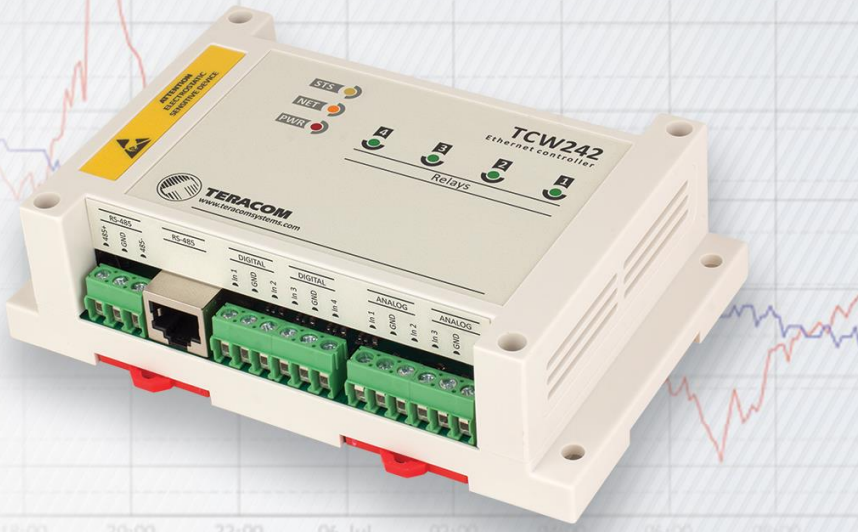
control solutions
TERACOM



TCW242 Industrial IoT module

Version 1.4 / February 2023

www.teracomsystems.com

1. Introduction

The TCW242 is an industrial-grade IoT (Internet of Things) module that offers advanced monitoring and control capabilities for industrial applications. Equipped with Ethernet connectivity and data logging functionality, this device allows users to easily monitor and analyze a wide range of parameters. The device features 4 digital inputs, 4 analog inputs, an RS-485 interface for connecting MODBUS RTU sensors, and 4 relays. The analog inputs can be configured to work in either current loop (0-20mA) or voltage (0-10V) modes, providing flexibility in monitoring different types of parameters.

The TCW242 supports up to 24 Teracom and third-party MODBUS RTU sensors, allowing users to configure up to 24 independent monitoring channels. Each channel can be set up using up to 2 input parameters and/or constants and can be configured as either discrete or general channels. This provides users with a wide range of monitoring options.

In addition to monitoring capabilities, the device also offers robust alarm functionality. Users can configure up to 24 independent alarms with 5 different user-selectable states. Each alarm can be set up using up to 2 limits and hysteresis and can be assigned to a specific channel. In this case, in an alarm state, the assigned channel is colored on the monitoring page and graphs, providing users with a visual indication of the alarm state.

The device also supports a variety of communication protocols for M2M (machine-to-machine) communication, including SNMP, HTTP API, MODBUS TCP/IP, and MQTT. These protocols allow for easy integration with other industrial systems and devices, providing users with a wide range of options for data analysis and control. Overall, the TCW242 industrial IoT module is a powerful and versatile device that offers advanced monitoring and control capabilities for industrial applications.

2. Features

- Ethernet connectivity at 10/100 Mbps with automatic cable detection;
- Web-based configuration and control with password protection;
- 4 digital inputs for dry contacts;
- 4 analog inputs with 0-10V and 0-20mA options;
- 4 relays with normally open/normally closed contacts;
- MODBUS RTU for up to 24 sensors (registers);
- Monitor up to 24 channels;
- Set up to 24 independent alarms;
- SNMPv3 protocol support;
- Alarm notifications via SNMP traps;
- HTTP and SNMP port changing;
- HTTP API commands;
- Periodical HTTP/HTTPS Post of XML/JSON status files for client-server systems;
- MODBUS TCP/IP protocol support;
- MQTT 3.1.1 protocol support;
- Dynamic DNS with DynDNS, No-IP, and DNS-O-Matic support;
- NTP protocol support;
- Data logging capacity for up to 70,000 records;
- Schedule single or repeating tasks;
- Custom functions available;
- Mounts on DIN rail;
- Option to backup and restore settings;
- Remote firmware update capability.

3. Applications

The industrial IoT module is a versatile solution for a variety of applications in the light industry. Its robust design and advanced features make it ideal for remote monitoring and control in communication facilities, food and beverage storage, greenhouses, water stations, and other industrial settings. With its advanced communication options, the module can easily be integrated into existing systems, and its flexible design allows it to be configured to meet the specific needs of each application. Whether you need to monitor temperature and humidity in a greenhouse or manage water levels at a water station, this industrial IoT module is a reliable and cost-effective solution.

4. Specifications

- Physical characteristics
  Dimensions: 145 x 90 x 40 mm
  Weight: 200 g

- Environmental limits
  Operating temperature range: -20 to 55°C
  Storage temperature range: -25 to 60°C
  Operating relative humidity range: 10 to 80% (non-condensing)

- Warranty
  Warranty period: 3 years

- Power supply
  Operating voltage range (including -15/+20% according to IEC 62368-1): 10 to 28 VDC
  Current consumption: 0.35A @ 12VDC (without MODBUS RTU sensors powering)

- RS-485 interface for MODNUS RTU sensors
  Isolation: Non-isolated
  Output voltage (pin 7 of RJ-45): 5.0 ± 0.3 VDC
  Maximum output current (pin 7 of RJ-45): 0.2 A

- Digital inputs
  Isolation: Non-isolated
  Mode: OPEN/CLOSED (“Dry contact”) or COUNTER
  Maximum input voltage: +5.5VDC
  Sampling rate: 1mS
  Digital filtering time interval: 5 to 60000mS

- Analog inputs
  Isolation: Non-isolated
  Type: Single-ended
  Resolution: 12 bits
  Mode: Voltage or current loop
  Input Range: 0/10V or 0/20mA
  Accuracy: ±1%
  Input Impedance: 1 mega-ohm (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: 0.1 Hz at rated load

CAUTION: The device does not contain any internal overcurrent protection facilities on the relay's contact lines.
External fuses or short circuit current limiting circuit breakers, rated to 3 Amps, are to be used for overcurrent protection of the connecting lines.

- Internal FLASH memory
  Settings segment endurance: 100 000 cycles (Every setting change is a memory cycle).
  Data logger segment endurance: 100 000 full scrolls (70000 records) of the logger.
  Update segment endurance: 100 000 cycles (updates).

- Lithium battery
  Type: CR1220

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

5. LED indicators

The device has 3 LED indicators that show its status:

- **PWR** (red) – lights up during normal operation, and blinks together with STS in case of a hardware error;
- **STS** (yellow) – blinks while the controller's main program is running;
- **NET** (orange) – indicates network status. It lights up when a connection is established and blinks with network activity.

6. Installation and configuration

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

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

6.1. Mounting

The TCW242 must be installed in a clean, dry, and non-flammable location. Ventilation is recommended for high ambient temperature environments.

To mount the device, use two plastic dowels (e.g. Würth GmbH 0912 802 002) and two dowel screws (e.g. Würth GmbH 0157 06 70) to secure it to a wall. Refer to fig.1 in Appendix A for mechanical details.

Leave 50 mm of space on all sides for ventilation and electrical isolation. Refer to fig.2 in Appendix A.

The device can also be attached to a standard DIN rail (35mm by 7.55mm) by hooking the back of the enclosure onto the rail and snapping the bottom into place.

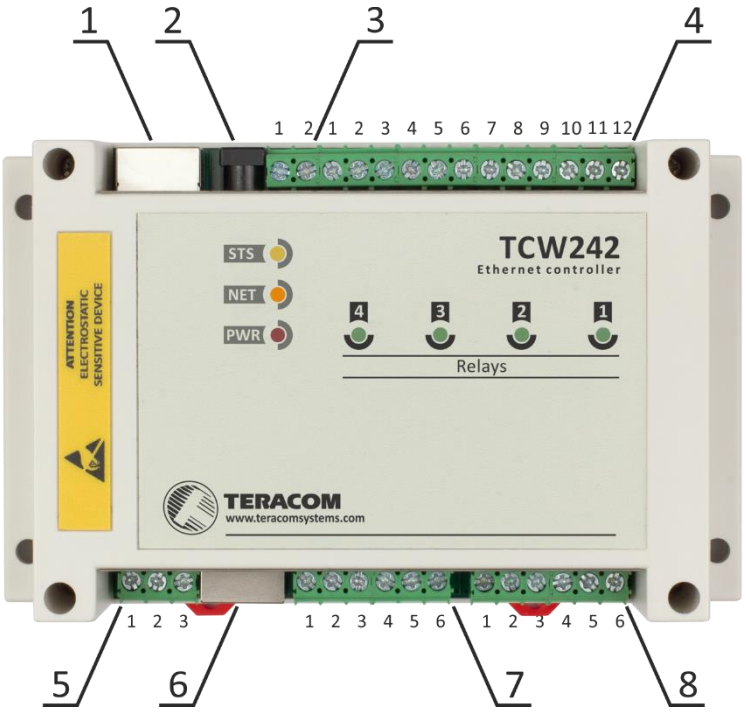
6.2. Connection

Warning! Power off before wiring.

Follow these steps for correct wiring:

- Turn off power;
- Connect wires to terminals;
- Turn on the power.

Ensure that wires are securely attached to terminals and tightened. Improper wiring or configuration can cause permanent damage to TCW242 or connected equipment.



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 Relay4 **Pin2** – COM Relay4 **Pin3** – NO Relay4 **Pin4** – NC Relay3 **Pin5** – COM Relay3 **Pin6** – NO Relay3 **Pin7** – NC Relay2 **Pin8** – COM Relay2 **Pin9** – NO Relay2 **Pin10** – NC Relay1 **Pin11** – COM Relay1 **Pin12** – NO Relay1
Connector 5	**Pin1** – RS485+ **Pin2** – GND **Pin3** – RS485-
Connector 6	**Pin1** – not connected (most left) **Pin2** – not connected **Pin3** – not connected **Pin4** – RS485- **Pin5** – RS485+ **Pin6** – not connected **Pin7** – +VDD **Pin8** – GND
Connector 7	**Pin1** – Digital In 1 **Pin2** – GND **Pin3** – Digital In 2 **Pin4** – Digital In 3 **Pin5** – GND **Pin6** – Digital In 4
Connector 8	**Pin1**– Analog In 1 **Pin2** – GND **Pin3** – Analog In 2 **Pin4** – Analog In 3 **Pin5** – GND **Pin6** – Analog In 4

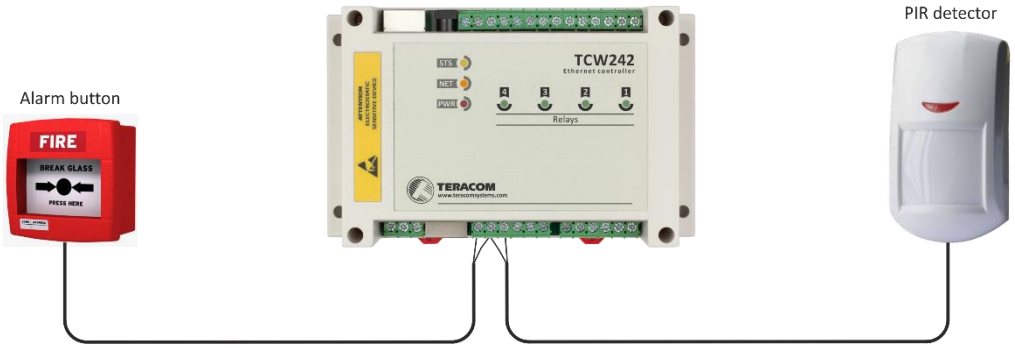
6.2.1. Power supply connection

TCW242 must be powered by the adapter SYS1308(N)-2412-W2E or equivalent, suitable for overvoltage category II and certified for safety compliance. The power supply device should be able to withstand short circuits and secondary circuit overloads. Ensure the equipment is easily accessible for disconnecting from the power supply during use.

6.2.2. Digital inputs connection

Note that all inputs are not isolated from the power supply.

The digital inputs of TCW242 in OPEN/CLOSED mode can monitor devices with "dry contact" outputs such as door contacts, push buttons, PIR detectors, etc. To connect an alarm button and PIR detector to TCW242, connect one side to the "Digital In" terminal and the other side to the "GND" terminal, as illustrated in a picture:

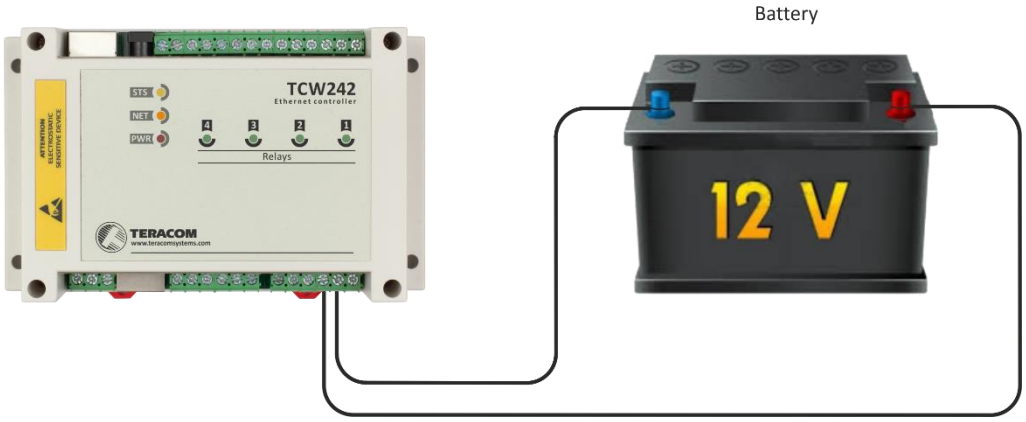


For COUNTER mode, the inputs can monitor devices with a "dry contact" pulse output interface. The maximum cable length should not exceed 30 meters.

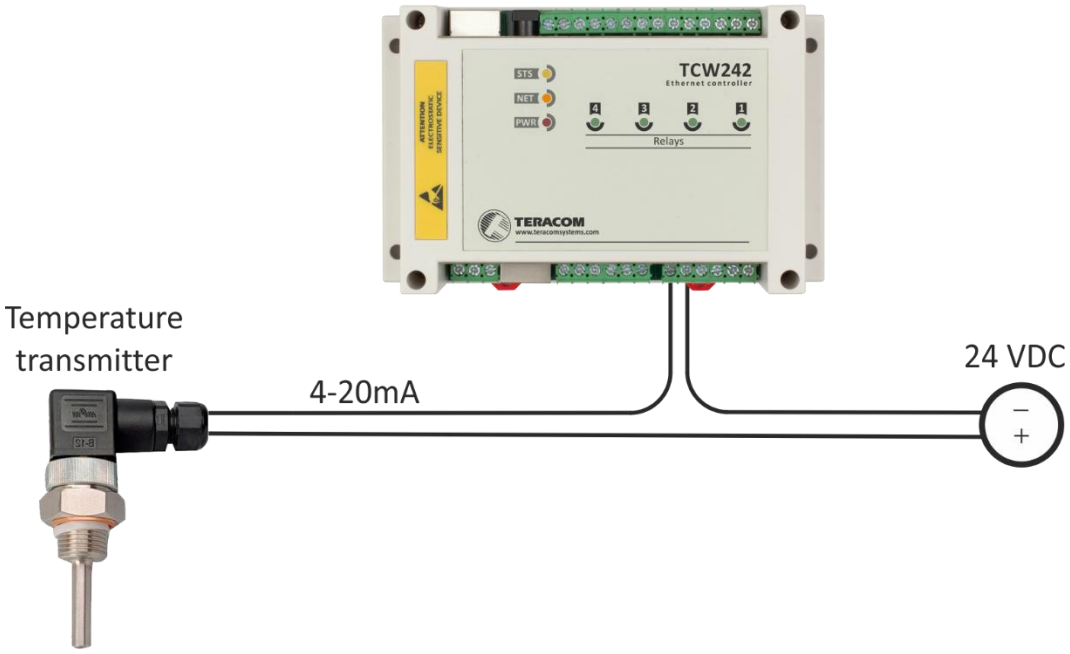
6.2.3. Analog inputs connection

Note that all inputs are not isolated from the power supply.

The analog inputs of TCW242 can be used to monitor devices with voltage and current loop outputs and connect directly to sensors, such as temperature and humidity sensors, voltage and current transducers, etc. The illustration shows how to connect a battery to the analog input of TCW242 in voltage mode, with the positive terminal connected to "Analog In" and the negative terminal to "GND."



