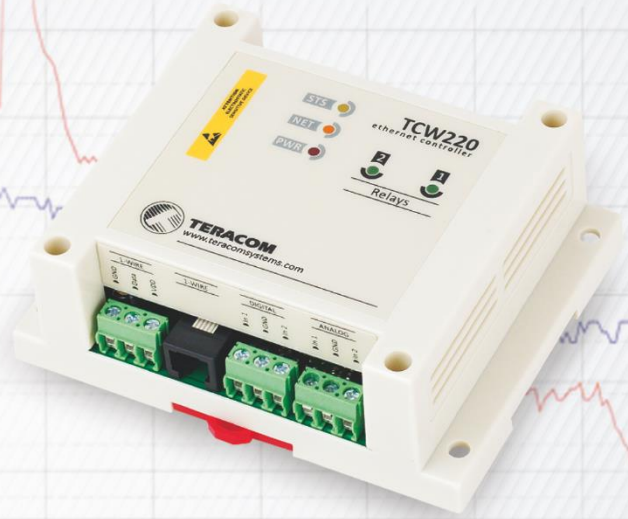
control solutions
TERACOM



TCW220 Ethernet data logger

Revision 4.20 / October 2023

USER MANUAL

www.teracomsystems.com

Ethernet data logger TCW220

1. Introduction

TCW220 is a general purpose and stand-alone Ethernet data logger for data acquisition applications. It has 2 analog inputs, with 10-bit resolution and 2 digital "dry contact" inputs. It also has a 1-Wire interface for up to 8 Teracom 1-Wire sensors - temperature, humidity, CO2, current, 4/20mA, galvanic isolated analog voltage, etc.

TCW220 has 2 relays with normally open and normally closed contacts. The relays can be activated either remotely (WEB, SNMP, HTTP API, MODBUS etc.) or locally – from the status of a monitored parameter (1 Wire sensor, analog voltage, and dry contact)

All monitored parameters can be logged, in internal FLASH memory, on a previous set time interval and/or on an alarm condition. The memory is large enough for at least 36 days with records on every 1 minute. The log file can be periodically uploaded to 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. Like answer server can return commands for relay's control. This is a way for building SCADA system.

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

2. Features

- 10/100 Mb Ethernet connectivity;
- Auto-MDIX;
- Ethernet data logger with up to 70000 records;
- Graphical display of logged data;
- Password protected web-based configuration and control;
- 2 digital "dry contact" inputs;
- 2 analog inputs with 0 to 10VDC range, with 10-bit resolution;
- Settable multiplier, offset, and dimension for analog inputs;
- 2 relays with NO and NC contacts;
- 1-Wire interface for up to 8 Teracom sensors;
- SNMP v.2 support;
- SNMP traps to up to 5 recipients like alarm alert;
- MODBUS TCP/IP support;
- SMTP with TLS security;
- TLS 1.0, TLS 1.1 and TLS 1.2 support;
- e-mails to up to 5 recipients like alarm alert;
- XML/JSON status files;
- HTTP API commands;
- Dynamic DNS with DynDNS, No-IP and DNS-O-Matic support;
- NTP support;
- Periodical HTTP/HTTPS Post of XML/JSON status files for client-server systems;
- Port changing for HTTP, SNMP and MODBUS;
- Wide power supply voltage range;
- Remote firmware update.

3. Applications

TCW220 is suitable for industrial automation, data acquisition systems, environmental monitoring and local control of an electrical and non-electrical parameter, building automation etc.

It works very well as a standalone device that can be controlled using a web browser or as a part of small to large industrial control systems for SCADA (supervisory control and data acquisition).

A few example applications include:

- Stand-alone or system Ethernet data logger;
- Industrial process automation;
- Industrial environmental monitoring;
- Mushroom and winery environmental monitoring and control;
- Small to large SCADA systems;
- Ethernet data logger for storage facilities.

4. Specifications

- Physical characteristics

  Dimensions: 115 x 90 x 40 mm
  Weight: 170 g

- Environmental limits

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

- Warranty

  Warranty period: 3 years

- Power requirements

  Input Voltage: 10 to 28 VDC
  Input Current: 240 mA @ 12 VDC (with all relays ON)

- Digital inputs

  Isolation: Non isolated
  Mode: Dry contact
  Maximum input voltage: +5.5VDC
  Sampling rate: 10mS
  Digital filtering time interval: 30mS

- Analog inputs

  Isolation: Non isolated
  Type: Single ended
  Resolution: 10 bits
  Mode: Voltage
  Input Range: 0 to 10 VDC
  Accuracy: ±1%
  Sampling Rate: 500mS per channel (averaged value of 500 samples)
  Input Impedance: 150 kilo-ohms (min.)

- Relay outputs

  Type: Form C (N.O. and N.C. contacts)
  Contact current rating: 3 A @ 24 VDC/30 VAC (resistive load)
  Initial insulation resistance: 100 mega-ohms (min.) @ 500 VDC
  Mechanical endurance: 10 000 000 operations
  Electrical endurance: 100 000 operations @ 3 A resistive load
  Contact resistance: 50 milli-ohms max. (initial value)
  Minimum pulse output: 1 Hz at rated load

- 1-Wire 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:

- **Relay1-Relay2** (green) – these LEDs are illuminated whenever the corresponding relay is activated (the NO contact is closed and the NC contact is open);
- **PWR** (red) – in working mode shines, blinks together with STS if there is a hardware error;
- **STS** (yellow) – flashes when the main program of the controller is executed;
- **NET** (orange) – network status - ON when a link is established, blinks if 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

TCW220 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.

Mount the device to a wall by using two plastic dowels 8x60mm (example Würth GmbH 0912 802 002) and two dowel screws 6x70mm (example Würth GmbH 0157 06 70). Attach the screws to the surface vertically. See Appendix-A, fig. 1 for mechanical details.

Maintain spacing from adjacent equipment. Allow 50 mm of space on all sides, as shown in fig.2 in Appendix A, this provides ventilation and electrical isolation

TCW220 can be mounted to a standard (35mm by 7.55mm) DIN rail. Attach the controller to the DIN rail by hooking the hook on the back of the enclosure to the DIN rail and then snap the bottom hook into place.

6.2. Connection

Attention! Disconnect power supply before wiring.

The correct wiring procedure is as follows:

- Make sure power is turned off;
- Make wiring connections to the terminals;
- Apply power.

It is recommended to test and configure TCW220 without any controlled device. In this case, unexpected turn on will be avoided.

Make sure that wires are properly attached to the terminals and that the terminals are tightened. Not proper wiring and configuration can cause permanent damage to TCW220 or the equipment to which it is connected or both.

Connector 1	Ethernet - RJ45
Connector 2	Power - 2.1x5.5mm connector, central positive
Connector 3	Pin1 – Power positive Pin2 – Power negative
Connector 4	Pin1 – NC Relay2 Pin2 – COM Relay2 Pin3 – NO Relay2 Pin4 – NC Relay1 Pin5 – COM Relay1 Pin6 – NO Relay1
Connector 5	Pin1 – 1-Wire GND Pin2 – 1-Wire Data Pin3 – 1-Wire +VDD
Connector 6	Pin1 – GND (most left) Pin2 – GND Pin3 – 1-Wire Data Pin4 – 1-Wire GND Pin5 – 1-Wire +VDD Pin6 – 1-Wire +VDD (most right)
Connector 7	Pin1 – Digital In 1 Pin2 – GND Pin3 – Digital In 2
Connector 8	Pin1– Analog In 1 Pin2 – GND Pin3 – Analog In 2

6.2.1. Power supply

TCW220 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.

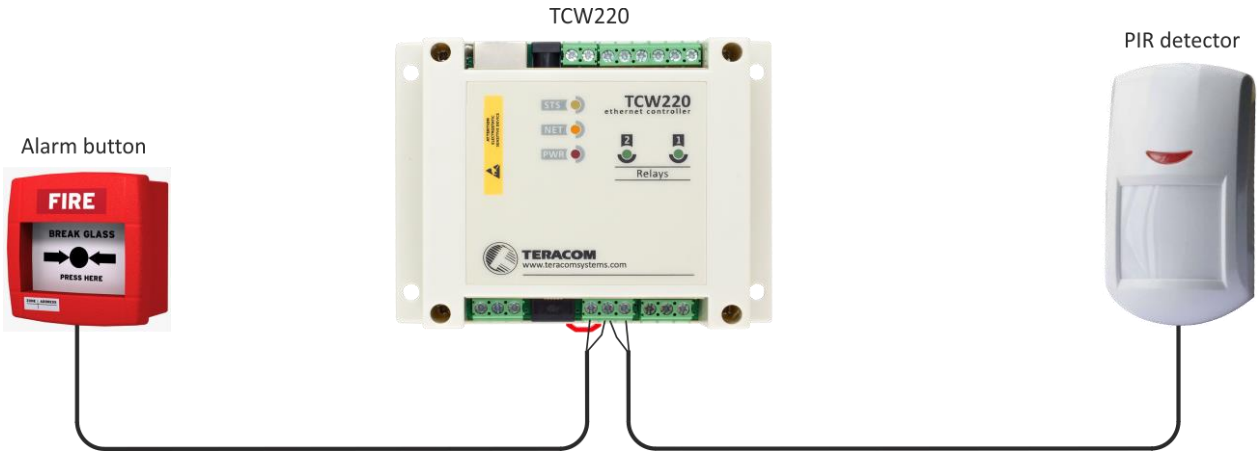
When in use, do not position the equipment so that it is difficult to disconnect the device from the power supply.

6.2.2. Digital inputs

Attention! Digital inputs are NOT galvanic isolated.

The digital inputs of TCW220 can be used for monitoring of devices with “dry contact” outputs – door contact switch, push button, PIR detector etc.

The picture below illustrates how a dry contact can be connected to inputs of TCW220.



One side of the contact is connected to “Digital In” and another side is connected to “GND” terminals.

The maximum cable length should be up to 30 meters.

6.2.3. Analog inputs

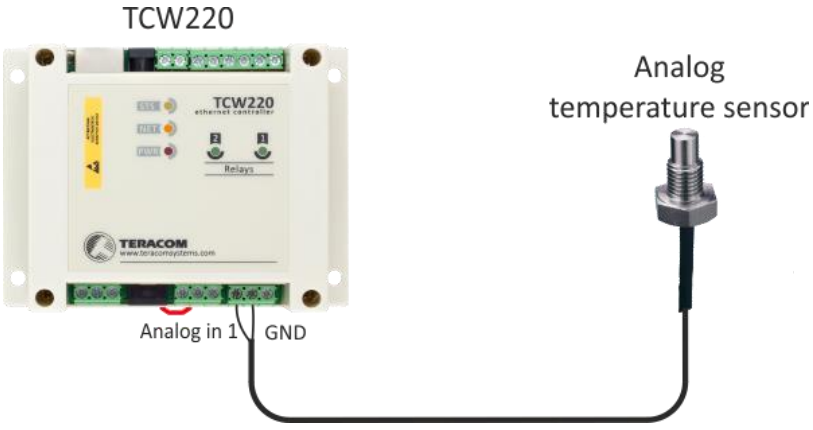
Attention! Analog inputs are NOT galvanic isolated.

Analog inputs of TCW220 have standard industrial range 0/10VDC. They can be connected directly to analog sensors with 0/5VDC or 0/10VDC. The inputs are factory calibrated for better accuracy.

Built-in functionality “Multiplier”, “Offset” and “Dimension” for every analog input gives a possibility to assign almost all analog sensors while the directly measured parameter is shown.

It is also possible to monitor voltages bigger than 10 VDC with external resistive dividers.

The following picture illustrates how a high-temperature sensor can be connected to the analog input of TCW220. The output of a sensor is connected to “Analog In” and the shield is connected to “GND” terminal.



The maximum cable length should be up to 30 meters.

6.2.4. 1-Wire interface

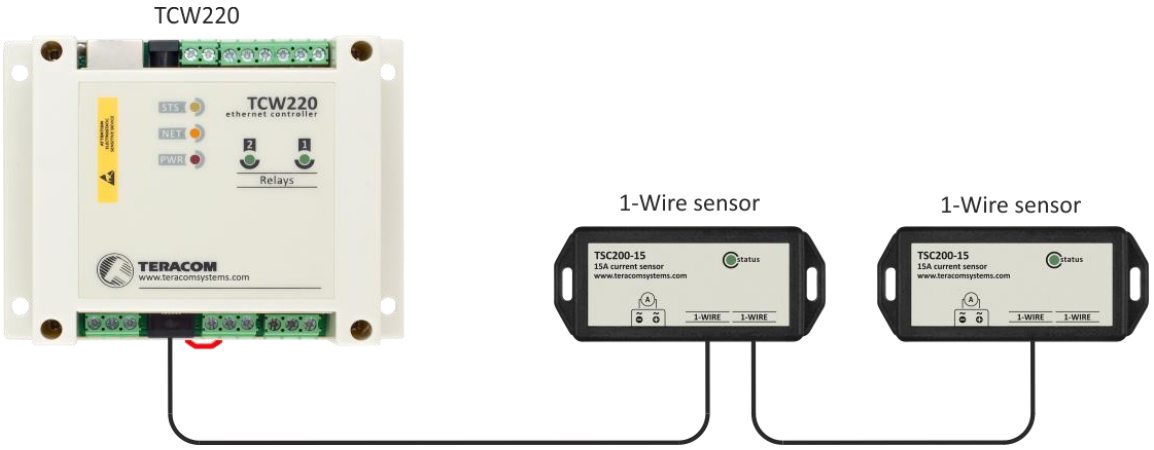
Up to eight 1-Wire sensors can be connected to TCW220. The device supports following sensors - temperature, temperature/humidity, CO2, DC current, AC current, 4/20mA transmitter, galvanic isolated analog voltage, barometric pressure etc. Connected sensors are automatically detected and appropriate dimension is assigned.

For some sensors, like example TSV2xx and TSA2xx, multiplier, offset and dimension can be set. This allows direct measured physical parameter monitoring.

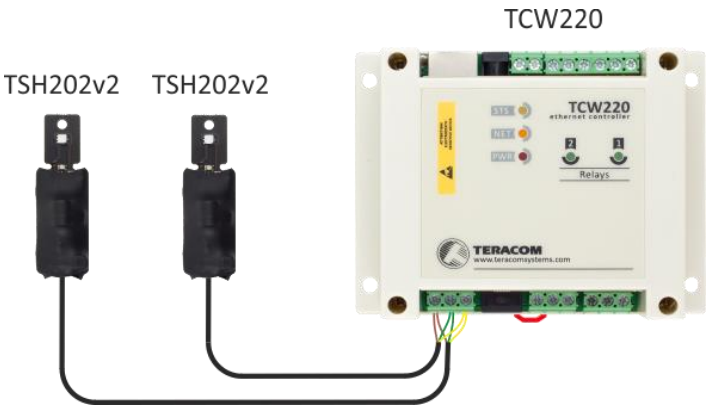
1-Wire is a registered trademark of Maxim Integrated Products, Inc. It is designed to connect several sensors over a short wiring. It is not suitable for long distances or environments with EMC interference. We recommend reading Maxim's 1-Wire tips at https://www.teracomsystems.com/wp-content/uploads/1-wire/guidelines-for-reliable-long-line-1-wire-networks.pdf.

The sensors have three wires – positive voltage (+VDD), ground (GND) and bidirectional data (Data). The colors of wires or pinot for every sensor are specified in its user manual.

It is strongly recommended to use "daisy-chained" (linear topology) for multiple sensors:



"Star" topology can be used only as a last resort for up to 4 sensors and total cable length up to 10 meters:



Connections can be realized either by screw terminal connector or by a standard RJ-11 connector.

