control solutions
TERACOM

TCW210-TH Temperature and humidity data logger

Revision 1.21 / October 2023

USER MANUAL

www.teracomsystems.com

1. Introduction

TCW210-TH is a temperature and humidity data logger with an embedded WEB server. Real-time data and charts of temperature, humidity and dew point can be monitored with a standard web browser (no special software is needed). Standard protocols as SNMP, MODBUS/TCP, and HTTP/API are available for M2M applications. The device supports also interface to popular IoT analytics – ThingSpeak.

The Ethernet temperature logger supports up to eight temperature or humidity-temperature sensors. All they can be connected either to the 1-Wire interface, popular for home automation or to more robust MODBUS RTU over RS-485.

All monitored parameters can be recorded, in internal FLASH memory. The records are made on the previous set time interval and/or on an alarm condition. The memory is large enough for at least 36 days with records on every minute. The log file can be periodically uploaded on a dedicated server by HTTP/HTTPS Post. The stored data can be monitored on 4 graph pages.

XML/JSON file with all monitored parameters can periodically upload to a dedicated server by HTTP/HTTPS Post.

For every monitored parameter e-mails and SNMP traps for up to 5 recipients can be sent. Alarm alert can also be sent by HTTP/HTTPS Post with XML/JSON status files.

2. Features

- Data logger for up to 70000 records;
- 1-Wire and MODBUS RTU sensors support;
- HTTP API commands;
- Periodical HTTP/HTTPS Post of XML/JSON status files for client-server systems;
- SNMP v.2 support;
- SNMP traps to up to 5 recipients like alarm alert;
- MODBUS TCP/IP support;
- SMTP with TLS encryption;
- TLS 1.0, TLS 1.1 and TLS 1.2 support;
- e-mails to up to 5 recipients like alarm alert;
- ThingSpeak service support;
- NTP support;
- Back-up/Restore of configurations;
- Dynamic DNS support;
- 10/100 Mb Ethernet connectivity;
- Auto-MDIX;
- Port changing for HTTP, SNMP and MODBUS TCP/IP;
- Password protected WEB based configuration and control;
- Extended working temperature range;
- Wide power supply voltage range;
- Remote firmware update.

3. Applications

TCW210-TH is suitable for environmental monitoring, building, and industrial automation.

It works very well for monitoring temperature and humidity as a standalone device using a WEB browser only or as a part of small to large industrial control systems for SCADA (supervisory control and data acquisition).

A few application examples - pharmaceutical and food processing and storage, clean rooms, laboratories, HVAC systems, greenhouses and farms, electronic assembly etc.

4. Specifications

- Physical characteristics

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

- Environmental limits

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

- Warranty

  Warranty period: 3 years

- Power requirements

  Input Voltage: 10 to 28 VDC
  Input Current: 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.)

5. LED indicators

The following indicators show the status of the controller:

- **PWR** (red) – in working mode shines, flashes together with STS if there is a hardware error;
- **STS** (yellow) – flashes when the main program of the controller is executed;
- **NET** (orange) – indicates the network connection status - ON when a link is established, flashing when there is an activity.

6. Installation and setup

This device must be installed by qualified personnel.

This device must not be installed directly outdoors.

The installation consists of mounting the device, connecting to an IP network, connecting inputs and outputs, providing power and configuring via a web browser.

6.1. Mounting

TCW210-TH should be mounted in a clean and dry location on a not flammable surface. Ventilation is recommended for installations where the ambient air temperature is expected to be high.

Maintain spacing between adjacent equipment. Allow 50 mm of space on all sides, as shown in Appendix A, this provides ventilation and electrical isolation.

6.2. Connection

Attention! Disconnect power supply before wiring.

The correct wiring procedure is as follows:

- Make sure power is turned off;
- Apply all sensors;
- Apply power.

Make sure that cables are properly attached. Not proper wiring and configuration can cause permanent damage to TCW210-TH or the equipment to which it is connected or both.

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 – Line B-
	Pin3 – 1-Wire Data		Pin5 – Line A+
	Pin4 – 1-Wire GND		Pin6 – not connected
	Pin5 – +VDD		Pin7 – +VDD
	Pin6 – +VDD (most right)		Pin8 – GND

6.2.1. Power supply

TCW210-TH is designed to be supplied by adapter SYS1421-0612-W2E or similar, intended for use in the conditions of overvoltage category II, and prior assessed for compliance with safety

requirements. The power supply equipment shall be resistant to short circuit and overload in a secondary circuit.

When in use, do not position the equipment so that it is difficult to disconnect the device from the power supply.

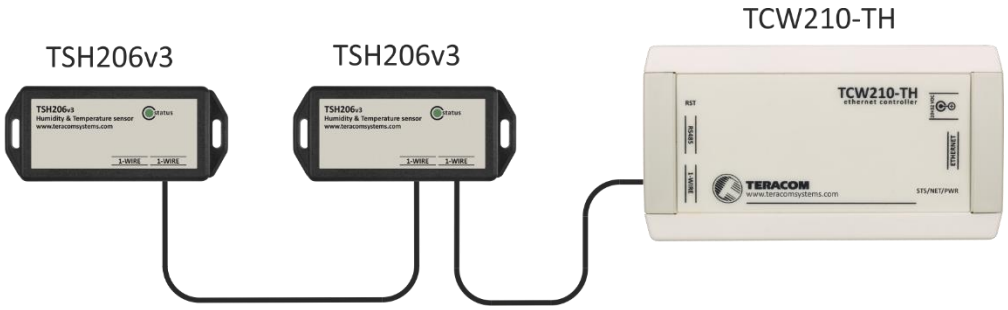
6.2.2. 1-Wire interface

1-Wire is a registered trademark of Maxim Integrated Products, Inc. It is designed to connect several sensors over short wiring. It is not suitable for long distances or environments with EMC interference.

The maximum number of sensors (1-Wire or RS-485) connected to TCW210-TH is 8.

The device supports temperature and humidity-temperature sensors. Connected sensors are automatically detected and the appropriate dimension is assigned.

It is strongly recommended to use “daisy-chained” (linear topology) for multi-sensors systems:



It is strongly recommended to use only UTP/FTP cables and keep total cable length up to 30m. Although functionality has been achieved in the longer distance, we cannot guarantee error-free operation over mentioned wiring length. We recommend reading Maxim’s 1-Wire tips at https://www.teracomsystems.com/wp-content/uploads/1-wire/guidelines-for-reliable-long-line-1-wire-networks.pdf.

We guarantee proper operation only with Teracom 1-Wire sensors.

6.2.3. RS-485 interface

RS-485 is a standard for serial communications systems defined by Telecommunications Industry Association (TIA) and Electronic Industries Alliance (EIA). Implementing the standard, communication systems can be used effectively over long distances and in electrically noisy (industrial) environments.

The maximum number of sensors (1-Wire or RS-485) connected to TCW210-TH is 8.

The device supports temperature and humidity-temperature sensors.

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:

Interconnections are realized by UTP/FTP cables with RJ-45 connectors. The popular Ethernet wiring standard ANSI/TIA/EIA T568B is used:

Pin#	RJ45
1	Orange/White
2	Orange
3	Green/White
4	Blue
5	Blue/White
6	Green
7	Brown/White
8	Brown

It is recommended to use standard patch cables for LAN networks.

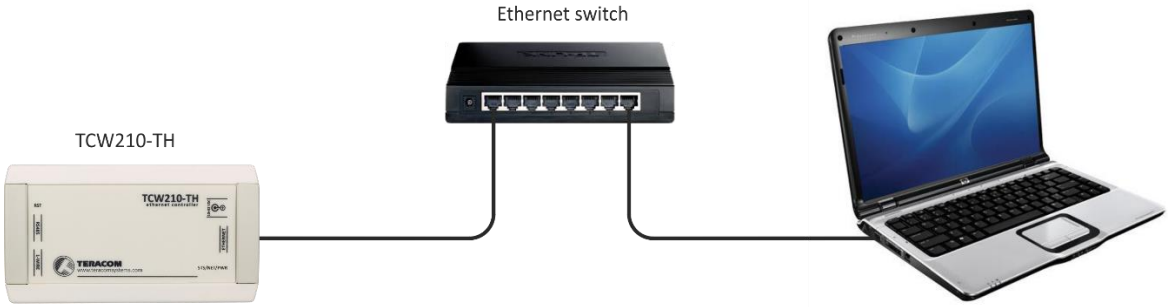
Special attention should be paid on termination of the bus in the last sensor.

We recommend keeping total cable length up to 30 m, although the RS-485 interface works over a much longer distance.

Attention! Special care should be paid on termination of the bus. The last sensor in the chain should have a 120-ohm terminator installed on the free RJ-45 socket. The terminator is delivered with the module.

