm(1)} |
| x.4.3.1.8.5.2.0 | s82Al | read-only | S82 alarm status | INTEGER {normal(0),alarm(1)} |
| x.4.3.1.8.5.3.0 | s83Al | read-only | S83 alarm status | INTEGER {normal(0),alarm(1)} |
| x.4.3.1.8.6.0 | s81RawInt | read-only | S81 raw value x1000 in Integer format | Integer32 |
| x.4.3.1.8.7.0 | s82RawInt | read-only | S82 raw value x1000 in Integer format | Integer32 |

**monitorNcontrol**

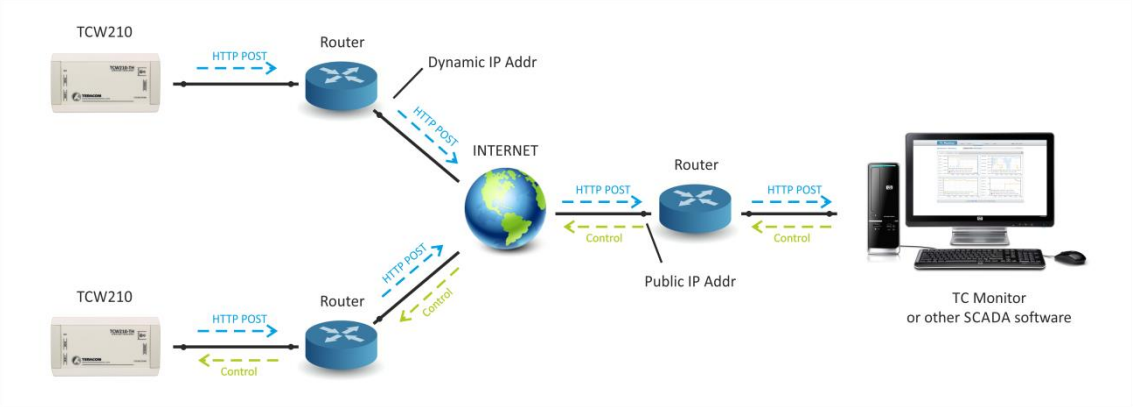
| OID | Name | Access | Description | Syntax |
|---|---|---|---|---|
| x.4.3.5.0 | configurationSaved | read-write | Configuration save status SAVED/UNSAVED | INTEGER { unsaved(0), saved(1) } |
| x.4.3.6.0 | restartDevice | read-write | Restart Device | INTEGER { cancel(0), restart(1) } |
| x.4.3.7.0 | temperatureUnit | read-only | Unit of the all temperature values | INTEGER { celcius(0), fahrenheit(1) } |
| x.4.3.8.0 | hardwareErr | read-only | Hardware Error | INTEGER { noErr(0), owErr(1), hwErr(2) } |

## 8.2. HTTP API

### 8.2.1. HTTP POST

TCW210-TH can execute HTTP/HTTPS POST requests to upload XML or JSON files to a dedicated server. This functionality is particularly useful when the controller is situated behind a router without a public IP address or when the user lacks access to the router's configuration. It is important to note that the server should possess a public IP address.

The typical monitoring application is shown in the picture below:




HTTP/HTTPS POSTs can be sent either periodically or periodically in conjunction with alarm conditions. In response, the server can send an HTTP GET request with the appropriate command—refer to section 8.2.3 for details on HTTP API commands.

To test HTTP/HTTPS POST functionality, follow these steps:

- Save following code like post.php:

```
<?php
    define("FILENAME", 'status.xml');
    define("FOLDER", '');
    define("SEPARATOR", '');
    define("STR_SUCCESS", 'set FIN');
    define("STR_ERROR", 'error');

    if($_SERVER['REQUEST_METHOD'] == 'POST'){
        $datePrefix = date('YmdHis', strtotime('now'));
        $pathname = FOLDER.SEPARATOR.$datePrefix.'_'.FILENAME;
        $postdata = file_get_contents("php://input");
        $handle = fopen($pathname, 'w+');
        $content = var_export($postdata, true);
        fwrite($handle, substr($content, 1, strlen($content)-2));
        fclose($handle);
        echo (($handle === false) ? STR_ERROR : STR_SUCCESS)."\r\n";
    }
        else {
            echo "The PHP script is working!";
        }
?>
```

- Upload the post.php file to a public web server that supports PHP. To ensure the script is functioning correctly, enter the URL (for example, www.yourserverURL.com/post.php) in your web browser. If successful, a webpage displaying “The PHP script is working!” will appear.
- Configure the TCW210-TH controller to send HTTP/HTTPS POST requests to your web server. Input the URL (yourserverURL.com/post.php) in the URL field, and click the “Test HTTP POST” button.
- If the HTTP/HTTPS POST request is received and processed successfully, “OK” will be displayed next to the button. Additionally, an XML file will be created in the same directory where post.php is located. The filename will include a timestamp, formatted as 20151120103318_status.xml.

### 8.2.2. HTTP GET

HTTP GET can be utilized to monitor the TCW210-TH by retrieving XML or JSON files. The format for accessing the status is as follows:

http://device.ip.address/status.xml

http://device.ip.address/status.json

For further details regarding the structure of these files, refer to sections 8.2.4 XML file structure and 8.2.5 JSON file structure.

HTTP GET requests can be sent at any time to the TCW210-TH if it is on the same network or has appropriate routing. If there is no direct access to the device, an HTTP GET can be sent immediately after receiving an HTTP POST from the same device.

#### 8.2.2.1. Commands

All commands used with HTTP POST are also applicable to HTTP GET. The correct format for sending commands is:

http://device.ip.address/status.xml?yyy=xxx

Where:

yyy is the command;

xxx is the parameter

Example:

http://device.ip.address/status.xml?pper=120 will set the POST period to 120 seconds.