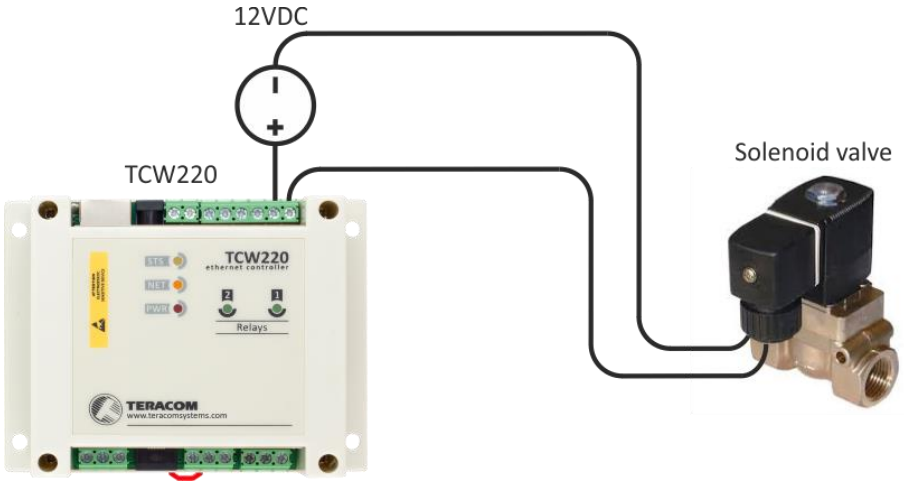
There are many parameters which determine the maximum length of the wires – the type of cable, the number of sensors, ambient electromagnetic noise and sensor network topology.

It is strongly recommended to use only UTP/FTP cables and keep total cable length up to 30 m. Although functionality has been achieved on a longer distance, we cannot guarantee error-free operation over mentioned wiring length.

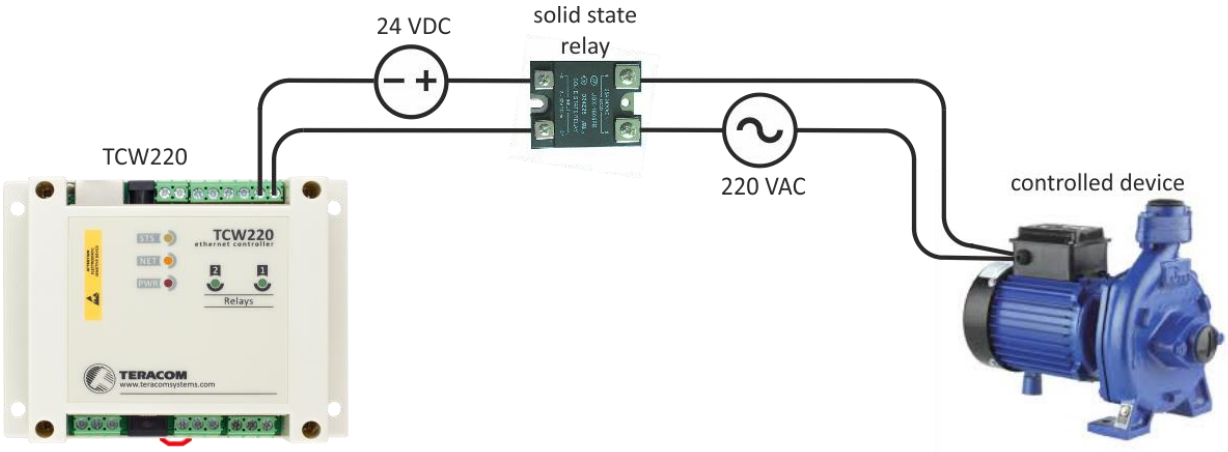
We guarantee proper operation only with Teracom 1-Wire sensors.

6.2.5. Relays

The relay contacts are internally connected directly to the terminal connectors. For all relays normally open, normally closed and common contacts are available.



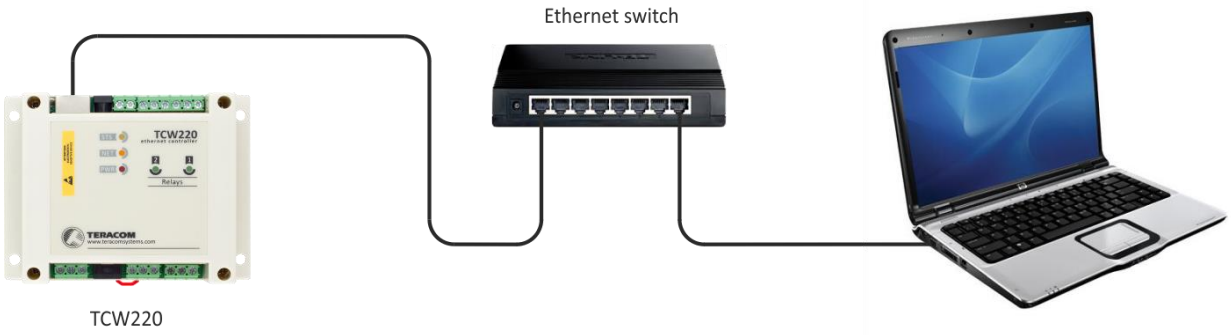
For loads with higher switchable current/voltage than specified, an external relay should be used.



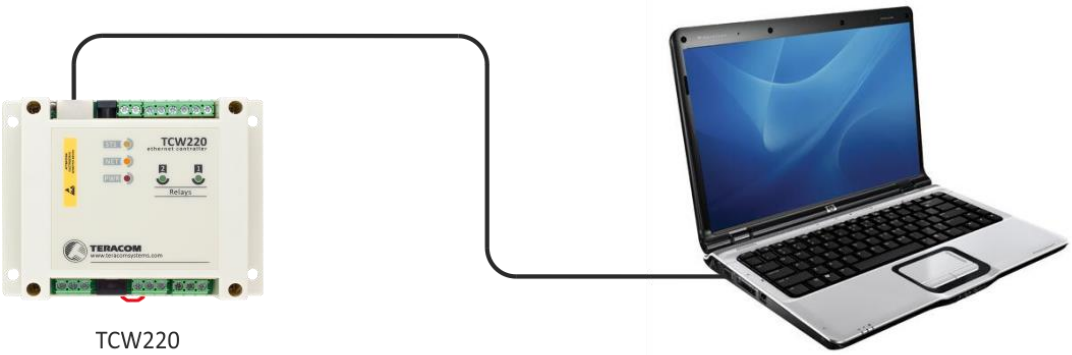
When mechanical relays switch inductive loads such as motors, transformers, relays, etc., the current will arc across the relay contacts each time the contacts open. Over time, this cause wears on the relay contacts which shorten their life. When switching an inductive load, it is recommended that relay contact protection devices are used.

6.2.6. Network connection

The Ethernet port of TCW220 should be connected to 10/100 Base-T Ethernet hub, switch or router.



For configuration, TCW220 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.



TCW220

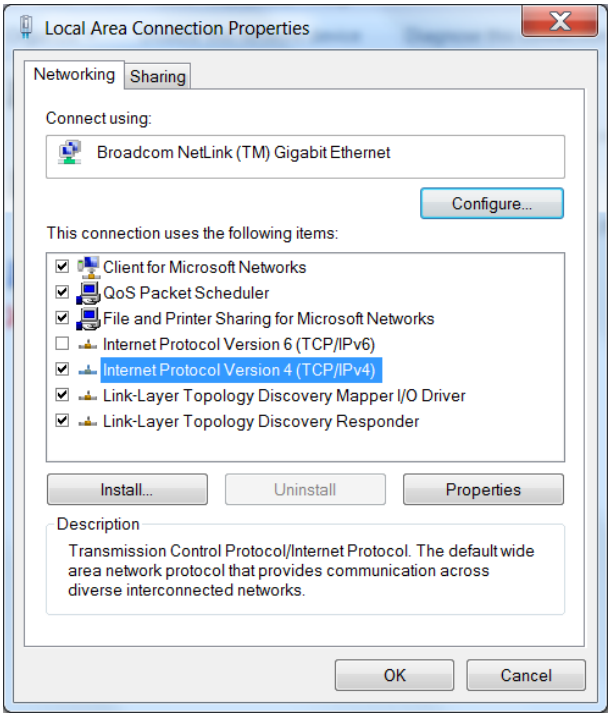
TCW220 can be used in a wireless network by connecting through a wireless router.

6.3. Communication setup

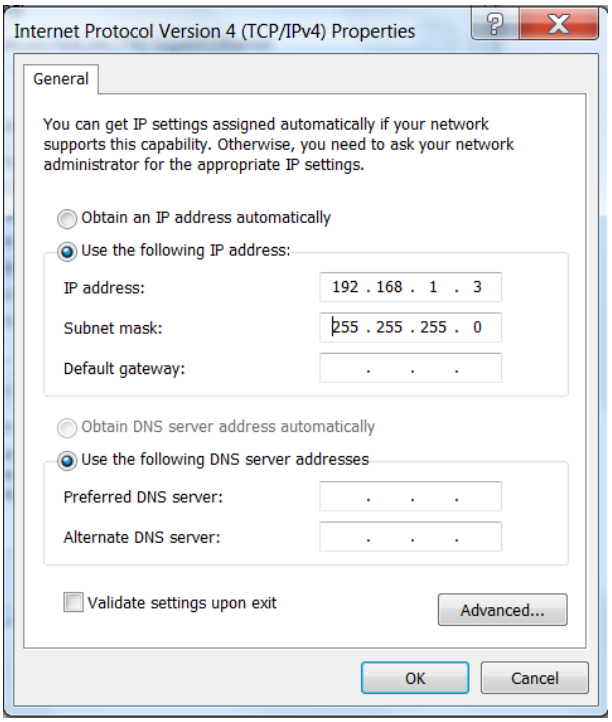
By default TCW220 is delivered with the following network settings:

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

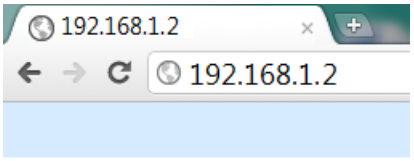
Communication with TCW220 can be established by assigning a temporary IP address to the computer. For computers with Windows OS assigning of IP address is made in “Local area connection properties”:



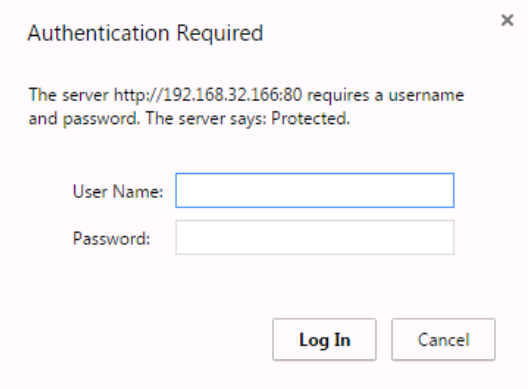
This address should be on the same network - for example 192.168.1.3:



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

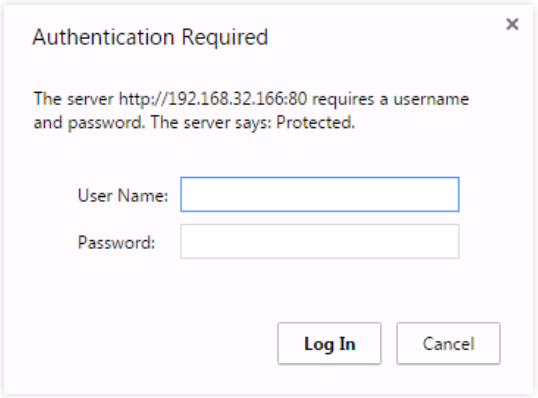
The WEB interface allows configuration, monitoring, and control. All pages are UTF-8 encoded.

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 sessions.

7.1. Monitoring

This menu allows monitoring all parameters as data (Monitoring->Data) and graphics (Monitoring->Graph).

Data page displays the current state of TCW220.

The page has 4 sections – “Sensors”, “Digital inputs”, “Analog inputs” and “Relays”. All they can be added/removed from monitoring page independently by appropriate setup - see “Setup-System-Display” section.

For every parameter (sensor, input, relay) there is a description of up to 15 characters. Default ones can be changed in “Setup-Input/Output”.

The Monitoring 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.

Graph pages can display all logged data in 2D. There are 4 graph pages. Every graph page can show up to 4 parameters with up to 2 dimensions.

7.1.1. Data - sensors section

All detected 1-Wire sensors are shown in this section.

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

For every sensor, there are a description, value, and ID information.

Teracom 1-Wire sensors readings are shown in the Value 1 column. Dual sensors such as (TSH2xx) temperature/humidity sensors have the 2nd parameter shown in the Value 2 column.

It is possible to lock sensors 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 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 TCW220 is used as a part of monitoring and control system managed either by SNMP or HTTP API commands.

For some sensors “Unit”, “Multiplier” and “Offset” can be set in section “Setup-Input/Output”.

Pos	Description	Value 1	Value 2	ID	Lock
1	S1:TST1xx	23.188°C	------	[2867895F07000058]	☐
2	S2:TSH2xx	24.375°C	33.313%RH	[015225B71700FF45]	☐
3	S3	------	------	[0000000000000000]	☐
4	S4	------	------	[0000000000000000]	☐
5	S5	------	------	[0000000000000000]	☐
6	S6	------	------	[0000000000000000]	☐
7	S7	------	------	[0000000000000000]	☐
8	S8	------	------	[0000000000000000]	☐

Scan for new sensors

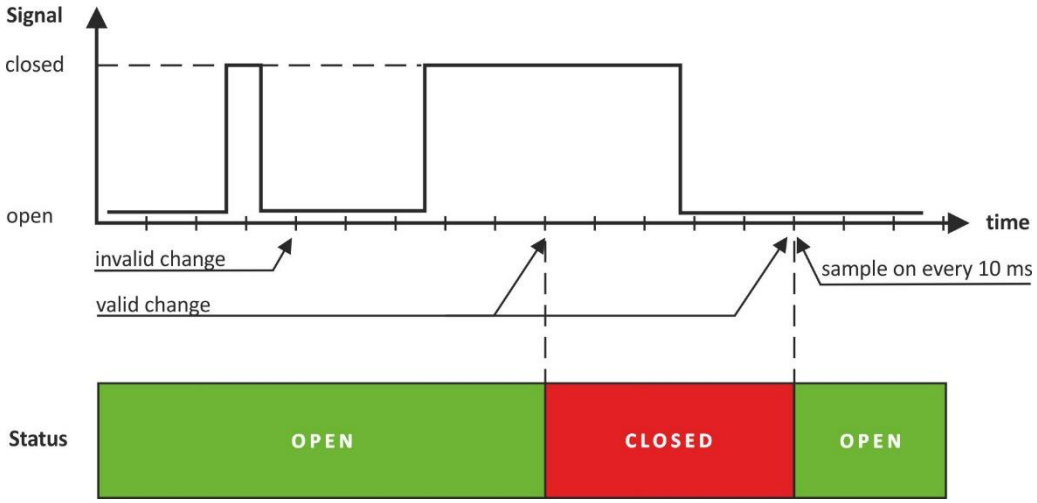
7.1.2. Data - digital inputs section

Digital inputs can be used for monitoring the state of discrete devices – motion sensor, door contact, relay contact, alarm output etc. All digital inputs are not galvanic isolated.

One side of the contact is connected to “Digital In” and another side is connected to “GND” pins.

Digital inputs are sampled every 10mS. The change of input status is considered valid if the same value is read in 3 consecutive samples (30mS) and low-to-high/high-to-low delays

(Setup->Conditions) are zero.



Status of every input is shown by text and by color. The color is red if the input is in an alarm condition.

Digital input	Status	Digital input	Status
Digital Input 1	CLOSED	Digital Input 2	OPEN

Default descriptions can be changed on “Setup->Input/Output” page.

7.1.3. Data - analog inputs section

Analog inputs can be used for monitoring of analog sensors with 0/5 or 0/10 voltage outputs.

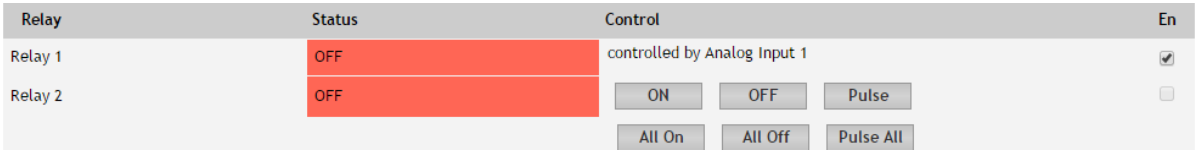
All analog inputs are not galvanic isolated.

Analog input	Value	Analog input	Value
Analog Input 1	0.031V	Analog Input 2	5.046V

For every analog input “Unit”, “Multiplier” and “Offset” can be set in section “Setup->Input/Output”.

7.1.4. Data - relays section

The section displays the current state of relays and presents buttons that can be used to change their status.



Each relay can be activated either remotely by WEB, SNMP, HTTP API, and MODBUS or locally, from the status of a monitored parameter (1 Wire sensor, analog voltage, and dry contact).