The illustration shows how to connect a temperature sensor with a current loop output to the analog input. Connect the active terminal to "Analog In" and the shield terminal to "GND."



The maximum cable length should not exceed 30 meters.

6.2.4. RS-485 connection

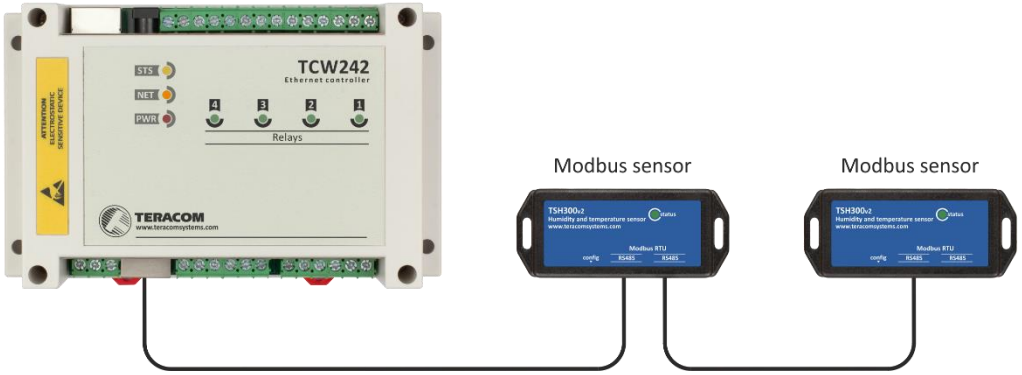
The RS-485 interface is not isolated from the power source.

Up to 24 MODBUS RTU sensors can be connected to TCW242, including both Teracom and third-party sensors, using a standard RJ-45 connector with the recommended pinout found in the "MODBUS over Serial Line Specification and Implementation Guide" on www.modbus.org.

A 120-ohm line terminator must be used at both ends of the bus, with TCW242 incorporating one terminator and the client responsible for the other.

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

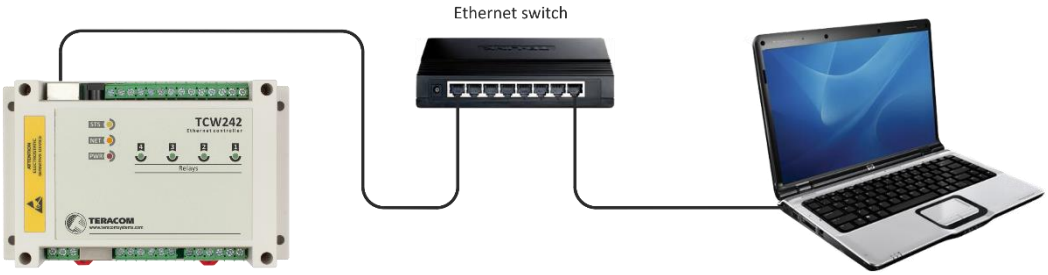
RS-485 interface is non-isolated from the power supply.



It is advised to utilize only UTP/FTP cables and limit the overall cable length to 30 m, although longer distances have been proven to work as well.

6.2.5. Ethernet connectivity

TCW242's Ethernet port should be connected to a 10/100 Base-T Ethernet hub, switch, or router.



Direct connection to a computer's Ethernet port is possible for configuration purposes, and TCW242 supports Auto-MDIX, allowing you to use either a standard straight-through or a crossover cable.

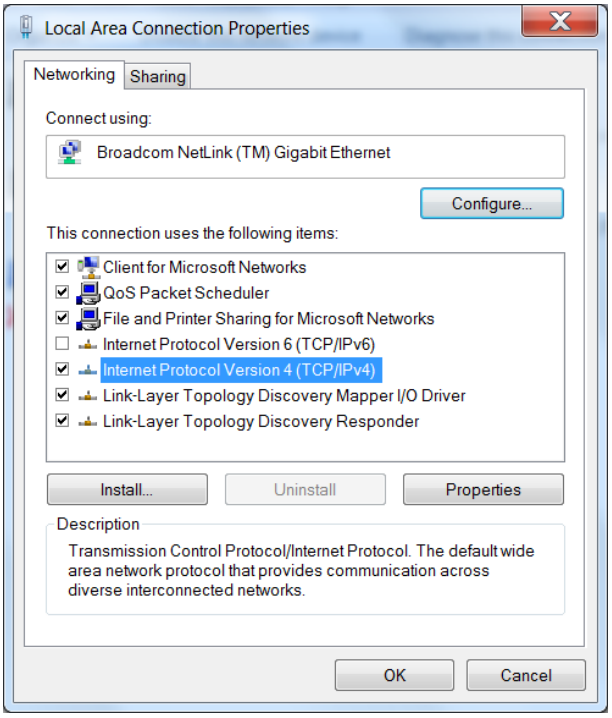
TCW242 can be integrated into a wireless network by connecting to a wireless router.

6.3. Communication setup

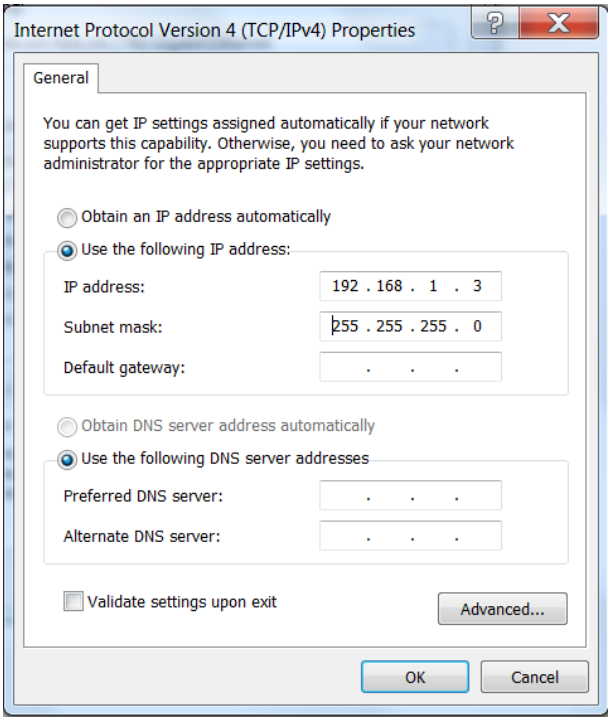
By default, TCW242 is delivered with these network settings:

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

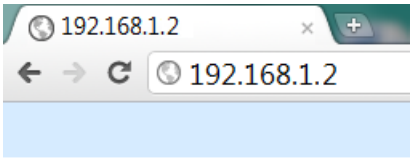
To communicate with TCW242, assign a temporary IP address to your computer. For Windows, it can be done in "Local Area Connection Properties":



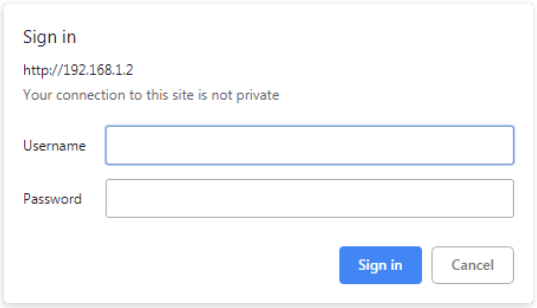
The assigned address should be on the same network, e.g. 192.168.1.3:



To access the web interface, type http://192.168.1.2 into your browser.



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



Authorization is required, the default login is admin/admin.

Change username/password for security.

You can easily locate all TCW controllers connected to the LAN using the free tool "TCW discoverer." It is compatible with Windows and Mac operating systems and can be downloaded from www.teracomsystems.com.

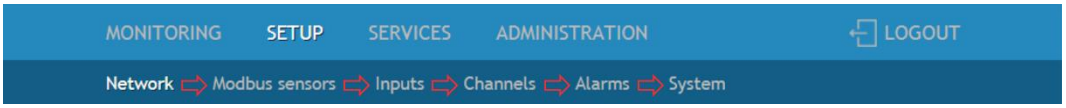
7. Concept of setup

The primary setup method of the device is via its web interface, although additional settings can be made through SNMP and HTTP API commands.

First, network settings are configured, then primary parameters such as MODBUS RTU sensors, analog inputs, and digital inputs are set up. Next, channels are created from the primary parameters, allowing up to two parameters and constants to be combined with mathematical operations, or consisting of just one primary parameter.

The alarm setup comes after tuning the channels. It's crucial to note that alarms operate with channels, not primary parameters. An alarm can have a maximum of two conditions and can involve multiple channels. Alarms are standalone but can be assigned to a channel.

Hence, the proper setup order is as follows:



Finally, desired services such as data logging, SNMP, HTTP API, etc. can be enabled.

8. Web interface

The web interface provides both configuration and monitoring capabilities. Multiple active sessions are supported by the controller. The web interface only supports HTTP, HTTPS is not supported.

8.1. Monitoring

In the monitoring section, the status of all channels, relays, and alarms are displayed both textually and graphically. The "Data" and "Alarms" pages refresh automatically at an interval of 0 to 253 seconds, with zero meaning no refresh. This interval can be set in the "Setup-System-Refresh of channels and alarms pages" section, with a default of 1 second.

8.1.1. Channels and relays

This page shows the current values and alarm status of all monitored channels and relays. The information is updated at the refresh interval.

Channels

Channel	Description	Value	Unit	Status
1	V01-Analog	3.054	V	Normal
2	V02-Temperature	30.424	°C	Warning
3	V03-Humidity	52.147	%RH	Minor
4	V04-D1	20011		Critical

Relays

Relay	Description	Status	Control
1	Relay 1	ON	ON OFF Pulse
2	Relay 2	OFF	ON OFF Pulse
3	Relay 3	OFF	ON OFF Pulse
4	Relay 4	OFF	ON OFF Pulse
			All On All Off Pulse All

8.1.2. Alarms

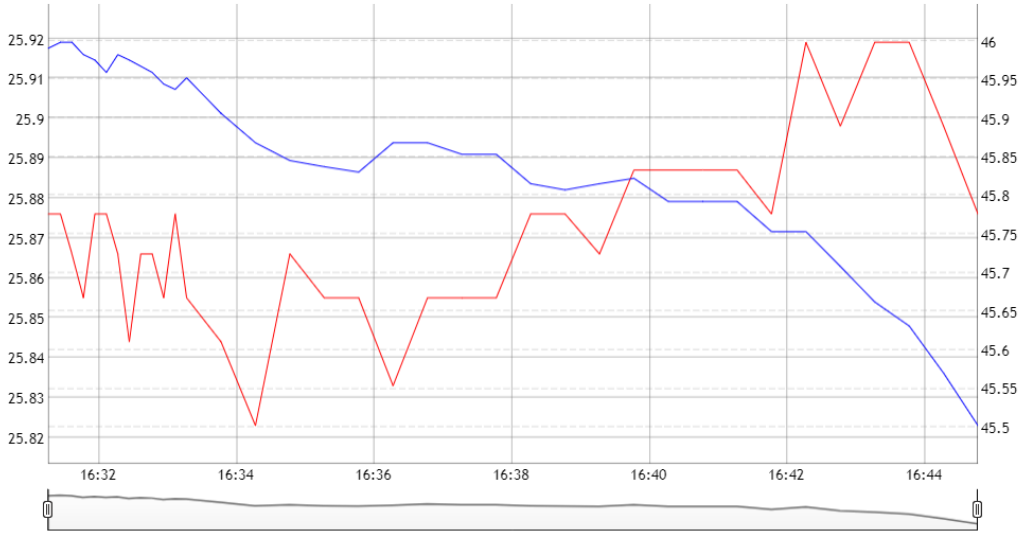
The page shows the status of all alarms. The information is updated at the refresh interval.

Alarms	Description	Status
1	AL01-AN	Normal
2	AL02-T	Warning
3	AL03-H	Minor
4	AL04-D1	Critical

8.1.3. Graphs

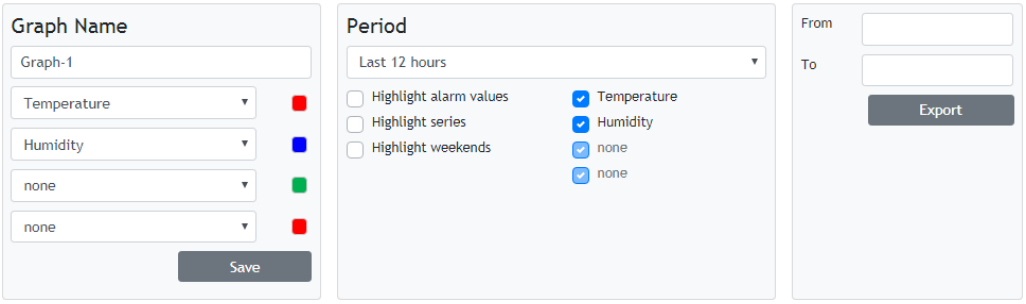
Channels and their alarm statuses can be visually monitored through tabs Graph-1 to Graph-6.

Each graph supports up to 4 channels with 2 different dimensions, and the curve color for each channel can be selected. Alarm statuses are shown with fixed colors. Modify the display with checkboxes.



To view past information, the data logger must be active.

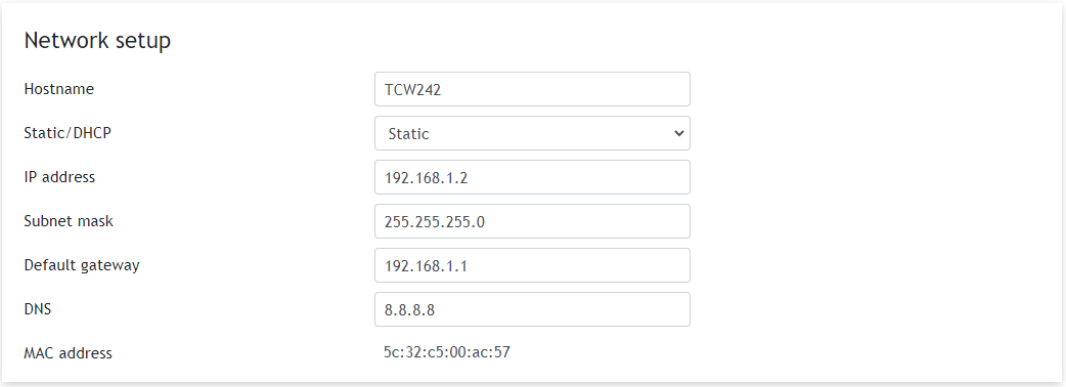
Note that graph information is static and not updated in real time. To view the latest information, reload the page. Exporting information as a CSV file is possible.



8.2. Setup

8.2.1. Network

The network parameters can be configured on this page, including static and dynamic IP address assignments.



It's recommended to change the default IP of the controller after the initial power-up to prevent collisions on the same network. Clearing the ARP cache by typing "arp -d" in the computer's command prompt may be necessary after connecting a new device.

The hostname, with a maximum length of 15 characters, is displayed in TCW discoverer search results. Using a public DNS server such as 8.8.8.8 or 8.8.4.4 is recommended over the default gateway.

8.2.2. Modbus sensors

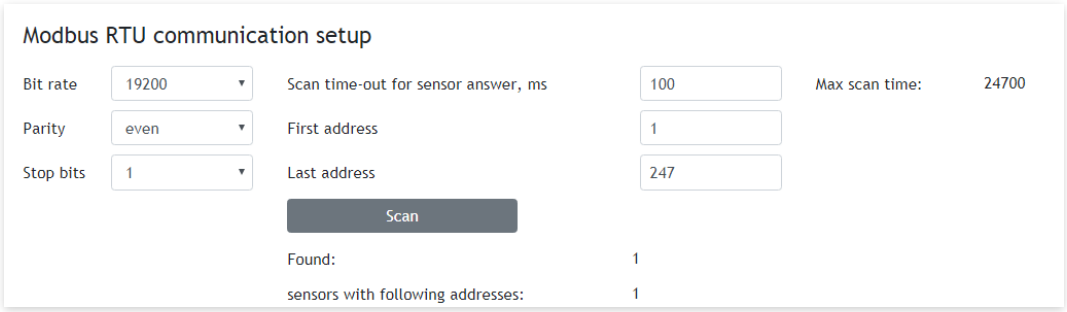
8.2.2.1. Modbus RTU communication configuration

This page enables you to configure the RS-485 interface communication parameters such as bit rate, parity, and stop bits. The default settings are 19200, even parity, and 1 stop bit.

All sensors on the bus must have the same bit rate, parity, and stop-bit settings. Before adding any sensors, ensure their parameters are properly configured.

A bus scanning tool is available to report the number of detected sensors and their addresses. This tool is useful when adding new sensors and it is recommended to use a small address range to speed up the scanning process.