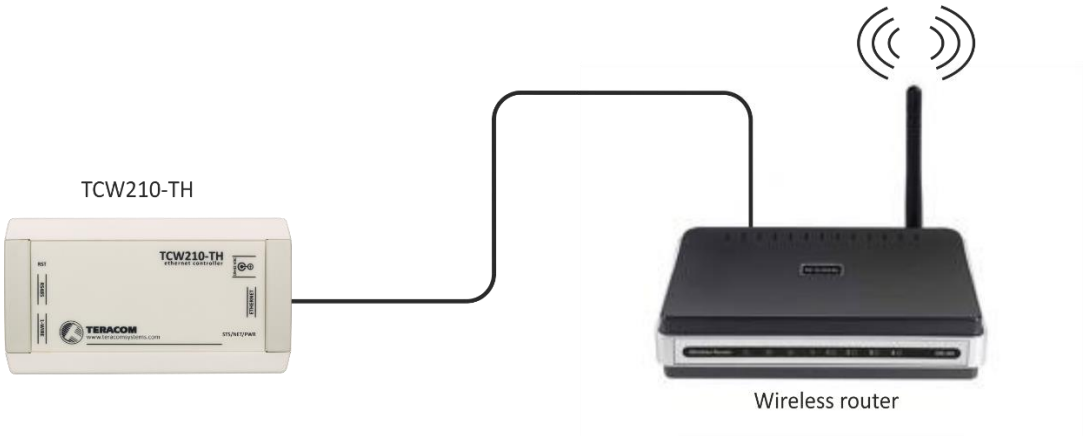
6.2.4. Network connection

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



For setup, TCW210-TH may be connected directly to the Ethernet port on a computer. The device support Auto-MDIX and it is not necessary to use “crossover” cable, standard “straight-through” can be also used.

TCW210-TH can be used in a wireless network by connecting through a wireless router.

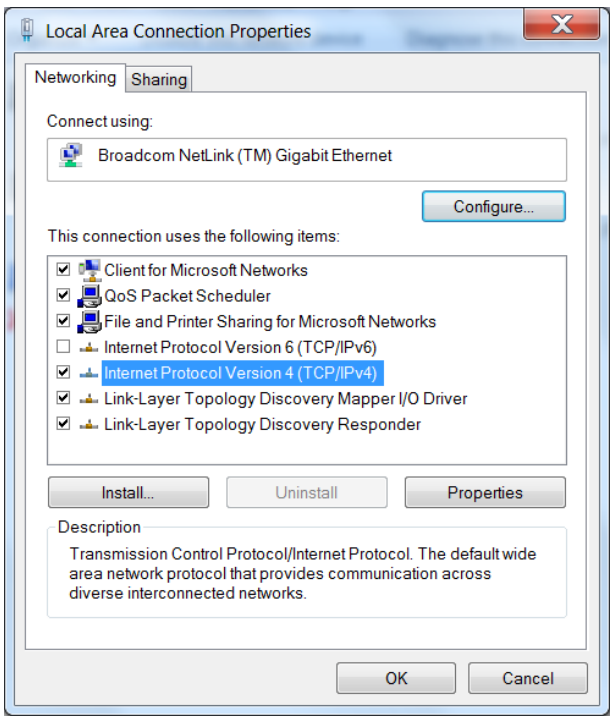


6.3. Communication setup

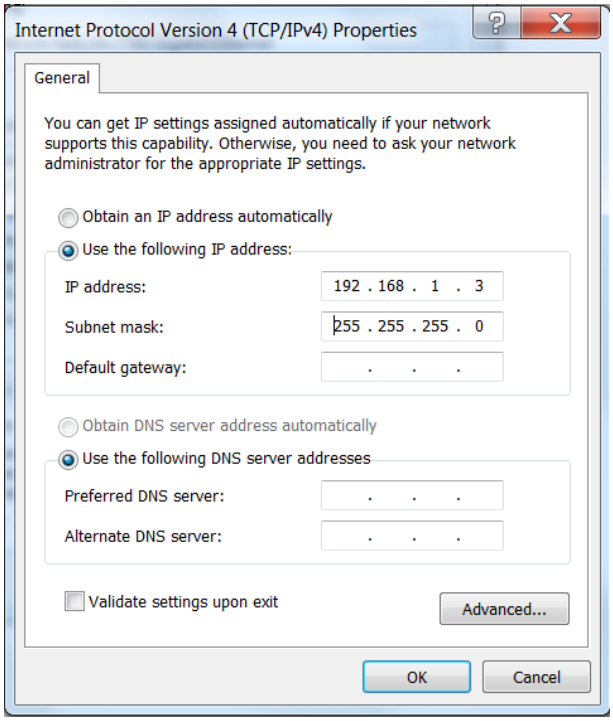
By default TCW210-TH is delivered with the following network settings:

IP address: 192.168.1.2, Subnet Mask: 255.255.255.0, Default Gateway: 192.168.1.1

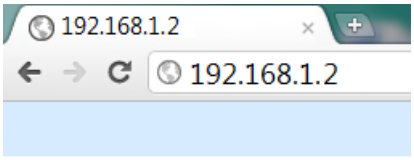
Communication with TCW210-TH can be established by assigning a temporary IP address to the computer. For computers with Windows OS assigning of IP address is made in “Local area connection properties”:



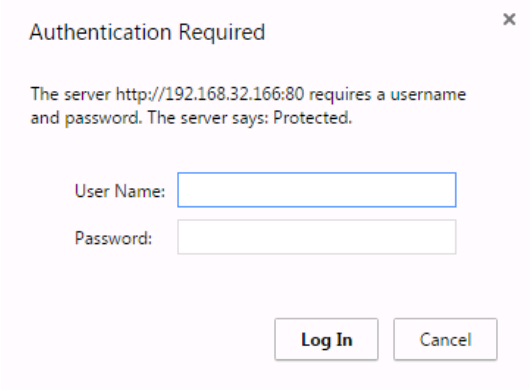
The address should be on the same network - for example 192.168.1.3:



To get access to the web interface, you should type http://192.168.1.2 into the browser:



If the network settings are correct, the login pop-up window will appear:



All TCW controllers connected to LAN can be easily found by the free tool “TCW discoverer”.

It is available for Win and Mac operating systems and can be downloaded from www.teracomsystems.com.

7. Web interface

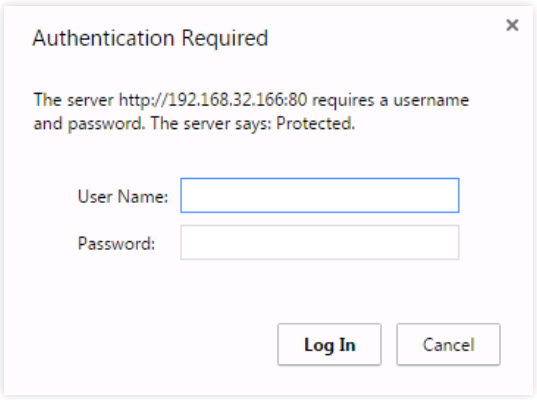
The WEB interface allows configuration, monitoring, and control. All pages are UTF-8 encoded.

For the WEB interface, the device supports HTTP only (HTTPS is not supported).

If the controller is properly addressing, login pop-up window appears.

Authorization data must be entered (by default username=admin, password=admin).

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



The controller supports a few active session.

7.1. Monitoring

Monitoring page displays the current state of TCW210-TH. It has one data and four graphs tabs.

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 TCW210-TH can be monitored on this page. There are two sections on the page - one for 1-Wire sensors and one for MODBUS RTU sensors.

TCW210-TH supports up to eight sensors. They can be connected to both interfaces in a random ratio, settable in section “Sensors ratio setup” on Setup->Sensors page. By default, the number of MODBUS RTU sensors are 4.

All detected 1-Wire sensors are shown in “1-Wire sensors” section. The sensors should be setup in section “1-Wire sensors setup” on Setup->Sensors page.

Teracom 1-Wire temperature sensors readings are shown in the column “Temperature”, while for dual sensors (TSH2xx) both column “Temperature” and “Humidity” are used. For dual sensors, the parameter Dew point is calculated.

All MODBUS RTU sensors are shown in “Modbus sensors” section. The sensors should be added and set up on Setup->Sensors page.

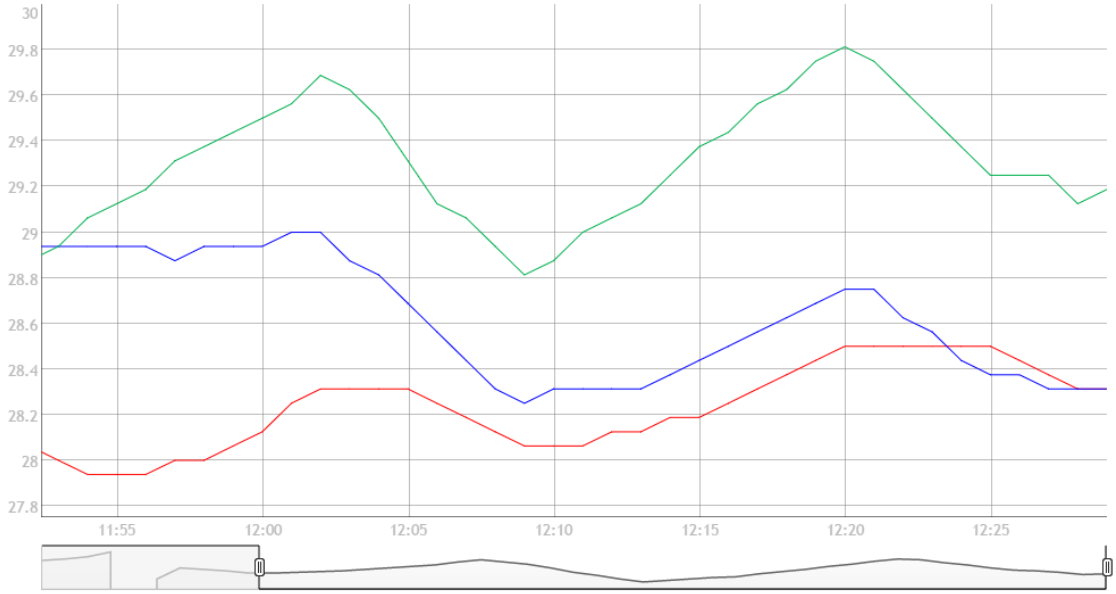
For every sensor, there are a description, value, and ID information. The description length is up to 15 characters. Default descriptions can be changed in Setup->Conditions page.