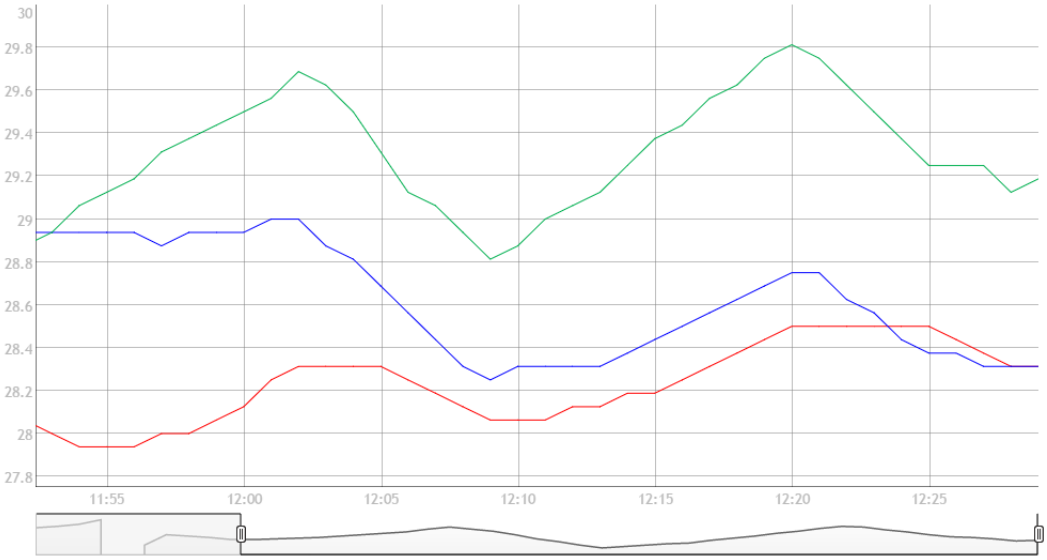
For WEB control every relay has “On”, “Off” and “Pulse” buttons. There are also “All On”, “All Off” and “Pulse All” for common control of relays. Pulse duration in seconds can be set separately for each relay in “Setup-Input/Output-RelayOutputs”.

For locally activated relays a text description of the controlling parameter is displayed rather than buttons. Parameters for local relay activation can be set in “Setup-Input/Output-RelayOutputs”. Control of relays follows conditions set in “Setup-Alarm conditions”.

For every locally activated relay, there is checkbox “En”. It allows temporarily to turn off the automatic control, make manual changes by buttons and then again return to automatic control. By default, this checkbox is turned off.

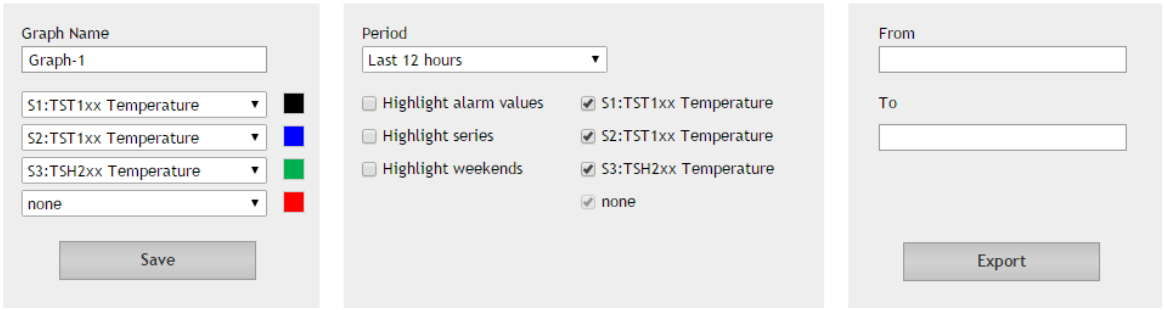
7.1.5. Graph

Every graph page can display up to 4 parameters (including relay status) with up to 2 different dimension.



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

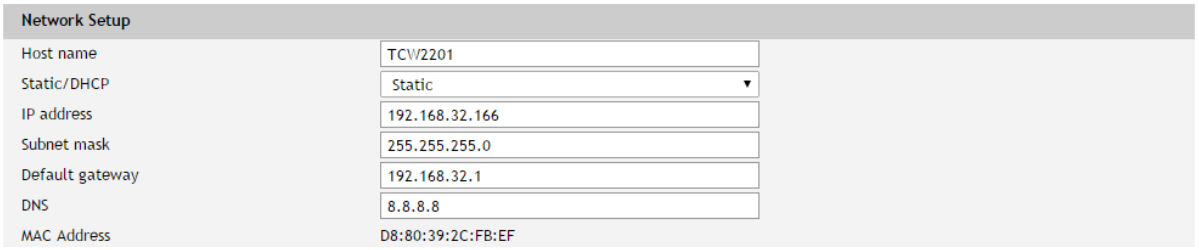
Export of monitored can be made from the page.



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 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.

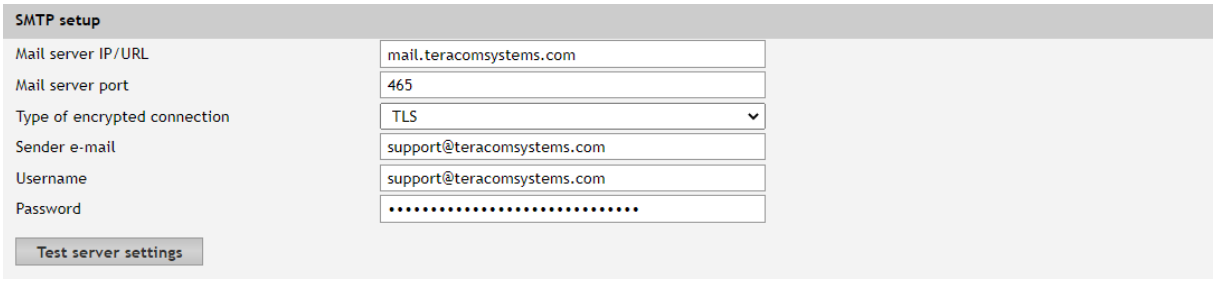
The "Host name" is up to 15 characters. It is shown in search results of TCW discoverer.

It is recommended to use public DNS server (8.8.8.8, 8.8.4.4 etc.) rather than default gateway.

7.2.2. SMTP

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

7.2.2.1. 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.

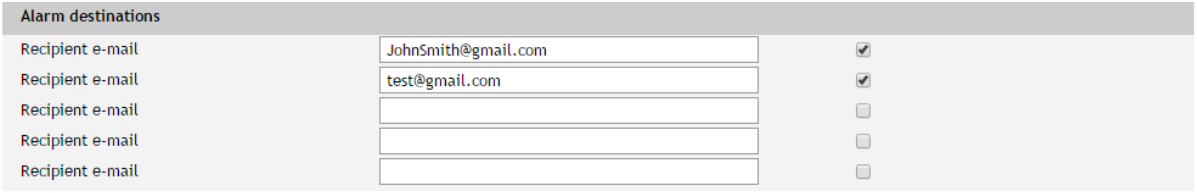
Sender e-mail, username, and password are standard authentication details. For most SMTP servers, the sender's e-mail and username are the same.

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.

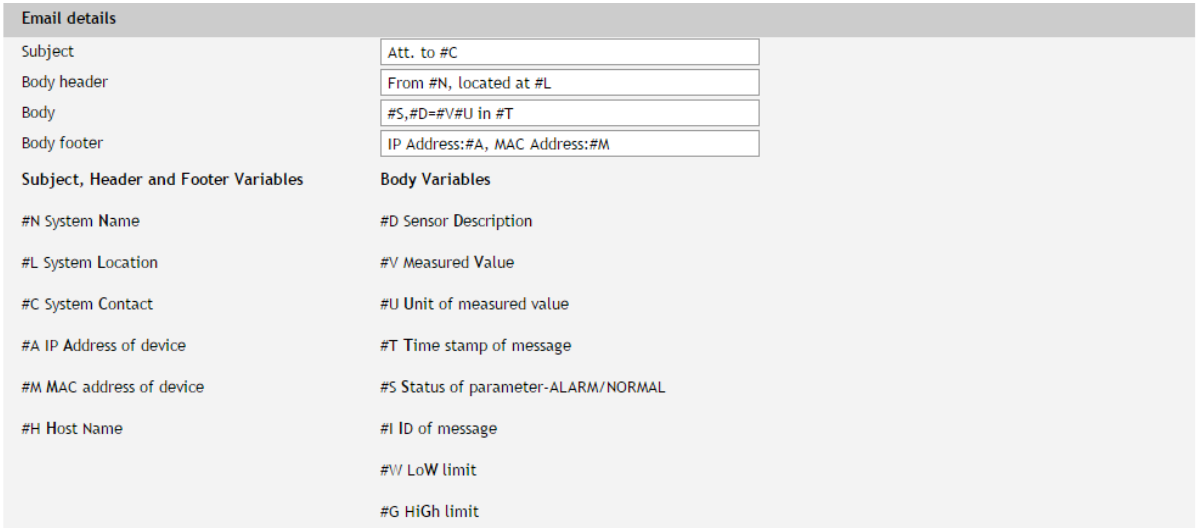
7.2.2.2. Alarm destination



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

7.2.2.3. 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.



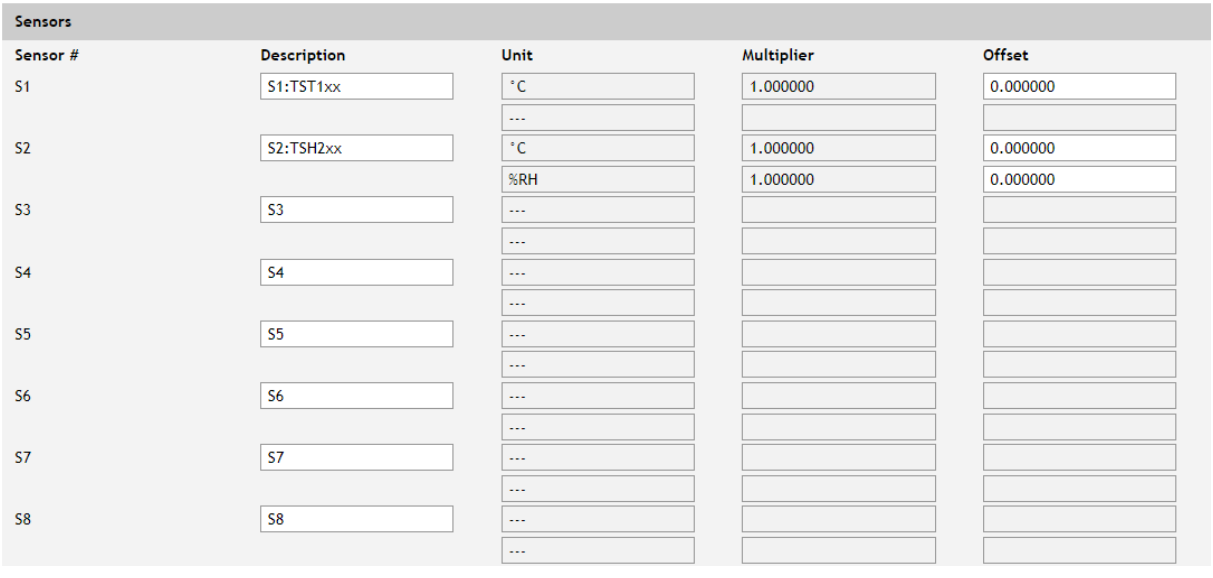
7.2.3. Input/Output

7.2.3.1. 1-Wire sensors

For every 1-Wire 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 some specific sensors (TSA200, TSV200 etc.) fields “Unit” and “Multiplier” are also available.



7.2.3.2. Digital inputs

For every digital input, a description up to 15 characters can be set.

The text, written in “Low level” and “High level” is displayed on monitoring page for this input. These fields accept up to 15 characters.

Digital inputs			
Input #	Description	Low level	High level
DI1	Digital Input 1	CLOSED	OPEN
DI2	Digital Input 2	CLOSED	OPEN

7.2.3.3. Analog inputs

For every analog input, a description up to 15 characters can be set.

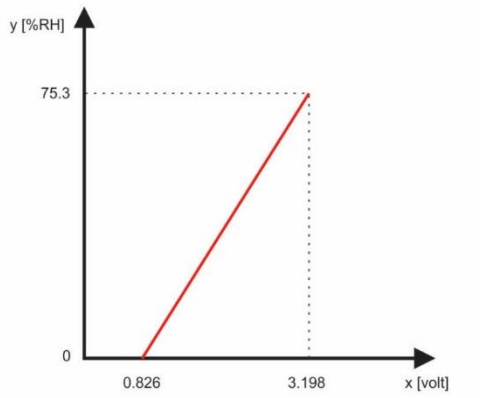
Analog inputs				
Input #	Description	Unit	Multiplier	Offset
AI1	Server room	%RH	31.740	0.8260
AI2	Analog Input 2	V	1.000	0.0000

For every analog input, fields “Unit”, “Multiplier” and “Offset” are available to convert the raw voltage/current into meaningful engineering units. The scaled value is calculated by:

SV[Un] = (RV - OF) * MU
Where:
SV – scaled (displayed) value;
Un – unit;
RV – raw voltage from the source;
MU – multiplier;
OF – offset.

Example:
For humidity sensor HIH-4000-003 following data (from the datasheet) is available:

VOUT = 0.826 at 0% RH
VOUT = 3.198 at 75.3% RH



The sensor provides raw voltage values as output, but what we actually need is the corresponding relative humidity values. To achieve this, we use a multiplier and an offset. These two parameters allow us to calculate the relative humidity for any given voltage within the sensor's working range.

The multiplier (MU) is determined by the ratio of the change in relative humidity (ΔRH%) to the change in voltage (ΔV). In geometric terms, this is akin to finding the slope of a line. For this particular sensor, the line is described by the equation ΔRH%/ΔV. We can calculate the multiplier as follows:

MU = (75.3 - 0) / (3.198 - 0.826) = 75.3 / 2.372 = 31.745 %RH/V

The offset (OF) is calculated using the multiplier and the relation between one of the known points. By substituting the scaled value (SV) and the corresponding raw value (RV) into the equation SV = (RV - OF) * MU, we can solve for the offset:

OF = RV - SV/MU

Using the point where SV = 0 and RV = 0.826, we find:

OF = 0.826 - (0 / 31.745) = 0.826 - 0 = 0.826

Similarly, we can calculate the offset using the other point, where SV = 75.3 and RV = 3.198:

OF = 3.198 - (75.3 / 31.745) = 3.198 - 2.372 = 0.826

Therefore, the formula for this sensor becomes:

SV = (RV – 0.826) * 31.745

To verify the accuracy of this formula, let's check the case where VOUT = 0.826 V (0%RH):

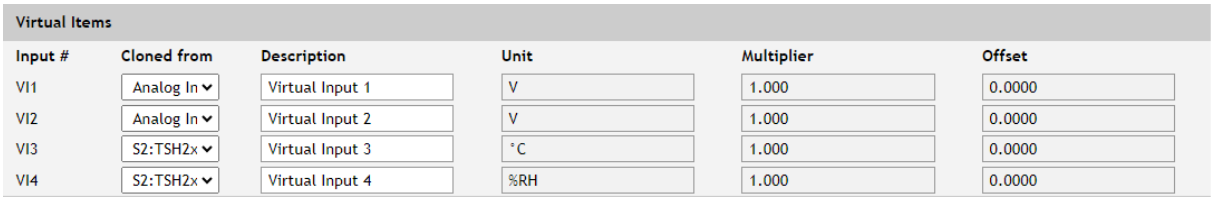
SV = (0.826 – 0.826) * 31.745 = 0 *31.745 = 0 %RH

This confirms that the formula correctly converts the voltage to the corresponding relative humidity value

By default and after "Factory default settings" procedure:

Unit - V
Offset - 0.00
Multiplier - 1.00

7.2.3.4. Virtual inputs



Virtual item is an additional feature that gives the ability to clone a monitored parameter - analog input or 1-Wire sensor.