You can adjust the "Scan Timeout for Sensor Answer" to find the optimal setting for an unknown sensor. The test begins with a large timeout (e.g. 500ms) and gradually reduces the time until the sensor stops responding. To ensure stability, increase the found timeout by, for example, 20%.

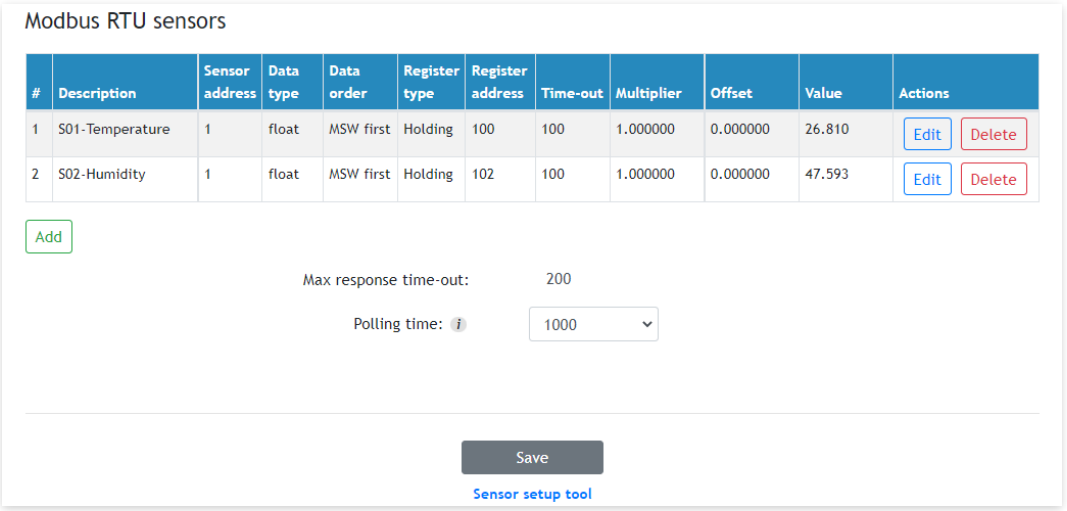


8.2.2.2. Modbus RTU sensors

This section allows for the management of MODBUS RTU sensors/registers, including adding, deleting, or editing them. These serve as the primary parameters that can be utilized in creating channels.

It is advisable to add sensors/registers individually using the scan tool outlined in 8.2.2.1.

A maximum of 24 sensors/registers can be added and displayed in a table.



According to the MODBUS convention, the address range for slaves is 1 to 247.

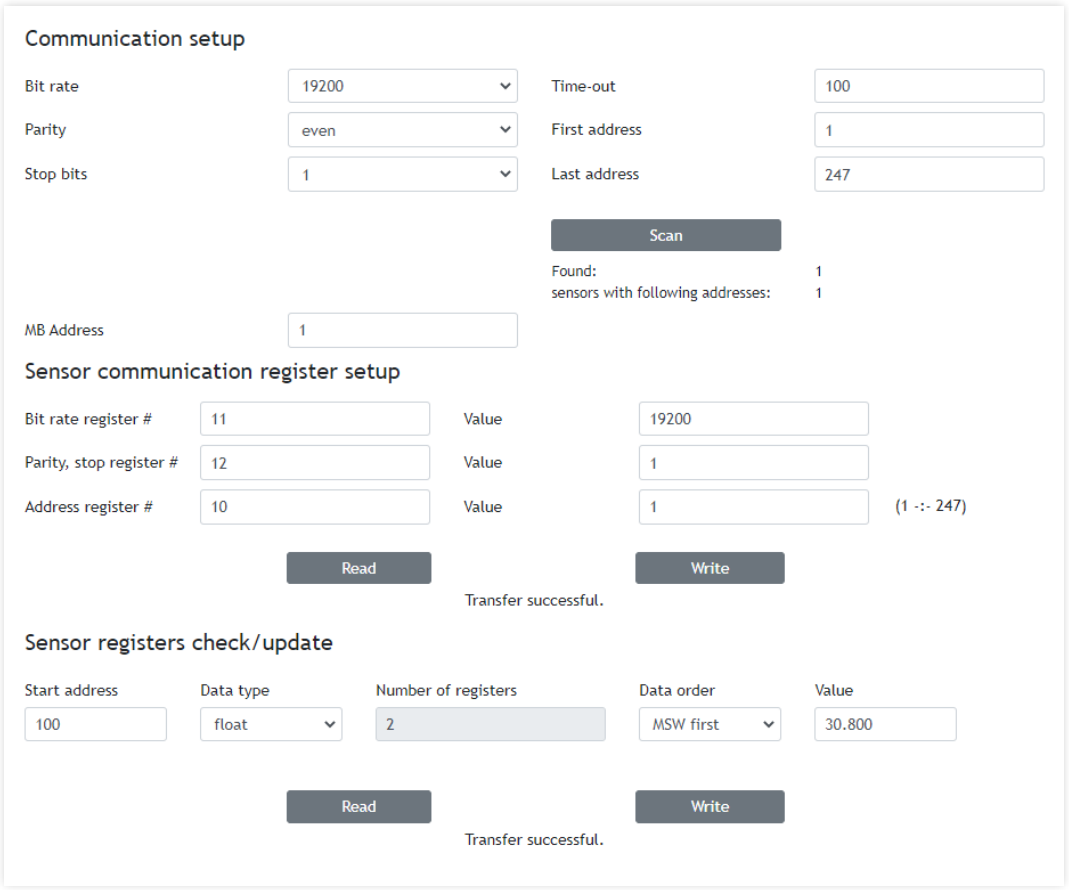
The value of the sensor can be calculated using the formula: Value = (Raw_Value * Multiplier) + Offset. To view the raw value, set Multiplier to 1 and Offset to 0.

The controller continuously polls all sensors and expects a response within a specified "response time-out". If the same sensor fails to respond three times, it is considered not present. The "maximum response time-out" for the system is determined by the sum of response time-outs for all sensors, and the system's response is based on this value. It is strongly recommended that "Polling time" be greater than or equal to "Maximum response time-out".

The "Polling time" can be set to 1, 2, 3, 4, 5, 10, 15, 30, 60, 120, or 180 seconds and is 1 second by default.

8.2.2.3. Sensor setup tool

The sensor setup tool allows you to configure and control MODBUS RTU sensors from different manufacturers by changing their addresses and communication parameters. It is a simple yet useful tool.



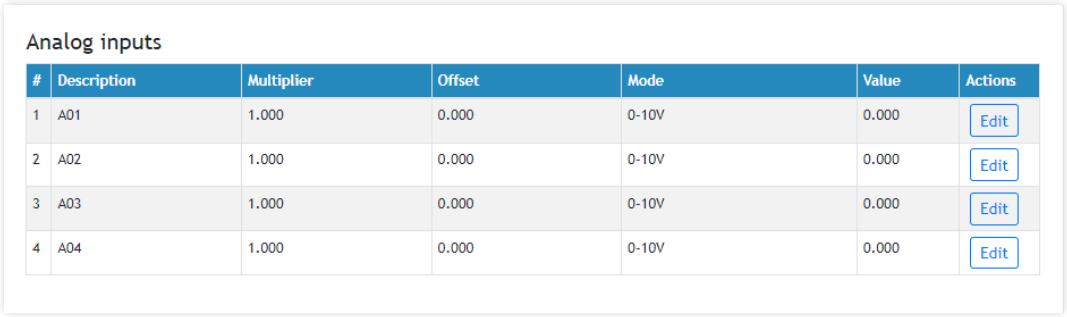
8.2.3. Input/Output

This page allows for the configuration of relays, analog, and digital inputs.

8.2.3.1. Analog inputs

The TCW242 has 4 non-isolated analog inputs. Each input can operate as either voltage (0-10V) or current loop (0-20mA). The Multiplier and Offset can be adjusted for each input with the formula: "Value = (Raw_Value * Multiplier) + Offset".

To view the raw value, set Multiplier to 1 and Offset to 0. The default settings after a "Factory default" procedure are Multiplier=1.00, Offset=0.00, and Mode=0-10V.



Example:

To get the HIH-4000-003 humidity sensor to work properly, set the following parameters based on the datasheet:

Multiplier: 31.74

Offset: 0.826

Mode: 0-10V

Unit: %RH (set on the Channels web page)

Note that the multiplier value is the inverse of the slope parameter (1/0.0315).

For example, if the sensor's output voltage is 3.198V, the analog input value will be 75.28% RH:

75.28 = (3.198 - 0.826) * 31.74

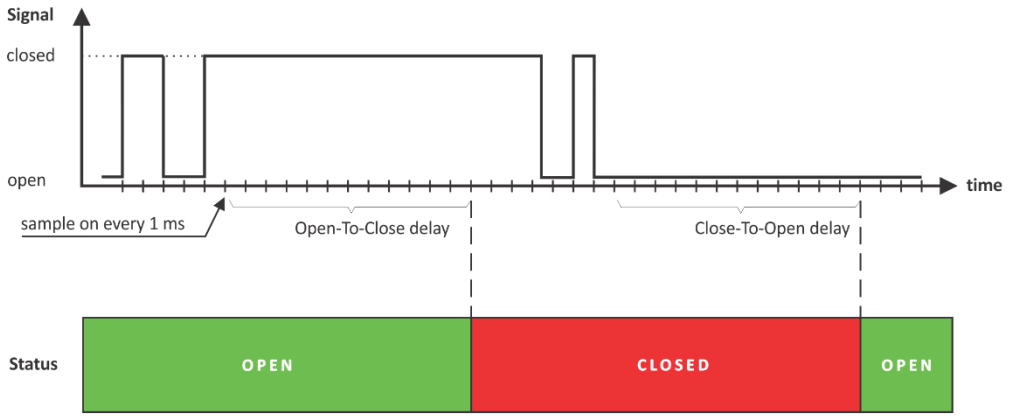
8.2.3.2. Digital inputs

The TCW242 features 4 non-isolated digital inputs. Each input can operate as either OPEN/CLOSE or COUNTER mode. In the COUNTER mode, counting can occur on rising, falling, or both edges, with the ability to set an initial counter value.

Digital inputs

#	Description	Closed state	Open state	C/O delay	O/C delay	Mode	Value	Actions
1	D1	CLOSED	OPEN	5	5	Discrete(Open/Closed)	OPEN	Edit
2	D2	CLOSED	OPEN	5	5	Counter(Rising edge)	10158	Edit
3	D3	CLOSED	OPEN	5	5	Counter(Falling edge)	11	Edit
4	D4	CLOSED	OPEN	5	5	Counter(Both edges)	16	Edit

There are two delays, Open-to-Close and Close-to-Open, that can be set between 5ms and 60000ms for additional digital filtering and are applicable in both modes.



In the illustration, the Open-to-Close and Close-to-Open delays are set to 13ms.

8.2.3.3. Relay outputs

The TCW242 has 4 relay outputs which can be controlled through the WEB interface, HTTP API, SNMP, or locally using alarm conditions. The WEB control for each relay offers "On", "Off", and "Pulse" buttons, as well as "All On", "All Off", and "Pulse All" for collective control. The pulse duration can be set individually for each relay in the "Setup-Input/Output-Relay Outputs" section.

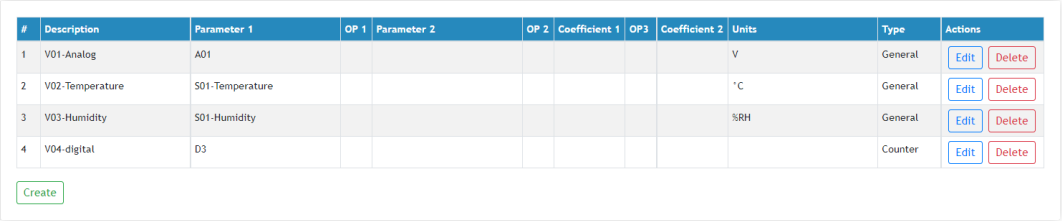
Relays

Relay	Description	Status	Control
1	Relay 1	ON	ON OFF Pulse
2	Relay 2	OFF	ON OFF Pulse
3	Relay 3	OFF	ON OFF Pulse
4	Relay 4	OFF	ON OFF Pulse
			All On All Off Pulse All

For local activation, a description of the controlling parameter is displayed instead of buttons, and the parameters for local activation can be set in the "Setup-Input/Output-Relay Outputs" section.

8.2.4. Channels

The section allows for adding, editing, and deleting channels, with a max of 24, for monitoring and periodic data logging.



#	Description	Parameter 1	OP 1	Parameter 2	OP 2	Coefficient 1	OP3	Coefficient 2	Units	Type	Actions
1	V01-Analog	A01							V	General	Edit Delete
2	V02-Temperature	S01-Temperature							°C	General	Edit Delete
3	V03-Humidity	S01-Humidity							%RH	General	Edit Delete
4	V04-digital	D3								Counter	Edit Delete

Create

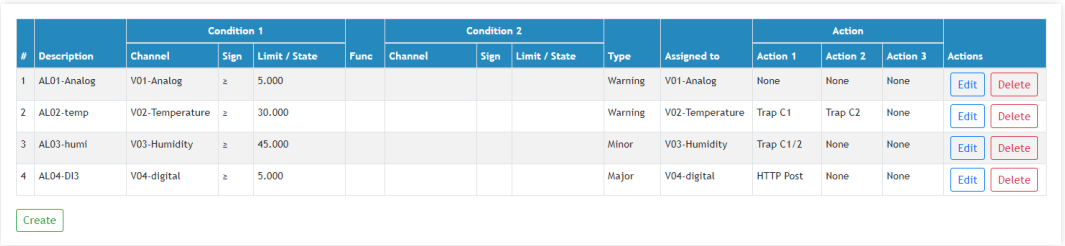
Discrete channels consist of a single digital input in OPEN/CLOSE mode, while general channels are made up of up to two primary parameters and constants.

Note that for general channels, operations are performed in the sequence of OP1, OP2, and OP3. A digital input in OPEN/CLOSE mode can also be used to form a general channel with values of 0 for CLOSE and 1 for OPEN.

8.2.5. Alarms

The Alarms section allows you to manage up to 24 alarms, including adding, deleting, or editing. Only channels can be used to create alarms, with 4 types available: Warning, Minor, Major, and Critical.