Dual sensors (humidity-temperature) have the two parameters. For these sensors, Dew point parameter is calculated automatically.

The page can be automatically refreshed on an interval of 0 to 253 seconds. Zero means no automatic refresh. This parameter is set in section Setup->System->Monitoring page automatic refresh. By default, it is 1 second.

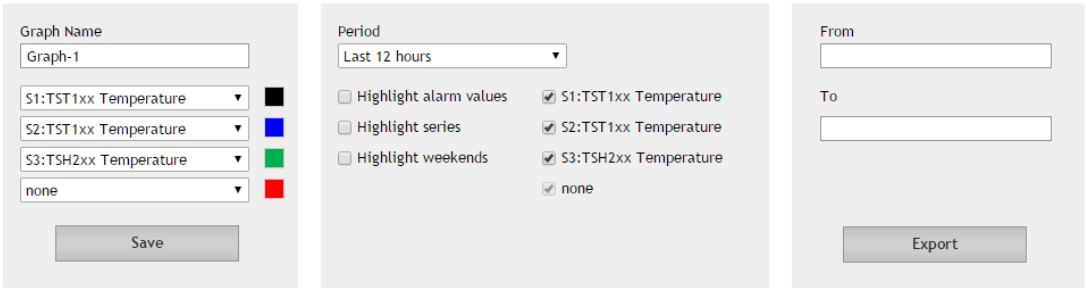
7.1.2. Graphs

Every graph page can display up to 4 parameters with up to 2 different dimension.



For every parameter different color can be set. There are a few checkboxes for display modification.

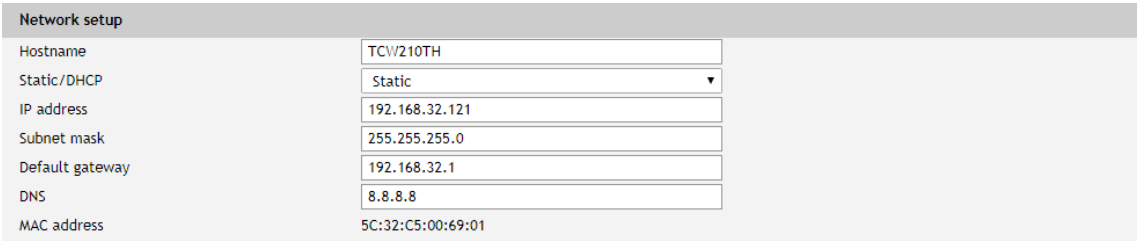
Export of monitored parameters can be made from the page.



7.2. Setup

7.2.1. Network

The network parameters are set on this page.



The controller supports static and dynamic IP addresses.

It is good practice to change the default IP address of the controller immediately after first power-on. This will avoid collisions if many devices are used on the same network.

It may be necessary to clear the arp cache, each time you connect a new device to the network. This is done by typing *arp -d* in the command prompt window of the computer.

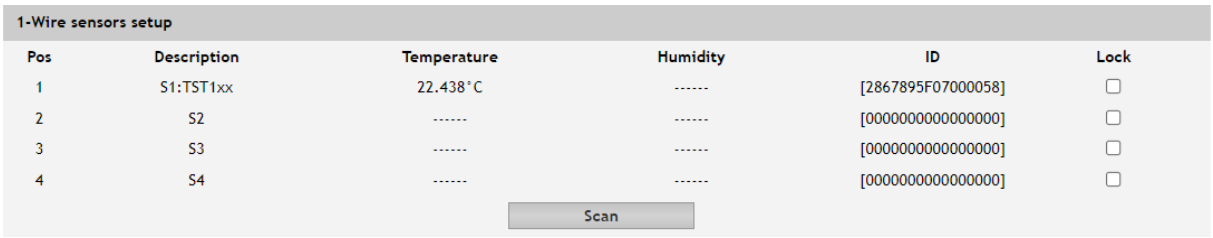
The “Hostname” is up to 15 characters. It is shown in the search results of TCW discoverer.

7.2.2. Sensors

7.2.2.1. Sensors ratio setup

The ratio between 1-Wire and MODBUS RTU sensors can be set here. By default, it is 4:4.

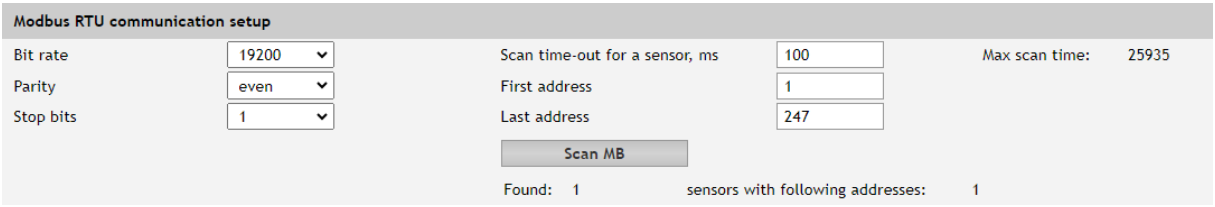
7.2.2.2. 1-Wire sensors setup



Detection is made either after power on or by the button “Scan for new sensors”. All found sensors are shown in ascending order refer their unique ID number.

It is possible to lock a 1-Wire sensor in a specific position. To do this all sensors should be added one by one. After every addition, a new scan should be made and a newly found sensor should be locked in its position. If all sensors are locked, removing one “in the middle” will not change the positions of other sensors after reset. This option is very useful when TCW210-TH is used as a part of a monitoring and control system managed either by SNMP or HTTP API commands.

7.2.2.3. MODBUS RTU communication setup

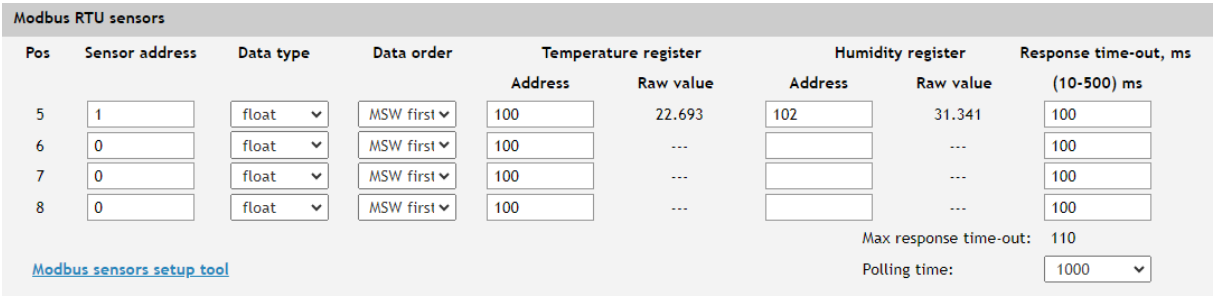


TCW210-TH supports MODBUS RTU over RS-485 interface. All sensors connected to this interface should work with the same communication settings.

By default, TCW210-TH works with the standard for MODBUS RTU settings – 19200, E, 1.

In the right part of the section, there is a tool for scan the MODBUS RTU interface. To optimize scan time, the appropriate address segment should be set up after this, the button “Save” on the bottom of the page should be pressed.

7.2.2.4. MODBUS RTU sensors



TCW210-TH supports Teracom and third-party MODBUS RTU sensors.

Before to add MODBUS RTU sensors, the user should take care of their address setting. It is not allowed to use two sensors with the same address. It is recommended to scan for new sensors before to make any changes.

For every sensor, an appropriate register address together with the data type and data order should be set. All changes should be saved. If the settings are OK in the columns “Raw value” the right data will be shown.

TCW210-TH supports MODBUS RTU sensors with response time-out between 10 and 500mS. The response time-out for a new sensor is 100mS by default but it is recommended to use minimum response time guaranteed by the sensor’s manufacturer.

The sum of chosen response time-out for every sensor forms the maximum response time-out for the system.

The polling time is the time between two sequential readings of the same sensor. The chosen polling time determines the system's time of reaction. By default, it is 1 second.

Important: The maximum response time-out can’t be lower than the polling time.

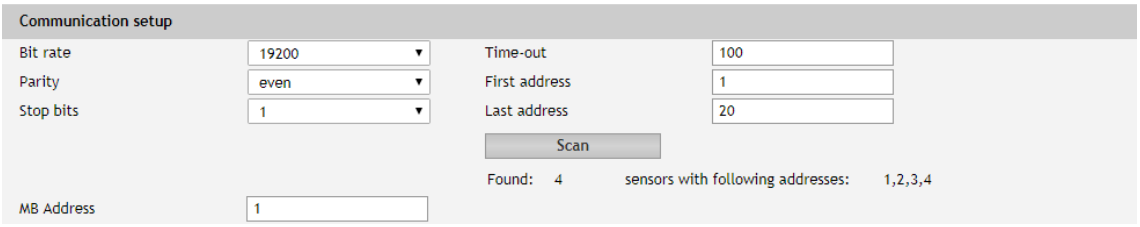
7.2.2.5. Sensor setup tool

The link for the tool is available on the bottom of the Modbus RTU sensors paragraph. It can be used for sensor communication setup changes or just to read information from a register.