#### 8.2.2.2. HTTP GET authentication

If HTTP API authentication is enabled, basic access authentication is required to access the status.xml file. The format of the command is detailed in the table below:

| XML/HTTP API authentication | Format |
|---|---|
| enabled | http://device.ip.address/status.xml?a=uuuu:pppp |
| disabled | http://device.ip.address/status.xml |

Example:

To set the POST period to 120 seconds, the following command can be used, assuming the username is admin and the password is admin

http://device.ip.address/status.xml?a=admin:admin&pper=120

### 8.2.3. List of HTTP API commands

| Command | Description |
|---|---|
| sn**pt**=30.0 | Set Min of sensor to 30.0<br>(**p** is 1,2,3,4,5,6,7 or 8 for the respective sensor<br>**t** is 1 or 2 for the respective parameter of sensor)<br>sn12=30.0 will set Min for sensor 1, parameter 2 |
| sx**pt**=40.0 | Set Max of sensor to 40.0<br>(**p** is 1,2,3,4,5,6,7 or 8 for the respective sensor<br>**t** is 1 or 2 for the respective parameter of sensor)<br>sx42=40.0 will set Min for sensor 4, parameter 2 |
| sy**pt**=2.0 | Set Hys of sensor to 2.0<br>(**p** is 1,2,3,4,5,6,7 or 8 for the respective sensor<br>**t** is 1 or 2 for the respective parameter of sensor)<br>sy81=2.0 will set Hys for sensor 8, parameter 1 |
| delsen=xxxx | Notification delay for sensors<br>(xxxx is between 0 and 3600) |
| dataf=x | Data format XML/JSON for HHTP POST – 0 XML, 1 JSON |

| | |
|---|---|
| pushtls=x | HTTP/HTTPS protocol, where x is 0 for HTTP and 1 for HTTPS |
| purl=yyy | URL for HTTP POST to Server 1, where yyy is a full path to php file. Example: purl=212.25.45.120:30181/xampp/test/posttest.php |
| pper=x | HTTP POST period in seconds (x is between 10 and 14400) |
| dk=xxx | HTTP POST key – xxx is up to 17 characters |
| save | Save all previous changes (except relays' one) in the FLASH memory. **As every save reflects the FLASH cycles (endurance), this command should be used very carefully.** pper=120&save – will set Post period to 120 seconds and save it |
| FIN | Terminate session. (It works with HTTP/HTTPS POST, but not with HTTP GET) |

### 8.2.4. XML file structure

```
<Monitor>
  <DeviceInfo>
      <DeviceName>TCW210-TH</DeviceName>
      <HostName>TCW210TH</HostName>
      <ID>54:10:EC:0C:1D:E1</ID>
      <FwVer>TCW210TH-v1.252</FwVer>
      <MnfInfo>www.teracomsystems.com</MnfInfo>
      <SysContact>info@teracomsystems.com</SysContact>
      <SysName>SysName</SysName>
      <SysLocation>SysLocation</SysLocation>
  </DeviceInfo>
  <S>
      <S1>
          <description>S1:TST1xx</description>
          <id>2867895F07000058</id>
          <SenType>1W</SenType>
          <addr>---</addr>
          <item1>
              <value>22.875</value>
              <unit>°C</unit>
              <alarm>0</alarm>
              <min>19.000</min>
              <max>85.000</max>
              <hys>0.500</hys>
          </item1>
          <item2>
              <value>---</value>
```

<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item2>
<item3>
<value>---</value>
<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item3>
</S1>
<S2>
<description>S2</description>
<id>0000000000000000</id>
<SenType>1W</SenType>
<addr>---</addr>
<item1>
<value>---</value>
<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item1>
<item2>
<value>---</value>
<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item2>
<item3>
<value>---</value>
<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item3>
</S2>
<S3>
<description>S3</description>
<id>0000000000000000</id>
<SenType>1W</SenType>
<addr>---</addr>
<item1>
<value>---</value>

<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item1>
<item2>
<value>---</value>
<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item2>
<item3>
<value>---</value>
<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item3>
</S3>
<S4>
<description>S4</description>
<id>0000000000000000</id>
<SenType>1W</SenType>
<addr>---</addr>
<item1>
<value>---</value>
<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item1>
<item2>
<value>---</value>
<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item2>
<item3>
<value>---</value>
<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item3>

```
</S4>
<S5>
    <description>S5</description>
    <id>CD00000000000000</id>
    <SenType>MB</SenType>
    <addr>1</addr>
    <item1>
        <value>22.963</value>
        <unit>°C</unit>
        <alarm>0</alarm>
        <min>-40.000</min>
        <max>85.000</max>
        <hys>8.500</hys>
    </item1>
    <item2>
        <value>31.279</value>
        <unit>%RH</unit>
        <alarm>0</alarm>
        <min>0.000</min>
        <max>100.000</max>
        <hys>10.000</hys>
    </item2>
    <item3>
        <value>5.073</value>
        <unit>°C</unit>
        <alarm>0</alarm>
        <min>0.000</min>
        <max>25.000</max>
        <hys>2.500</hys>
    </item3>
</S5>
<S6>
    <description>S6</description>
    <id>0000000000000000</id>
    <SenType>MB</SenType>
    <addr>2</addr>
    <item1>
        <value>24.742</value>
        <unit>°C</unit>
        <alarm>0</alarm>
        <min>-40.000</min>
        <max>85.000</max>
        <hys>8.500</hys>
    </item1>
    <item2>
        <value>---</value>
        <unit>---</unit>
        <alarm>0</alarm>
        <min>---</min>
        <max>---</max>
        <hys>---</hys>
    </item2>
```

<item3>
<value>---</value>
<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item3>
</S6>
<S7>
<description>S7</description>
<id>0000000000000000</id>
<SenType>MB</SenType>
<addr>0</addr>
<item1>
<value>---</value>
<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item1>
<item2>
<value>---</value>
<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item2>
<item3>
<value>---</value>
<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item3>
</S7>
<S8>
<description>S8</description>
<id>0000000000000000</id>
<SenType>MB</SenType>
<addr>0</addr>
<item1>
<value>---</value>
<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item1>

```
            <item2>
                <value>---</value>
                <unit>---</unit>
                <alarm>0</alarm>
                <min>---</min>
                <max>---</max>
                <hys>---</hys>
            </item2>
            <item3>
                <value>---</value>
                <unit>---</unit>
                <alarm>0</alarm>
                <min>---</min>
                <max>---</max>
                <hys>---</hys>
            </item3>
        </S8>
    </S>
    <HTTPPush>
        <Key/>
        <PushPeriod>300</PushPeriod>
    </HTTPPush>
    <hwerr/>
    <Alarmed>0</Alarmed>
    <Scannig/>
    <Time>
        <Date>11.10.2023</Date>
        <Time>08:45:18</Time>
    </Time>
</Monitor>
```

Where:

<value>---</value> and <unit>---</unit> means no sensor on this position;

<alarm>1</alarm> means there is trigger condition.