The alarms can be assigned to any channel and can have up to 2 conditions, joined by logical operators AND and OR, with no restrictions on which channel to use in each condition. It's possible to combine conditions from different channels in one alarm



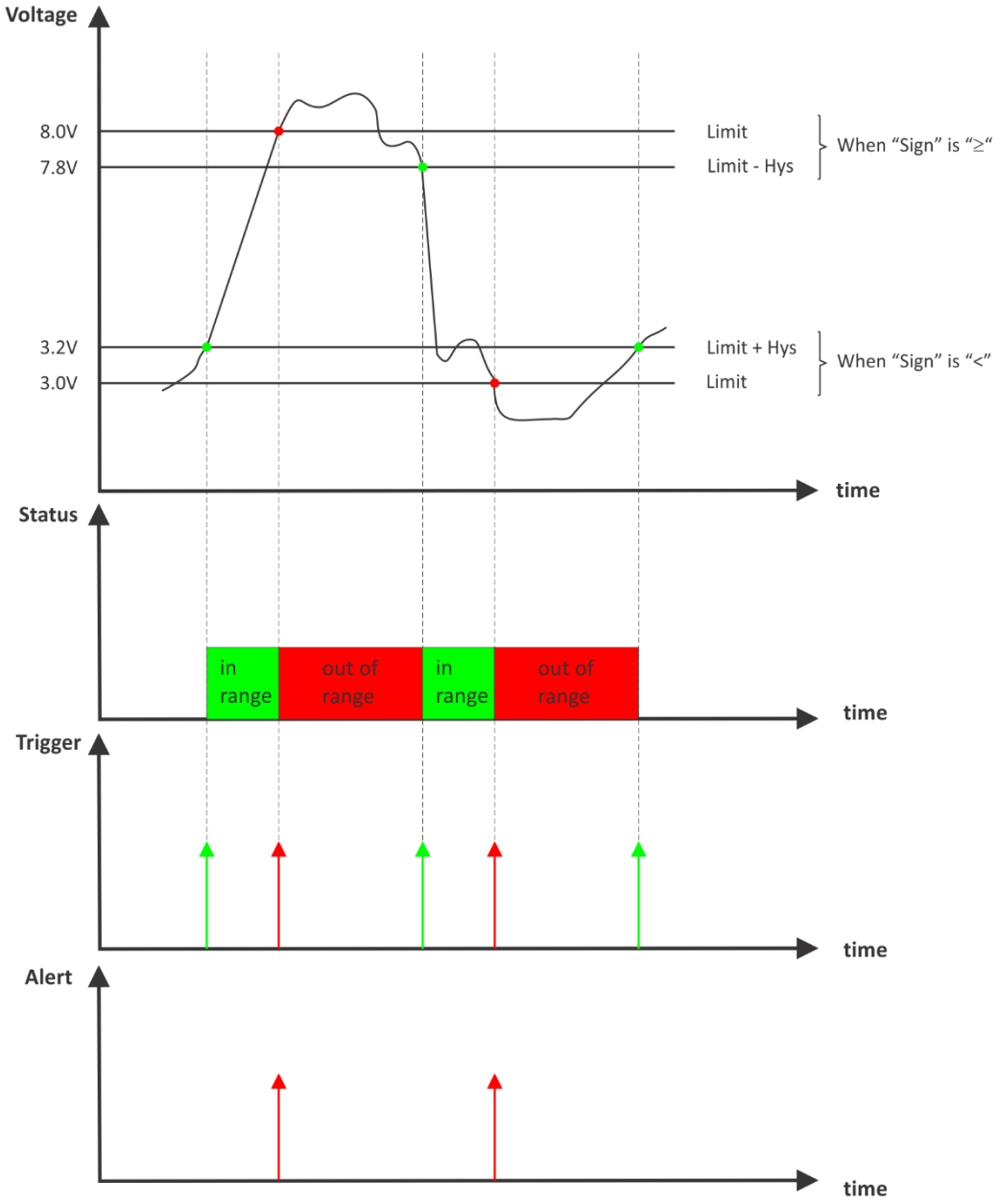
#	Description	Condition 1			Func	Condition 2			Type	Assigned to	Action			Actions
		Channel	Sign	Limit / State		Channel	Sign	Limit / State			Action 1	Action 2	Action 3	
1	AL01-Analog	V01-Analog	≥	5.000					Warning	V01-Analog	None	None	None	Edit Delete
2	AL02-temp	V02-Temperature	≥	30.000					Warning	V02-Temperature	Trap C1	Trap C2	None	Edit Delete
3	AL03-humi	V03-Humidity	≥	45.000					Minor	V03-Humidity	Trap C1/2	None	None	Edit Delete
4	AL04-DI3	V04-digital	≥	5.000					Major	V04-digital	HTTP Post	None	None	Edit Delete

Create

The "Limit" sets the boundary for the normal operating range of the monitored channel. If the channel value exceeds the limit (exceeds it by going higher with "Sign" set to "≥" or goes lower with "Sign" set to "<"), a trigger event occurs.

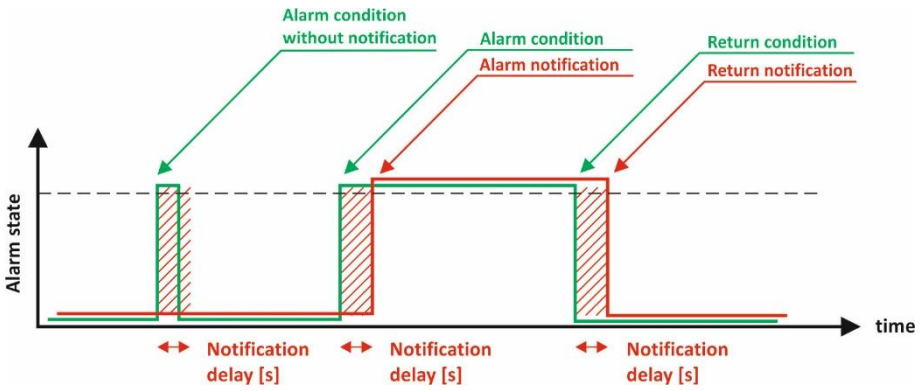
The trigger event is considered resolved when the value returns within the range, either by going higher than the Limit + Hysteresis with "Sign" set to "<" or lower than Limit - Hysteresis with "Sign" set to "≥".

Hysteresis helps prevent frequent triggering due to small fluctuations around the trigger point and it is highly recommended to not use a value of 0.0 for Hysteresis.



Every alarm has a "Return notification" option, enabling a notification when the parameter returns to the normal range.

Additionally, each alarm has a "Notification delay" parameter, serving as a filter for brief alarm occurrences.

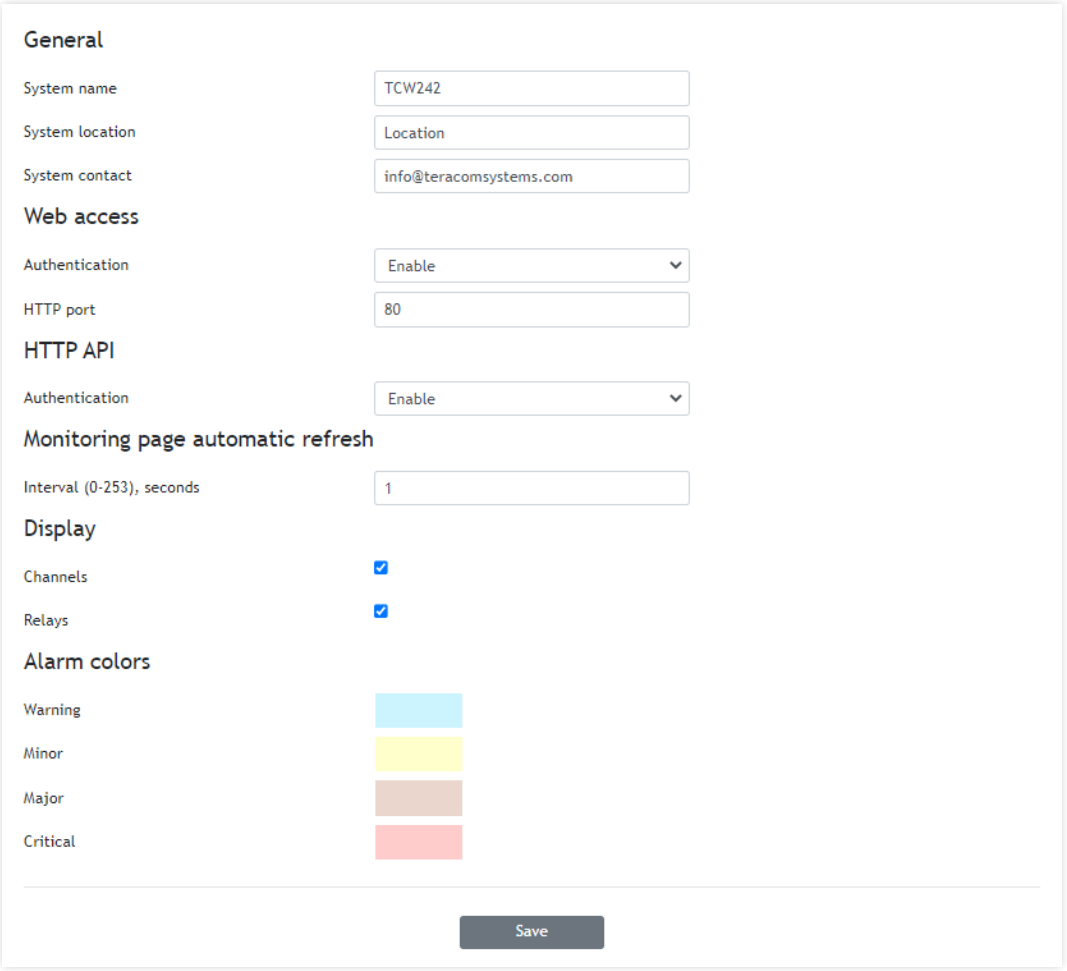


8.2.6. System

On the "System" page, you can configure general system settings.

This includes the system name, location, and contact for device identification, which appear in SNMP and XML/JSON status files.

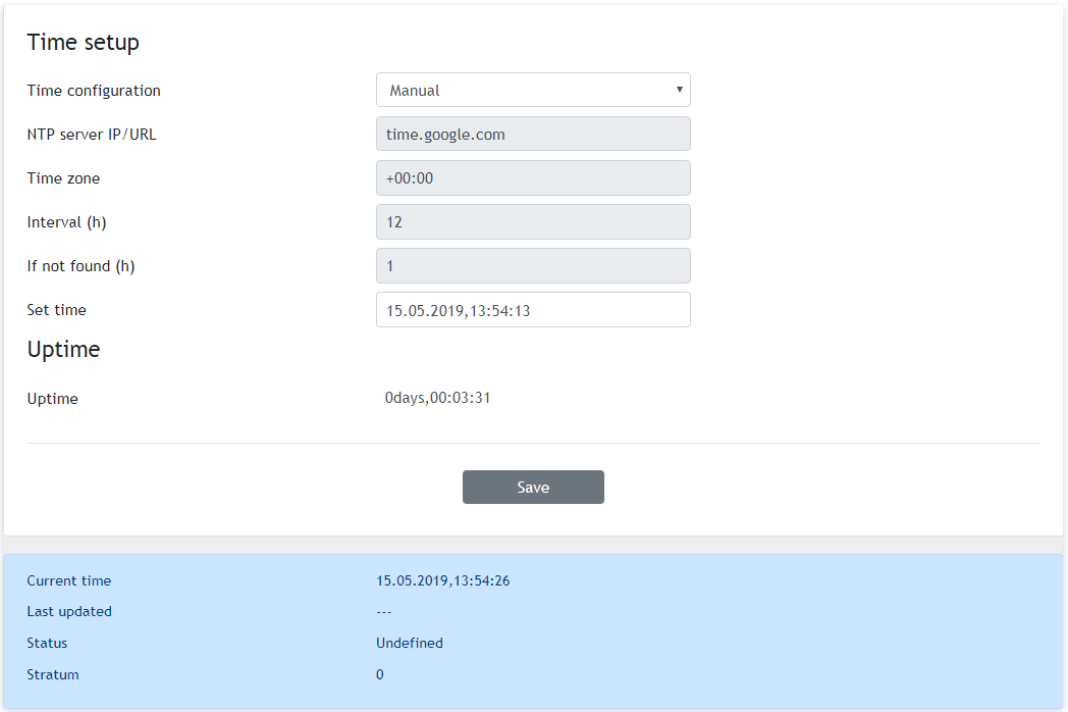
The default WEB access requires authentication with the username and password "admin/admin". The default HTTP port for WEB access is 80, but it can be changed for routers that don't support different ports for port forwarding. The HTTP API also requires authentication with the same username and password as WEB access. The controller supports two types of authentication. The refresh interval, which can range from 0 to 253 seconds (0 meaning no refresh), can also be set. The color coding for alarms is fixed and is shown for information purposes.



8.3. Services

8.3.1. NTP

The internal clock of the controller can be set either manually or using NTP (Network Time Protocol).



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

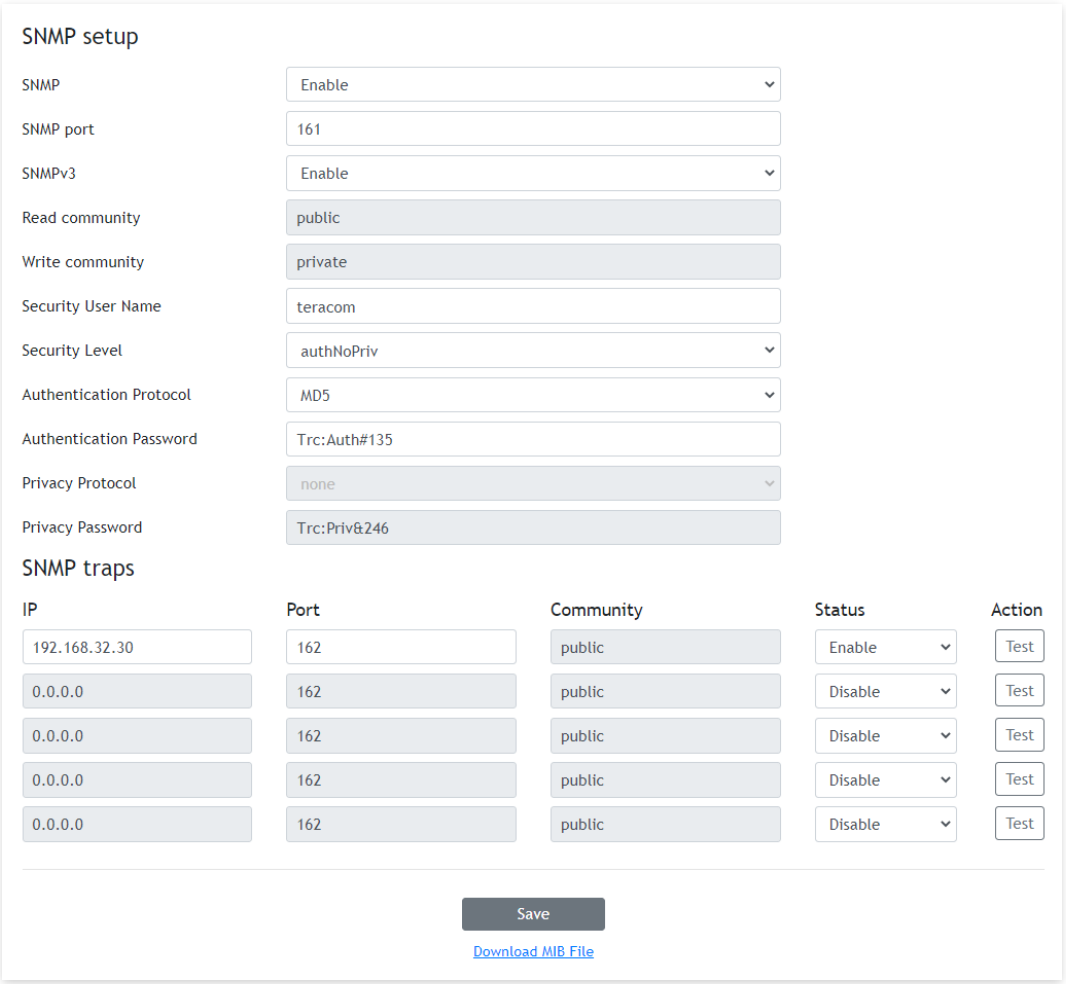
8.3.2. SNMP

TCW242 supports SNMPv2 and SNMPv3 and the default settings are:

- SNMP disabled
- Port 161
- SNMPv3 disabled
- read community public
- write community private
- Security User Name teracom
- Security Level noAuthNoPriv
- Authentication Protocol none
- Authentication Password Trc:Auth#135
- Privacy Protocol none
- Privacy Password Trc:Priv&246

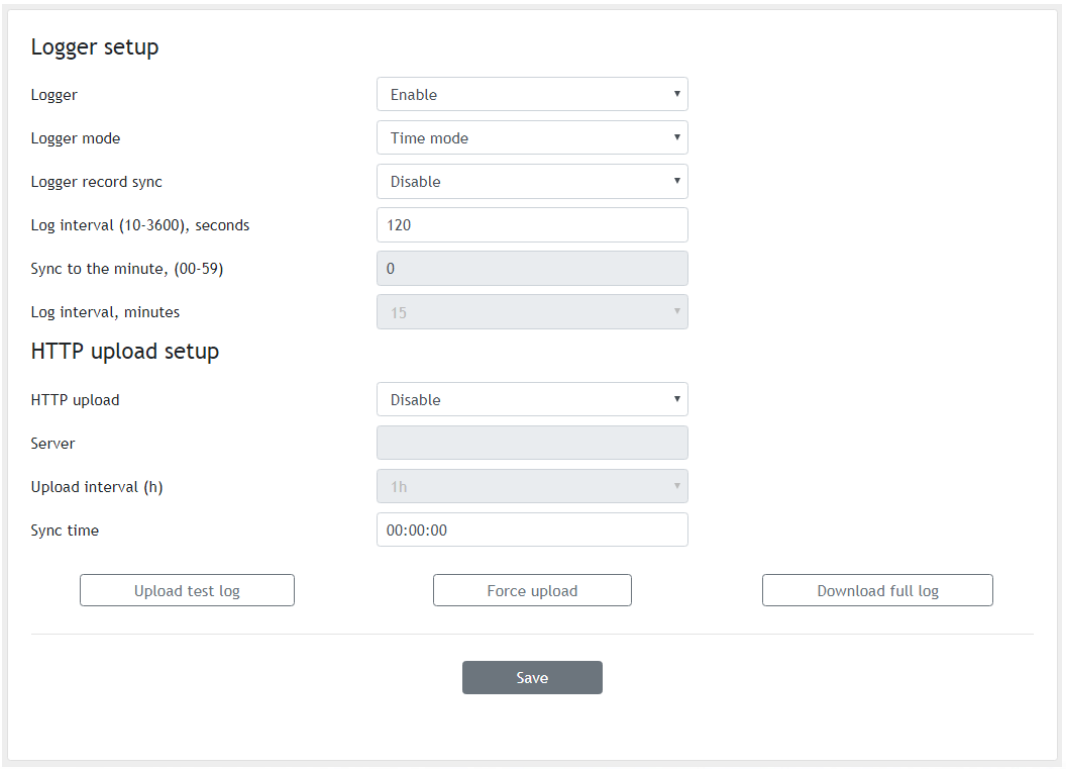
SNMPv3 provides improved security features including authentication and privacy options (noAuthNoPriv, authNoPriv, authPriv). SNMPv3 also supports user-based authentication mechanisms (MD5, SHA, none) and privacy mechanisms (DES, AES, none).