7.2.2.5.1. Communication setup

The section is similar to the general MODBUS RTU communication setup. The only new field is the sensor address.

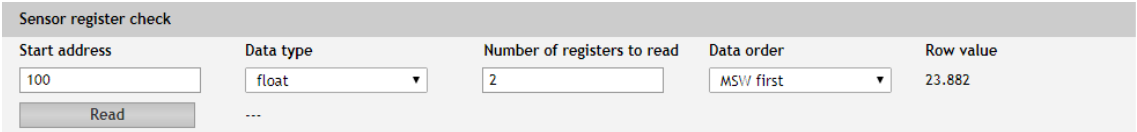
The changes of settings in this section are not memorized and don’t change the general settings of TCW210-TH.



7.2.2.5.2. Sensor communication register setup

This part of the tool is used for check and change the status of communication registers of the sensor.

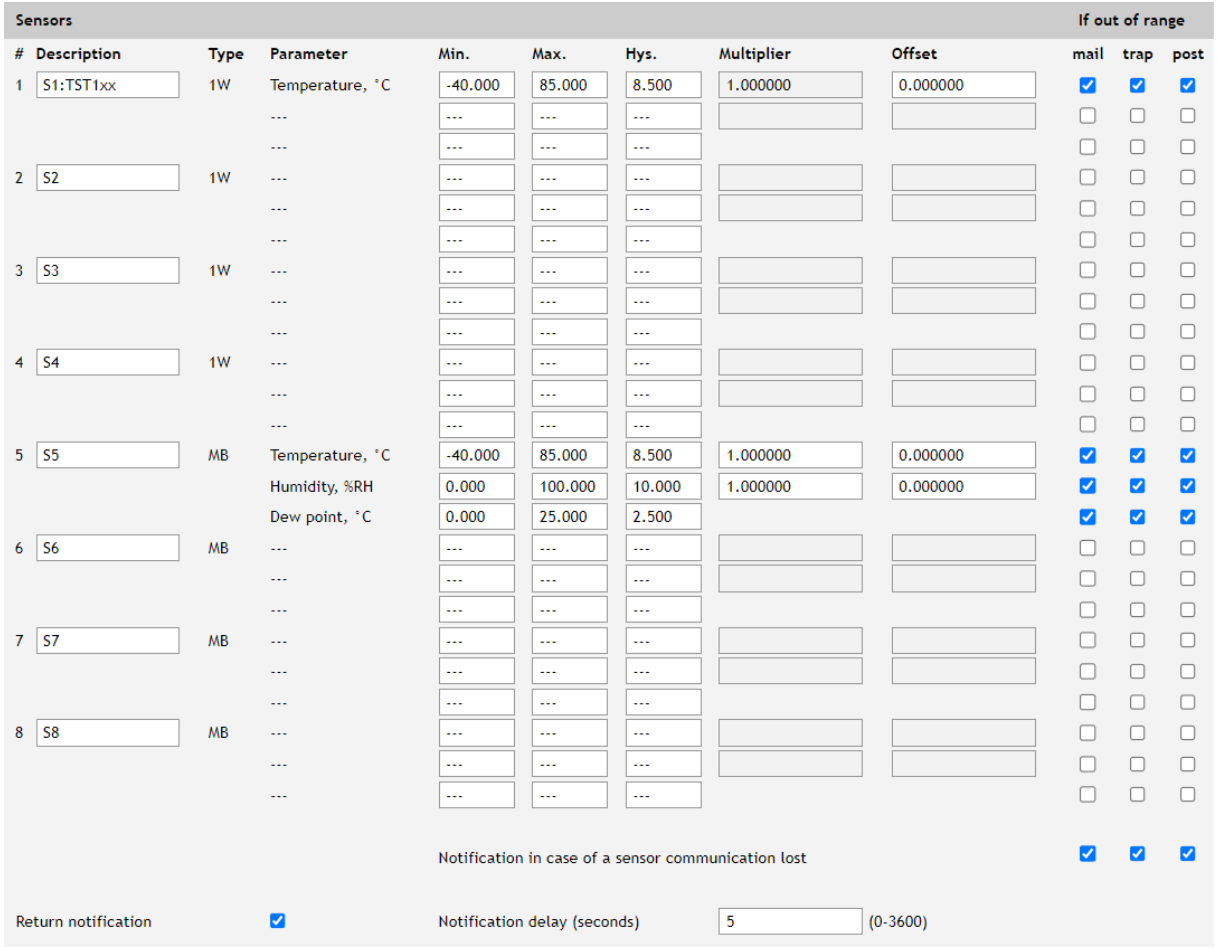
7.2.2.5.3. Sensor register check



This part of the tool is used for general sensor register check.

7.2.3. Conditions

This section is used for parameterization of 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 every sensor, a description up to 15 characters can be set.

For all sensors “Offset” field is enabled. The offset is used for simple correction of displayed value.

For all Modbus RTU sensors multiplier is enabled, but for Teracom sensors it should be 1.

For every parameter, there is a field for trigger conditions (“Min”, “Max” and “Hys.”).

“Min” and “Max” indicate the border of the working range for the observed parameter.

A “Max” trigger condition occurs when the value exceeds the trigger set point. A “Min” trigger condition occurs when the value is lower than the trigger set point. In both cases, the monitored parameter goes out of range.

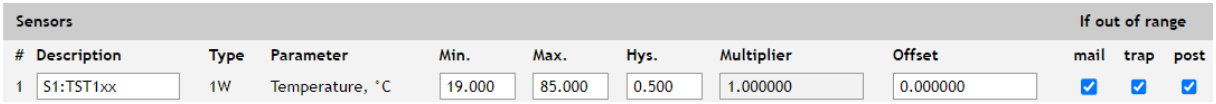
Coming back in range for the observed parameter is considered when the value goes higher than (Min + Hys) or lower than (Max – Hys). Hysteresis (“Hys”) is used to prevent excessively triggering when the value vacillates around trigger point.

Example:

TCW210-TH and TST103 are used to monitoring of room temperature. The wanted minimum temperature is 19°C. The initial temperature is 17°C.

TST100 is assigned to the first position for 1-Wire sensors.

Following parameters are set for Sensor1: Min=19, Max=85 and Hys=0.5.

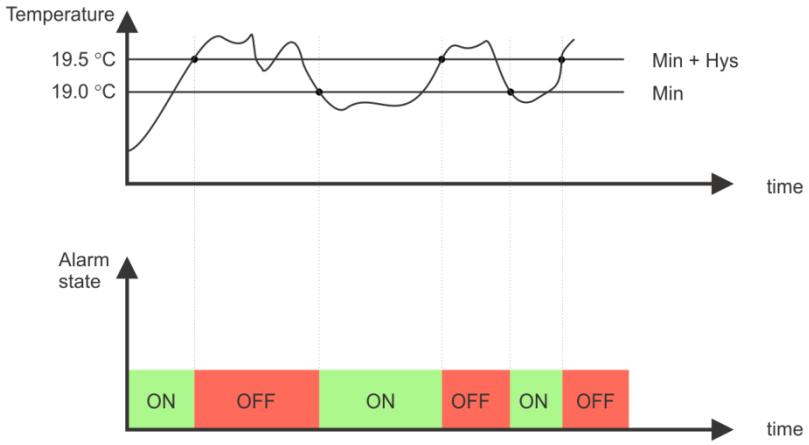


Sensors								If out of range		
#	Description	Type	Parameter	Min.	Max.	Hys.	Multiplier	Offset	mail	trap	post
1	S1:TST1xx	1W	Temperature, °C	19.000	85.000	0.500	1.000000	0.000000	☑	☑	☑

When the controller is switched on, Alarm is immediately activated because the monitored temperature is out of range.

When the temperature reaches 19.5°C (19.0 + 0.5) it goes in range (trigger condition) and Alarm is deactivated.

The temperature falls and when it reached 19°C it goes out of range (trigger and alert conditions). E-mail is sent.



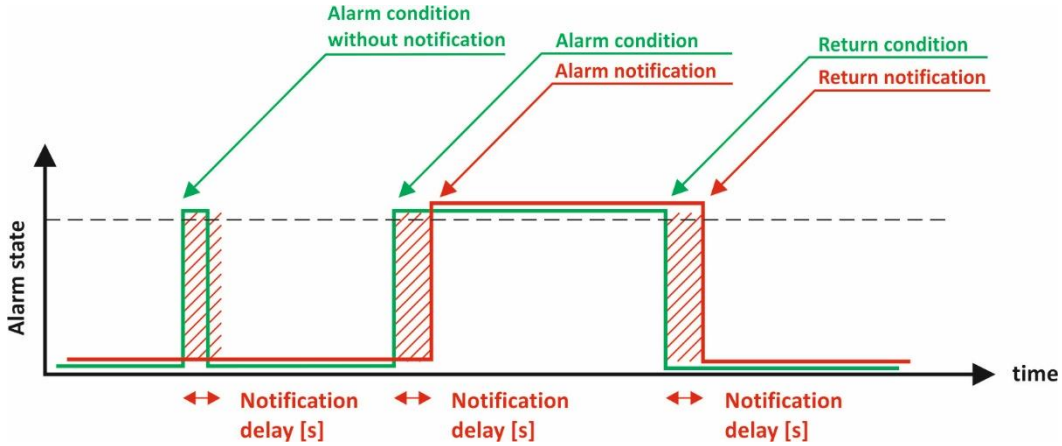
The “Max” value is set far enough from the wanted temperature to avoid trigger/alert conditions around it.