### 8.2.5. JSON file structure

```
{
  "Monitor": {
    "DeviceInfo": {
      "DeviceName": "TCW210-TH",
      "HostName": "TCW210TH",
      "ID": "54:10:EC:0C:1D:E1",
      "FwVer": "TCW210TH-v1.252",
      "MnfInfo": "www.teracomsystems.com",
      "SysContact": "info@teracomsystems.com",
      "SysName": "SysName",
      "SysLocation": "SysLocation"
    },
    "S": {
      "S1": {
        "description": "S1:TST1xx",
        "id": "2867895F07000058",
        "SenType": "1W",
        "addr": "---",
        "item1": {
          "value": "23.063",
          "unit": "°C",
          "alarm": "0",
```

```
      "min": "19.000",
      "max": "85.000",
      "hys": "0.500"
    },
    "item2": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    },
    "item3": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    }
  },
  "S2": {
    "description": "S2",
    "id": "0000000000000000",
    "SenType": "1W",
    "addr": "---",
    "item1": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    },
    "item2": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    },
    "item3": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    }
  },
  "S3": {
    "description": "S3",
    "id": "0000000000000000",
    "SenType": "1W",
    "addr": "---",
    "item1": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    },
    "item2": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    },
    "item3": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
```

```
      "max": "---",
      "hys": "---"
    }
  },
  "S4": {
    "description": "S4",
    "id": "0000000000000000",
    "SenType": "1W",
    "addr": "---",
    "item1": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    },
    "item2": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    },
    "item3": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    }
  },
  "S5": {
    "description": "S5",
    "id": "CD00000000000000",
    "SenType": "MB",
    "addr": "1",
    "item1": {
      "value": "23.160",
      "unit": "°C",
      "alarm": "0",
      "min": "-40.000",
      "max": "85.000",
      "hys": "8.500"
    },
    "item2": {
      "value": "31.803",
      "unit": "%RH",
      "alarm": "0",
      "min": "0.000",
      "max": "100.000",
      "hys": "10.000"
    },
    "item3": {
      "value": "5.483",
      "unit": "°C",
      "alarm": "0",
      "min": "0.000",
      "max": "25.000",
      "hys": "2.500"
    }
  },
  "S6": {
    "description": "S6",
    "id": "0000000000000000",
    "SenType": "MB",
    "addr": "2",
    "item1": {
      "value": "25.125",
      "unit": "°C",
      "alarm": "0",
      "min": "-40.000",
      "max": "85.000",
      "hys": "8.500"
    },
```

```
    "item2": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    },
    "item3": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    }
  },
  "S7": {
    "description": "S7",
    "id": "0000000000000000",
    "SenType": "MB",
    "addr": "0",
    "item1": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    },
    "item2": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    },
    "item3": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    }
  },
  "S8": {
    "description": "S8",
    "id": "0000000000000000",
    "SenType": "MB",
    "addr": "0",
    "item1": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    },
    "item2": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    },
    "item3": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    }
  }
```

```
    },
    "HTTPPush": {
      "Key": "",
      "PushPeriod": "300"
    },
    "hwerr": "",
    "Alarmed": "0",
    "Scannig": "",
    "Time": {
      "Date": "11.10.2023",
      "Time": "09:00:12"
    }
  }
}
```

## 8.3. Modbus TCP/IP

Modbus TCP/IP is a serial communications protocol originally published by Modicon in 1979. It facilitates master-slave/client-server communication between intelligent devices. This protocol is commonly employed to connect a supervisory computer with a remote terminal unit (RTU) within supervisory control and data acquisition (SCADA) systems.

Modbus TCP/IP operates over Ethernet, enabling communication between devices such as sensors, actuators, and controllers. The protocol is widely adopted due to its simplicity and reliability in industrial and automation applications. It allows for the exchange of data and commands, making it an essential component for integrating and monitoring various devices within a networked environment.

### 8.3.1. Codes and answers

#### 8.3.1.1. Read Holding Registers (FC=03)

Request

This command is requesting the content of holding registers 19800.

**03 4D58 0002**

03: The Function Code 3 (read Holding Registers)
4D58: The Data Address of the first register requested (4D58 hex = 19800)
0002: The total number of registers requested. (read 2 registers each 2 byte = 4 bytes)

Response

**03 04 41BD 0655**

03: The Function Code 3 (read Sensor 1 Part 1 Holding Registers)
04: The number of data bytes to follow (2 registers x 2 bytes each = 4 bytes)
41BD 0655: 4 bytes value

All holding registers with float value are sent in big-endian.

In the example, the above value of 23.628 is sent.

Request

This command is requesting the content of holding registers 18100.

**03 46B4 0020**

03: Function Code 3 (read Sensor 1 description Holding Registers)
46B4: The Data Address of the first register requested (46B4 hex = 18100)
0020: The total number of registers requested (read 32 registers each 2 byte = 64 bytes)

Response

**03 40 5365 6E73 6F72 3100 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000**

03: Function Code 3 (read Holding Registers)
40: The number of data bytes to follow (32 registers x 2 bytes each = 64 bytes)
5365 6E73 6F72 3100 0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000: 64 bytes value

All holding registers with strings are sent in big-endian.

The answer is padded with 0.

In the example above string “Sensor1” is sent.

#### 8.3.1.2. Exception codes

All exceptions are signaled by adding 0x80 to the function code of the request, and following this byte by a single reason byte for example as follows:

**01 Illegal function**

The function code received in the query is not an allowable action for the controller.

**02 Illegal data address**

The data address received in the query is not an allowable address for the slave. More specifically, the combination of the reference number and transfer length is invalid. For a controller with 100 registers, a request with offset 96 and length 4 would succeed, a request with offset 96 and length 5 will generate exception 02.