For alarm notifications, up to 5 independent recipients can receive SNMP traps, each with its own port and community. A trap is also sent after reset. The MIB file can be downloaded from the web page.



8.3.3. Logger

The logger operates in three modes: Time, Alarm, and Time & Alarm. The mode determines the trigger for creating a record in the logger's memory. In Time mode, records are made at regular intervals based on the "Log Interval." In Alarm mode, records are made whenever an alarm condition is met. In Time & Alarm mode, a combination of both triggers is used.



The time between two log entries is determined by the log interval. Lowering the log interval increases the accuracy of the records, but reduces the amount of history captured. The logger can be synced to a specific minute in an hour, useful for monitoring utilities like electricity and gas. The log interval can be set from 1 to 60 minutes in a drop-down menu. "Sync to the minute" specifies the minute for syncing and it's best 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.

With the current settings, 4 records will be recorded per hour at HH:00, HH:15, HH:30, and HH:45. The device has been powered up at 09:12, so the first record will be taken immediately after power-up at 09:12. The subsequent records will be taken at 09:15, 09:30, 09:45, 10:00, 10:15, and so on.

Data can be accessed in two ways: downloading the full log file through the WEB interface or uploading the last un-sent records to a designated HTTP server. The upload frequency can be set to any value between 1 and 24 hours, but real-time clock synchronization is necessary (using the NTP service). The HTTP server can be either a domain or IP address, but proper DNS settings are important.

The "Sync Time" is the time of day when the upload period is synchronized.

For example, if the current time is 19:31, the upload period is 3 hours, and the Sync Time is 9:00, then the upload times will be: 9:00, 12:00, 15:00, 18:00, 21:00, 24:00, 3:00, and 6:00. The first upload after enabling the logger at 19:31 will occur at 21:00.

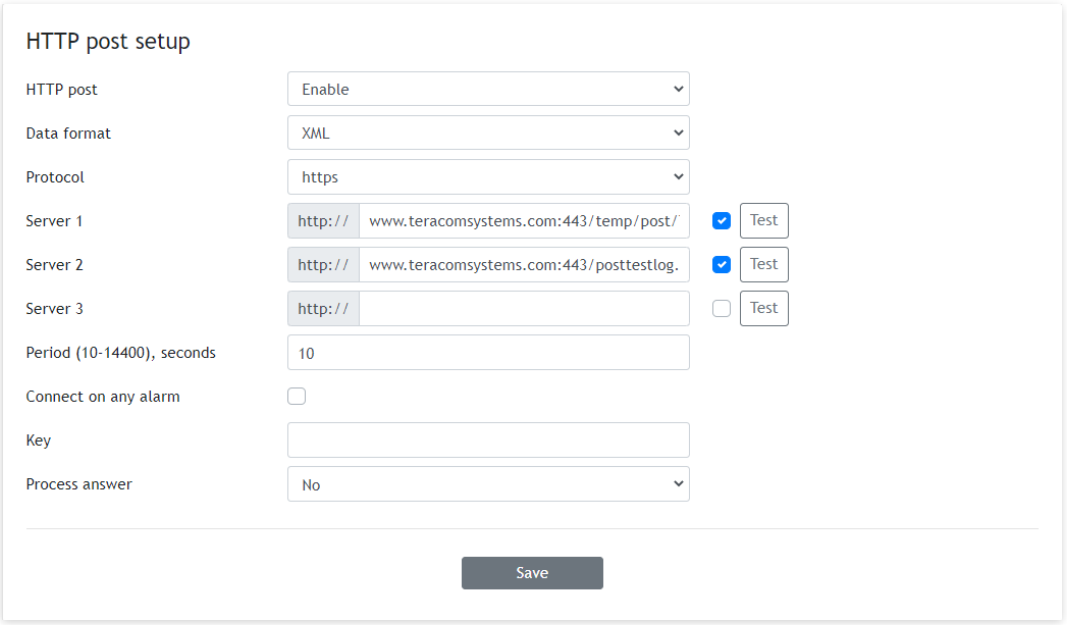
The "Force Upload" button triggers an upload of recorded data between the previous periodical upload and the present moment.

By default, the logger is disabled.

Additional information on the logger can be found in the "Datalogger" section.

8.3.4. HTTP Post

TCW242 can regularly transfer an XML/JSON file to a designated server through HTTP or HTTPS Post. HTTPS uses TLS 1.0, 1.1, or 1.2 with RSA key exchange/agreement and authentication. The XML/JSON file includes current information on all monitored parameters and additional system details. You can select the file format from a drop-down menu.



TCW242 can send an XML/JSON file to 3 servers via HTTP or HTTPS Post. HTTP servers can be addressed by domain name or IP address.

The "Period" of sending, which can range from 10 to 14400 seconds, can be changed remotely via HTTP API. Shortening the "Period" leads to closer real-time operation, but also higher data traffic.

If the "Connect on any alarm" box is checked, an HTTP/HTTPS Post request will be sent during any alarm.

The "Key" field is user-defined and sent in the XML/JSON file for device identification.

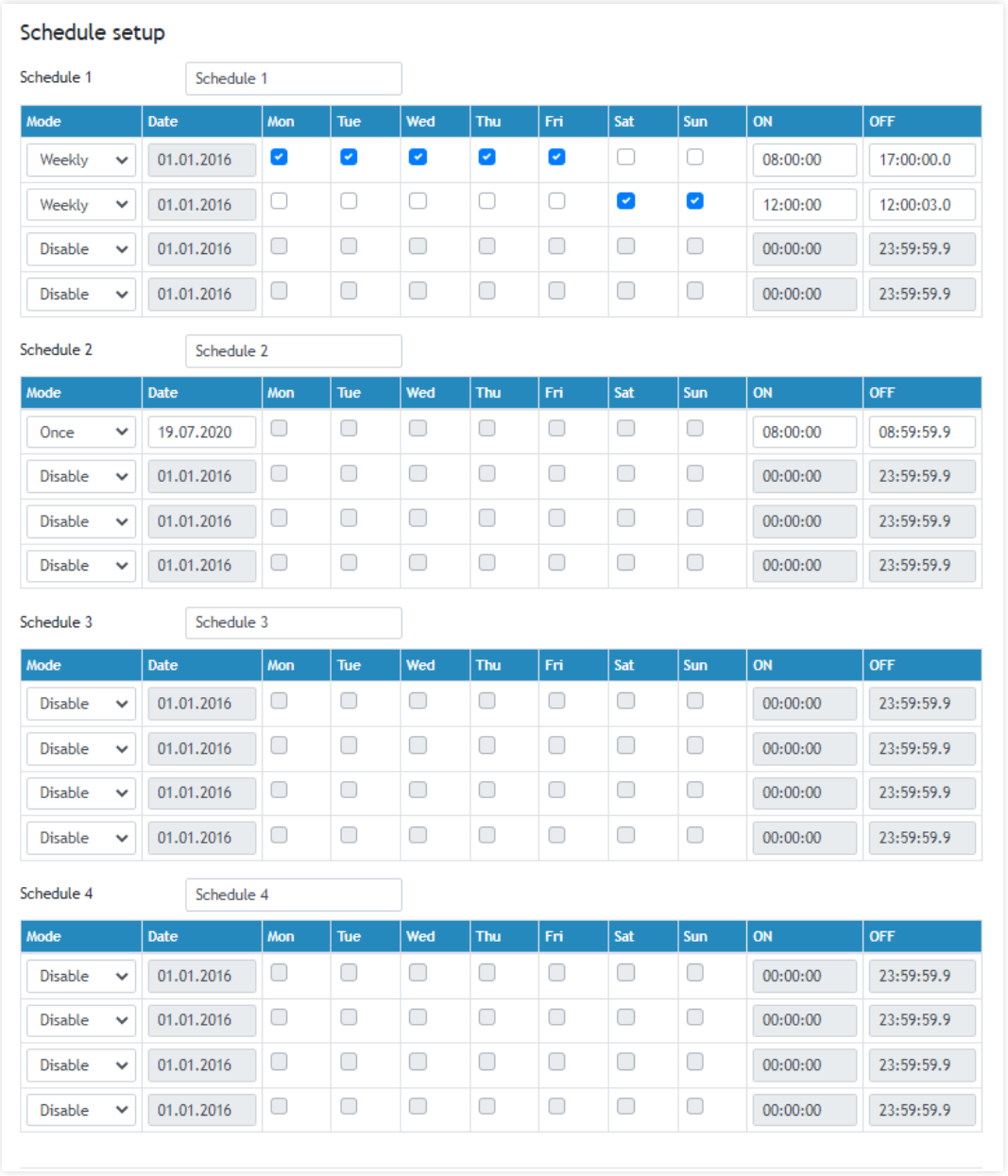
If "Process Answer" is enabled, TCW242 will execute commands sent by the remote server as a response to HTTP/HTTPS Post.

Learn more about HTTP/HTTPS Post in the HTTP API section.

8.3.5. Schedule

TCW242 has 4 schedules, each with up to 4 tasks.

The schedules allow you to create tasks based on calendar dates and can be combined with alarms for more advanced control. Two relays can control a single device, with one following the alarm and the other following the schedule.



TCW242 has two types of schedules: single and weekly.

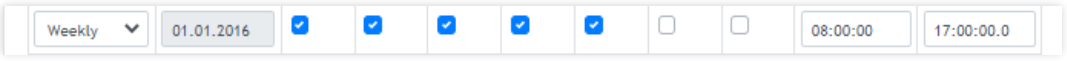
There are two types of schedules depending on repetition and duration – single and weekly tasks.

Here are some examples:

- A single task for a time period:

  An example of a single task with a set time period could look like this: Event on 06.04.2020 from 08:00 to 09:00, with a resolution of 0.1 seconds for "OFF time." This allows for short pulse support.

- A weekly task for a time period:



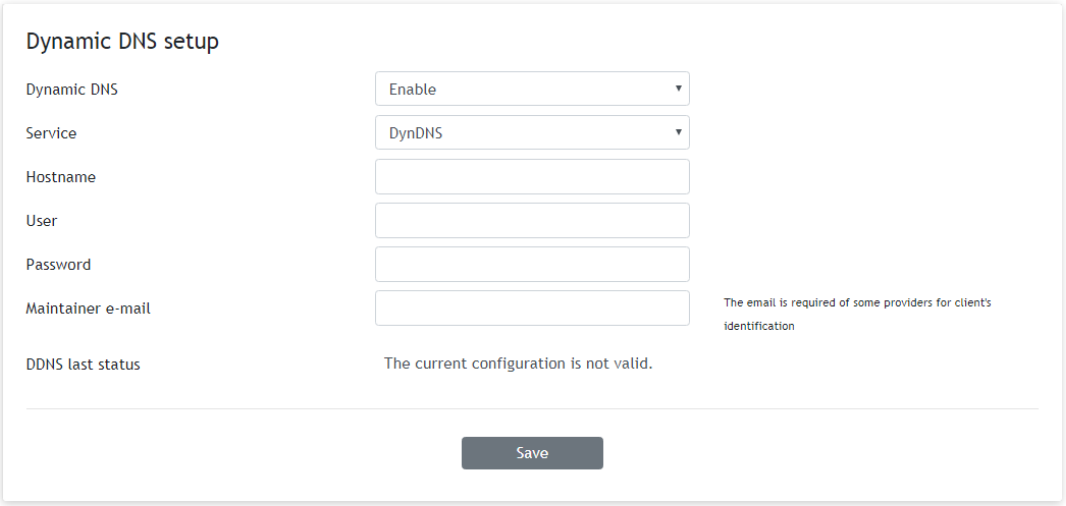
  With the above setting, an event will occur every weekday starting at 08:00 and ending at 17:00.

- A weekly task for a time period which includes midnight:

With the above configuration, there will be an event lasting 2 hours, occurring from Monday 23:00:00 to Tuesday 01:00:00.

8.3.6. Dynamic DNS

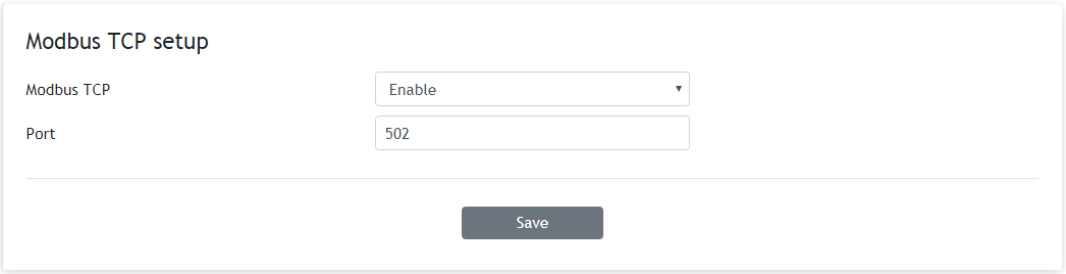
TCW242 supports DynDNS, No-IP, and DNS-O-Matic for dynamic DNS.



This enables access to TCW242 from the internet using only a dynamic public IP address.

8.3.7. MODBUS

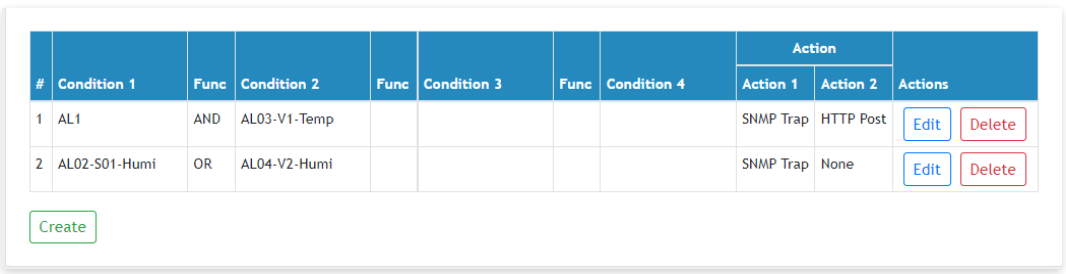
TCW242 supports MODBUS over TCP/IP, using port 502 as the default.



Modbus is disabled by default, but more information about this feature can be found in the MODBUS section.