For every sensor, there are 3 independent ways of alert for alarm condition – e-mail, SNMP trap and HTTP/HTTPS post of an XML/JSON file. Each alarm notification method is activated by a checkbox.

In case of sensors communication loss e-mail, SNMP trap and post (HTTP/HTTPS post of XML file) notification can be send. Each notification method is activated by a checkbox.

Globally for all sensors, there is a checkbox “Return notification”. If this option is chosen there will be notification also when parameter returns in range.

Globally for all sensors, there are “Notification delay” parameter. It is very useful as a filter for short alarm conditions.

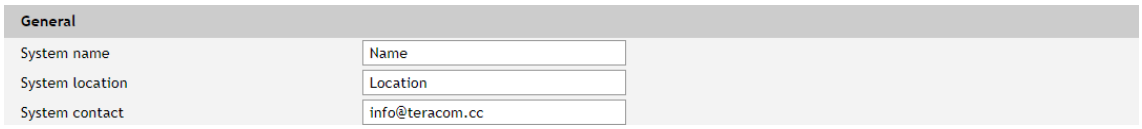


7.2.4. System

On this page, some general settings can be made.

7.2.4.1. General

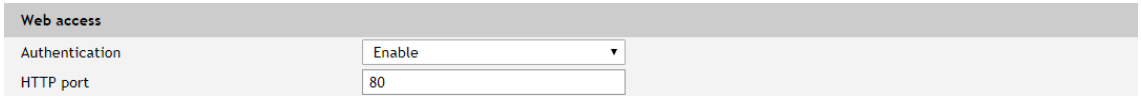
The system name, location, and contact can be used for automatic identification of device via M2M protocols.



7.2.4.2. WEB access

In this section, WEB access authentication can be deactivated. By default, it is activated with admin/admin authentication details.

HTTP port for WEB access can be changed. This is useful for some routers which don't support different outside/inside ports for port forwarding. By default, HTTP port is 80.



7.2.4.3. HTTP API

In this section, HTTP API access authentication can be activated/deactivated. By default it is active.

Authentication details are the same for WEB access. The controller support two types of authentication – see the explanation for HTTP API below.

7.2.4.4. Monitoring page automatic refresh

Monitoring page refresh interval can be set between 0 and 253 seconds. Zero means no automatic refresh.

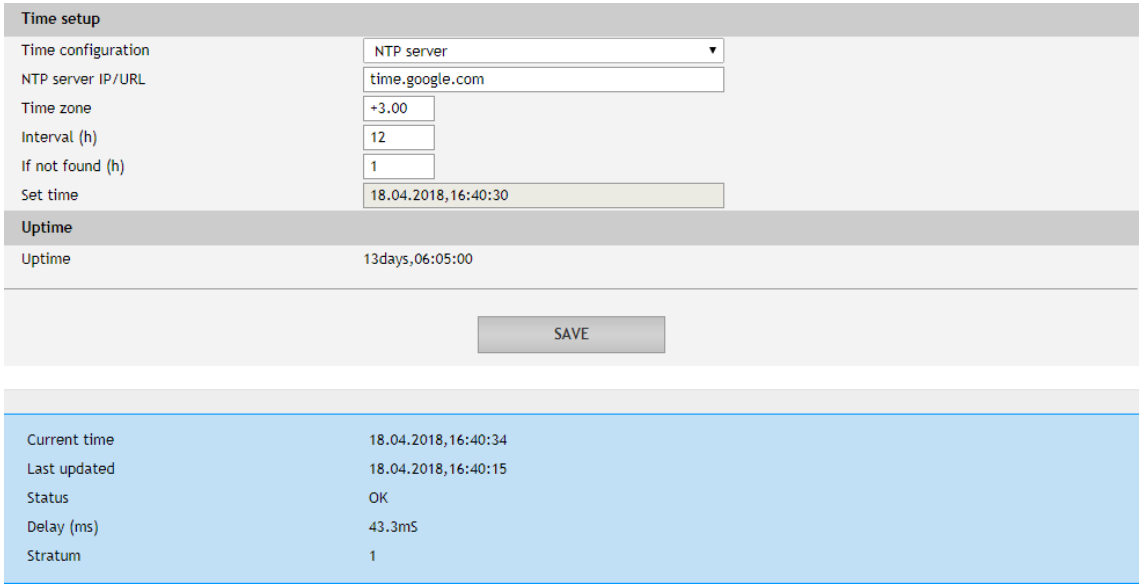
7.2.4.5. Display

The unit for observed temperatures can be selected between Celsius and Fahrenheit temperature scales.

If all sensors are attached to the one interface, the other section can be removed from the Monitoring page by the appropriate checkbox.

7.2.5. Time

Internal RTC (Real Time clock) of the controller can be set either manually or automatically.



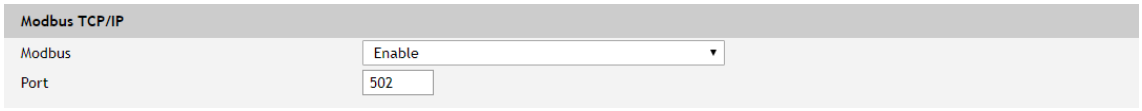
For automatic clock synchronization, the controller supports NTP (Network Time Protocol) and all necessary parameters for automatic synchronization are available in this section.

By default, NTP synchronization is disabled, server – time.google.com, Time zone +0.00 and interval of 12 hours.

7.3. Services

7.3.1. MODBUS

TCW210-TH supports MODBUS TCP/IP over the Ethernet interface.

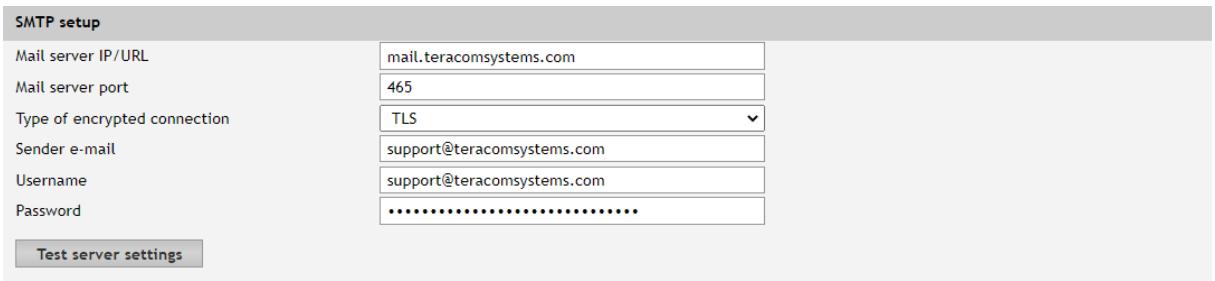


By default, MODBUS TCP/IP is disabled. The standard port for this protocol is 502. The table with the registers' addresses can be found in section 8.3. MODBUS TCP/IP.

7.3.2. SMTP

This page is used to enter valid SMTP settings for email alerts and recipients' addresses.

7.3.2.1. SMTP setup



The mail server address can be set either by hostname (for example mail.teracomsystems.com) or IP address.

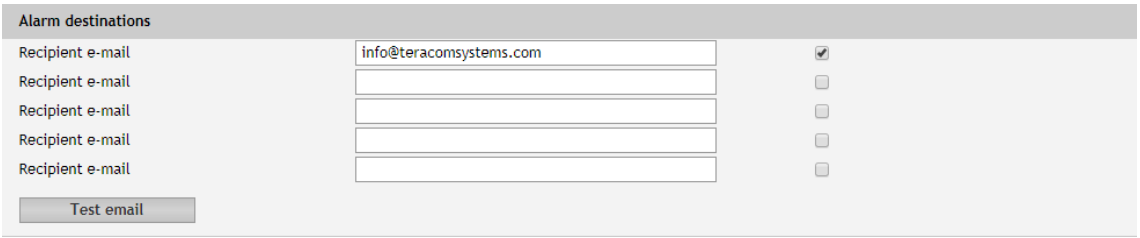
By default, without an encrypted connection, the SMTP port is 25. Ask ISP if the default port doesn't work.

Sender e-mail, username, and password are standard authentication details. For most SMTP servers, the sender's e-mail and username are the same.

There is a button for server settings test with feedback. In this test sender and recipient of the e-mail are the same.

Transport Layer Security protocol is used for secure communication with public mail servers. TCW220 supports TLS 1.0, TLS 1.1, and TLS 1.2 with RSA as a key exchange/agreement and authentication, which ensures successful operation with almost all public servers. STARTTLS is not supported.