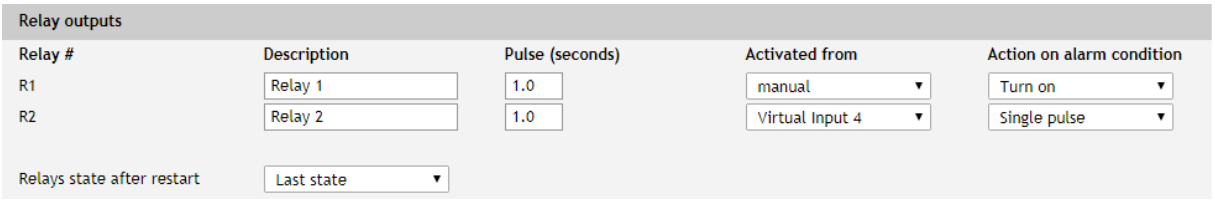
For the virtual item, different alarm borders from the original can be set. In this way, more alarm borders (alarm notifications) can be organized for the same parameter.

The values of Unit, Multiplier, and Offset are presented for information only. They are inherited from the original parameter and can't be edited.

Virtual items can be used for alarm notifications, in Functions, and for local relay activation.

7.2.3.5. Relay outputs

For every relay, a description up to 15 characters can be set.



For every relay different time for pulse duration can be set. The resolution is 0.1 second.

Every relay can be activated remotely or locally – by the value of the monitored parameter.

By default, all relays are activated remotely and in the field "Activated from" is written "manual".

For local activation, alarm conditions for different sources are used. They are set up in section "Setup-Alarm conditions". To assign a parameter to relay, following choices are possible:

- Sxy – “S” stands for “**S**ensor 1-Wire”, “x” is a number from 1 to 8, “y” is a number from 1 to 2. The relay is activated from a value measured from specified 1-Wire sensor and rules for ranges specified in “Setup-Alarm conditions”;
- “Analog input z” - the relay is activated from a value measured from specified analog input and rules for ranges specified in “Setup-Alarm conditions”; z is number from 1 to 2;
- “Virtual input z” - the relay is activated from a value from specified virtual item (cloned analog input or 1-Wire sensor) and rules for ranges specified in “Setup->Alarm conditions”; z is number from 1 to 4;
- “Digital input z” - the relay follows the state of specified digital input; z is number from 1 to 2;
- Any alarm - the relay is activated from any of set alarm conditions.

7.2.4. Conditions

This section is used for parameterization of the trigger and alert conditions for 1-Wire sensors, analog inputs, virtual items and digital inputs.

7.2.4.1. 1-Wire sensors, analog inputs and virtual items

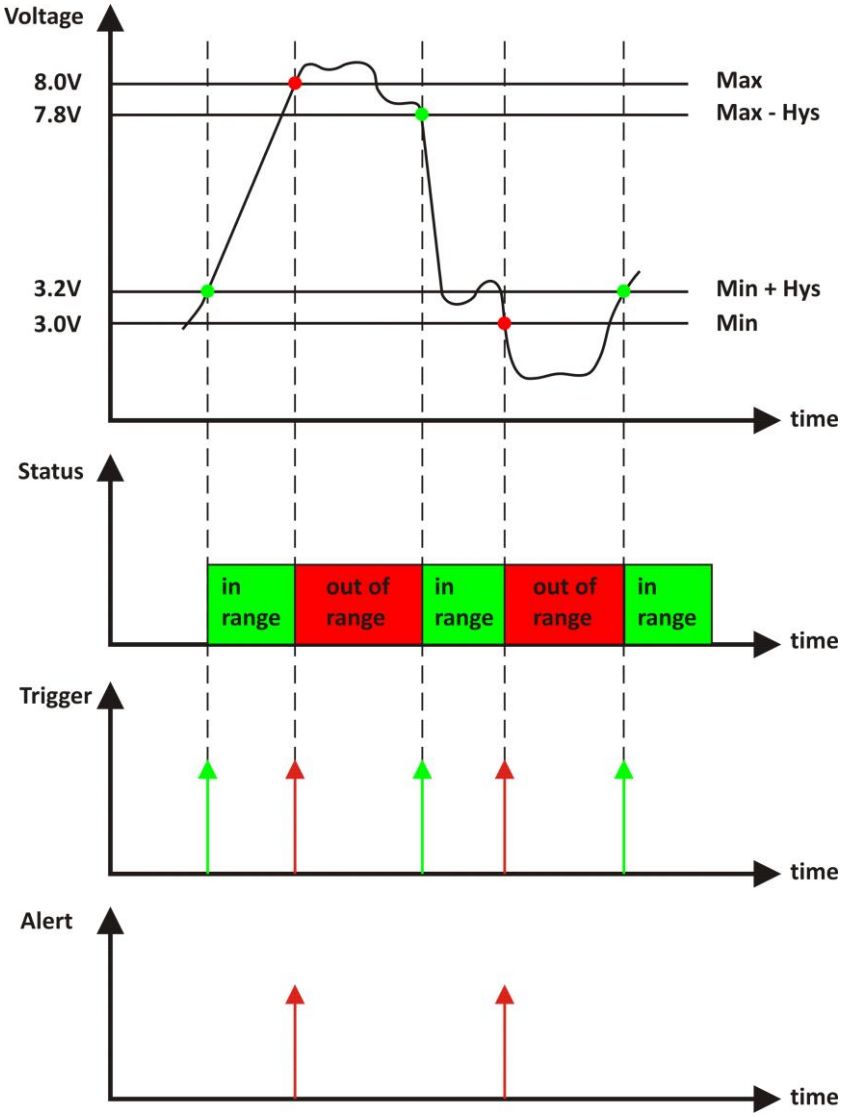
For every sensor two type of fields are presented – one is for setup of trigger conditions (“Min”, “Max” and “Hys.”) and another for alert notification (“If out of range”).

Sensors						If out of range		
#	Description	Type	Min.	Max.	Hys.	mail	trap	post
1	S1:TST1xx	Temperature, °C	-40.000	85.000	8.500	☑	☑	☑
		---	---	---	---	☐	☐	☐
2	S2:TSH2xx	Temperature, °C	-40.000	85.000	8.500	☑	☑	☑
		Humidity, %RH	0.000	100.000	10.000	☑	☑	☑
3	S3	---	---	---	---	☐	☐	☐
		---	---	---	---	☐	☐	☐
4	S4	---	---	---	---	☐	☐	☐
		---	---	---	---	☐	☐	☐
5	S5	---	---	---	---	☐	☐	☐
		---	---	---	---	☐	☐	☐
6	S6	---	---	---	---	☐	☐	☐
		---	---	---	---	☐	☐	☐
7	S7	---	---	---	---	☐	☐	☐
		---	---	---	---	☐	☐	☐
8	S8	---	---	---	---	☐	☐	☐
		---	---	---	---	☐	☐	☐
			Notification in case of a sensor communication lost			☑	☑	☑
Return notification		☑	Notification delay (seconds)		3	(0-3600)		
Analog inputs						If out of range		
#	Description	Dimension	Min.	Max.	Hys.	mail	trap	post
1	Analog Input 1	V	0.000	10.000	0.100	☐	☐	☐
2	Analog Input 2	V	0.000	10.000	0.100	☐	☐	☐
Return notification		☐	Notification delay (seconds)		0	(0-3600)		
Virtual items						If out of range		
#	Description	Dimension	Min.	Max.	Hys.	mail	trap	post
1	Virtual Input 1	V	4.500	5.500	0.010	☐	☐	☐
2	Virtual Input 2	V	6.000	8.000	0.100	☐	☐	☐
3	Virtual Input 3	°C	25.000	35.000	0.010	☐	☐	☐
4	Virtual Input 4	%RH	45.000	55.000	0.100	☐	☐	☐
Return notification		☐	Notification delay (seconds)		0	(0-3600)		

“Min” and “Max” indicate the border of 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 the trigger point.



Example:

TCW220, TST100, and appropriate heater are used to control the 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.

For Relay1 local activation from Sensor1 is set.

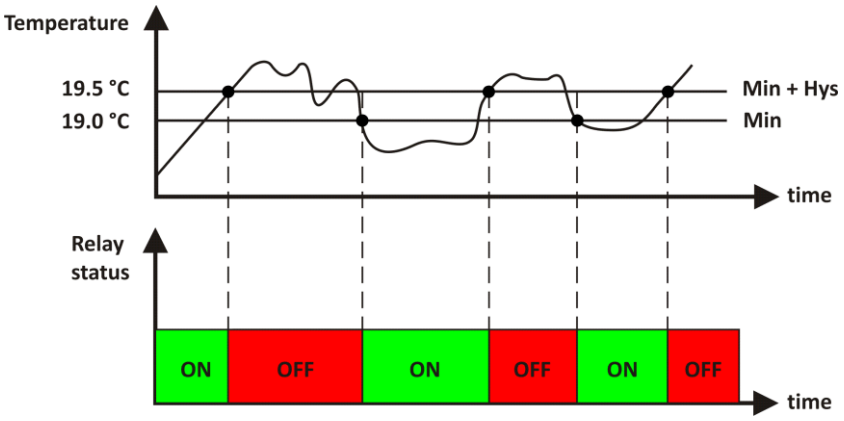
Following parameters are set for Sensor1: Min=19, Max=100 and Hys=0.5.

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

When the controller is switched on, Relay1 is immediately activated because the monitored temperature is out of range. This switches the heater on. The temperature is going higher.

When temperature reaches 19.5°C (19.0 + 0.5) it goes in range (trigger condition) and Relay1 is deactivated. The heater is switched off.

The temperature falls and when it reached 19°C it goes out of range (trigger and alert conditions). The relay is activated (heater is switched on) and e-mail is sent.



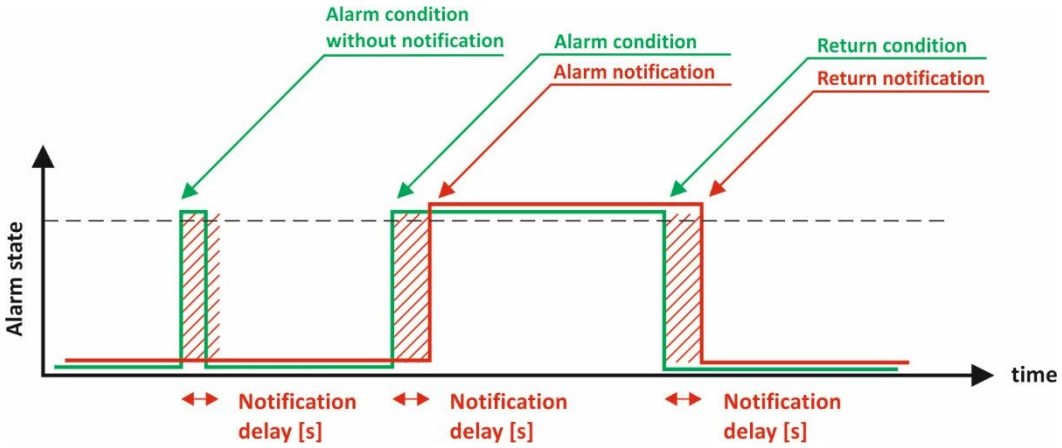
The “Max” value is set far enough from the wanted temperature to avoid trigger/alert conditions around it.

For every sensor or analog input, there are 3 independent ways of alert when there is an alarm condition – e-mail, SNMP trap, and post (HTTP/HTTPS post of XML file). Each alarm notification method is activated by 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 and for all analog inputs, 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 and for all analog inputs, there is a “Notification delay” parameter. It is very useful like a filter for short alarm conditions.

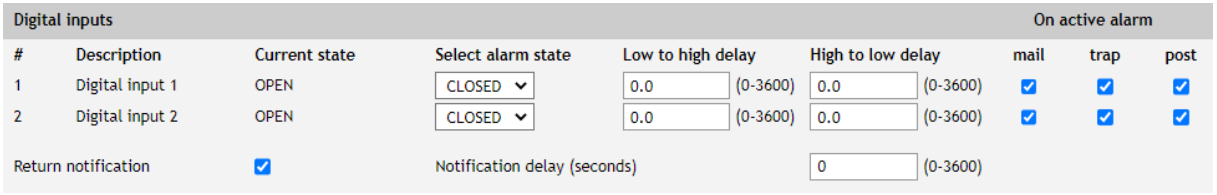


7.2.4.2. Digital inputs

For every digital input, the alarm state should be chosen – Open or Close. When the input goes in alarm state 3 independent way of alert are possible – e-mail, SNMP trap, and post (HTTP/HTTPS post of XML file). Globally for all digital inputs, there is a checkbox “Return notification”. If this option is chosen there will be notification also when parameter returns in range.

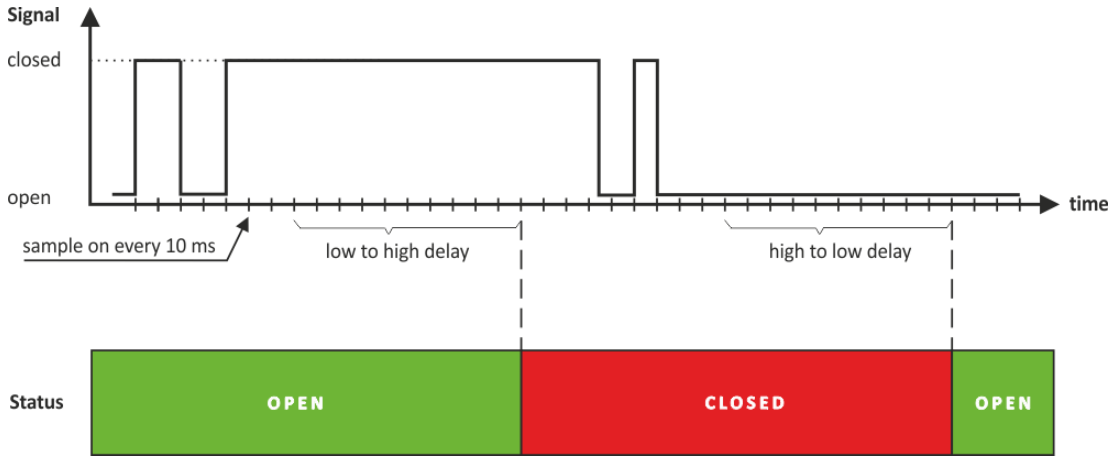
Globally for all digital inputs, there is “Notification delay” parameter. It is very useful like a filter for short alarm conditions.

In the time when the input is in an alarm state, on Monitoring page appropriate input will be colored in red.



There are two delays - low-to-high and high-to-low for digital input change. These delays are added to the standard delay of 30mS. They have 100mS resolution and by default are zero.

These options can be used for additional debouncing.



On the picture above low-to-high and high-to-low delays are set to 0.1 seconds.

7.2.5. System

On this page, some general settings can be made.

7.2.5.1. 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.

7.2.5.2. HTTP API

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

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

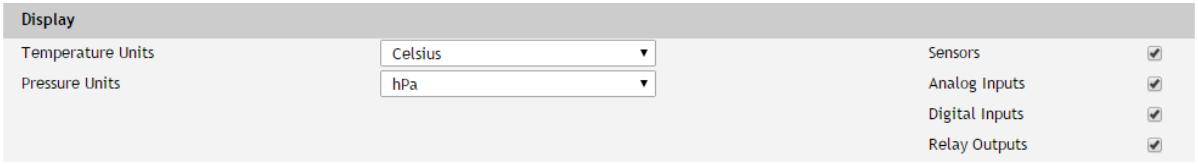
7.2.5.3. Monitoring page automatic refresh

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

7.2.5.4. Display

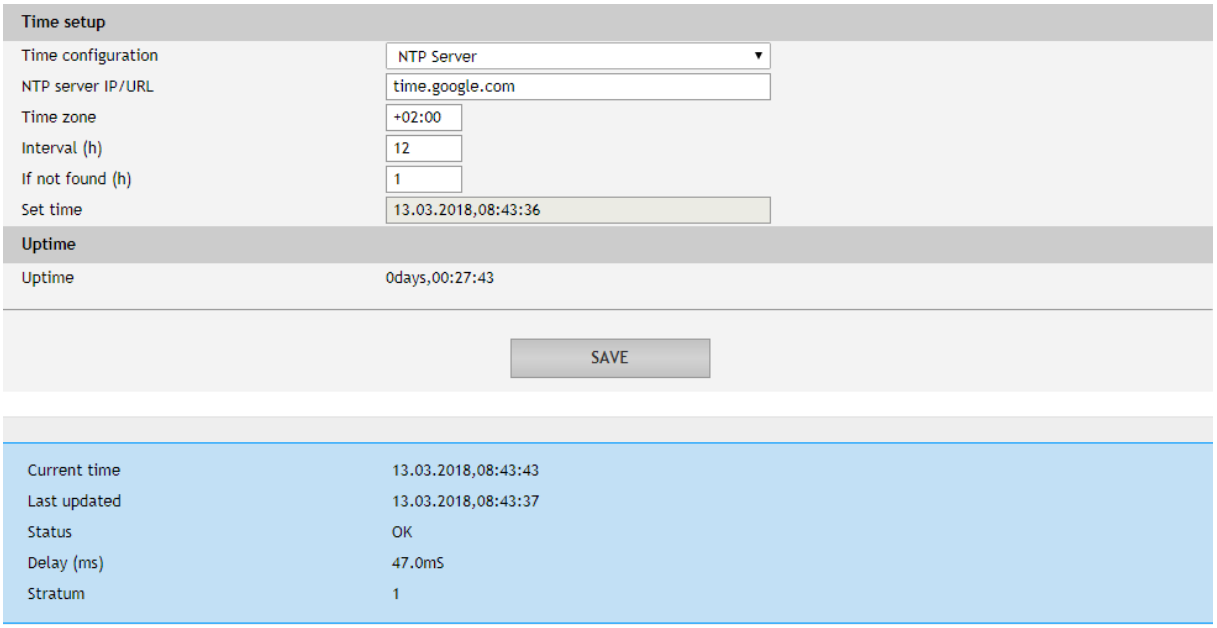
The unit for observed temperatures can be selected from different scales.