8.3.8. Functions.

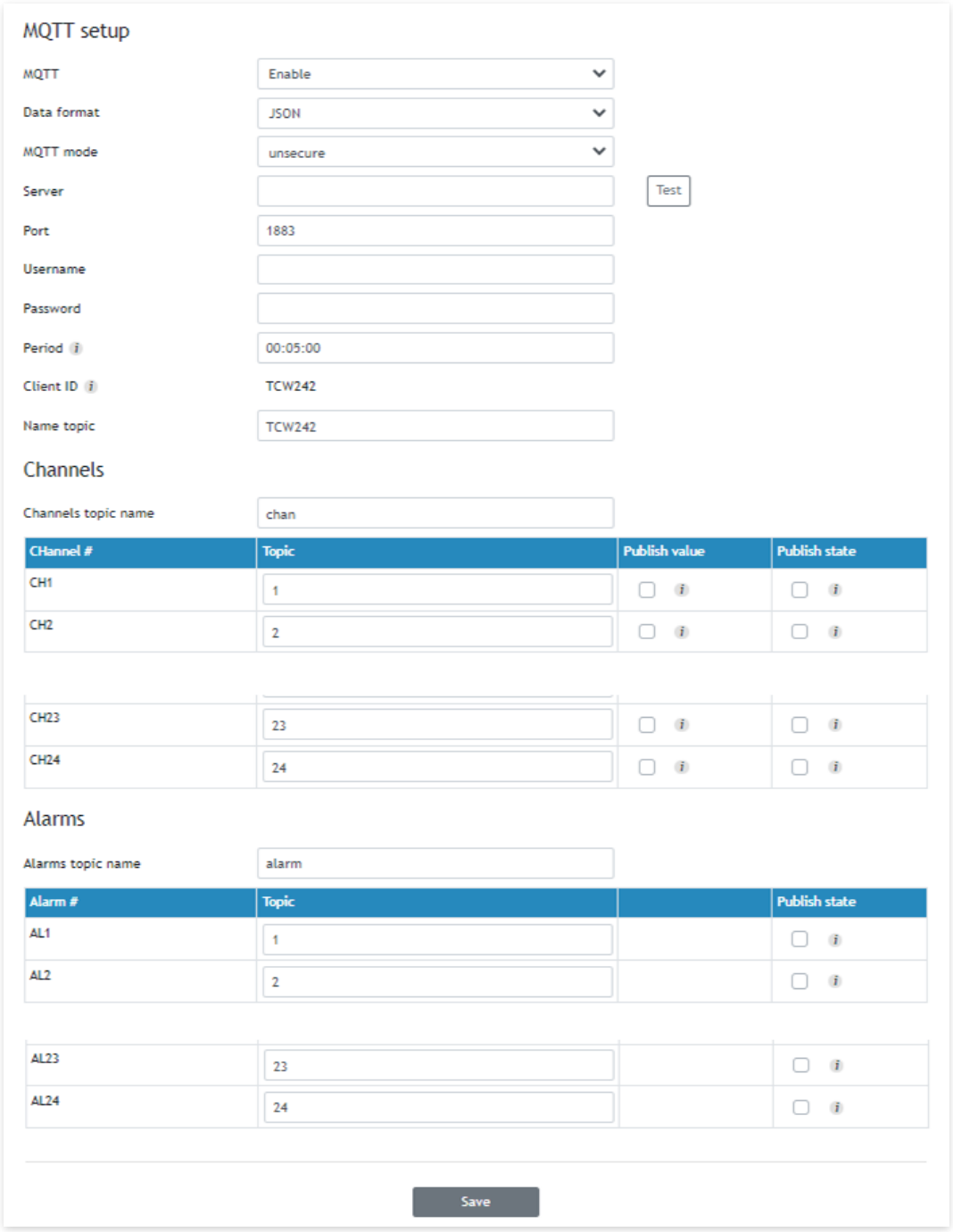
This section allows you to set up four separate functions, each of which can include up to four alarm conditions connected by AND and OR operators. The functions can be selected from a drop-down menu to activate local relays. Each function has parameters for "Action delay" and "Action on return".



8.3.9. MQTT

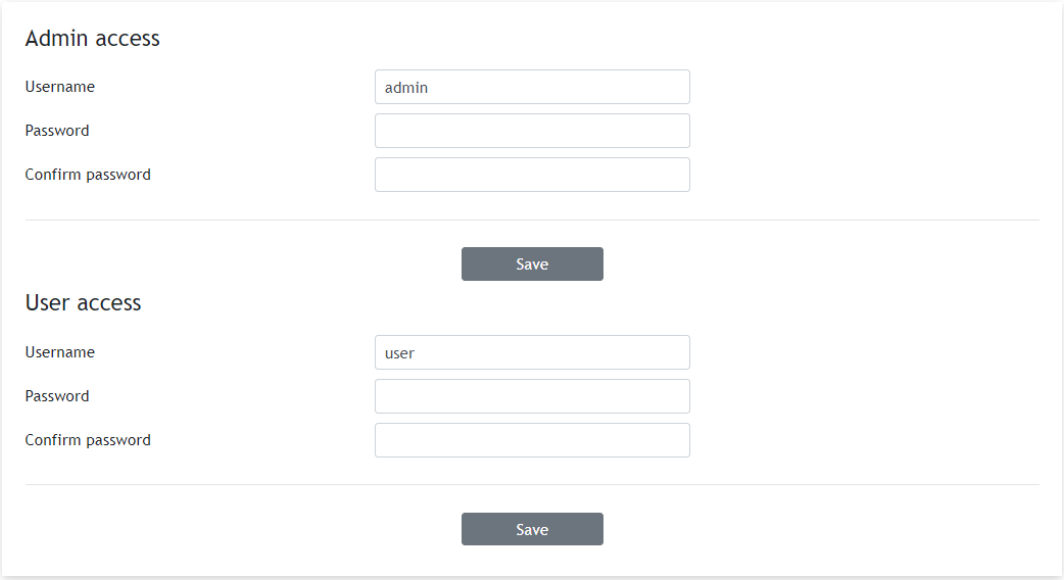
The device supports MQTT 3.1.1 in unsecured and encrypted (SSL/TLS) communication, using JSON and XML data format.

There are independent topics for every channel and alarm.



8.4. Administration

8.4.1. User/Pass

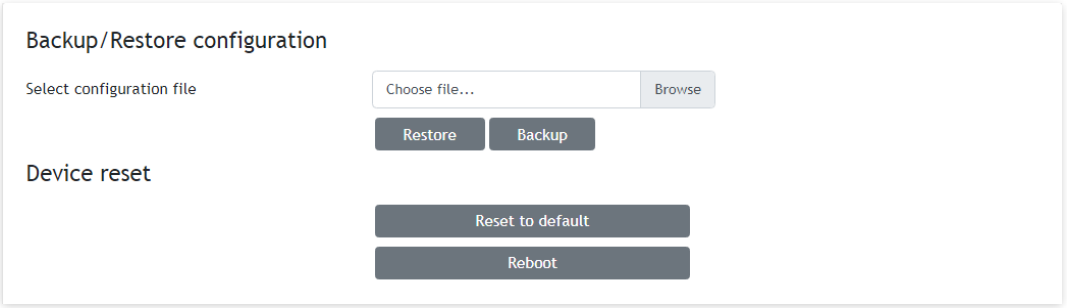


The TCW242 has two user levels: "Admin" with full control and "User" with limited access.

Both usernames and passwords can be up to 31 characters long.

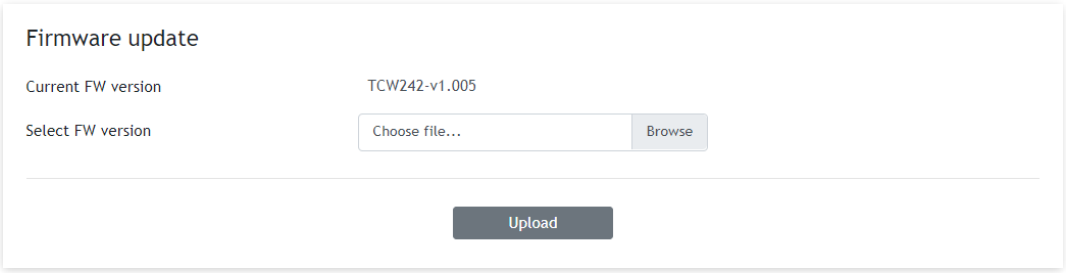
8.4.2. Backup/Restore

The TCW242 supports backing up and restoring all user settings via an XML backup file, making it useful for replicating similar settings across multiple controllers..



8.4.3. FW update

The TCW242 can be updated through its web interface.



To update the device:

- Download the latest firmware from www.teracomsystems.com

- Go to Administration->FW update and select the downloaded .cod file and press "UPLOAD"
- Wait until the update is complete.

Attention! Do not turn off the power during the update, it may damage the device

8.5. Logout

The TCW242 supports multiple sessions, but it's recommended to log out after finishing work for best practice.

9. Protocols and API

9.1. SNMP

The TCW242 supports SNMP (Simple Network Management Protocol), a standard protocol for managing IP network devices, allowing administrative computers (managers) to monitor and control them. The device can be configured and monitored using an SNMP v.2 or v.3 compatible program, with parameters grouped by function. To obtain a valid OID number, replace "!" with "1.3.6.1.4.1.38783". To save changes, set configurationSaved (OID !.8.6.3.0) to "1".

product

OID	Name	Access	Description	Syntax
!.8.1.1.0	name	read-only	Device name	DisplayString
!.8.1.2.0	version	read-only	Firmware version	DisplayString
!.8.1.3.0	date	read-only	Release date	DisplayString

setup -> network

replace "**!**" with "1.3.6.1.4.1.38783" in the table below

OID	Name	Access	Description	Syntax
!.8.2.1.1.0	deviceID	read-only	Device ID (default MAC address)	MacAddress
!.8.2.1.2.0	hostName	read-only	Hostname	DisplayString (SIZE (0..38))
!.8.2.1.3.0	deviceIP	read-only	Device IP address	IpAddress

setup -> parameters -> mbSensors -> mbSensorsTable -> mbSensorsEntry -> msSensIndex 1 to 24

replace "**?**" with a number from 1 to 24 and "**!**" with "1.3.6.1.4.1.38783" in the table below

OID	Name	Access	Description	Syntax
!.8.2.2.1.1.1.2.**?**.0	mbSenDescription.**?**	read-write	Sensor description	DisplayString
!.8.2.2.1.1.1.3.**?**.0	mbSenMult.**?**	read-write	Sensor multiplier x1000 in Integer format	Integer32
!.8.2.2.1.1.1.4.**?**.0	mbSenOffset.**?**	read-write	Sensor offset x1000 in Integer format	Integer32
!.8.2.2.1.1.1.5.**?**.0	mbSenVal.**?**	read-only	Sensor value x1000 in Integer format	Integer32
!.8.2.2.1.1.1.6.**?**.0	mbSenCounter.**?**	read-only	Sensor as 32-bit Counter	Counter32

setup -> parameters -> analogInputs -> analogInpTable -> analogInpEntry -> analogInpIndex 1 to 4

replace "**?**" with a number from 1 to 4 and "**!**" with "1.3.6.1.4.1.38783" in the table below

OID	Name	Access	Description	Syntax
!.8.2.2.2.1.1.2.**?**.0	analogInpDescription.**?**	read-write	Analog input description	DisplayString
!.8.2.2.2.1.1.3.**?**.0	analogInpMult.**?**	read-write	Analog input multiplier x1000 in Integer format	Integer32
!.8.2.2.2.1.1.4.**?**.0	analogInpOffset.**?**	read-write	Analog input offset x1000 in Integer format	Integer32
!.8.2.2.2.1.1.5.**?**.0	analogInpMode.**?**	read-write	Analog input mode - 0-10V or 4-20mA	Integer32
!.8.2.2.2.1.1.6.**?**.0	analogInpValue.**?**	read-only	Analog input value x1000 in Integer format	Integer32

setup -> parameters -> digitalInputs -> digitalInpTable -> digitalInpEntry -> digitalInpIndex 1 to 4

replace "**?**" with a number from 1 to 4 and "**!**" with "1.3.6.1.4.1.38783" in the table below

OID	Name	Access	Description	Syntax
!.8.2.2.3.1.1.2.**?**.0	digInpDescription.**?**	read-write	Digital Input description	DisplayString
!.8.2.2.3.1.1.3.**?**.0	digInplowLevel.**?**	read-write	Digital Input closed state	DisplayString
!.8.2.2.3.1.1.4.**?**.0	digInphighLevel.**?**	read-write	Digital Input open state	DisplayString
!.8.2.2.3.1.1.5.**?**.0	digInpMode.**?**	read-write	Digital Input mode - Discrete or Counter	INTEGER { openclosed(0),

				risingEdge(1), fallingEdge(2), bothEdges(3) }
!.8.2.2.3.1.1.6.**?**.0	digInpCloseToOpenDelay.**?**	read-write	Digital input Close To Open delay	Integer32(0..60000)
!.8.2.2.3.1.1.7.**?**.0	digInpOpenToCloseDelay.**?**	read-write	Digital input Open To Close delay	Integer32(0..60000)
!.8.2.2.3.1.1.8.**?**.0	digInpCounterInitValue.**?**	read-only	Digital input counter initial value	Integer32
!.8.2.2.3.1.1.9.**?**.0	digInpValue.**?**	read-only	Digital input value	Unsigned32

setup -> parameters -> relaySetup -> relOutSetupTable -> relOutSetupEntry -> relOutSetupIndex 1 to 4
replace "**?**" with a number from 1 to 4 and "**!**" with "1.3.6.1.4.1.38783" in the table below

OID	Name	Access	Description	Syntax
!.8.2.2.4.1.1.2.**?**.0	relOutDescription.**?**	read-write	Relay description	DisplayString
!.8.2.2.4.1.1.3.**?**.0	relOutPulseWidth.**?**	read-write	Relay Pulse x100ms	Integer32
!.8.2.2.4.1.1.4.**?**.0	relOutActivation.**?**	read-write	Relay activated from	INTEGER {webHttpApi(0), al01(1), al02(2), al03(3), al04(4), al05(5), al06(6), al07(7), al08(8), al09(9), al10(10), al11(11), al12(12), al13(13), al14(14), al15(15), al16(16), al17(17), al18(18), al19(19), al20(20), al21(21), al22(22), al23(23), al24(24), anyAlarm(25), schedule1(26), schedule2(27), schedule3(28), schedule4(29) }
!.8.2.2.4.1.1.5.**?**.0	relOutAction.**?**	read-write	Relay action on alarm condition	INTEGER {on(0), pulse(2)}

OID	Name	Access	Description	Syntax
!.8.2.2.4.2.0	relOutAfterRst	read-write	Relay state after restart	INTEGER {off(0), on(1), laststate(2)}

monitorNcontrol -> channels -> chanTable -> chanEntry -> chIndex 1 to 24
replace "**?**" with a number from 1 to 24 and "**!**" with "1.3.6.1.4.1.38783" in the table below

OID	Name	Access	Description	Syntax
!.8.3.1.1.1.2.**?**.0	chType.**?**	read-write	Channel type	INTEGER {general(0), cumulative(1), discrete(2), counter(3)}
!.8.3.1.1.1.3.**?**.0	chdescription.**?**	read-write	Channel description	DisplayString
!.8.3.1.1.1.4.**?**.0	chParam1.**?**	read-write	Channel parameter 1	INTEGER {none(0), s01(3), s02(4), s03(5), s04(6), s05(7), s06(8), s07(9), s08(10),s09(11), s10(12), s11(13), s12(14), s13(15), s14(16), s15(17), s16(18), s17(19), s18(20), s19(21), s20(22), s21(23), s22(24), s23(25), s24(26), a01(27), a02(28), a03(29), a04(30), a05(31), a06(32), d01(33), d02(34), d03(35), d04(36)}
!.8.3.1.1.1.5.**?**.0	chOP1.**?**	read-write	Channel operand 1	INTEGER{none(0), multiplication(1), division(2), sum(3), subtract(4)}
!.8.3.1.1.1.6.**?**.0	chParam2.**?**	read-write	Channel parameter 2	INTEGER {none(0), one(1), null(2), s01(3),

				s02(4), s03(5), s04(6), s05(7), s06(8), s07(9), s08(10),s09(11), s10(12), s11(13), s12(14), s13(15), s14(16), s15(17), s16(18), s17(19), s18(20), s19(21), s20(22), s21(23), s22(24), s23(25), s24(26),a01(27), a02(28), a03(29), a04(30), a05(31), a06(32),d01(33), d02(34), d03(35), d04(36)}
!.8.3.1.1.1.7.?.0	chOP2.?	read-write	Channel operand 2	INTEGER {none(0), multiplication(1), division(2), sum(3), subtract(4)}
!.8.3.1.1.1.8.?.0	chCoef1.?	read-write	Channel coefficient 1 x1000 in Integer format	Integer32
!.8.3.1.1.1.9.?.0	chOP3.?	read-write	Channel operand 3	INTEGER {none(0), multiplication(1), division(2), sum(3), subtract(4)}
!.8.3.1.1.1.10.?.0	chCoef2.?	read-write	Channel coefficient 2 x1000 in Integer format	Integer32
!.8.3.1.1.1.11.?.0	chUnit.?	read-write	Channel unit	DisplayString
!.8.3.1.1.1.12.?.0	chCumulInitValue.?	read-write	Channel cumulative initial value	Integer32
!.8.3.1.1.1.13.?.0	chValue.?	read-only	Channel value x1000 in Integer format	Integer32
!.8.3.1.1.1.14.?.0	chCounter.?	read-only	Channel as 32-bit counter	Counter32
!.8.3.1.1.1.15.?.0	chAlarmStatus.?	read-only	Channel alarm status	INTEGER {undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)}

monitorNcontrol -> relays -> relOutTable -> relOutEntry -> relayIndex 1 to 4

replace "**?**" with a number from 1 to 4 and "**!**" with "1.3.6.1.4.1.38783" in the table below

OID	Name	Access	Description	Syntax
!.8.3.2.1.1.2.?.0	relayState.?	read-write	Relay state	INTEGER {off(0), on(1)}
!.8.3.2.1.1.3.?.0	relayPulse.?	read-write	Relay pulse	INTEGER {off(0), on(1)}

monitorNcontrol -> alarmsTable -> alarmsEntry -> alIndex 1 to 24

replace "**?**" with a number from 1 to 24 and "**!**" with "1.3.6.1.4.1.38783" in the table below