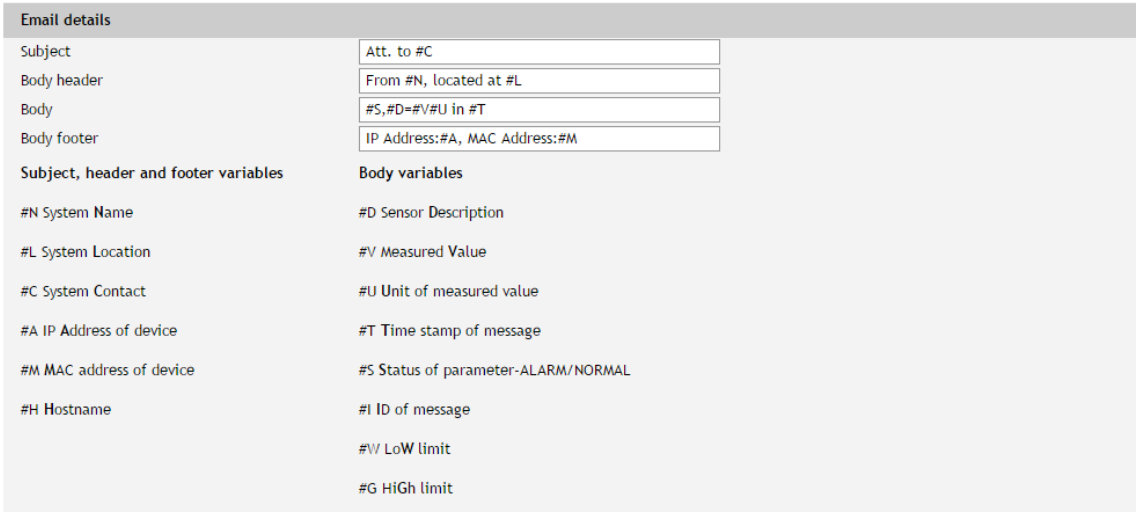
7.3.2.2. Alarm destination



Up to 5 mail recipients can be set. All they can be activated independently by a checkbox.

7.3.2.3. E-mail details

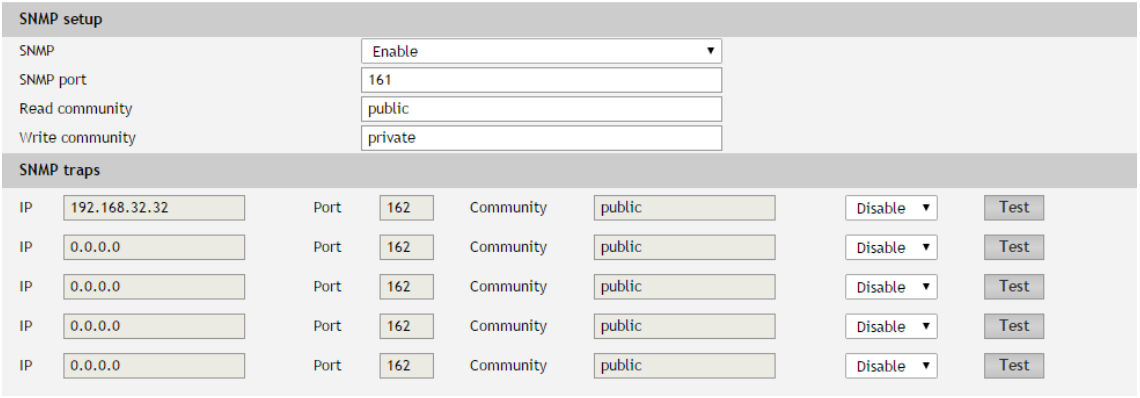
The subject, body header, body and body footer can be customized. For this customization, a set of keys is used. All they are described on the page.



7.3.3. SNMP

The TCW210-TH supports SNMP v.2.

In this section, all necessary parameters for proper operation of SNMP can be set.



By default SNMP is disabled, the port is 161, read community is public and write community is private.

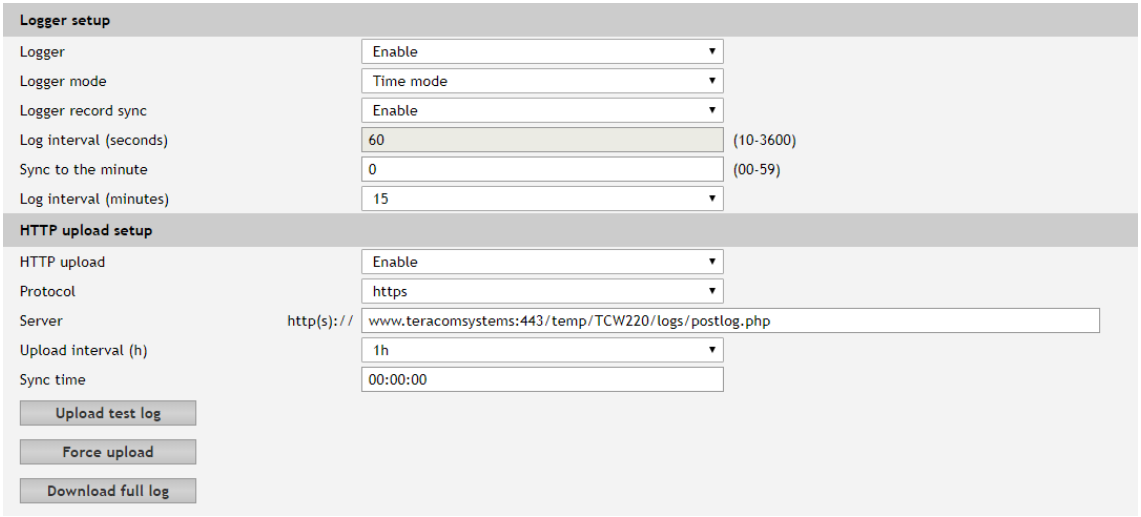
In an alarm condition, SNMP trap can be sent up to 5 independent recipients. All they can be with different port and community. There is an independent button for trap test. SNMP traps can be sent if:

- the measured parameter of the sensor goes outside the range;
- restart;
- SW reset.

7.3.4. Logger

The TCW210-TH supports logger for all monitored parameters. The records are made in a circular buffer within the internal flash memory. When the buffer is full, the oldest values are overwritten with the newest ones.

7.3.4.1. Logger



The logger works in three modes – Time, Alarm and Time&Alarm. The mode specifies what initiates a record in the logger's memory.

In Time mode, records are made periodically on "Log interval" time. In Alarm mode, records are made on every alarm condition. In Time&Alarm mode, a mix of both conditions for records is used.

The log interval determines the time between two log entries. It is good to remember that by reducing the log interval, we increase the resolution, but we also reduce the past period for which we have records.

The logger records can be synchronized with a specific minute in an hour. Synchronization is very useful when monitoring electricity, water, gas meters, etc. The log interval can be chosen from a drop-down menu between 1 and 60 minutes. The field „Sync to the minute“ determines which minute of every hour is used for synchronization. Although any minute can be selected, it is better to use the default value - 00.

Example:

The current settings are:

- Current time = 09:12
- Logger record sync = Enable;
- Sync to the minute = 00;
- Sync interval = 15 minutes.

The settings determine 4 records per hour in **HH:00**, HH:15, HH:30, and HH:45.

The device is powered up.

The first record will be immediately after power-up - 09:12. The next records will be in 09:15, 09:30, 09:45, 10:00, 10:15, etc.

There are two ways to reach the logger records:

- download of the full log file, using “Download full log” in the WEB interface*;*
- periodical upload the last unsent records to the dedicated HTTP server.

The records are uploaded in CSV file format using HTTP or HTTPS protocol. The HTTPS upload is over TLS 1.0, TLS 1.1 and TLS 1.2 with RSA as a key exchange/agreement and authentication.

The period of the upload can be chosen from the menu between 1 and 24 hours. If you enable this service, take care of the real-time clock (NTP service).

The HTTP server for upload can be domain or IP address but take care about DNS settings.

“Sync time” is a moment in the day when a period of upload is synchronized.

Example:

Current time is 19:31, Upload period is 3 hours and Sync time is 9:00.

To synchronize the logger to 9:00 it means that time for uploads will be: 09:00, 12:00, 15:00, 18:00, 21:00, 24:00, 03:00 and 06:00. The first upload, after enabling the logger in 19:31, will be in 21:00.

The button “Force upload” initiates upload recorded information between previous periodical upload and now.

By default, the logger is disabled.

More about the logger can be found in the Data logger section.

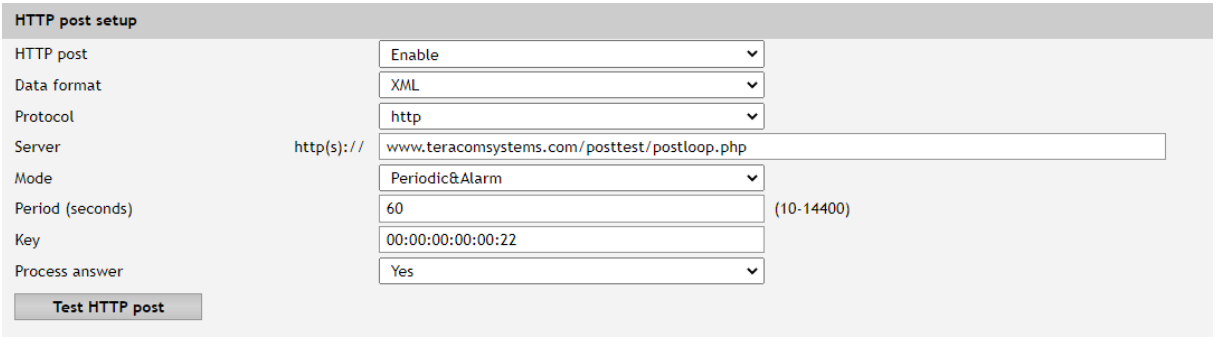
7.3.5. HTTP post

TCW210-TH can periodically upload a file to a dedicated server using HTTP or HTTPS Post.

The HTTPS is over TLS 1.0, TLS 1.1 and TLS 1.2 with RSA as a key exchange/agreement and authentication.

The posting period is between 10 and 14400 seconds.

The file format can be XML or JSON.



By default, Periodic&Alarm is selected as the mode. In addition to the periodic posts, a file can be uploaded at any alarm condition.

If Periodic only is selected as the mode, then periodic posts are performed without alarm posts.