All four sections on “Monitoring page” can be added or removed independently by appropriate setup here.



7.2.6. NTP

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 +00:00 and interval of 12 hours.

7.3. Services

7.3.1. Modbus

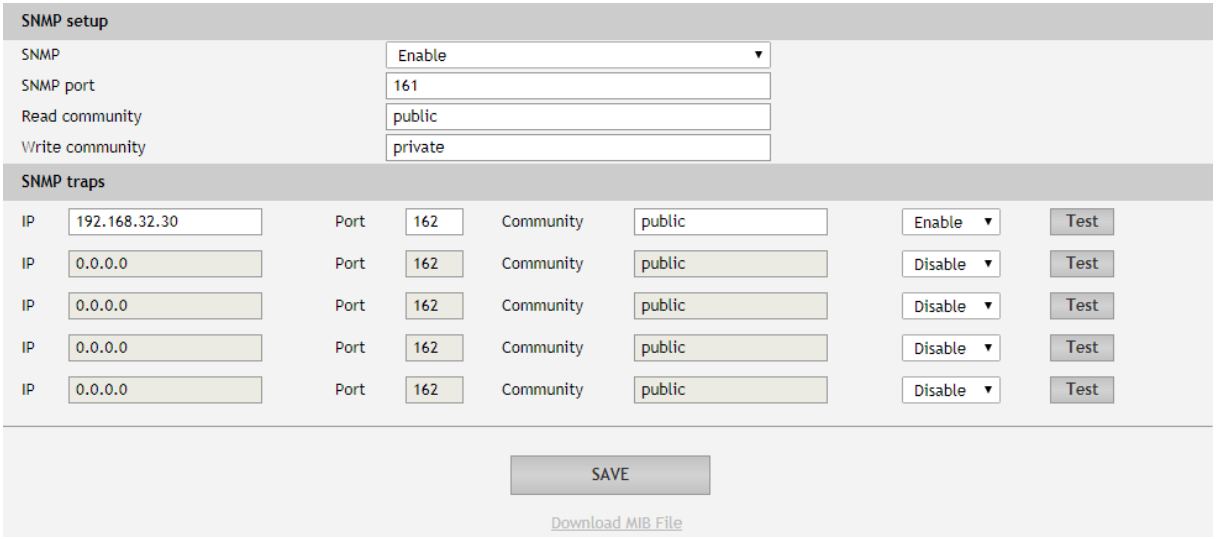
TCW220 supports Modbus TCP/IP over the Ethernet interface.

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

SNMP

The TCW220 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:

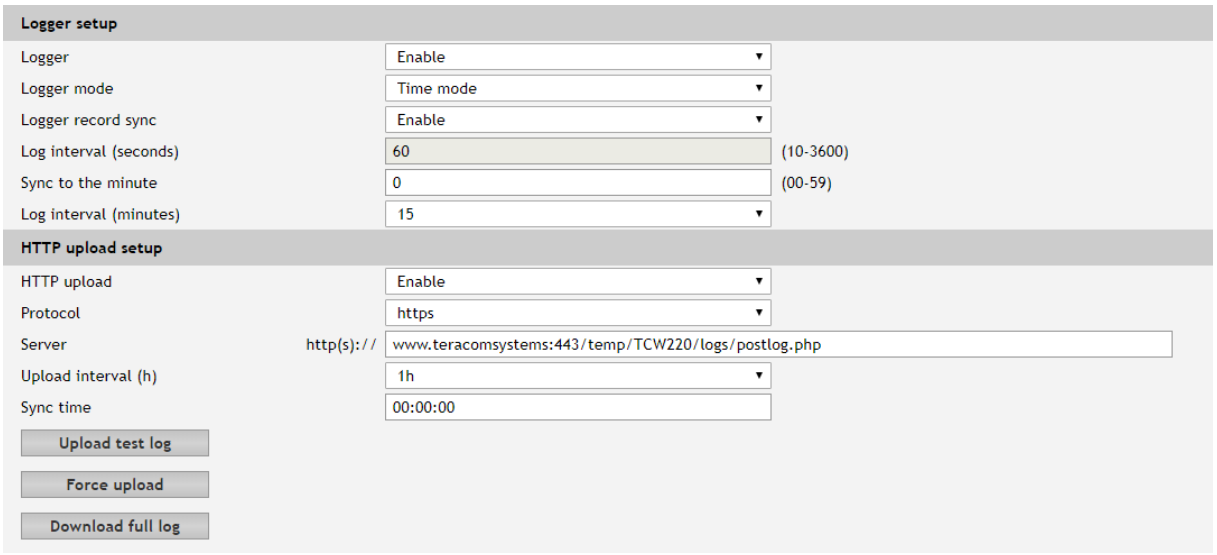
- event occurs (status change) on Digital Inputs;
- measured parameter on Analog Inputs goes outside the range;
- measured parameter on the 1-Wire bus goes outside the range;
- restart condition.

SNMP trap is sent after reset.

Actual MIB file can be downloaded from here.

7.3.2. Logger

The TCW220 supports logger for all monitored parameters and status of relays.



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.

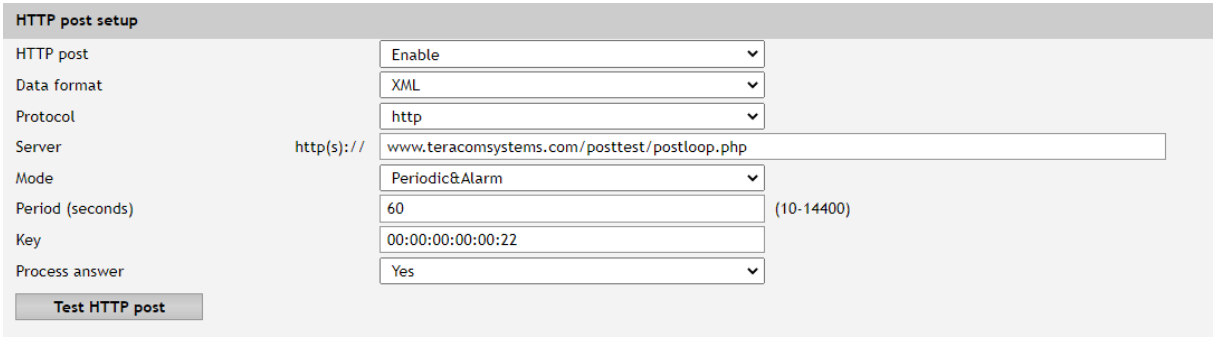
By default, the logger is disabled.

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

7.3.3. HTTP post

TCW220 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 period of the post 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.

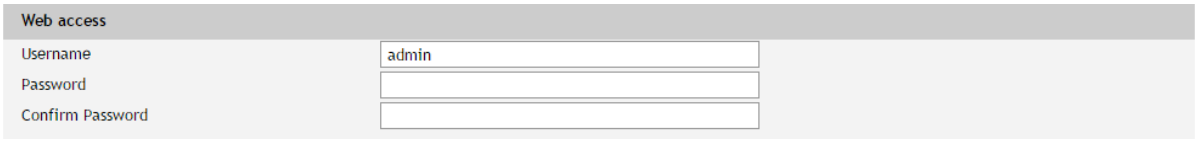
If “Process Answer” option is enabled, the TCW220 will process the answer of the remote server. The list of valid commands is described in section “HTTP API commands”.

7.4. Administration

7.4.1. User/Pass

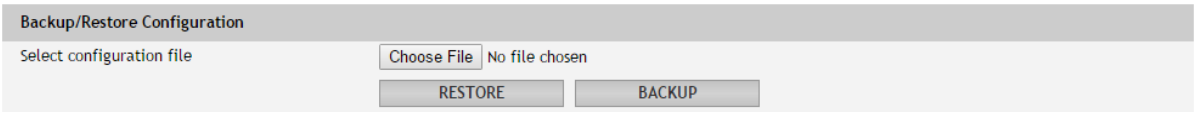
The TCW220 supports one user only. It has administrative rights.

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



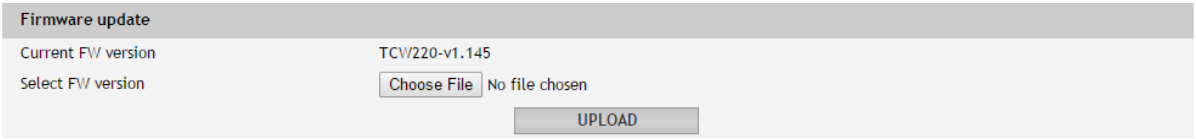
7.4.2. Backup/Restore

The TCW220 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 TCW220 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 TCW220 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 TCW220 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.2.1.1.0	name	read-only	Device name	DisplayString
x.2.1.2.0	version	read-only	Firmware version	DisplayString
x.2.1.3.0	date	read-only	Release date	DisplayString

setup -> network

OID	Name	Access	Description	Syntax
x.2.2.1.1.0	deviceID	read-only	Device ID (default MAC address)	MacAddress
x.2.2.1.2.0	hostName	read-only	Hostname	DisplayString
x.2.2.1.3.0	deviceIP	read-only	Device IP address	IpAddress

setup -> io -> sensorsSetup -> sensor1setup

OID	Name	Access	Description	Syntax
x.2.2.2.1.1.1.0	s1description	read-write	Sensor 1 description	DisplayString
x.2.2.2.1.1.2.1.0	s11MAXInt	read-write	S11 maximum value x1000 in Integer format	Integer32
x.2.2.2.1.1.2.2.0	s11MINInt	read-write	S11 minimum value x1000 in Integer format	Integer32
x.2.2.2.1.1.2.3.0	s11HYSTInt	read-write	S11 hysteresis value x1000 in Integer format	Integer32
x.2.2.2.1.1.3.1.0	s12MAXInt	read-write	S12 maximum value x1000 in Integer format	Integer32
x.2.2.2.1.1.3.2.0	s12MINInt	read-write	S12 minimum value x1000 in Integer format	Integer32
x.2.2.2.1.1.3.3.0	s12HYSTInt	read-write	S12 hysteresis value x1000 in Integer format	Integer32

setup -> io -> sensorsSetup -> sensor2setup

OID	Name	Access	Description	Syntax
x.2.2.2.1.2.1.0	s2description	read-write	Sensor2 description	DisplayString
x.2.2.2.1.2.2.1.0	s21MAXInt	read-write	s21 maximum value x1000 in Integer format	Integer32
x.2.2.2.1.2.2.2.0	S21MINInt	read-write	S21 minimum value x1000 in Integer format	Integer32
x.2.2.2.1.2.2.3.0	S21HYSTInt	read-write	S21 hysteresis value x1000 in Integer format	Integer32
x.2.2.2.1.2.3.1.0	S22MAXInt	read-write	S22 maximum value x1000 in Integer format	Integer32
x.2.2.2.1.2.3.2.0	S22MINInt	read-write	S22 minimum value x1000 in Integer format	Integer32
x.2.2.2.1.2.3.3.0	S22HYSTInt	read-write	S22 hysteresis value x1000 in Integer format	Integer32

setup -> io -> sensorsSetup -> sensor3setup

OID	Name	Access	Description	Syntax
x.2.2.2.1.3.1.0	S3description	read-write	Sensor 3 description	DisplayString
x.2.2.2.1.3.2.1.0	S31MAXInt	read-write	S31 maximum value x1000 in Integer format	Integer32
x.2.2.2.1.3.2.2.0	S31MINInt	read-write	S31 minimum value x1000 in Integer format	Integer32
x.2.2.2.1.3.2.3.0	S31HYSTInt	read-write	S31 hysteresis value x1000 in Integer format	Integer32
x.2.2.2.1.3.3.1.0	S32MAXInt	read-write	S32 maximum value x1000 in Integer format	Integer32
x.2.2.2.1.3.3.2.0	S32MINInt	read-write	S32 minimum value x1000 in Integer format	Integer32
x.2.2.2.1.3.3.3.0	S32HYSTInt	read-write	S32 hysteresis value x1000 in Integer format	Integer32

setup -> io -> sensorsSetup -> sensor4setup

OID	Name	Access	Description	Syntax
x.2.2.2.1.4.1.0	S4description	read-write	Sensor 4 description	DisplayString
x.2.2.2.1.4.2.1.0	S41MAXInt	read-write	S41 maximum value x1000 in Integer format	Integer32
x.2.2.2.1.4.2.2.0	S41MINInt	read-write	S41 minimum value x1000 in Integer format	Integer32
x.2.2.2.1.4.2.3.0	S41HYSTInt	read-write	S41 hysteresis value x1000 in Integer format	Integer32
x.2.2.2.1.4.3.1.0	S42MAXInt	read-write	S42 maximum value x1000 in Integer format	Integer32
x.2.2.2.1.4.3.2.0	S42MINInt	read-write	S42 minimum value x1000 in Integer format	Integer32
x.2.2.2.1.4.3.3.0	S42HYSTInt	read-write	S42 hysteresis value x1000 in Integer format	Integer32

setup -> io -> sensorsSetup -> sensor5setup

OID	Name	Access	Description	Syntax
x.2.2.2.1.5.1.0	S5description	read-write	Sensor 5 description	DisplayString
x.2.2.2.1.5.2.1.0	S51MAXInt	read-write	S51 maximum value x1000 in Integer format	Integer32
x.2.2.2.1.5.2.2.0	S51MINInt	read-write	S51 minimum value x1000 in Integer format	Integer32

x.2.2.2.1.5.2.3.0	S51HYSTInt	read-write	S51 hysteresis value x1000 in Integer format	Integer32
x.2.2.2.1.5.3.1.0	S52MAXInt	read-write	S52 maximum value x1000 in Integer format	Integer32
x.2.2.2.1.5.3.2.0	S52MINInt	read-write	S52 minimum value x1000 in Integer format	Integer32
x.2.2.2.1.5.3.3.0	S52HYSTInt	read-write	S52 hysteresis value x1000 in Integer format	Integer32

setup -> io -> sensorsSetup -> sensor6setup

OID	Name	Access	Description	Syntax
x.2.2.2.1.6.1.0	S6description	read-write	Sensor 6 description	DisplayString
x.2.2.2.1.6.2.1.0	S61MAXInt	read-write	S61 maximum value x1000 in Integer format	Integer32
x.2.2.2.1.6.2.2.0	S61MINInt	read-write	S61 minimum value x1000 in Integer format	Integer32
x.2.2.2.1.6.2.3.0	S61HYSTInt	read-write	S61 hysteresis value x1000 in Integer format	Integer32
x.2.2.2.1.6.3.1.0	S62MAXInt	read-write	S62 maximum value x1000 in Integer format	Integer32
x.2.2.2.1.6.3.2.0	S62MINInt	read-write	S62 minimum value x1000 in Integer format	Integer32
x.2.2.2.1.6.3.3.0	S62HYSTInt	read-write	S62 hysteresis value x1000 in Integer format	Integer32