OID	Name	Access	Description	Syntax
!.8.3.3.1.1.2.?.0	alDescription.?	read-write	Alarm description	DisplayString
!.8.3.3.1.1.3.?.0	alCond1Channel.?	read-write	Alarm condition 1 channel	INTEGER {none(0), v01(1), v02(2), v03(3), v04(4), v05(5), v06(6), v07(7), v08(8), v09(9), v10(10), v11(11), v12(12), v13(13), v14(14), v15(15), v16(16), v17(17), v18(18), v19(19), v20(20), v21(21), v22(22), v23(23), v24(24)}
!.8.3.3.1.1.4.?.0	alCond1Operand.?	read-write	Alarm condition 1 operand	INTEGER{larger(1), less(2)}
!.8.3.3.1.1.5.?.0	alCond1Limit.?	read-write	Alarm condition 1 limit x1000 in Integer format	Integer32
!.8.3.3.1.1.6.?.0	alCond1Hys.?	read-write	Alarm condition 1 hysteresis x1000 in Integer format	Integer32
!.8.3.3.1.1.7.?.0	alCond1AlarmState.?	read-write	Alarm condition 1 discrete alarm state	INTEGER {open(0), closed(1)}
!.8.3.3.1.1.8.?.0	alCondLogic.?	read-write	Alarm conditions logic	INTEGER{none(0), and(1), or(2)}

!.8.3.3.1.1.9.?.0	alCond2Channel.?	read-write	Alarm condition 2 channel	INTEGER {none(0), v01(1), v02(2), v03(3), v04(4), v05(5), v06(6), v07(7), v08(8), v09(9), v10(10), v11(11), v12(12), v13(13), v14(14), v15(15), v16(16), v17(17), v18(18), v19(19), v20(20), v21(21), v22(22), v23(23), v24(24)}
!.8.3.3.1.1.10.?.0	alCond2Operand.?	read-write	Alarm condition 2 operand	INTEGER{larger(1), less(2)}
!.8.3.3.1.1.11.?.0	alCond2Limit.?	read-write	Alarm condition 2 limit x1000 in Integer format	Integer32
!.8.3.3.1.1.12.?.0	alCond2Hys.?	read-write	Alarm condition 2 hysteresis x1000 in Integer format	Integer32
!.8.3.3.1.1.13.?.0	alCond2AlarmState.?	read-write	Alarm condition 2 discrete alarm state	INTEGER {open(0), closed(1)}
!.8.3.3.1.1.14.?.0	alType.?	read-write	Alarm type	INTEGER {warning(3), minor(4), major(5), critical(6)}
!.8.3.3.1.1.15.?.0	alAssigned.?	read-write	Alarm assigned to	INTEGER {none(0), v01(1), v02(2), v03(3), v04(4), v05(5), v06(6), v07(7), v08(8),v09(9), v10(10), v11(11), v12(12), v13(13), v14(14), v15(15), v16(16),v17(17), v18(18), v19(19), v20(20), v21(21), v22(22), v23(23), v24(24)}
!.8.3.3.1.1.16.?.0	alActionDelay.?	read-write	Alarm action delay	Integer32
!.8.3.3.1.1.17.?.0	alActionOnReturn.?	read-write	Alarm action on return	INTEGER {no(0), yes(1)}
!.8.3.3.1.1.18.?.0	alAction1.?	read-write	Alarm action 1	INTEGER {none(0), trapcond1(1), trapcond2(2), trapcond1and2(3), postiostate(4)}
!.8.3.3.1.1.19.?.0	alAction2.?	read-write	Alarm action 2	INTEGER {none(0), trapcond1(1), trapcond2(2), trapcond1and2(3), postiostate(4)}
!.8.3.3.1.1.20.?.0	alAction3.?	read-write	Alarm action 3	INTEGER {none(0), trapcond1(1), trapcond2(2), trapcond1and2(3), postiostate(4)}
!.8.3.3.1.1.21.?.0	alStatus.?	read-write	Alarm status	INTEGER {undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)}

monitorNcontrol

replace "**!**" with "1.3.6.1.4.1.38783" in the table below

OID	Name	Access	Description	Syntax
!.8.3.4.0	configurationSaved	read-write	Configuration save status SAVED/UNSAVED	INTEGER { unsaved(0), saved(1) }
!.8.3.5.0	restartDevice	read-write	Restart Device	INTEGER { cancel(0), restart(1) }
!.8.3.6.0	hardwareErr	read only	Hardware Error	INTEGER { noErr(0), hwErr(1) }

monitorNcontrol-> funcTable -> funcEntry -> funcIndex 1 to 4

replace "**?**" with a number from 1 to 4 and "**!**" with "1.3.6.1.4.1.38783" in the table below

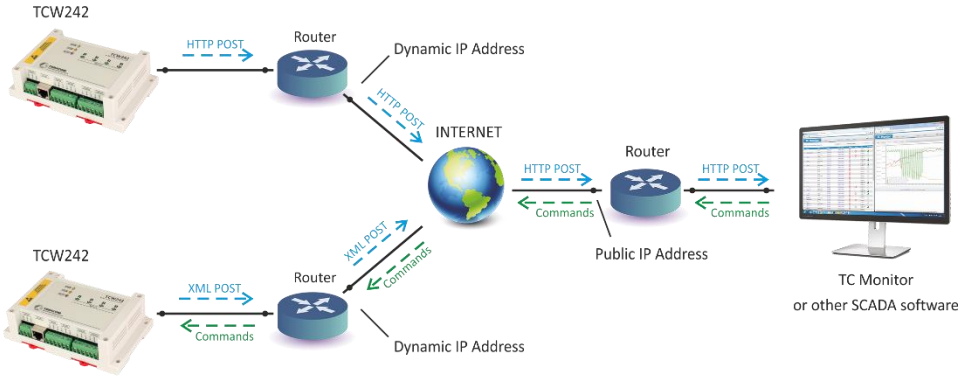
OID	Name	Access	Description	Syntax
!.8.3.7.1.1.2.?.0	funcState	read only	Function state	INTEGER { false(0), true(1) }
!.8.3.7.1.1.3.?.0	funcAl	read only	Function alarm	INTEGER { normal(0), alarm(1) }

9.2. HTTP API

9.2.1. HTTP Post

The TCW242 can perform HTTP/HTTPS Posts to upload XML/JSON files to a designated server. This feature is useful when the controller is located behind a router without a public IP address or when access to the router configuration is unavailable. The server must have a public IP address.

A typical monitoring application is depicted in the following illustration:



HTTP/HTTPS Posts can be sent at set intervals or upon the trigger of an alarm.

The server can also send HTTP Gets with specific commands in response, as described in section 9.2.3. HTTP commands.

Steps to test HTTP/HTTPS Post:

- Save the following code as 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 to a public web server with PHP support. Verify the script by typing the URL (e.g. www.yourserverURL.com/post.php) in a web browser. If successful, a page with "The PHP script is working!" will be displayed.
- Configure the controller to send HTTP/HTTPS POST to your web server. Enter the URL (yourserverURL.com/post.php) in the URL field and click "Test HTTP Post".
- If the HTTP/HTTPS POST is received and processed, "OK" will appear near the button. Additionally, an XML file with time information (e.g. 20220420103318_status.xml) will be created in the same directory as post.php.

9.2.2. HTTP Get

The TCW242 can be monitored using HTTP Get and XML or JSON files. The format of the files is as follows:

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

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

For more details on the structure of the files, see Appendix A XML file structure and Appendix B JSON file structure.

HTTP Get can be sent at any time if the device is on the same network or if appropriate routing is in place. If direct access to the device is not possible, HTTP Get can be sent immediately after receiving an HTTP/HTTPS Post from the same device.

9.2.2.1. Commands

HTTP Get can also be used with all the commands used in HTTP/HTTPS Post, using the format:

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

where:

yyy is the command

xxx is the parameter

Example:

http://device.ip.address/status.xml?pper=300 will set the post period to 300 seconds.

9.2.2.2. HTTP GET authentication

If enabled, HTTP API authentication requires basic access authentication to access the status.xml file. The format for the command is as follows:

Enabled: http://device.ip.address/status.xml?a=uuuu:pppp

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

Example:

To set the post period to 120 sec with username "admin" and password "admin", the command would be

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

9.2.3. List of HTTP API commands

Command	Description
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
purl1=yyy	URL for HTTP Post to Server 1, where yyy is a full path to the PHP file. Example: purl=212.25.45.120:30181/xampp/test/posttest1.php
purl2=yyy	URL for HTTP Post to Server 1, where yyy is a full path to the PHP file. Example: purl=212.25.45.120:30181/xampp/test/posttest2.php
purl3=yyy	URL for HTTP Post to Server 1, where yyy is a full path to the PHP file. Example: purl=212.25.45.120:30181/xampp/test/posttest3.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
ron=**n**	Turn relay ON (**n** is 1,2,4 or 8) ron=1 - will turn ON relay 1 ron=2 - will turn ON relay 2 ron=4 - will turn ON relay 3 ron=8 - will turn ON relay 4
ron=1&ron=2&ron=4&ron=8	Turn four relays ON
rof=**n**	Turn relay OFF (**n** is 1,2,4 or 8) rof=1 - will turn OFF relay 1 rof=2 - will turn OFF relay 2 rof=4 - will turn OFF relay 3 rof=8 - will turn OFF relay 4
rof=1&rof=2&rof=4&rof=8	Turn four relays OFF
rpl=**n**	Toggle relay state (**n** is 1,2,4 or 8) rpl=1 - will toggle relay 1 state rpl=2 - will toggle relay 2 state rpl=4 - will toggle relay 3 state rpl=8 - will toggle relay 4 state
mdata=x	Data format JSON/Plain text for MQTT Publish – 0 JSON, 1 Plain text
mmode=x	Publish protocol, where x is 0 for unsecure and 1 for TLS/SSL
muser=xxxx	Username authentication for MQTT, where xxxx is a username
mpass=xxxx	Password authentication for MQTT, where xxxx is a password
murl=yyy	URL for MQTT publish, where yyy is a path murl=212.25.45.120
mport=yyyy	Port for MQTT publish, where yyyy is a port mport=1883
mper=x	MQTT publish period in seconds (x is between 60 and 172800) mper=600 – will set MQTT publish period to 600 seconds
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.)

9.3. MODBUS TCP/IP

The Modbus protocol is a communication standard that was first introduced in 1979 by Modicon. It enables communication between a master device and a slave device in SCADA systems

9.3.1. Codes and answers

9.3.1.1. Read Coil Status (FC=01)

Request
Command to obtain ON/OFF status of discrete coils at address 100.
01 0064 0001
Response
01 01 01

9.3.1.2. Force Single Coil (FC=05)

Request
This command sets the state of discrete address 100 to "ON".
05 0064 FF00
Response
The typical response is a copy of the request, sent back after writing to the coil.
05 0064 FF00

9.3.1.1. Read Discrete Inputs (FC=02)

Request
This command is asking for the status of 4 discrete inputs.

02 0064 0004
Response
02 01 0C

9.3.1.3. Read Holding Registers (FC=03)

Request
This command requests the value of holding register 19300.
03 4B64 0002
Response
03 04 41DD 4210

Request
This command requests the value of holding register 18100.
03 46B4 0008
Response
03 10 54 65 6D 70 65 72 61 74 75 72 65 00 00 00 00 00

9.3.1.2. Write Single Register (FC=06)

Request
This command writes a value to the single register at address 18300.
06 477C 0003
Response
06 477C 0003

9.3.1.3. Write Multiple Registers (FC=16)

Request
This command writes a value to the single register at address 18300.
10 477C 0002 04 0003 0004
Response
10 477C 0002

9.3.1.4. Exception codes

All exceptions are indicated by adding 0x80 to the function code in the request, followed by a single reason byte. For example:

01 Illegal function:

The function code received in the query is not a valid action for the controller.

02 Illegal data address:

The data address received in the query is not a valid address for the slave. Specifically, the reference number and transfer length combination is invalid. A request with offset 96 and length 4 would succeed, but a request with offset 96 and length 5 would result in exception code 02.

9.3.2. Short address table

Note: Changes can be saved by setting "Configuration Saved" to 1.

Parameter	FC	PDU decimal address	Data size	Data
Relay 1	01,05,15	100	Discrete	
Relay 2	01,05,15	101	Discrete	
Relay 3	01,05,15	102	Discrete	
Relay 4	01,05,15	103	Discrete	
Digital input 1	02	100	Discrete	
Digital input 2	02	101	Discrete	
Digital input 3	02	102	Discrete	
Digital input 4	02	103	Discrete	
Analog input 1 value	03	14400	32-bit Float	
Analog input 2 value	03	14402	32-bit Float	
Analog input 3 value	03	14404	32-bit Float	
Analog input 4 value	03	14406	32-bit Float	
MB sensor 1 value	03	16400	32-bit Float	
MB sensor 2 value	03	16402	32-bit Float	
MB sensor 3 value	03	16404	32-bit Float	
MB sensor 4 value	03	16406	32-bit Float	
MB sensor 5 value	03	16408	32-bit Float	
MB sensor 6 value	03	16410	32-bit Float	
MB sensor 7 value	03	16412	32-bit Float	
MB sensor 8 value	03	16414	32-bit Float	
MB sensor 9 value	03	16416	32-bit Float	
MB sensor 10 value	03	16418	32-bit Float	
MB sensor 11 value	03	16420	32-bit Float	

MB sensor 12 value	03	16422	32-bit Float	
MB sensor 13 value	03	16424	32-bit Float	
MB sensor 14 value	03	16426	32-bit Float	
MB sensor 15 value	03	16428	32-bit Float	
MB sensor 16 value	03	16430	32-bit Float	
MB sensor 17 value	03	16432	32-bit Float	
MB sensor 18 value	03	16434	32-bit Float	
MB sensor 19 value	03	16436	32-bit Float	
MB sensor 20 value	03	16438	32-bit Float	
MB sensor 21 value	03	16440	32-bit Float	
MB sensor 22 value	03	16442	32-bit Float	
MB sensor 23 value	03	16444	32-bit Float	
MB sensor 24 value	03	16446	32-bit Float	
Channel 1 value	03	19200	32-bit Float	
Channel 2 value	03	19202	32-bit Float	
Channel 3 value	03	19204	32-bit Float	
Channel 4 value	03	19206	32-bit Float	
Channel 5 value	03	19208	32-bit Float	
Channel 6 value	03	19210	32-bit Float	
Channel 7 value	03	19212	32-bit Float	
Channel 8 value	03	19214	32-bit Float	
Channel 9 value	03	19216	32-bit Float	
Channel 10 value	03	19218	32-bit Float	
Channel 11 value	03	19220	32-bit Float	
Channel 12 value	03	19222	32-bit Float	
Channel 13 value	03	19224	32-bit Float	
Channel 14 value	03	19226	32-bit Float	
Channel 15 value	03	19228	32-bit Float	
Channel 16 value	03	19230	32-bit Float	
Channel 17 value	03	19232	32-bit Float	
Channel 18 value	03	19234	32-bit Float	
Channel 19 value	03	19236	32-bit Float	
Channel 20 value	03	19238	32-bit Float	
Channel 21 value	03	19240	32-bit Float	
Channel 22 value	03	19242	32-bit Float	
Channel 23 value	03	19244	32-bit Float	
Channel 24 value	03	19246	32-bit Float	
Channel 1 alarm status	03	19400	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Channel 2 alarm status	03	19401	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Channel 3 alarm status	03	19402	16-bit unsign int	undefined(0), normal(1), indeterminate(2),

				warning(3), minor(4), major(5), critical(6)
Channel 4 alarm status	03	19403	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Channel 5 alarm status	03	19404	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Channel 6 alarm status	03	19405	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Channel 7 alarm status	03	19406	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Channel 8 alarm status	03	19407	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Channel 9 alarm status	03	19408	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Channel 10 alarm status	03	19409	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Channel 11 alarm status	03	19410	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Channel 12 alarm status	03	19411	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Channel 13 alarm status	03	19412	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Channel 14 alarm status	03	19413	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Channel 15 alarm status	03	19414	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Channel 16 alarm status	03	19415	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)

Channel 17 alarm status	03	19416	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Channel 18 alarm status	03	19417	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Channel 19 alarm status	03	19418	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Channel 20 alarm status	03	19419	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Channel 21 alarm status	03	19420	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Channel 22 alarm status	03	19421	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Channel 23 alarm status	03	19422	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Channel 24 alarm status	03	19423	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 1 status	03	22000	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 2 status	03	22001	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 3 status	03	22002	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 4 status	03	22003	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 5 status	03	22004	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 6 status	03	22005	16-bit unsign int	undefined(0), normal(1), indeterminate(2),

				warning(3), minor(4), major(5), critical(6)
Alarm 7 status	03	22006	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 8 status	03	22007	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 9 status	03	22008	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 10 status	03	22009	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 11 status	03	22010	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 12 status	03	22011	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 13 status	03	22012	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 14 status	03	22013	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 15 status	03	22014	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 16 status	03	22015	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 17 status	03	22016	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 18 status	03	22017	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 19 status	03	22018	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)

Alarm 20 status	03	22019	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 21 status	03	22020	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 22 status	03	22021	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 23 status	03	22022	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 24 status	03	22023	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Function 1 state	03	23000	16-bit unsign int	false(0), true(1)
Function 2 state	03	23001	16-bit unsign int	false(0), true(1)
Function 3 state	03	23002	16-bit unsign int	false(0), true(1)
Function 4 state	03	23003	16-bit unsign int	false(0), true(1)
Function 1 alarm state	03	23100	16-bit unsign int	normal(0), alarm(1)
Function 2 alarm state	03	23101	16-bit unsign int	normal(0), alarm(1)
Function 3 alarm state	03	23102	16-bit unsign int	normal(0), alarm(1)
Function 4 alarm state	03	23103	16-bit unsign int	normal(0), alarm(1)
Save configuration	03,06	24000	16-bit unsign int	unsaved(0), saved(1)
Restart device	03,06	24001	16-bit unsign int	cancel(0), restart(1)
HW error	03	24002	16-bit unsign int	noErr(0), hwErr(1)
Device ID	03	24100	18 bytes UTF-8	Example: 5c:32:c5:00:ac:52
Hostname	03	24200	16 bytes UTF-8	
Device IP	03	24300	16 bytes UTF-8	Example: 192.168.1.2

10. Data Logger

The logger uses a FLASH memory circular buffer for storage, where new data overwrites the oldest when full. The full log is always available for download and cannot be cleared.
The number of records stored varies depending on the description length and characters used (up to 52371 in the worst case, enough for 36 days with 1-minute records). Typically, the logger can store 71400 records (enough for 49 days with 1-minute records).
The records can be uploaded as a CSV file with a semicolon delimiter to an HTTP server in intervals of 1 to 24 hours, with the first row being a header including the record ID and time stamp.