If Alarm only is selected as the mode, then alarm posts are performed without periodic posts.

The "Key" field value is sent in the XML/JSON and can be used for device identification.

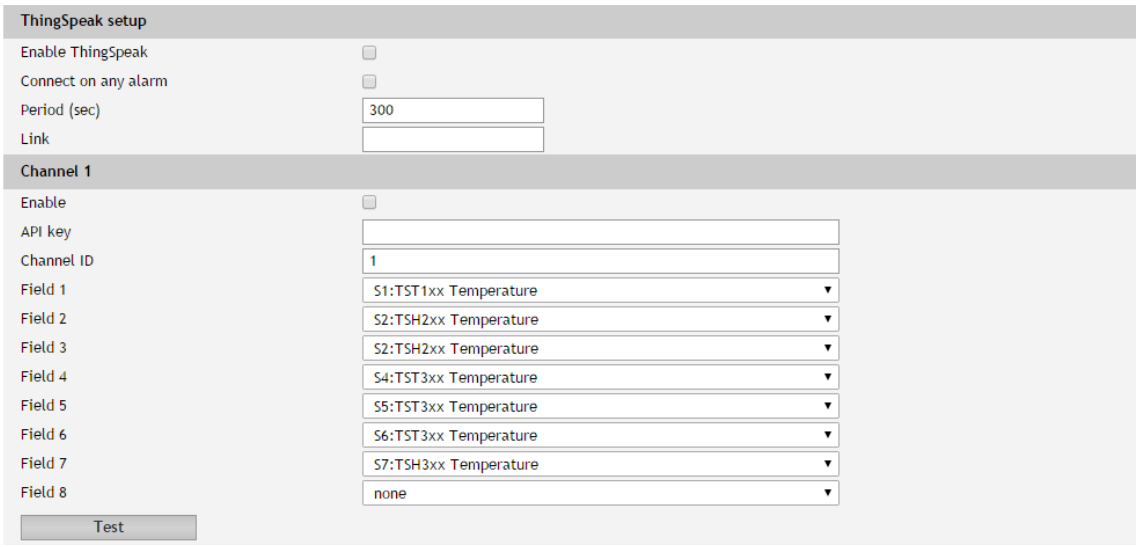
If "Process Answer" option is enabled, the TCW210-TH will process the answer of the remote server. The list of valid commands is described in section "HTTP API commands".

7.3.6. Cloud

ThingSpeak server is an open data platform and API for the Internet of Things that enables you to collect, store, analyze, visualize, and act on data from sensors.

The primary element of ThingSpeak activity is the channel, which contains API key, channel ID, and eight data fields.

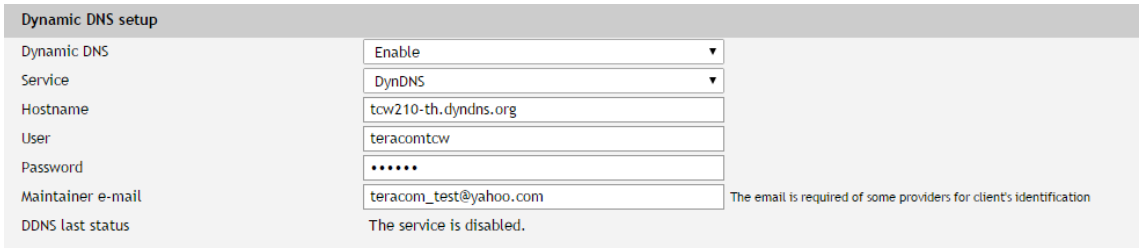
TCW210-TH has four channel sections – Channel 1, Channel 2, Channel 3 and Channel 4.



7.3.7. Dynamic DNS

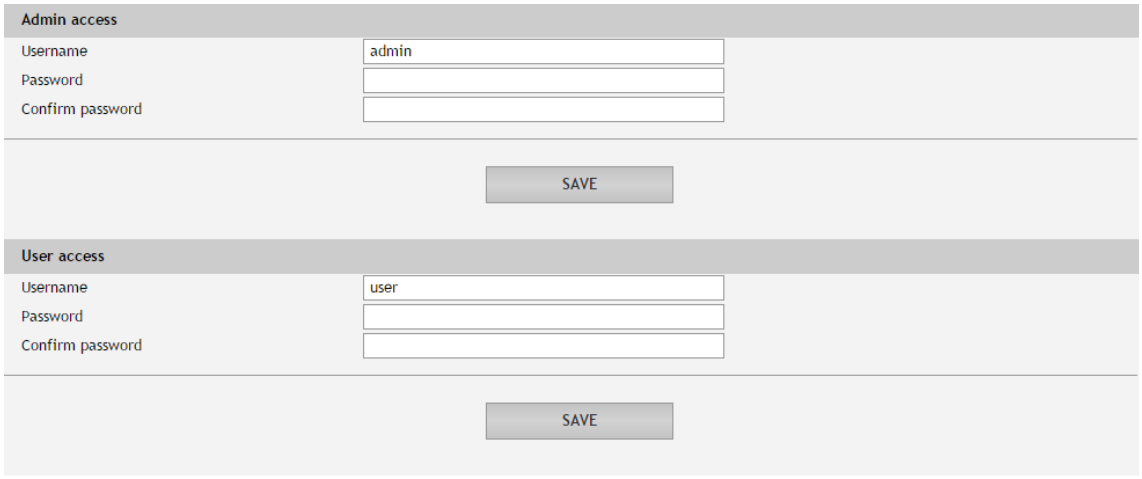
With dynamic DNS can access TCW210-TH from the public Internet without investing in a broadband account that has a static IP address.

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



7.4. Administration

7.4.1. User/Pass



The TCW210-TH supports two users – “Admin” and “User”.

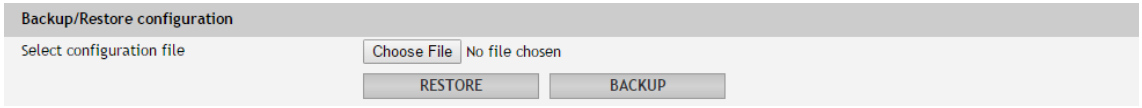
“Admin” has administrative rights.

“User” shall not modify any settings.

The username and password can be up to 31 characters long.

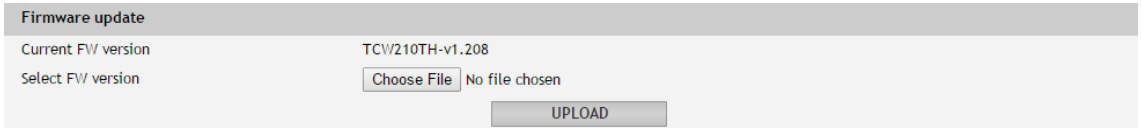
7.4.2. Backup/Restore

TCW210-TH supports backup and restore of all user setting. All settings are saved in XML backup file. This file can be used after this for restore on many devices. This is very useful for multiplying similar settings to a batch of controllers.



7.4.3. FW update

The TCW210-TH can be updated via the WEB interface.



To update the device follow the steps below:

- Go to www.teracomsystems.com and download the latest firmware;
- From Administration->FW update select downloaded .cod file and press “upload” button;
- After the firmware update is completed, the Login page will appear.

Attention! Don’t turn off the power supply during the update. Turning off the power supply will damage the device.

7.5. Logout

The TCW210-TH support multisession, but the good practice is to log out after finish the work.

8. Protocols and API

8.1. SNMP

Simple Network Management Protocol (SNMP) is a standard internet protocol for managing devices on IP networks. In typical uses of SNMP, one or more administrative computers, called managers, monitor and control devices on LAN. Each controlled device, at all times, executes a software component called an agent which reports information via SNMP to the manager.

The TCW210-TH can be configured and monitored through SNMP.

This could be done using every SNMP v.2 compatible program. Parameters that can be changed, are grouped according to their functions in the tables below. To obtain a valid OID number it is necessary to replace the “x” symbol with ”1.3.6.1.4.1.38783”.

To save the changes configurationSaved (OID x.2.3.5.0) should be set 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) }

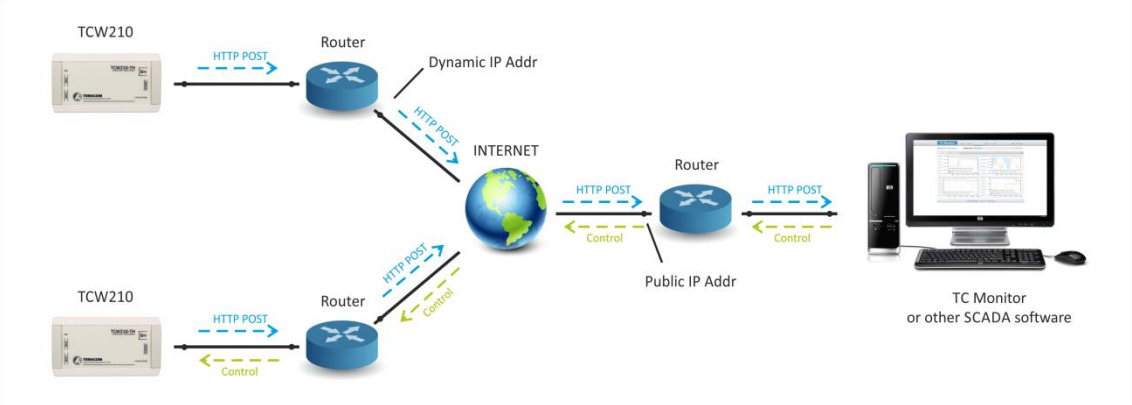
8.2. HTTP API

8.2.1. HTTP Post

TCW210-TH can execute HTTP/HTTPS Post to upload XML/JSON file to a dedicated server.