### 8.3.2. Address table

| Parameter | FC | PDU decimal address | Data size | Data |
|---|---|---|---|---|
| Number of installed sensors | 03 | 18000 | 16-bit Integer | |
| | | | | |
| Sensor 1 description | 03,16 | 18100 | 64 bytes UTF-8 | |
| Sensor 2 description | 03,16 | 18132 | 64 bytes UTF-8 | |
| Sensor 3 description | 03,16 | 18164 | 64 bytes UTF-8 | |
| Sensor 4 description | 03,16 | 18196 | 64 bytes UTF-8 | |
| Sensor 5 description | 03,16 | 18228 | 64 bytes UTF-8 | |
| Sensor 6 description | 03,16 | 18260 | 64 bytes UTF-8 | |
| Sensor 7 description | 03,16 | 18292 | 64 bytes UTF-8 | |
| Sensor 8 description | 03,16 | 18324 | 64 bytes UTF-8 | |
| | | | | |
| Sensor 1, S11 dimension | 03 | 18400 | 64 bytes UTF-8 | |
| Sensor 1, S12 dimension | 03 | 18432 | 64 bytes UTF-8 | |
| Sensor 1, S13 dimension | 03 | 18464 | 64 bytes UTF-8 | |
| Sensor 2, S21 dimension | 03 | 18496 | 64 bytes UTF-8 | |
| Sensor 2, S22 dimension | 03 | 18528 | 64 bytes UTF-8 | |
| Sensor 2, S23 dimension | 03 | 18560 | 64 bytes UTF-8 | |
| Sensor 3, S31 dimension | 03 | 18592 | 64 bytes UTF-8 | |
| Sensor 3, S32 dimension | 03 | 18624 | 64 bytes UTF-8 | |
| Sensor 3, S33 dimension | 03 | 18656 | 64 bytes UTF-8 | |
| Sensor 4, S41 dimension | 03 | 18688 | 64 bytes UTF-8 | |
| Sensor 4, S42 dimension | 03 | 18720 | 64 bytes UTF-8 | |
| Sensor 4, S43 dimension | 03 | 18752 | 64 bytes UTF-8 | |
| Sensor 5, S51 dimension | 03 | 18784 | 64 bytes UTF-8 | |
| Sensor 5, S52 dimension | 03 | 18816 | 64 bytes UTF-8 | |
| Sensor 5, S53 dimension | 03 | 18848 | 64 bytes UTF-8 | |
| Sensor 6, S61 dimension | 03 | 18880 | 64 bytes UTF-8 | |
| Sensor 6, S62 dimension | 03 | 18912 | 64 bytes UTF-8 | |
| Sensor 6, S63 dimension | 03 | 18944 | 64 bytes UTF-8 | |

| | | | | |
|---|---|---|---|---|
| Sensor 7, S71 dimension | 03 | 18976 | 64 bytes UTF-8 | |
| Sensor 7, S72 dimension | 03 | 19008 | 64 bytes UTF-8 | |
| Sensor 7, S73 dimension | 03 | 19040 | 64 bytes UTF-8 | |
| Sensor 8, S81 dimension | 03 | 19072 | 64 bytes UTF-8 | |
| Sensor 8, S82 dimension | 03 | 19104 | 64 bytes UTF-8 | |
| Sensor 8, S83 dimension | 03 | 19136 | 64 bytes UTF-8 | |
| Sensor 1, S11 max | 03,16 | 19200 | 32-bit Float | |
| Sensor 1, S12 max | 03,16 | 19202 | 32-bit Float | |
| Sensor 1, S13 max | 03,16 | 19204 | 32-bit Float | |
| Sensor 2, S21 max | 03,16 | 19206 | 32-bit Float | |
| Sensor 2, S22 max | 03,16 | 19208 | 32-bit Float | |
| Sensor 2, S23 max | 03,16 | 19210 | 32-bit Float | |
| Sensor 3, S31 max | 03,16 | 19212 | 32-bit Float | |
| Sensor 3, S32 max | 03,16 | 19214 | 32-bit Float | |
| Sensor 3, S33 max | 03,16 | 19216 | 32-bit Float | |
| Sensor 4, S41 max | 03,16 | 19218 | 32-bit Float | |
| Sensor 4, S42 max | 03,16 | 19220 | 32-bit Float | |
| Sensor 4, S43 max | 03,16 | 19222 | 32-bit Float | |
| Sensor 5, S51 max | 03,16 | 19224 | 32-bit Float | |
| Sensor 5, S52 max | 03,16 | 19226 | 32-bit Float | |
| Sensor 5, S53 max | 03,16 | 19228 | 32-bit Float | |
| Sensor 6, S61 max | 03,16 | 19230 | 32-bit Float | |
| Sensor 6, S62 max | 03,16 | 19232 | 32-bit Float | |
| Sensor 6, S63 max | 03,16 | 19234 | 32-bit Float | |
| Sensor 7, S71 max | 03,16 | 19236 | 32-bit Float | |
| Sensor 7, S72 max | 03,16 | 19238 | 32-bit Float | |
| Sensor 7, S73 max | 03,16 | 19240 | 32-bit Float | |
| Sensor 8, S81 max | 03,16 | 19242 | 32-bit Float | |
| Sensor 8, S82 max | 03,16 | 19244 | 32-bit Float | |
| Sensor 8, S83 max | 03,16 | 19246 | 32-bit Float | |
| | | | | |
| Sensor 1, S11 min | 03,16 | 19300 | 32-bit Float | |
| Sensor 1, S12 min | 03,16 | 19302 | 32-bit Float | |
| Sensor 1, S13 min | 03,16 | 19304 | 32-bit Float | |
| Sensor 2, S21 min | 03,16 | 19306 | 32-bit Float | |
| Sensor 2, S22 min | 03,16 | 19308 | 32-bit Float | |
| Sensor 2, S23 min | 03,16 | 19310 | 32-bit Float | |
| Sensor 3, S31 min | 03,16 | 19312 | 32-bit Float | |
| Sensor 3, S32 min | 03,16 | 19314 | 32-bit Float | |
| Sensor 3, S33 min | 03,16 | 19316 | 32-bit Float | |
| Sensor 4, S41 min | 03,16 | 19318 | 32-bit Float | |
| Sensor 4, S42 min | 03,16 | 19320 | 32-bit Float | |
| Sensor 4, S43 min | 03,16 | 19322 | 32-bit Float | |
| Sensor 5, S51 min | 03,16 | 19324 | 32-bit Float | |
| Sensor 5, S52 min | 03,16 | 19326 | 32-bit Float | |
| Sensor 5, S53 min | 03,16 | 19328 | 32-bit Float | |
| Sensor 6, S61 min | 03,16 | 19330 | 32-bit Float | |
| Sensor 6, S62 min | 03,16 | 19332 | 32-bit Float | |