setup -> io -> sensorsSetup -> sensor7setup

OID	Name	Access	Description	Syntax
x.2.2.2.1.7.1.0	S7description	read-write	Sensor 7 description	DisplayString
x.2.2.2.1.7.2.1.0	S71MAXInt	read-write	S71 maximum value x1000 in Integer format	Integer32
x.2.2.2.1.7.2.2.0	S71MINInt	read-write	S71 minimum value x1000 in Integer format	Integer32
x.2.2.2.1.7.2.3.0	S71HYSTInt	read-write	S71 hysteresis value x1000 in Integer format	Integer32
x.2.2.2.1.7.3.1.0	S72MAXInt	read-write	S72 maximum value x1000 in Integer format	Integer32
x.2.2.2.1.7.3.2.0	S72MINInt	read-write	S72 minimum value x1000 in Integer format	Integer32
x.2.2.2.1.7.3.3.0	S72HYSTInt	read-write	S72 hysteresis value x1000 in Integer format	Integer32

setup -> io -> sensorsSetup -> sensor8setup

OID	Name	Access	Description	Syntax
x.2.2.2.1.8.1.0	S8description	read-write	Sensor 8 description	DisplayString
x.2.2.2.1.8.2.1.0	S81MAXx10Int	read-write	S81 maximum value x1000 in Integer format	Integer32
x.2.2.2.1.8.2.2.0	S81MINx10Int	read-write	S81 minimum value x1000 in Integer format	Integer32
x.2.2.2.1.8.2.3.0	S81HYSTx10Int	read-write	S81 hysteresis value x1000 in Integer format	Integer32
x.2.2.2.1.8.3.1.0	S82MAXx10Int	read-write	S82 maximum value x1000 in Integer format	Integer32
x.2.2.2.1.8.3.2.0	S82MINx10Int	read-write	S82 minimum value x1000 in Integer format	Integer32
x.2.2.2.1.8.3.3.0	S82HYSTx10Int	read-write	S82 hysteresis value x1000 in Integer format	Integer32

setup -> io -> analogSetup -> analog1setup

OID	Name	Access	Description	Syntax
x.2.2.2.2.1.1.0	voltage1description	read-write	Voltage 1 description	DisplayString
x.2.2.2.2.1.2.0	voltage1max	read-write	Voltage 1 maximum	Integer32
x.2.2.2.2.1.3.0	voltage1min	read-write	Voltage 1 minimum	Integer32
x.2.2.2.2.1.4.0	voltage1hyst	read-write	Voltage 1 hysteresis	Integer32

setup -> io -> analogSetup -> analog2setup

OID	Name	Access	Description	Syntax
x.2.2.2.2.2.1.0	voltage2description	read-write	Voltage 2 description	DisplayString
x.2.2.2.2.2.2.0	voltage2max	read-write	Voltage 2 maximum	Integer32
x.2.2.2.2.2.3.0	voltage2min	read-write	Voltage 2 minimum	Integer32
x.2.2.2.2.2.4.0	voltage2hyst	read-write	Voltage 2 hysteresis	Integer32

setup -> io -> digitalSetup

OID	Name	Access	Description	Syntax
x.2.2.2.3.1.0	digitalInput1description	read-write	Digital Input 1 description"	DisplayString
x.1.2.2.3.2.0	digitalInput2description	read-write	Digital Input 2 description	DisplayString

setup -> io -> relaysSetup -> relay1setup

OID	Name	Access	Description	Syntax
x.2.2.2.4.1.1.0	relay1description	read-write	Relay 1 description	DisplayString
x.2.2.2.4.1.2.0	relay1pulseWidth	read-write	Relay1 Pulse x100ms	Integer32
x.2.2.2.4.1.3.0	relay1controlledBy	read-write	Relay1 control logic	Integer32 { manual(0),sensor11(1),sensor12(2),sensor21(3),sensor22(4), sensor31(5),sensor32(6),sensor41(7),sensor42(8),sensor51(9), sensor52(10),sensor61(11),sensor62(12),sensor71(13),sensor72(14), sensor81(15),sensor82(16),analog1(17),analog2(18), digital1(19),digital2(20),anyAlarm(21), virtual1(22),virtual2(23),virtual3(24),virtual4(25)}

setup -> io-> relaysSetup -> relay2setup

OID	Name	Access	Description	Syntax
x.2.2.2.4.2.1.0	relay2description	read-write	Relay 2 description	DisplayString
x.2.2.2.4.2.2.0	relay2pulseWidth	read-write	Relay 2 Pulse x100ms	Integer32
x.2.2.2.4.2.3.0	relay2controlledBy	read-write	Relay 2 control logic	Integer32 { manual(0),sensor11(1),sensor12(2),sensor21(3),sensor22(4), sensor31(5),sensor32(6),sensor41(7),sensor42(8),sensor51(9), sensor52(10),sensor61(11),sensor62(12),sensor71(13),sensor72(14), sensor81(15),sensor82(16),analog1(17),analog2(18), digital1(19),digital2(20),anyAlarm(21), virtual1(22),virtual2(23),virtual3(24),virtual4(25)}

setup -> io -> virtualSetup -> virtual1setup

OID	Name	Access	Description	Syntax
x.2.2.2.5.1.1.0	virtualInput1description	read-write	Virtual input 1 description	DisplayString
x.2.2.2.5.1.2.0	virtualInput1max	read-write	Virtual input 1 maximum	Integer32
x.2.2.2.5.1.3.0	virtualInput1min	read-write	Virtual input 1 minimum	Integer32
x.2.2.2.5.1.4.0	virtualInput1hyst	read-write	Virtual input 1 hysteresis	Integer32
x.2.2.2.5.1.5.0	virtualInput1Parent	read-write	Virtual input 1 parent	INTEGER{none(0),sensor11(1),sensor12(2),sensor21(3),sensor22(4),sensor31(5),sensor32(6),sensor41(7),sensor42(8),sensor51(9),sensor52(10),sensor61(11),sensor62(12),sensor71(13),sensor72(14), sensor81(15),sensor82(16),analog1(17),analog2(18)}

setup -> io -> virtualSetup -> virtual2setup

OID	Name	Access	Description	Syntax
x.2.2.2.5.2.1.0	virtualInput2description	read-write	Virtual input 2 description	DisplayString
x.2.2.2.5.2.2.0	virtualInput2max	read-write	Virtual input 2 maximum	Integer32
x.2.2.2.5.2.3.0	virtualInput2min	read-write	Virtual input 2 minimum	Integer32
x.2.2.2.5.2.4.0	virtualInput2hyst	read-write	Virtual input 2 hysteresis	Integer32
x.2.2.2.5.2.5.0	virtualInput2Parent	read-write	Virtual input 2 parent	INTEGER{none(0),sensor11(1),sensor12(2),sensor21(3),sensor22(4),sensor31(5),sensor32(6),sensor41(7),sensor42(8),sensor51(9),sensor52(10),sensor61(11),sensor62(12),sensor71(13),sensor72(14),sensor81(15),sensor82(16),analog1(17),analog2(18)}

setup -> io -> virtualSetup -> virtual3setup

OID	Name	Access	Description	Syntax
x.2.2.2.5.3.1.0	virtualInput3description	read-write	Virtual input 3 description	DisplayString
x.2.2.2.5.3.2.0	virtualInput3max	read-write	Virtual input 3 maximum	Integer32
x.2.2.2.5.3.3.0	virtualInput3min	read-write	Virtual input 3 minimum	Integer32
x.2.2.2.5.3.4.0	virtualInput3hyst	read-write	Virtual input 3 hysteresis	Integer32
x.2.2.2.5.3.5.0	virtualInput3Parent	read-write	Virtual input 3 parent	INTEGER{none(0),sensor11(1),sensor12(2),sensor21(3),sensor22(4),sensor31(5),sensor32(6),sensor41(7),sensor42(8),sensor51(9),sensor52(10),sensor61(11),sensor62(12),sensor71(13),sensor72(14),sensor81(15),sensor82(16),analog1(17),analog2(18)}

setup -> io -> virtualSetup -> virtual4setup

OID	Name	Access	Description	Syntax
x.2.2.2.5.4.1.0	virtualInput4description	read-write	Virtual input 4 description	DisplayString
x.2.2.2.5.4.2.0	virtualInput4max	read-write	Virtual input 4 maximum	Integer32
x.2.2.2.5.4.3.0	virtualInput4min	read-write	Virtual input 4 minimum	Integer32
x.2.2.2.5.4.4.0	virtualInput4hyst	read-write	Virtual input 4 hysteresis	Integer32
x.2.2.2.5.4.5.0	virtualInput4Parent	read-write	Virtual input 4 parent	INTEGER{none(0),sensor11(1),sensor12(2),sensor21(3),sensor22(4),sensor31(5),sensor32(6),sensor41(7),sensor42(8),sensor51(9),sensor52(10),sensor61(11),sensor62(12),sensor71(13),sensor72(14),sensor81(15),sensor82(16),analog1(17),analog2(18)}

monitorNcontrol -> sensors -> sensor1

OID	Name	Access	Description	Syntax
x.2.3.1.1.1.0	s11Int	read-only	S11 value x1000 in Integer format	Integer32
x.2.3.1.1.2.0	s12Int	read-only	S12 value x1000 in Integer format	Integer32
x.2.3.1.1.3.0	s1ID	read-only	S1 ID value	Mac Address

monitorNcontrol -> sensors -> sensor2

OID	Name	Access	Description	Syntax
x.2.3.1.2.1.0	s21Int	read-only	S21 value x1000 in Integer format	Integer32
x.2.3.1.2.2.0	s22Int	read-only	S22 value x1000 in Integer format	Integer32
x.2.3.1.2.3.0	s2ID	read-only	S2 ID value	OCTET STRING (SIZE (16))

monitorNcontrol -> sensors -> sensor3

OID	Name	Access	Description	Syntax
x.2.3.1.3.1.0	s31Int	read-only	S31 value x1000 in Integer format	Integer32
x.2.3.1.3.2.0	s32Int	read-only	S32 value x1000 in Integer format	Integer32
x.2.3.1.3.3.0	s3ID	read-only	S3 ID value	OCTET STRING (SIZE (16))

monitorNcontrol -> sensors -> sensor4

OID	Name	Access	Description	Syntax
x.2.3.1.4.1.0	s41Int	read-only	S41 value x1000 in Integer format	Integer32
x.2.3.1.4.2.0	s42Int	read-only	S42 value x1000 in Integer format	Integer32
x.2.3.1.4.3.0	s4ID	read-only	S4 ID value	OCTET STRING (SIZE (16))

monitorNcontrol -> sensors -> sensor5

OID	Name	Access	Description	Syntax
x.2.3.1.5.1.0	s51Int	read-only	S51 value x1000 in Integer format	Integer32
x.2.3.1.5.2.0	s52Int	read-only	S52 value x1000 in Integer format	Integer32
x.2.3.1.5.3.0	s5ID	read-only	S5 ID value	OCTET STRING (SIZE (16))

monitorNcontrol -> sensors -> sensor6

OID	Name	Access	Description	Syntax
x.2.3.1.6.1.0	s61Int	read-only	S61 value x1000 in Integer format	Integer32
x.2.3.1.6.2.0	s62Int	read-only	S62 value x1000 in Integer format	Integer32
x.2.3.1.6.3.0	s6ID	read-only	S6 ID value	OCTET STRING (SIZE (16))

monitorNcontrol -> sensors -> sensor7

OID	Name	Access	Description	Syntax
x.2.3.1.7.1.0	s71Int	read-only	S71 value x1000 in Integer format	Integer32
x.2.3.1.7.2.0	s72Int	read-only	S72 value x1000 in Integer format	Integer32
x.2.3.1.7.3.0	s7ID	read-only	S7 ID value	OCTET STRING (SIZE (16))

monitorNcontrol -> sensors -> sensor8

OID	Name	Access	Description	Syntax
x.2.3.1.8.1.0	s81Int	read-only	S81 value x1000 in Integer format	Integer32
x.2.3.1.8.2.0	s82Int	read-only	S82 value x1000 in Integer format	Integer32
x.2.3.1.8.3.0	s8ID	read-only	S8 ID value	OCTET STRING (SIZE (16))

monitorNcontrol -> analog

OID	Name	Access	Description	Syntax
x.2.3.2.1.0	voltage1Int	read-only	Voltage1 x1000 in Integer format	Integer32
x.2.3.2.2.0	voltage2Int	read-only	Voltage2 x1000 in Integer format	Integer32

monitorNcontrol -> digital

OID	Name	Access	Description	Syntax
x.2.3.3.1.0	digitalInput1State	read-only	Digital1 Input State	INTEGER {closed(0), open(1)}
x.2.3.3.2.0	digitalInput2State	read-only	Digital2 Input State	INTEGER {closed(0), open(1)}

monitorNcontrol -> relays -> relay1

OID	Name	Access	Description	Syntax
x.2.3.4.1.1.0	relay1State	read-write	Relay1 State	INTEGER {off(0), on(1)}
x.2.3.4.1.2.0	relay1Pulse	read-write	Relay1 Pulse	INTEGER {off(0), on(1)}

monitorNcontrol -> relays -> relay2

OID	Name	Access	Description	Syntax
x.2.3.4.2.1.0	relay2State	read-write	Relay2 State	INTEGER {off(0), on(1)}
x.2.3.4.2.2.0	relay2pulse	read-write	Relay2 pulse length	INTEGER {off(0), on(1)}

monitorNcontrol

OID	Name	Access	Description	Syntax
x.2.3.5.0	configurationSaved	read-write	Configuration save status SAVED/UNSAVED	INTEGER { unsaved(0), saved(1)}
x.2.3.6.0	restartDevice	read-write	Restart Device	INTEGER { cancel(0), restart(1)}
x.2.3.7.0	temperatureUnit	read-only	Unit of the all temperature values	INTEGER {celcius(0), fahrenheit(1)}
x.2.3.8.0	hardwareErr	read-only	Hardware Error	INTEGER {noErr(0), owErr(1), hwErr(2)}
x.2.3.9.0	pressureUnit	read-only	Unit of the pressure value	INTEGER {hPa(0), mbar(1), mmhg(2)}

monitorNcontrol -> virtual

OID	Name	Access	Description	Syntax
x.2.3.11.1.0	virtualInput1Int	read-write	Virtual input 1 x1000 in Integer format	Integer32
x.2.3.11.2.0	virtualInput2Int	read-write	Virtual input 2 x1000 in Integer format	Integer32
x.2.3.11.3.0	virtualInput3Int	read-write	Virtual input 3 x1000 in Integer format	Integer32
x.2.3.11.4.0	virtualInput4Int	read-write	Virtual input 4 x1000 in Integer format	Integer32

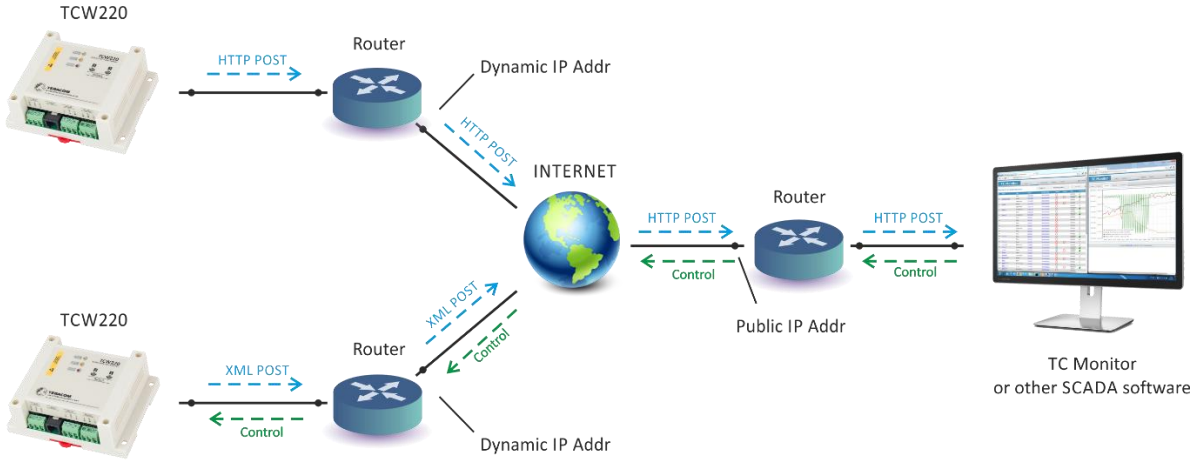
8.2. HTTP API

8.2.1. HTTP Post

TCW220 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. As an answer, the server can send HTTP Get with appropriate command – see **8.2.3. HTTP commands**

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 (for example www.yourserverURL.com/post.php) in your web browser. If all is OK, a web page with “The PHP script is working!” will be shown.
- Set the controller to send 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, a XML file will be created in the same directory, where post.php is located. The file name will contain time information and looks as 20151120103318_status.xml.