This functionality is very useful if the controller is behind the router without public IP address or the user don't have access to router configuration. The server should have a public IP address.

The typical monitoring application is shown in the picture below:



HTTP/HTTPS post can be sent periodically or periodically plus on an alarm condition.

To test HTTP/HTTPS Post follow the steps below:

- 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!";
        }
?>
```

- Copy the post.php file on a public web server with PHP support. To verify that the script is working properly, you can type the URL in your web browser (for example www.yourserverURL.com/post.php). If all is OK, a web page with "The PHP script is working!" will be shown.
- Set the controller to send an HTTP/HTTPS POST to your web server. Enter the address (yourserverURL.com/post.php) in the URL field. Click on "Test HTTP Post" button.
- If the HTTP/HTTPS POST is received and processed, "OK" will be shown close to the button. Along with this, an XML file will be created in the same directory, where post.php is located. The file name will contain time information and looks like 20171120103318_status.xml.

8.2.2. HTTP Get

HTTP Get can be used to monitor TCW210-TH via XML or JSON files. The format is as follows:

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

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

See sections **8.2.4 XML file structure** and **8.2.5 JSON file structure** for details of files.

HTTP Get can be sent at any time to TCW210-TH if it is on the same network or it has appropriate routing.

If there isn't direct access to the device, HTTP Get can be sent immediately after HTTP Post receiving from the same device.

8.2.2.1. Commands

All command used with HTTP Post can be used also with HTTP Get. The right format 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 post period of 120 sec.

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 shown 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:
http://device.ip.address/status.xml?a=admin:admin&pper=120 will set post period of 120 sec in case the username=admin and pass=admin

8.2.3. List of HTTP API commands

Command	Description
sn**pt**=30.0	Set Min of sensor to 30.0 (**p** is 1,2,3,4,5,6,7 or 8 for the respective sensor **t** is 1 or 2 for the respective parameter of sensor) sn12=30.0 will set Min for sensor 1, parameter 2
sx**pt**=40.0	Set Max of sensor to 40.0 (**p** is 1,2,3,4,5,6,7 or 8 for the respective sensor **t** is 1 or 2 for the respective parameter of sensor) sx42=40.0 will set Min for sensor 4, parameter 2

sy**pt**=2.0	Set Hys of sensor to 2.0 (**p** is 1,2,3,4,5,6,7 or 8 for the respective sensor **t** is 1 or 2 for the respective parameter of sensor) sy81=2.0 will set Hys for sensor 8, parameter 1
delsen=xxxx	Notification delay for sensors (xxxx is between 0 and 3600)
dataf=x	Data format XML/JSON for HHTP Post – 0 XML, 1 JSON
pushtls=x	http(s) 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 protocol is originally published by Modicon in 1979. It is used to establish master-slave/client-server communication between intelligent devices. MODBUS TCP/IP is often used to connect a supervisory computer with remote units in supervisory control and data acquisition (SCADA) systems.

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 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 – Odd or Even;

As factory default communication parameters the device uses standard ones for MODBUS RTU:

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

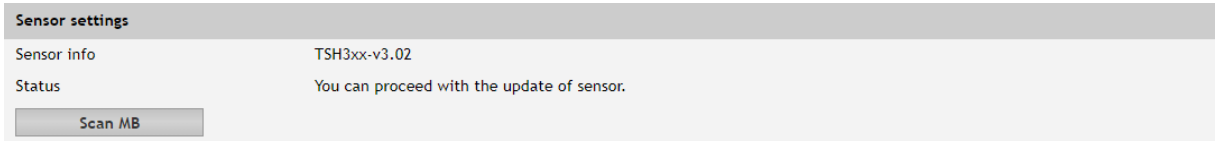
8.4.2. Teracom sensors update tool

TCW210-TH supports Teracom sensor FW update tool.

The tool is available at http://device.ip.address/teracom485.htm.

Attention! To make any changes to MODBUS RTU sensor it should be alone on the RS-485 bus.

8.4.2.1. Sensor settings



The tool works with the current MODBUS RTU communication parameters.

To avoid collisions the good practice is to set TCW210-TH and the sensor with the factory default MODBUS RTU communication parameters. This will ensure smooth operation. The default MODBUS RTU communication parameters for TCW210-TH are described in 8.4.1. Communication parameters.

Before to make any changes it is strongly recommended to scan for sensor settings. This will inform about the current FW version of the sensor but also will check if the sensor is alone on the bus.

8.4.2.2. Sensor FW update

To arrange the FW update, the appropriate file should be uploaded to the sensor first and after this, the button “Update” should be pressed.

8.5. Logger

The logger utilizes circular buffer in FLASH memory. When it is full, the new data overwrites the oldest one. In this manner FLASH memory stores full log all the time. There isn't a command to clear the log. Copy of full log is always available for download.

The number of records depends on how long descriptions and what kind of characters are used. In the worst case (15 bytes description with characters from the highest part of UTF-8) the number of records is about 52000. This number is enough for 36 days with records every 1 minute.

The new data can be periodically uploaded as a file to the dedicated HTTP server in time intervals – 1, 2, 3, 4, 6, 8, 12 and 24 hours. The data is sent in CSV format. The semicolon is used for a delimiter.

The first row of the log file is always header. All rows, including the header, start with record ID and time stamp.

Structure of one row (record) of the log is as follows:

ID	Time	Type of record	Inputs value	Relays	Alarm conditions

ID	32-bit unique number for every row (record).	
Time	time stamp of record, in format yyyy.mm.dd, hh:mm:ss.	
Type of record	following types of records are available:	
	"Time"	for periodical record;
	"Event"	for record initiate by alarm condition;
	"Type"	for header record;
	"Start"	after power-up condition;
	"Restart"	after reset condition;
	"Power Down"	after power-down condition;
	"Bad"	for problematic record.
Inputs value	sensors.	
Alarm conditions	show condition for every input, "1" means an active alarm.	

Example of 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;
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;
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 3 different ways.

9.1. Factory default from WEB interface

If the button “Factory default” from Administration->Backup/Restore is pressed, all parameters return to factory default except 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 working, all Network settings go to factory default.

9.3. General factory default with the reset button

For factory default reset of all parameters following steps should be executed:

- Press and hold the RESET button, then turn on the power supply;
- Yellow LED shines and red LED blinks about 5 times on a second;
- After about 5 seconds red LED will turn off, the button can be released;
- Yellow LED flashes on 1 second and red LED shines – 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 up to 2000 meters.

When the controller is a part of a system, the other elements of the system shall comply with the EMC requirements and shall be intended for use in the same ambient conditions.

11. Safety

This device must not be used for medical, life-saving purposes or for any purpose where its failure could cause serious injury or the loss of life.

To reduce the risk of fire, only flexible stranded wire, with cross section 0.5mm² or larger for wiring of digital and analog inputs and relay output of the device should be used.

To avoid electric shock and fire hazard, 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 (damage) of the controller, if recommended free spaces to adjacent devices are not ensured. The joint part with external component shall have space for attachment/removal of the cable after installation.

Teracom does not guarantee successful operation of the product if the product was used under conditions deviating from the product specifications.

To ensure that the device works correctly follow the steps below:

- ensure that the device is installed correctly, refer to this user manual;
- log in to the devices via a browser program;
- make proper setup;
- install sensor TSH1XX or TST1XX on the 1-Wire bus;
- install sensor TSH3XX or TST3XX on the RS-485 bus;
- go to “Monitoring page” of WEB interface – proper parameters value should be displayed at the same time flashing “STS” led should indicate the 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

Upon completion of any service or repairs to the device or once per year, a safety check must be performed to determine that this product is in proper operating condition.

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

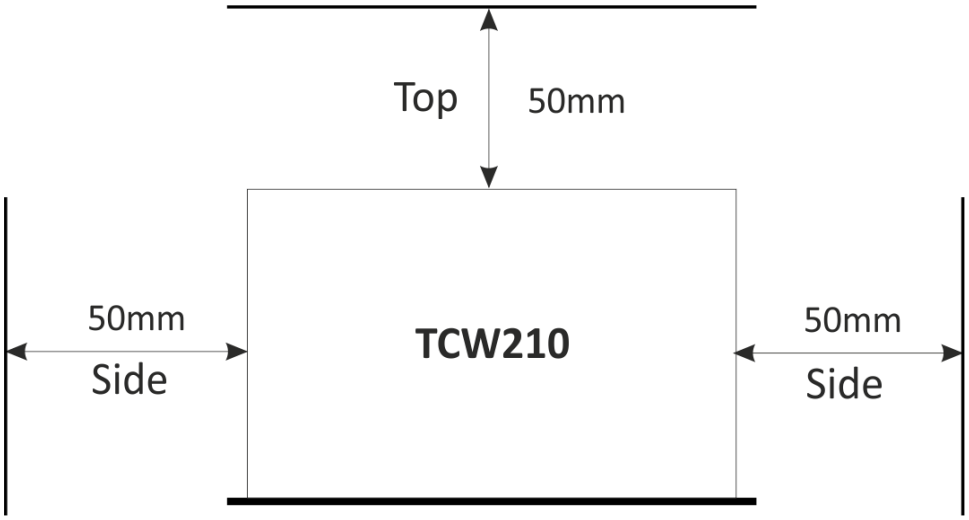


Fig.1