| | | | | |
|---|---|---|---|---|
| Sensor 6, S63 min | 03,16 | 19334 | 32-bit Float | |
| Sensor 7, S71 min | 03,16 | 19336 | 32-bit Float | |
| Sensor 7, S72 min | 03,16 | 19338 | 32-bit Float | |
| Sensor 7, S73 min | 03,16 | 19340 | 32-bit Float | |
| Sensor 8, S81 min | 03,16 | 19342 | 32-bit Float | |
| Sensor 8, S82 min | 03,16 | 19344 | 32-bit Float | |
| Sensor 8, S83 min | 03,16 | 19346 | 32-bit Float | |
| | | | | |
| Sensor 1, S11 hysteresis | 03,16 | 19400 | 32-bit Float | |
| Sensor 1, S12 hysteresis | 03,16 | 19402 | 32-bit Float | |
| Sensor 1, S13 hysteresis | 03,16 | 19404 | 32-bit Float | |
| Sensor 2, S21 hysteresis | 03,16 | 19406 | 32-bit Float | |
| Sensor 2, S22 hysteresis | 03,16 | 19408 | 32-bit Float | |
| Sensor 2, S23 hysteresis | 03,16 | 19410 | 32-bit Float | |
| Sensor 3, S31 hysteresis | 03,16 | 19412 | 32-bit Float | |
| Sensor 3, S32 hysteresis | 03,16 | 19414 | 32-bit Float | |
| Sensor 3, S33 hysteresis | 03,16 | 19416 | 32-bit Float | |
| Sensor 4, S41 hysteresis | 03,16 | 19418 | 32-bit Float | |
| Sensor 4, S42 hysteresis | 03,16 | 19420 | 32-bit Float | |
| Sensor 4, S43 hysteresis | 03,16 | 19422 | 32-bit Float | |
| Sensor 5, S51 hysteresis | 03,16 | 19424 | 32-bit Float | |
| Sensor 5, S52 hysteresis | 03,16 | 19426 | 32-bit Float | |
| Sensor 5, S53 hysteresis | 03,16 | 19428 | 32-bit Float | |
| Sensor 6, S61 hysteresis | 03,16 | 19430 | 32-bit Float | |
| Sensor 6, S62 hysteresis | 03,16 | 19432 | 32-bit Float | |
| Sensor 6, S63 hysteresis | 03,16 | 19434 | 32-bit Float | |
| Sensor 7, S71 hysteresis | 03,16 | 19436 | 32-bit Float | |
| Sensor 7, S72 hysteresis | 03,16 | 19438 | 32-bit Float | |
| Sensor 7, S73 hysteresis | 03,16 | 19440 | 32-bit Float | |
| Sensor 8, S81 hysteresis | 03,16 | 19442 | 32-bit Float | |
| Sensor 8, S82 hysteresis | 03,16 | 19444 | 32-bit Float | |
| Sensor 8, S83 hysteresis | 03,16 | 19446 | 32-bit Float | |
| | | | | |
| Sensor 1, S11 multiplier | 03,16 | 19500 | 32-bit Float | |
| Sensor 1, S12 multiplier | 03,16 | 19502 | 32-bit Float | |
| Sensor 1, S13 multiplier | 03,16 | 19504 | 32-bit Float | |
| Sensor 2, S21 multiplier | 03,16 | 19506 | 32-bit Float | |
| Sensor 2, S22 multiplier | 03,16 | 19508 | 32-bit Float | |
| Sensor 2, S23 multiplier | 03,16 | 19510 | 32-bit Float | |
| Sensor 3, S31 multiplier | 03,16 | 19512 | 32-bit Float | |
| Sensor 3, S32 multiplier | 03,16 | 19514 | 32-bit Float | |
| Sensor 3, S33 multiplier | 03,16 | 19516 | 32-bit Float | |
| Sensor 4, S41 multiplier | 03,16 | 19518 | 32-bit Float | |
| Sensor 4, S42 multiplier | 03,16 | 19520 | 32-bit Float | |
| Sensor 4, S43 multiplier | 03,16 | 19522 | 32-bit Float | |
| Sensor 5, S51 multiplier | 03,16 | 19524 | 32-bit Float | |
| Sensor 5, S52 multiplier | 03,16 | 19526 | 32-bit Float | |
| Sensor 5, S53 multiplier | 03,16 | 19528 | 32-bit Float | |

| | | | | |
|---|---|---|---|---|
| Sensor 6, S61 multiplier | 03,16 | 19530 | 32-bit Float | |
| Sensor 6, S62 multiplier | 03,16 | 19532 | 32-bit Float | |
| Sensor 6, S63 multiplier | 03,16 | 19534 | 32-bit Float | |
| Sensor 7, S71 multiplier | 03,16 | 19536 | 32-bit Float | |
| Sensor 7, S72 multiplier | 03,16 | 19538 | 32-bit Float | |
| Sensor 7, S73 multiplier | 03,16 | 19540 | 32-bit Float | |
| Sensor 8, S81 multiplier | 03,16 | 19542 | 32-bit Float | |
| Sensor 8, S82 multiplier | 03,16 | 19544 | 32-bit Float | |
| Sensor 8, S83 multiplier | 03,16 | 19546 | 32-bit Float | |
| | | | | |
| Sensor 1, S11 offset | 03,16 | 19600 | 32-bit Float | |
| Sensor 1, S12 offset | 03,16 | 19602 | 32-bit Float | |
| Sensor 1, S13 offset | 03,16 | 19604 | 32-bit Float | |
| Sensor 2, S21 offset | 03,16 | 19606 | 32-bit Float | |
| Sensor 2, S22 offset | 03,16 | 19608 | 32-bit Float | |
| Sensor 2, S23 offset | 03,16 | 19610 | 32-bit Float | |
| Sensor 3, S31 offset | 03,16 | 19612 | 32-bit Float | |
| Sensor 3, S32 offset | 03,16 | 19614 | 32-bit Float | |
| Sensor 3, S33 offset | 03,16 | 19616 | 32-bit Float | |
| Sensor 4, S41 offset | 03,16 | 19618 | 32-bit Float | |
| Sensor 4, S42 offset | 03,16 | 19620 | 32-bit Float | |
| Sensor 4, S43 offset | 03,16 | 19622 | 32-bit Float | |
| Sensor 5, S51 offset | 03,16 | 19624 | 32-bit Float | |
| Sensor 5, S52 offset | 03,16 | 19626 | 32-bit Float | |
| Sensor 5, S53 offset | 03,16 | 19628 | 32-bit Float | |
| Sensor 6, S61 offset | 03,16 | 19630 | 32-bit Float | |
| Sensor 6, S62 offset | 03,16 | 19632 | 32-bit Float | |
| Sensor 6, S63 offset | 03,16 | 19634 | 32-bit Float | |
| Sensor 7, S71 offset | 03,16 | 19636 | 32-bit Float | |
| Sensor 7, S72 offset | 03,16 | 19638 | 32-bit Float | |
| Sensor 7, S73 offset | 03,16 | 19640 | 32-bit Float | |
| Sensor 8, S81 offset | 03,16 | 19642 | 32-bit Float | |
| Sensor 8, S82 offset | 03,16 | 19644 | 32-bit Float | |
| Sensor 8, S83 offset | 03,16 | 19646 | 32-bit Float | |
| | | | | |
| Sensor 1 ID | 03 | 19700 | 16 bytes UTF-8 | Example: 2860B85F07000094 |
| Sensor 2 ID | 03 | 19708 | 16 bytes UTF-8 | |
| Sensor 3 ID | 03 | 19716 | 16 bytes UTF-8 | |
| Sensor 4 ID | 03 | 19724 | 16 bytes UTF-8 | |
| Sensor 5 ID | 03 | 19732 | 16 bytes UTF-8 | |
| Sensor 6 ID | 03 | 19740 | 16 bytes UTF-8 | |
| Sensor 7 ID | 03 | 19748 | 16 bytes UTF-8 | |
| Sensor 8 ID | 03 | 19756 | 16 bytes UTF-8 | |
| | | | | |
| Sensor 1, S11 value | 03 | 19800 | 32-bit Float | |
| Sensor 1, S12 value | 03 | 19802 | 32-bit Float | |
| Sensor 1, S13 value | 03 | 19804 | 32-bit Float | |
| Sensor 2, S21 value | 03 | 19806 | 32-bit Float | |