8.2.2. HTTP Get

HTTP Get can be used to monitor TCW220 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 TCW220 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?ron=1, will turn Relay 1 ON.

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 on 120 sec in case the username=admin and pass=admin

8.2.3. List of HTTP API commands

Command	Description
ron=**n**	Turn relay **n** ON (**n** is 1 or 2 for the respective relay) ron=1 - will turn ON relay 1 ron=2 - will turn ON relay 2
ron=1&ron=2	Turn both relays ON
rof=**n**	Turn relay **n** OFF (**n** is 1 or 2 for the respective relay) rof=1 - will turn OFF relay 1 rof=2 - will turn OFF relay 2
rof=1&rof=2	Turn both relays OFF
rtg=**n**	Toggle relay **n** state (**n** is 1 or 2 for the respective relay) rtg=1 - will toggle relay 1 state rtg=2 - will toggle relay 2 state
rpl=**n**	Pulse relay **n** (**n** is 1 or 2 for the respective relay) rpl=1 – will pulse relay 1 rpl=2 – will pulse relay 2
vn**f**=10.0	Set Min of analog input to 10.0 (**f** is 1 or 2 for the respective input) vn1=10.0 will set Min for analog input 1
vx**f**=20.0	Set Max of analog input to 20.0 (**f** is 1 or 2 for the respective input) vx2=20.0 will set Max for analog input 2
vy**f**=1.0	Set Hys of analog input to 1.0 (**f** is 1 or 2 for the respective input) vy1=1.0 will set Hys for analog input 1
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)

delanl=xxxx	Notification delay for analog inputs (xxxx is between 0 and 3600)
deldig=xxxx	Notification delay for digital inputs (xxxx is between 0 and 3600)
dda1=xxxx	Low to high delay for digital input 1 (xxxx is between 0 and 3600)
ddd1=xxxx	High to low delay for digital input 1 (xxxx is between 0 and 3600)
dda2=xxxx	Low to high delay for digital input 2 (xxxx is between 0 and 3600)
ddd2=xxxx	High to low delay for digital input 2 (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>TCW220</DeviceName>
        <HostName>TCW220</HostName>
        <ID>5C:32:C5:00:69:02</ID>
        <FwVer>TCW220-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>
            <item1>
                <value>23.313</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>
        </S1>
        <S2>
            <description>S2:TSH2xx</description>
            <id>015225B71700FF45</id>
            <item1>
                <value>24.563</value>
                <unit>°C</unit>
                <alarm>0</alarm>
                <min>-40.000</min>
                <max>85.000</max>
                <hys>8.500</hys>
            </item1>
            <item2>
                <value>34.313</value>
                <unit>%RH</unit>
                <alarm>0</alarm>
                <min>0.000</min>
                <max>100.000</max>
                <hys>10.000</hys>
            </item2>
        </S2>
        <S3>
            <description>S3</description>
            <id>0000000000000000</id>
            <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>
        </S3>
        <S4>
            <description>S4</description>
            <id>0000000000000000</id>
```

```
    <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>
</S4>
<S5>
    <description>S5</description>
    <id>0000000000000000</id>
    <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>
</S5>
<S6>
    <description>S6</description>
    <id>0000000000000000</id>
    <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>
</S6>
<S7>
    <description>S7</description>
    <id>0000000000000000</id>
    <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>
</S7>
```

```
    <S8>
        <description>S8</description>
        <id>0000000000000000</id>
        <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>
    </S8>
</S>
<AI>
    <AI1>
        <description>Analog Input 1</description>
        <value>0.033</value>
        <unit>V</unit>
        <multiplier>1.000</multiplier>
        <offset>0.0000</offset>
        <alarm>0</alarm>
        <min>0.000</min>
        <max>10.000</max>
        <hys>0.100</hys>
    </AI1>
    <AI2>
        <description>Analog Input 2</description>
        <value>0.033</value>
        <unit>V</unit>
        <multiplier>1.000</multiplier>
        <offset>0.0000</offset>
        <alarm>0</alarm>
        <min>0.000</min>
        <max>10.000</max>
        <hys>0.100</hys>
    </AI2>
</AI>
<VI>
    <VI1>
        <description>Virtual Input 1</description>
        <value>0.033</value>
        <unit>V</unit>
        <multiplier>1.000</multiplier>
        <offset>0.0000</offset>
        <alarm>1</alarm>
        <min>4.500</min>
        <max>5.500</max>
        <hys>0.010</hys>
    </VI1>
    <VI2>
        <description>Virtual Input 2</description>
        <value>0.033</value>
        <unit>V</unit>
        <multiplier>1.000</multiplier>
        <offset>0.0000</offset>
        <alarm>1</alarm>
        <min>6.000</min>
        <max>8.000</max>
        <hys>0.100</hys>
    </VI2>
    <VI3>
        <description>Virtual Input 3</description>
        <value>24.563</value>
        <unit>°C</unit>
        <multiplier>1.000</multiplier>
        <offset>0.0000</offset>
        <alarm>1</alarm>
        <min>25.000</min>
        <max>35.000</max>
```

```
            <hys>0.010</hys>
        </VI3>
        <VI4>
            <description>Virtual Input 4</description>
            <value>34.313</value>
            <unit>%RH</unit>
            <multiplier>1.000</multiplier>
            <offset>0.0000</offset>
            <alarm>1</alarm>
            <min>45.000</min>
            <max>55.000</max>
            <hys>0.100</hys>
        </VI4>
    </VI>
    <DI>
        <DI1>
            <description>Digital Input 1</description>
            <value>OPEN</value>
            <valuebin>1</valuebin>
            <alarmState>CLOSED</alarmState>
            <alarm>0</alarm>
        </DI1>
        <DI2>
            <description>Digital Input 2</description>
            <value>OPEN</value>
            <valuebin>1</valuebin>
            <alarmState>CLOSED</alarmState>
            <alarm>0</alarm>
        </DI2>
    </DI>
    <R>
        <R1>
            <description>Relay 1</description>
            <value>OFF</value>
            <valuebin>0</valuebin>
            <pulseWidth>0.1</pulseWidth>
            <control>0</control>
        </R1>
        <R2>
            <description>Relay 2</description>
            <value>OFF</value>
            <valuebin>0</valuebin>
            <pulseWidth>0.2</pulseWidth>
            <control>0</control>
        </R2>
    </R>
    <HTTPPush>
        <Key/>
        <PushPeriod>60</PushPeriod>
    </HTTPPush>
    <hwerr/>
    <Alarmed>1</Alarmed>
    <Scannig/>
    <Time>
        <Date>11.10.2023</Date>
        <Time>09:54:36</Time>
    </Time>
</Monitor>
```

Where:

<value>--- </value> and *<unit>--- </unit>* means no 1-Wire sensor on this position;

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

8.2.5. JSON file structure


```
{
  "Monitor": {
    "DeviceInfo": {
      "DeviceName": "TCW220",
      "HostName": "TCW220",
      "ID": "5C:32:C5:00:69:02",
      "FwVer": "TCW220-v1.252",
      "MnfInfo": "www.teracomsystems.com",
      "SysContact": "info@teracomsystems.com",
      "SysName": "SysName",
      "SysLocation": "SysLocation"
    },
    "S": {
      "S1": {
        "description": "S1:TST1xx",
        "id": "2867895F07000058",
        "item1": {
          "value": "23.375",
          "unit": "°C",
          "alarm": "0",
          "min": "-40.000",
          "max": "85.000",
          "hys": "8.500"
        },
        "item2": {
          "value": "---",
          "unit": "---",
          "alarm": "0",
          "min": "---",
          "max": "---",
          "hys": "---"
        }
      },
      "S2": {
        "description": "S2:TSH2xx",
        "id": "015225B71700FF45",
        "item1": {
          "value": "24.625",
          "unit": "°C",
          "alarm": "0",
          "min": "-40.000",
          "max": "85.000",
          "hys": "8.500"
        },
        "item2": {
          "value": "34.438",
          "unit": "%RH",
          "alarm": "0",
          "min": "0.000",
          "max": "100.000",
          "hys": "10.000"
        }
      },
      "S3": {
        "description": "S3",
        "id": "0000000000000000",
        "item1": {
          "value": "---",
          "unit": "---",
          "alarm": "0",
          "min": "---",
          "max": "---",
          "hys": "---"
        },
        "item2": {
          "value": "---",
          "unit": "---",
          "alarm": "0",
          "min": "---",
          "max": "---",
          "hys": "---"
        }
      },
      "S4": {
        "description": "S4",
```

```
    "id": "0000000000000000",
    "item1": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    },
    "item2": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    }
  },
  "S5": {
    "description": "S5",
    "id": "0000000000000000",
    "item1": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    },
    "item2": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    }
  },
  "S6": {
    "description": "S6",
    "id": "0000000000000000",
    "item1": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    },
    "item2": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    }
  },
  "S7": {
    "description": "S7",
    "id": "0000000000000000",
    "item1": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    },
    "item2": {
      "value": "---",
      "unit": "---",
      "alarm": "0",
      "min": "---",
      "max": "---",
      "hys": "---"
    }
```

```
    },
    "S8": {
      "description": "S8",
      "id": "0000000000000000",
      "item1": {
        "value": "---",
        "unit": "---",
        "alarm": "0",
        "min": "---",
        "max": "---",
        "hys": "---"
      },
      "item2": {
        "value": "---",
        "unit": "---",
        "alarm": "0",
        "min": "---",
        "max": "---",
        "hys": "---"
      }
    }
  },
  "AI": {
    "AI1": {
      "description": "Analog Input 1",
      "value": "0.033",
      "unit": "V",
      "multiplier": "1.000",
      "offset": "0.0000",
      "alarm": "0",
      "min": "0.000",
      "max": "10.000",
      "hys": "0.100"
    },
    "AI2": {
      "description": "Analog Input 2",
      "value": "0.033",
      "unit": "V",
      "multiplier": "1.000",
      "offset": "0.0000",
      "alarm": "0",
      "min": "0.000",
      "max": "10.000",
      "hys": "0.100"
    }
  },
  "VI": {
    "VI1": {
      "description": "Virtual Input 1",
      "value": "0.033",
      "unit": "V",
      "multiplier": "1.000",
      "offset": "0.0000",
      "alarm": "1",
      "min": "4.500",
      "max": "5.500",
      "hys": "0.010"
    },
    "VI2": {
      "description": "Virtual Input 2",
      "value": "0.033",
      "unit": "V",
      "multiplier": "1.000",
      "offset": "0.0000",
      "alarm": "1",
      "min": "6.000",
      "max": "8.000",
      "hys": "0.100"
    },
    "VI3": {
      "description": "Virtual Input 3",
      "value": "24.625",
      "unit": "°C",
      "multiplier": "1.000",
      "offset": "0.0000",
      "alarm": "1",
      "min": "25.000",
```