| | | | | |
|---|---|---|---|---|
| Sensor 2, S22 value | 03 | 19808 | 32-bit Float | |
| Sensor 2, S23 value | 03 | 19810 | 32-bit Float | |
| Sensor 3, S31 value | 03 | 19812 | 32-bit Float | |
| Sensor 3, S32 value | 03 | 19814 | 32-bit Float | |
| Sensor 3, S33 value | 03 | 19816 | 32-bit Float | |
| Sensor 4, S41 value | 03 | 19818 | 32-bit Float | |
| Sensor 4, S42 value | 03 | 19820 | 32-bit Float | |
| Sensor 4, S43 value | 03 | 19822 | 32-bit Float | |
| Sensor 5, S51 value | 03 | 19824 | 32-bit Float | |
| Sensor 5, S52 value | 03 | 19826 | 32-bit Float | |
| Sensor 5, S53 value | 03 | 19828 | 32-bit Float | |
| Sensor 6, S61 value | 03 | 19830 | 32-bit Float | |
| Sensor 6, S62 value | 03 | 19832 | 32-bit Float | |
| Sensor 6, S63 value | 03 | 19834 | 32-bit Float | |
| Sensor 7, S71 value | 03 | 19836 | 32-bit Float | |
| Sensor 7, S72 value | 03 | 19838 | 32-bit Float | |
| Sensor 7, S73 value | 03 | 19840 | 32-bit Float | |
| Sensor 8, S81 value | 03 | 19842 | 32-bit Float | |
| Sensor 8, S82 value | 03 | 19844 | 32-bit Float | |
| Sensor 8, S83 value | 03 | 19846 | 32-bit Float | |
| | | | | |
| Sensor 1, S11 row value | 03 | 19900 | 32-bit Float | |
| Sensor 1, S12 row value | 03 | 19902 | 32-bit Float | |
| Sensor 1, S13 row value | 03 | 19904 | 32-bit Float | |
| Sensor 2, S21 row value | 03 | 19906 | 32-bit Float | |
| Sensor 2, S22 row value | 03 | 19908 | 32-bit Float | |
| Sensor 2, S23 row value | 03 | 19910 | 32-bit Float | |
| Sensor 3, S31 row value | 03 | 19912 | 32-bit Float | |
| Sensor 3, S32 row value | 03 | 19914 | 32-bit Float | |
| Sensor 3, S33 row value | 03 | 19916 | 32-bit Float | |
| Sensor 4, S41 row value | 03 | 19918 | 32-bit Float | |
| Sensor 4, S42 row value | 03 | 19920 | 32-bit Float | |
| Sensor 4, S43 row value | 03 | 19922 | 32-bit Float | |
| Sensor 5, S51 row value | 03 | 19924 | 32-bit Float | |
| Sensor 5, S52 row value | 03 | 19926 | 32-bit Float | |
| Sensor 5, S53 row value | 03 | 19928 | 32-bit Float | |
| Sensor 6, S61 row value | 03 | 19930 | 32-bit Float | |
| Sensor 6, S62 row value | 03 | 19932 | 32-bit Float | |
| Sensor 6, S63 row value | 03 | 19934 | 32-bit Float | |
| Sensor 7, S71 row value | 03 | 19936 | 32-bit Float | |
| Sensor 7, S72 row value | 03 | 19938 | 32-bit Float | |
| Sensor 7, S73 row value | 03 | 19940 | 32-bit Float | |
| Sensor 8, S81 row value | 03 | 19942 | 32-bit Float | |
| Sensor 8, S82 row value | 03 | 19944 | 32-bit Float | |
| Sensor 8, S83 row value | 03 | 19946 | 32-bit Float | |
| | | | | |
| Sensor 1, S11 alarm status | 03 | 20000 | 16-bit unsign int | normal (0), alarm (1) |
| Sensor 1, S12 alarm status | 03 | 20001 | 16-bit unsign int | normal (0), alarm (1) |

| | | | | |
|---|---|---|---|---|
| Sensor 1, S13 alarm status | 03 | 20002 | 16-bit unsign int | normal (0), alarm (1) |
| Sensor 2, S21 alarm status | 03 | 20003 | 16-bit unsign int | normal (0), alarm (1) |
| Sensor 2, S22 alarm status | 03 | 20004 | 16-bit unsign int | normal (0), alarm (1) |
| Sensor 2, S23 alarm status | 03 | 20005 | 16-bit unsign int | normal (0), alarm (1) |
| Sensor 3, S31 alarm status | 03 | 20006 | 16-bit unsign int | normal (0), alarm (1) |
| Sensor 3, S32 alarm status | 03 | 20007 | 16-bit unsign int | normal (0), alarm (1) |
| Sensor 3, S33 alarm status | 03 | 20008 | 16-bit unsign int | normal (0), alarm (1) |
| Sensor 4, S41 alarm status | 03 | 20009 | 16-bit unsign int | normal (0), alarm (1) |
| Sensor 4, S42 alarm status | 03 | 20010 | 16-bit unsign int | normal (0), alarm (1) |
| Sensor 4, S43 alarm status | 03 | 20011 | 16-bit unsign int | normal (0), alarm (1) |
| Sensor 5, S51 alarm status | 03 | 20012 | 16-bit unsign int | normal (0), alarm (1) |
| Sensor 5, S52 alarm status | 03 | 20013 | 16-bit unsign int | normal (0), alarm (1) |
| Sensor 5, S53 alarm status | 03 | 20014 | 16-bit unsign int | normal (0), alarm (1) |
| Sensor 6, S61 alarm status | 03 | 20015 | 16-bit unsign int | normal (0), alarm (1) |
| Sensor 6, S62 alarm status | 03 | 20016 | 16-bit unsign int | normal (0), alarm (1) |
| Sensor 6, S63 alarm status | 03 | 20017 | 16-bit unsign int | normal (0), alarm (1) |
| Sensor 7, S71 alarm status | 03 | 20018 | 16-bit unsign int | normal (0), alarm (1) |
| Sensor 7, S72 alarm status | 03 | 20019 | 16-bit unsign int | normal (0), alarm (1) |
| Sensor 7, S73 alarm status | 03 | 20020 | 16-bit unsign int | normal (0), alarm (1) |
| Sensor 8, S81 alarm status | 03 | 20021 | 16-bit unsign int | normal (0), alarm (1) |
| Sensor 8, S82 alarm status | 03 | 20022 | 16-bit unsign int | normal (0), alarm (1) |
| Sensor 8, S83 alarm status | 03 | 20023 | 16-bit unsign int | normal (0), alarm (1) |
| | | | | |
| Save configuration | 03,06 | 50000 | 16-bit unsign int | unsaved (0), saved (1) |
| Restart device | 03,06 | 50001 | 16-bit unsign int | cancel (0), restart (1) |
| Temperature unit | 03,06 | 50002 | 16-bit unsign int | Celsius (0), Fahrenheit (1) |
| HW error | 03,06 | 50003 | 16-bit unsign int | noErr (0), hwErr (1) |
| Device ID | 03 | 50100 | 18 bytes UTF-8 | Example: 5c:32:c5:00:ac:52 |
| Hostname | 03 | 50200 | 16 bytes UTF-8 | |
| Device IP | 03 | 50300 | 16 bytes UTF-8 | Example: 192.168.1.2 |

## 8.4. Modbus RTU

### 8.4.1. Communication parameters

For Modbus RTU, TCW210-TH supports the following communication parameters:

- Baud rate – 2400, 4800, 9600, 19200, 38400, or 57600;
- Data bits – 8;
- Stop bits – 1 or 2;
- Parity –None, Odd, or Even;

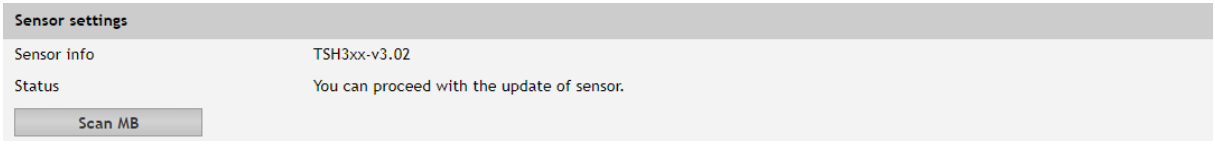
The factory default communication parameters for the device are the standard ones for Modbus RTU:

- Baud rate – 19200;
- Data bits – 8;
- Stop bits – 1;
- Parity – Even;

### 8.4.2. Teracom sensors update tool

The TCW210-TH supports the Teracom sensor firmware update tool, which is available at http://device.ip.address/teracom485.htm.
**Attention!** To make any changes to a Modbus RTU sensor, it should be the only device on the RS-485 bus.

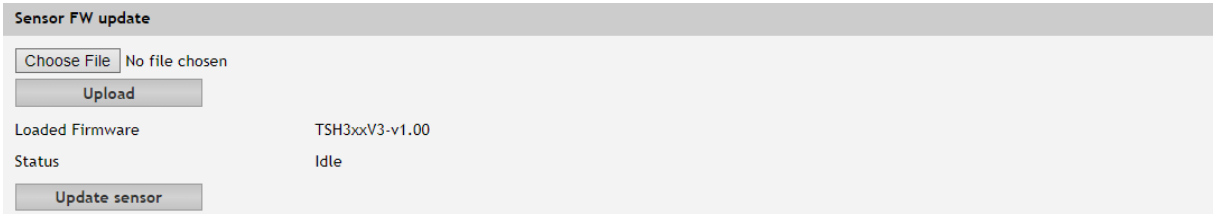
#### 8.4.2.1. Sensor settings




The tool operates using the current Modbus RTU communication parameters. To avoid communication collisions, it is advisable to configure both the TCW210-TH and the sensor with the factory default Modbus RTU communication parameters. This practice ensures smooth operation. The default Modbus RTU communication parameters for the TCW210-TH are detailed in Section 8.4.1.

Before making any changes, it is strongly recommended to scan for the sensor settings. This scan will provide information about the current firmware version of the sensor and verify whether the sensor is the only device on the bus.

#### 8.4.2.2. Sensor FW update




To initiate the firmware update, the appropriate file must first be uploaded to the sensor. After the upload is complete, click the “Update” button to proceed.

## 8.5. Logger

The logger uses a circular buffer in the device's FLASH memory. When the buffer becomes full, new data automatically overwrites the oldest entries, ensuring that the FLASH memory always holds the most recent complete log. There is no command to clear the log, and a copy of the full log is always available for download.

The number of log entries depends on the length of descriptions and the type of characters used. In the worst-case scenario (with 15-byte descriptions using characters from the upper range of UTF-8), the logger can store approximately 52,000 records, which is sufficient for 36 days with records logged every minute.

New log data can also be periodically uploaded as a file to a dedicated HTTP server at time intervals of 1, 2, 3, 4, 6, 8, 12, or 24 hours. The data is uploaded in CSV format, using a semicolon (;) as the delimiter.

Each log file begins with a header row, and all rows (including the header) start with a record ID and a timestamp.

The structure of each log entry (record) is as follows:

| ID | Time | Type of record | Inputs value | Relays | Alarm conditions |
|---|---|---|---|---|---|

| | |
|---|---|
| ID | A unique 32-bit number assigned to each row (record). |
| Time | The timestamp of the record in the format yyyy.mm.dd, hh:mm. |
| Type of record | Specifies the reason for the log entry. The available record types are: |
| | "Time" For periodical record; |
| | "Event" For records triggered by an alarm condition; |
| | "Type" For header records; |
| | "Start" Created after a power-up event; |
| | "Restart" Created after a reset event; |
| | "Power Down" Generated after a power-down event; |
| | "Bad" For problematic records. |
| Inputs value | Lists the values of all monitored inputs in the following order: sensors, analog inputs, and digital inputs. |
| Alarm conditions | Indicates the alarm status for each input. A value of "1" means an active alarm condition. |