```
            "max": "35.000",
            "hys": "0.010"
          },
          "VI4": {
            "description": "Virtual Input 4",
            "value": "34.438",
            "unit": "%RH",
            "multiplier": "1.000",
            "offset": "0.0000",
            "alarm": "1",
            "min": "45.000",
            "max": "55.000",
            "hys": "0.100"
          }
        },
        "DI": {
          "DI1": {
            "description": "Digital Input 1",
            "value": "OPEN",
            "valuebin": "1",
            "alarmState": "CLOSED",
            "alarm": "0"
          },
          "DI2": {
            "description": "Digital Input 2",
            "value": "OPEN",
            "valuebin": "1",
            "alarmState": "CLOSED",
            "alarm": "0"
          }
        },
        "R": {
          "R1": {
            "description": "Relay 1",
            "value": "OFF",
            "valuebin": "0",
            "pulseWidth": "0.1",
            "control": "0"
          },
          "R2": {
            "description": "Relay 2",
            "value": "OFF",
            "valuebin": "0",
            "pulseWidth": "0.2",
            "control": "0"
          }
        },
        "HTTPPush": {
          "Key": "",
          "PushPeriod": "60"
        },
        "hwerr": "",
        "Alarmed": "1",
        "Scannig": "",
        "Time": {
          "Date": "11.10.2023",
          "Time": "10:07:15"
        }
      }
    }
```

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 Coil Status (FC=01)

Request

This command is requesting the ON/OFF status of discrete coils on address 100.

01 0064 0001

01: The Function Code 1 (read Coil Status)
0064: The Data Address of the coil to read (0064 hex = 100)
0001: The total number of coils requested. (01 hex = 1)

Response

01 01 01

01: The Function Code 1 (read Coil Status)
01: The number of data bytes to follow
01: 7 space holders & Coils 1 (0000 0001)
Due to the number of coils requested, the last data field **01** contains the status of only 1 coil. The 7 most significant bits in this data field are filled in with zeroes. The activated relay is 1.

8.3.1.2. Force Single Coil (FC=05)

Request

This command is writing the contents of discrete on address 100 to ON.

05 0064 FF00

05: The Function Code 5 (Force Single Coil)
0064: The Data Address of the coil. (0064 hex = 100)
FF00: The status to write (FF00 = ON, 0000 = OFF)

Response

The normal response is an echo of the query, returned after the coil has been written.

05 0064 FF00

05: The Function Code 5 (Force Single Coil)
0064: The Data Address of the coil. (0064 hex = 100)
FF00: The status written (FF00 = ON, 0000 = OFF)

8.3.1.3. Read Input Status (FC=02)

Request

This command is requesting the ON/OFF status of discrete input 1

02 0064 0001

02: The Function Code 2 (read Input Status)
0064: The Data Address of the input to read (0064 hex = 100)
0001: The total number of coils requested.

Response

02 01 01

02: The Function Code 2 (read Input Status)
01: The number of data bytes to follow

01: 7 space holders & Discrete Input 100 (0000 0001)
The 7 most significant bits are filled in with zeroes.

8.3.1.4. 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 4 41BD 0655

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

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

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

Request
This command is requesting the content of holding registers 18100.
03 46B4 0020

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

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

03: Function Code 3 (read Analog Output 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.

8.3.1.5. 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 the 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
Relay 1	01,05,15	100	Discrete	
Relay 2	01,05,15	101	Discrete	
Digital input 1	02	100	Discrete	
Digital input 2	02	101	Discrete	
Relay 1 description	03,16	15000	64 bytes UTF-8	
Relay 2 description	03,16	15032	64 bytes UTF-8	
Relay 1 pulse width	03,16	15200	32-bit Float	
Relay 2 pulse width	03,16	15202	32-bit Float	
Relay 1 activated from	03,06,16	15300	16-bit unsign int	manual(0), sensor11(1), sensor12(2), sensor21(3), sensor22(4), sensor31(5), sensor32(6), sensor41(7), sensor42(8), sensor51(9), sensor52(10), sensor61(11), sensor62(12), sensor71(13), sensor72(14), sensor81(15), sensor82(16), analog1(17), analog2(18), digital1(19), digital2(20), anyAlarm(21), virtual1(22), virtual2(23), virtual3(24), virtual4(25)
Relay 2 activated from	03,06,16	15301	16-bit unsign int	—"—
Relay 1 action on alarm	03,06,16	15400	16-bit unsign int	on (0), pulse (2)
Relay 2 action on alarm	03,06,16	15401	16-bit unsign int	—"—
Relays state after restart	03,06	15500	16-bit unsign int	off (0), on (1), last state (2)

Digital input 1 description	03,16	16000	64 bytes UTF-8	
Digital input 2 description	03,16	16032	64 bytes UTF-8	
Digital input 1 alarm state	03	16200	16-bit unsign int	
Digital input 2 alarm state	03	16201	16-bit unsign int	
Analog input 1 description	03,16	17000	64 bytes UTF-8	
Analog input 2 description	03,16	17032	64 bytes UTF-8	
Analog input 1 max	03,16	17200	32-bit Float	
Analog input 2 max	03,16	17202	32-bit Float	
Analog input 1 min	03,16	17300	32-bit Float	
Analog input 2 min	03,16	17302	32-bit Float	
Analog input 1 hysteresis	03,16	17400	32-bit Float	
Analog input 2 hysteresis	03,16	17402	32-bit Float	
Analog input 1 value	03	17500	32-bit Float	
Analog input 2 value	03	17502	32-bit Float	
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 2, S21 dimension	03	18464	64 bytes UTF-8	
Sensor 2, S22 dimension	03	18496	64 bytes UTF-8	
Sensor 3, S31 dimension	03	18528	64 bytes UTF-8	
Sensor 3, S32 dimension	03	18560	64 bytes UTF-8	
Sensor 4, S41 dimension	03	18592	64 bytes UTF-8	
Sensor 4, S42 dimension	03	18624	64 bytes UTF-8	
Sensor 5, S51 dimension	03	18656	64 bytes UTF-8	
Sensor 5, S52 dimension	03	18688	64 bytes UTF-8	
Sensor 6, S61 dimension	03	18720	64 bytes UTF-8	
Sensor 6, S62 dimension	03	18752	64 bytes UTF-8	
Sensor 7, S71 dimension	03	18784	64 bytes UTF-8	
Sensor 7, S72 dimension	03	18816	64 bytes UTF-8	
Sensor 8, S81 dimension	03	18848	64 bytes UTF-8	
Sensor 8, S82 dimension	03	18880	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 2, S21 max	03,16	19204	32-bit Float	
Sensor 2, S22 max	03,16	19206	32-bit Float	
Sensor 3, S31 max	03,16	19208	32-bit Float	
Sensor 3, S32 max	03,16	19210	32-bit Float	
Sensor 4, S41 max	03,16	19212	32-bit Float	
Sensor 4, S42 max	03,16	19214	32-bit Float	
Sensor 5, S51 max	03,16	19216	32-bit Float	
Sensor 5, S52 max	03,16	19218	32-bit Float	
Sensor 6, S61 max	03,16	19220	32-bit Float	
Sensor 6, S62 max	03,16	19222	32-bit Float	
Sensor 7, S71 max	03,16	19224	32-bit Float	
Sensor 7, S72 max	03,16	19226	32-bit Float	
Sensor 8, S81 max	03,16	19228	32-bit Float	
Sensor 8, S82 max	03,16	19230	32-bit Float	
Sensor 1, S11 min	03,16	19300	32-bit Float	
Sensor 1, S12 min	03,16	19302	32-bit Float	
Sensor 2, S21 min	03,16	19304	32-bit Float	
Sensor 2, S22 min	03,16	19306	32-bit Float	
Sensor 3, S31 min	03,16	19308	32-bit Float	
Sensor 3, S32 min	03,16	19310	32-bit Float	
Sensor 4, S41 min	03,16	19312	32-bit Float	
Sensor 4, S42 min	03,16	19314	32-bit Float	
Sensor 5, S51 min	03,16	19316	32-bit Float	
Sensor 5, S52 min	03,16	19318	32-bit Float	
Sensor 6, S61 min	03,16	19320	32-bit Float	
Sensor 6, S62 min	03,16	19322	32-bit Float	
Sensor 7, S71 min	03,16	19324	32-bit Float	
Sensor 7, S72 min	03,16	19326	32-bit Float	
Sensor 8, S81 min	03,16	19328	32-bit Float	
Sensor 8, S82 min	03,16	19330	32-bit Float	
Sensor 1, S11 hysteresis	03,16	19400	32-bit Float	
Sensor 1, S12 hysteresis	03,16	19402	32-bit Float	
Sensor 2, S21 hysteresis	03,16	19404	32-bit Float	
Sensor 2, S22 hysteresis	03,16	19406	32-bit Float	
Sensor 3, S31 hysteresis	03,16	19408	32-bit Float	
Sensor 3, S32 hysteresis	03,16	19410	32-bit Float	
Sensor 4, S41 hysteresis	03,16	19412	32-bit Float	
Sensor 4, S42 hysteresis	03,16	19414	32-bit Float	
Sensor 5, S51 hysteresis	03,16	19416	32-bit Float	
Sensor 5, S52 hysteresis	03,16	19418	32-bit Float	
Sensor 6, S61 hysteresis	03,16	19420	32-bit Float	
Sensor 6, S62 hysteresis	03,16	19422	32-bit Float	
Sensor 7, S71 hysteresis	03,16	19424	32-bit Float	
Sensor 7, S72 hysteresis	03,16	19426	32-bit Float	

Sensor 8, S81 hysteresis	03,16	19428	32-bit Float	
Sensor 8, S82 hysteresis	03,16	19430	32-bit Float	
Sensor 1, S11 multiplier	03,16	19500	32-bit Float	
Sensor 1, S12 multiplier	03,16	19502	32-bit Float	
Sensor 2, S21 multiplier	03,16	19504	32-bit Float	
Sensor 2, S22 multiplier	03,16	19506	32-bit Float	
Sensor 3, S31 multiplier	03,16	19508	32-bit Float	
Sensor 3, S32 multiplier	03,16	19510	32-bit Float	
Sensor 4, S41 multiplier	03,16	19512	32-bit Float	
Sensor 4, S42 multiplier	03,16	19514	32-bit Float	
Sensor 5, S51 multiplier	03,16	19516	32-bit Float	
Sensor 5, S52 multiplier	03,16	19518	32-bit Float	
Sensor 6, S61 multiplier	03,16	19520	32-bit Float	
Sensor 6, S62 multiplier	03,16	19522	32-bit Float	
Sensor 7, S71 multiplier	03,16	19524	32-bit Float	
Sensor 7, S72 multiplier	03,16	19526	32-bit Float	
Sensor 8, S81 multiplier	03,16	19528	32-bit Float	
Sensor 8, S82 multiplier	03,16	19530	32-bit Float	
Sensor 1, S11 offset	03,16	19600	32-bit Float	
Sensor 1, S12 offset	03,16	19602	32-bit Float	
Sensor 2, S21 offset	03,16	19604	32-bit Float	
Sensor 2, S22 offset	03,16	19606	32-bit Float	
Sensor 3, S31 offset	03,16	19608	32-bit Float	
Sensor 3, S32 offset	03,16	19610	32-bit Float	
Sensor 4, S41 offset	03,16	19612	32-bit Float	
Sensor 4, S42 offset	03,16	19614	32-bit Float	
Sensor 5, S51 offset	03,16	19616	32-bit Float	
Sensor 5, S52 offset	03,16	19618	32-bit Float	
Sensor 6, S61 offset	03,16	19620	32-bit Float	
Sensor 6, S62 offset	03,16	19622	32-bit Float	
Sensor 7, S71 offset	03,16	19624	32-bit Float	
Sensor 7, S72 offset	03,16	19626	32-bit Float	
Sensor 8, S81 offset	03,16	19628	32-bit Float	
Sensor 8, S82 offset	03,16	19630	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 2, S21 value	03	19804	32-bit Float	
Sensor 2, S22 value	03	19806	32-bit Float	
Sensor 3, S31 value	03	19808	32-bit Float	
Sensor 3, S32 value	03	19810	32-bit Float	
Sensor 4, S41 value	03	19812	32-bit Float	
Sensor 4, S42 value	03	19814	32-bit Float	
Sensor 5, S51 value	03	19816	32-bit Float	
Sensor 5, S52 value	03	19818	32-bit Float	
Sensor 6, S61 value	03	19820	32-bit Float	
Sensor 6, S62 value	03	19822	32-bit Float	
Sensor 7, S71 value	03	19824	32-bit Float	
Sensor 7, S72 value	03	19826	32-bit Float	
Sensor 8, S81 value	03	19828	32-bit Float	
Sensor 8, S82 value	03	19830	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 2, S21 alarm status	03	20002	16-bit unsign int	normal (0), alarm (1)
Sensor 2, S22 alarm status	03	20003	16-bit unsign int	normal (0), alarm (1)
Sensor 3, S31 alarm status	03	20004	16-bit unsign int	normal (0), alarm (1)
Sensor 3, S32 alarm status	03	20005	16-bit unsign int	normal (0), alarm (1)
Sensor 4, S41 alarm status	03	20006	16-bit unsign int	normal (0), alarm (1)
Sensor 4, S42 alarm status	03	20007	16-bit unsign int	normal (0), alarm (1)
Sensor 5, S51 alarm status	03	20008	16-bit unsign int	normal (0), alarm (1)
Sensor 5, S52 alarm status	03	20009	16-bit unsign int	normal (0), alarm (1)
Sensor 6, S61 alarm status	03	20010	16-bit unsign int	normal (0), alarm (1)
Sensor 6, S62 alarm status	03	20011	16-bit unsign int	normal (0), alarm (1)
Sensor 7, S71 alarm status	03	20012	16-bit unsign int	normal (0), alarm (1)
Sensor 7, S72 alarm status	03	20013	16-bit unsign int	normal (0), alarm (1)
Sensor 8, S81 alarm status	03	20014	16-bit unsign int	normal (0), alarm (1)
Sensor 8, S82 alarm status	03	20015	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. 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. A copy of the 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) a number of records are about 52000. This number is enough for 36 days with records on every 1 minute.

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

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

The 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 initiated by alarm condition;
	"Type"	for header record;
	"Start"	after power-up condition;
	"Restart"	after reset condition;
	"Power Down"	after power-down condition;
	"Bad"	for a problematic record.
Inputs value	in orders – sensors, analog inputs and digital inputs.	
Relays	relays conditions.	
Alarm conditions	show condition for every input, "1" means an active alarm.	

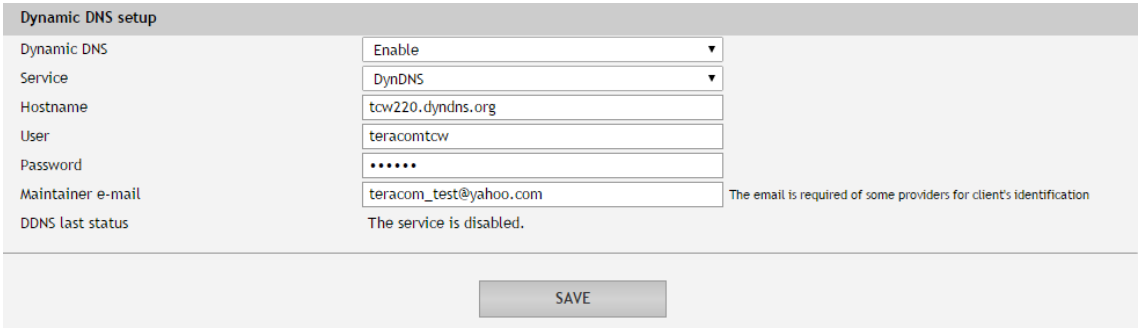
An example of the log file:

1131901;15.10.2015,01:02:23;Type;S11/°C;S12;S21/°C;S22;S31/°C;S32;S41/°C;S42;S51/°C;S52;S61/°C;S62;S71/°C;S72;S81/°C;S82;A1/V;A2/V;D1;D2;R1;R2;S1
1/°C;S12;S21/°C;S22;S31/°C;S32;S41/°C;S42;S51/°C;S52;S61/°C;S62;S71/°C;S72;S81/°C;S82;A1/V;A2/V;D1;D2;
1131902;15.10.2015,01:02:23;Time;18.250;;18.375;;18.125;;18.500;;18.188;;18.125;;18.375;;18.375;;11.352;0.065;1;0;1;0;1;;1;;1;;1;;1;;1;;1;;1;;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;;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;;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;;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;;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;;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;

8.5. Dynamic DNS

With dynamic DNS can access TCW220 from the public Internet without investing in a broadband account that has a static IP address.

TCW220 supports the following DNS services – DynDNS, No-IP, and DNS-O-Matric.



9. Factory default settings

TCW220 can be restored to its original factory default settings in 3 different ways.

9.1. Factory default from the 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:

User Name	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
Analog inputs unit	voltage
Analog inputs multiplier	1.000
Analog inputs ofset	0.000

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 this user manual;
- log into the devices via browser program;
- make proper setup;
- short the “Din1” and “GND”;
- install sensor TSH2XX or TST1XX on the 1-Wire 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, 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.

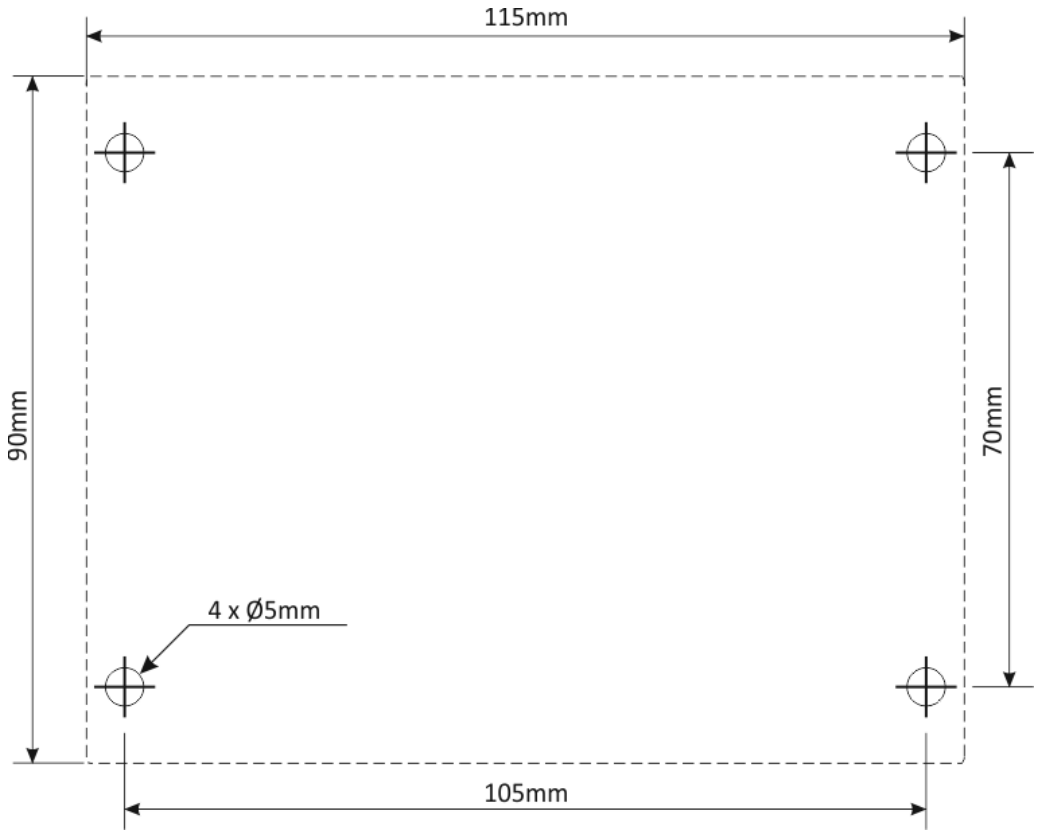

115mm
90mm
70mm
4 x Ø5mm
105mm

Fig.1

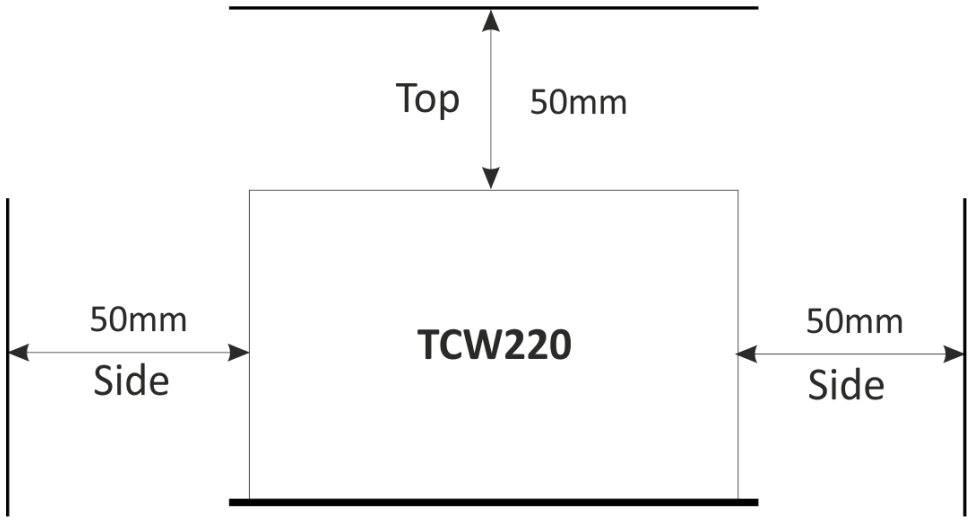

Top
50mm
50mm
Side
TCW220
50mm
Side

Fig.2