Example of the log file:

```
1131901;15.10.2015,01:02:23;Type;S11/°C;S12;S21/°C;S22;S31/°C;S32;S41/°C;S42;S51/°C;S52;S61/°C;S62;S71/°C;S72;S81/°C;S82;A1/V;A2/V;D1;D2;R1;R2;S1
1/°C;S12;S21/°C;S22;S31/°C;S32;S41/°C;S42;S51/°C;S52;S61/°C;S62;S71/°C;S72;S81/°C;S82;A1/V;A2/V;D1;D2;
1131902;15.10.2015,01:02:23;Time;18.250;;18.375;;18.125;;18.500;;18.188;;18.125;;18.375;;18.375;;11.352;0.065;1;0;1;0;1;;1;;1;;1;;1;;1;;1;;1;0;1;0;
1131903;15.10.2015,01:02:23;Event;18.250;;18.438;;18.125;;18.500;;18.188;;18.125;;18.313;;18.375;;11.352;0.066;0;1;0;1;1;;1;;1;;1;;1;;1;;1;;1;0;0;1;
1131904;15.10.2015,01:02:24;Time;18.250;;18.438;;18.125;;18.500;;18.188;;18.125;;18.313;;18.375;;11.352;0.066;0;1;0;1;1;;1;;1;;1;;1;;1;;1;;1;0;0;1;
1131905;15.10.2015,01:02:25;Time;18.250;;18.375;;18.125;;18.500;;18.188;;18.125;;18.313;;18.375;;11.352;0.066;0;1;0;1;1;;1;;1;;1;;1;;1;;1;;1;0;0;1;
1131906;15.10.2015,01:02:26;Time;18.250;;18.375;;18.125;;18.500;;18.188;;18.125;;18.313;;18.313;;11.352;0.066;0;1;0;1;1;;1;;1;;1;;1;;1;;1;;1;0;0;1;
1131907;15.10.2015,01:02:27;Time;18.250;;18.375;;18.125;;18.438;;18.188;;18.125;;18.313;;18.313;;11.352;0.066;0;1;0;1;1;;1;;1;;1;;1;;1;;1;;1;0;0;1;
1131908;15.10.2015,01:02:27;Event;18.250;;18.375;;18.125;;18.438;;18.188;;18.125;;18.313;;18.313;;2.198;9.092;0;1;0;1;1;;1;;1;;1;;1;;1;;1;;0;0;0;1;
```

# 9. Factory default settings

TCW210-TH can be restored to its original factory default settings in three different ways.

## 9.1. Factory default from WEB interface

Pressing the “Factory default” button in the Administration -> Backup/Restore section will reset all parameters to their factory defaults, except for network settings.

## 9.2. Factory default with the reset button

If the reset button is pressed for more than 5 seconds while the device is powered on, all network settings will be reverted to factory defaults.

## 9.3. General factory default with the reset button

To perform a complete factory reset of all parameters, follow these steps:

- Press and hold the RESET button, then turn on the power supply.
- The yellow LED will illuminate, and the red LED will blink approximately five times per second.
- After about 5 seconds, the red LED will turn off, and you can release the button.
- The yellow LED will flash once per second while the red LED remains illuminated, indicating that the device is in working mode with factory default settings.



The factory default settings are:

| | |
|---|---|
| Username | admin |
| Password | admin |
| IP Address | 192.168.1.2 |
| Subnet Mask | 255.255.255.0 |
| Default Gateway | 192.168.1.1 |
| SNMPConfiguration | disabled |
| readCommunity | public |
| writeCommunity | private |

## 10. Environment information

This equipment is intended for use in a Pollution Degree 2 environment, at altitudes of up to 2000 meters.

When the controller is part of a larger system, all other elements of the system must comply with EMC (Electromagnetic Compatibility) requirements and be suitable for use under the same ambient conditions.

## 11. Safety

This device must not be used for medical or life-saving purposes or for any application where its failure could result in serious injury or loss of life.

To reduce the risk of fire, use only flexible stranded wire with a cross-section of 0.5mm² or larger for wiring digital and analog inputs and relay outputs of the device.

To avoid electric shock and fire hazards, do not expose this product to liquids, rain, or moisture. Objects filled with liquids, such as vases, should not be placed on this device.

There is a risk of overheating (and potential damage) to the controller if the recommended free spaces next to adjacent devices are not maintained. Ensure that there is sufficient space for attaching and removing cables after installation.

Teracom does not guarantee the successful operation of the product if it is used under conditions that deviate from the product specifications.

To ensure that the device works correctly follow these steps:

- Ensure that the device is installed properly by referring to this user manual;
- Log in to the device using a web browser;
- Complete the necessary setup;
- Install TSH1XX or TST1XX sensor on the 1-Wire bus;
- Install TSH3XX or TST3XX sensor on the RS-485 bus;
- Navigate to the “Monitoring page” of the web interface. The correct parameter values should be displayed, and the flashing “STS” LED should indicate proper operation.

If the equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

In no event will Teracom Ltd. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

## 12. Maintenance

After any service or repair of the device, or at least once a year, a safety check must be performed to ensure that the product is in proper operating condition.

Clean the device only with dry cloth. Do not use liquid cleaners or aerosol cleaners. Do not use a magnetic/static cleaning device (dust remover) or any kind of abrasive materials to clean the device.

Following these guidelines will help maintain the device in optimal condition and ensure its longevity.

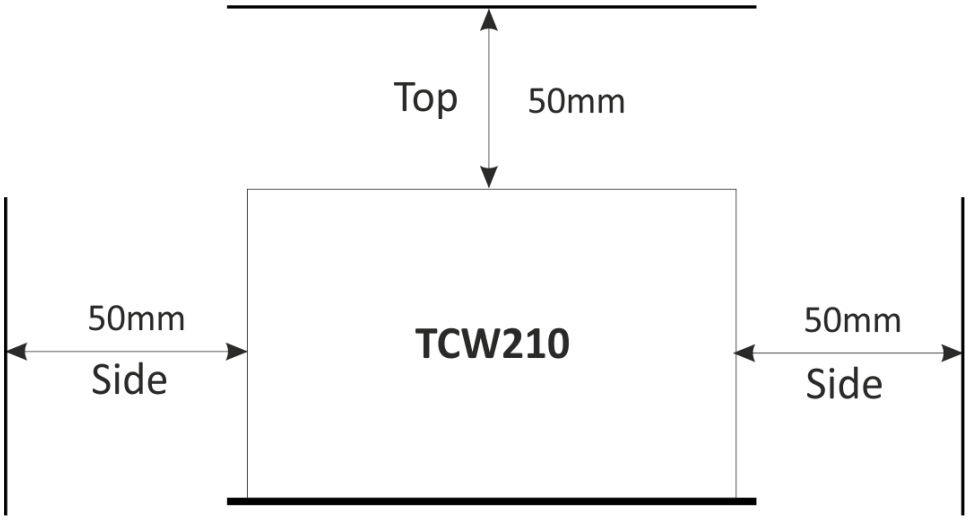



Fig.1