The structure of one row (a record) of the log is as follows:

ID	Time	Type of record	Channels - values/units	Relays - conditions	Channels - states/units	Alarms - values/descriptions

ID	32-bit unique number for every row (record).
Time	a time stamp of record, in format dd.mm.yyyy, 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
Channels - values/units	Channels 1 to 24 values/units
Relays - conditions	Relays 1 to 4 conditions
Channels - states/units	Channels 1 to 24 states/units For chanel types General, Cumulative and Discrete following states are available: 0 – “Undefined” 1 – “Normal” 2 – “Indeterminate” 3 – “Warning” 4 – “Minor” 5 – “Major” 6 – “Critical” For channel type Counter following states are available: 8 – “Undefined” 9 – “Normal” 10 – “Indeterminate” 11 – “Warning” 12 – “Minor” 13 – “Major” 14 – “Critical”
Alarms - values/descriptions	Alarms 1 to 24 values/descriptions The wollowing alarm values are available: 0 – “Undefined” 1 – “Normal” 2 – “Indeterminate” 3 – “Warning” 4 – “Minor” 5 – “Major”

6 – “Critical”

An example of the log file /fragment channels - values/units/:

ID;Time;Type;Ch1/°C;Ch2/%RH;Ch3;Ch4;Ch5;Ch6;Ch7;Ch8;Ch9;Ch10;Ch11;Ch12;Ch13;Ch14;Ch15;Ch16;Ch17;Ch18;Ch19;Ch20;Ch21;Ch22;Ch23;Ch24....
25114;19.07.2022,16:49:49; Time;25.319;54.512;118.833;229.877;0.000;6587.396;;;;;;;;;;;;;;;;;;;;;....

An example of the log file /fragment relays - conditions/:

ID;Time;Type;...R1,R2,R3,R4....
25114;19.07.2022,16:49:49; Time;1,1,0,0;;;;;;;;;;;;;;;;;;;;;....

An example of the log file /fragment channels - states/units/:

ID;Time;Type;....ChSt1/°C;ChSt2/%RH;ChSt3;ChSt4;ChSt5;ChSt6;ChSt7;ChSt8;ChSt9;ChSt10;ChSt11;ChSt12;ChSt13;ChSt14;ChSt15;ChSt16;ChSt17;ChSt18;ChSt19;ChSt20;ChSt21;ChSt22;ChSt23;ChSt24....
25114;19.07.2022,16:49:49;Time;....3;5;0;2;2;2;;;;;;;;;;;;;;;;;;;;;....

An example of the log file /fragment alarms - values/descriptions/:

ID;Time;Type;...Al1/AL1-T;Al2/AL2-H;Al3;Al4;Al5;Al6;Al7;Al8;Al9;Al10;Al11;Al12;Al13;Al14;Al15;Al16;Al17;Al18;Al19;Al20;Al21;Al22;Al23;Al24....
25114;19.07.2022,16:49:49;Time;....3;5;0;0;0;0;;;;;;;;;;;;;;;;;;

11. MQTT

MQTT is a publish/subscribe messaging protocol for client-server communication. It is lightweight, open-source, simple, and easy to implement. MQTT is widely used across industries including automotive, manufacturing, telecommunications, oil and gas.

Learn more about MQTT at www.mqtt.org.

12. Factory default settings

TCW242 can be reset to its original factory settings in three ways:

12.1. Factory reset without network settings

The "Factory Default" button under Administration > Backup/Restore restores all parameters to factory settings, excluding network settings.

12.2. Factory reset for network settings only

Pressing the reset button for over 5 seconds while the device is running will set all network settings to their factory defaults.

12.3. Full factory reset

For a full factory reset, follow these steps:

- Press and hold the RESET button and turn on the power supply
- The yellow LED lights and the red LED flashes about 5 times per second
- Release the reset button after about 5 seconds when the red LED turns off
- The yellow LED flashes once per second and the red LED lights, indicating the device is in working mode with factory defaults.

The factory default settings are:

Username	admin
Password	admin
IP Address	192.168.1.2
Subnet Mask	255.255.255.0
Default Gateway	192.168.1.1
SNMPConfiguration	disabled
readCommunity	public
writeCommunity	private
Analog inputs unit	voltage
Analog inputs multiplier	1.000
Analog inputs ofset	0.000
Analog inputs mode	Voltage
Digital inputs mode	Open/Closed

13. Operating environment

The device is designed for use in a pollution level 2 environment and at elevations up to 2000 meters. Any other equipment used in conjunction with the controller must also meet EMC requirements and be suitable for the same operating conditions.

14. Safety precautions

Do not use this device for medical or life-saving purposes or in any situation where its failure could result in serious injury or death.

Use only flexible stranded wire with a cross-section of 0.5mm² or greater for digital and analog inputs and relay output wiring to reduce the risk of fire.

To avoid electric shock and fire hazards, keep the device away from liquids, rain, and moisture. Do not place objects filled with liquids, such as vases, on the device.

Ensure recommended spacing between adjacent devices to prevent overheating of the controller. Leave space for cable attachment/removal after installation.

Teracom does not guarantee proper operation if the product is used outside of its specifications.

For proper device operation, follow these steps:

- Properly install the device, refer to the manual;
- Log in to the device using a browser program;
- Configure the device properly;
- Short the "Din1" and "GND";
- Go to the "Input/Output page" of the WEB interface, and check that the proper parameter value is displayed in the Digital Inputs section;
- Go to the "Modbus sensors page" of the WEB interface and install the Modbus RTU (TSH3XX/TST3XX or third-party) sensor, and check that the proper parameters are displayed;
- The flashing "STS" LED should indicate proper operation.

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

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

15. Maintenance

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

Clean the device only with a 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.

XML file structure

```
<Monitor>
  <DeviceInfo>
    <DeviceName>TCW242</DeviceName>
    <HostName>TCW242 </HostName>
    <ID>5c:32:c5:00:a8:0b</ID>
    <FwVer>TCW242-v1.000</FwVer>
    <MnfInfo>www.teracomsystems.com</MnfInfo>
    <SysContact>info@teracomsystems.com</SysContact>
    <SysName>TCW242</SysName>
    <SysLocation>Location</SysLocation>
  </DeviceInfo>
  <CH>
    <CH1>
      <type>0</type>
      <description>Temperature</description>
      <value>24.386</value>
      <valuebin/>
      <unit>°C</unit>
      <alarmbin>4</alarmbin>
      <alarm>Minor</alarm>
      <selch>3</selch>
    </CH1>
    <CH2>
      <type>0</type>
      <description>Humidity</description>
      <value>51.323</value>
      <valuebin/>
      <unit>RH</unit>
      <alarmbin>1</alarmbin>
      <alarm>Normal</alarm>
      <selch>4</selch>
    </CH2>
    <CH3>
      <type>2</type>
      <description>Digital Input 1</description>
      <value>OPEN</value>
      <valuebin>1<valuebin/>
      <unit/>
      <alarmbin>1</alarmbin>
      <alarm>Normal</alarm>
      <selch>31</selch>
    </CH3>
    <CH4>
      <type>0</type>
      <description>V04</description>
      <value>---</value>
      <valuebin/>
      <unit/>
      <alarmbin>0</alarmbin>
      <alarm/>
      <selch>0</selch>
    </CH4>
    <CH5>
      <type>0</type>
      <description>V05-Voltage</description>
      <value>0.000</value>
      <valuebin/>
      <unit>V</unit>
      <alarmbin>1</alarmbin>
      <alarm>Normal</alarm>
      <selch>27</selch>
    </CH5>
    <CH6>
      <type>0</type>
      <description>V06-Current</description>
      <value>0.000</value>
      <valuebin/>
      <unit>A</unit>
      <alarmbin>1</alarmbin>
      <alarm>Normal</alarm>
      <selch>28</selch>
    </CH6>
    <CH7>
```

```
    <type>1</type>
    <description>V07-Energy</description>
    <value>6587.396</value>
    <valuebin/>
    <unit>kWh</unit>
    <alarmbin>0</alarmbin>
    <alarm/>
    <selch>27</selch>
  </CH7>
  <CH8>
    <type>0</type>
    <description>V08</description>
    <value>---</value>
    <valuebin/>
    <unit/>
    <alarmbin>0</alarmbin>
    <alarm/>
    <selch>0</selch>
  </CH8>
  <CH9>
    <type>0</type>
    <description>V09</description>
    <value>---</value>
    <valuebin/>
    <unit/>
    <alarmbin>0</alarmbin>
    <alarm/>
    <selch>0</selch>
  </CH9>
  <CH10>
    <type>0</type>
    <description>V10</description>
    <value>---</value>
    <valuebin/>
    <unit/>
    <alarmbin>0</alarmbin>
    <alarm/>
    <selch>0</selch>
  </CH10>
  <CH11>
    <type>0</type>
    <description>V11</description>
    <value>---</value>
    <valuebin/>
    <unit/>
    <alarmbin>0</alarmbin>
    <alarm/>
    <selch>0</selch>
  </CH11>
  <CH12>
    <type>0</type>
    <description>V12</description>
    <value>---</value>
    <valuebin/>
    <unit/>
    <alarmbin>0</alarmbin>
    <alarm/>
    <selch>0</selch>
  </CH12>
  <CH13>
    <type>0</type>
    <description>V13</description>
    <value>---</value>
    <valuebin/>
    <unit/>
    <alarmbin>0</alarmbin>
    <alarm/>
    <selch>0</selch>
  </CH13>
  <CH14>
    <type>0</type>
    <description>V14</description>
    <value>---</value>
    <valuebin/>
    <unit/>
    <alarmbin>0</alarmbin>
    <alarm/>
```

```
    <selch>0</selch>
  </CH14>
  <CH15>
    <type>0</type>
    <description>V15</description>
    <value>---</value>
    <valuebin/>
    <unit/>
    <alarmbin>0</alarmbin>
    <alarm/>
    <selch>0</selch>
  </CH15>
  <CH16>
    <type>0</type>
    <description>V16</description>
    <value>---</value>
    <valuebin/>
    <unit/>
    <alarmbin>0</alarmbin>
    <alarm/>
    <selch>0</selch>
  </CH16>
  <CH17>
    <type>0</type>
    <description>V17</description>
    <value>---</value>
    <valuebin/>
    <unit/>
    <alarmbin>0</alarmbin>
    <alarm/>
    <selch>0</selch>
  </CH17>
  <CH18>
    <type>0</type>
    <description>V18</description>
    <value>---</value>
    <valuebin/>
    <unit/>
    <alarmbin>0</alarmbin>
    <alarm/>
    <selch>0</selch>
  </CH18>
  <CH19>
    <type>0</type>
    <description>V19</description>
    <value>---</value>
    <valuebin/>
    <unit/>
    <alarmbin>0</alarmbin>
    <alarm/>
    <selch>0</selch>
  </CH19>
  <CH20>
    <type>0</type>
    <description>V20</description>
    <value>---</value>
    <valuebin/>
    <unit/>
    <alarmbin>0</alarmbin>
    <alarm/>
    <selch>0</selch>
  </CH20>
  <CH21>
    <type>0</type>
    <description>V21</description>
    <value>---</value>
    <valuebin/>
    <unit/>
    <alarmbin>0</alarmbin>
    <alarm/>
    <selch>0</selch>
  </CH21>
  <CH22>
    <type>0</type>
    <description>V22</description>
    <value>---</value>
    <valuebin/>
```

```
      <unit/>
      <alarmbin>0</alarmbin>
      <alarm/>
      <selch>0</selch>
    </CH22>
    <CH23>
    <type>0</type>
      <description>V23</description>
      <value>---</value>
      <valuebin/>
      <unit/>
      <alarmbin>0</alarmbin>
      <alarm/>
      <selch>0</selch>
    </CH23>
    <CH24>
      <type>0</type>
      <description>V24</description>
      <value>---</value>
      <valuebin/>
      <unit/>
      <alarmbin>0</alarmbin>
      <alarm/>
      <selch>0</selch>
    </CH24>
  </CH>
  <AL>
    <AL1>
      <description>AL01-T</description>
      <alarmbin>4</alarmbin>
      <alarm>Minor</alarm>
      <assign>1</assign>
    </AL1>
    <AL2>
      <description>AL02-H</description>
      <alarmbin>1</alarmbin>
      <alarm>Normal</alarm>
      <assign>2</assign>
    </AL2>
    <AL3>
      <description>AL03</description>
      <alarmbin>0</alarmbin>
      <alarm/>
      <assign>0</assign>
    </AL3>
    <AL4>
      <description>AL04-DI1</description>
      <alarmbin>0</alarmbin>
      <alarm/>
      <assign>0</assign>
    </AL4>
    <AL5>
      <description>AL05-Humidity</description>
      <alarmbin>1</alarmbin>
      <alarm/>
      <assign>2</assign>
    </AL5>
    <AL6>
      <description>AL06</description>
      <alarmbin>0</alarmbin>
      <alarm/>
      <assign>0</assign>
    </AL6>
    <AL7>
      <description>AL07</description>
      <alarmbin>0</alarmbin>
      <alarm/>
      <assign>0</assign>
    </AL7>
    <AL8>
      <description>AL08</description>
      <alarmbin>0</alarmbin>
      <alarm/>
      <assign>0</assign>
    </AL8>
    <AL9>
      <description>AL09</description>
```

```
    <alarmbin>0</alarmbin>
    <alarm/>
    <assign>0</assign>
  </AL9>
  <AL10>
    <description>AL10</description>
    <alarmbin>0</alarmbin>
    <alarm/>
    <assign>0</assign>
  </AL10>
  <AL11>
    <description>AL11</description>
    <alarmbin>0</alarmbin>
    <alarm/>
    <assign>0</assign>
  </AL11>
  <AL12>
    <description>AL12</description>
    <alarmbin>0</alarmbin>
    <alarm/>
    <assign>0</assign>
  </AL12>
  <AL13>
    <description>AL13</description>
    <alarmbin>0</alarmbin>
    <alarm/>
    <assign>0</assign>
  </AL13>
  <AL14>
    <description>AL14</description>
    <alarmbin>0</alarmbin>
    <alarm/>
    <assign>0</assign>
  </AL14>
  <AL15>
    <description>AL15</description>
    <alarmbin>0</alarmbin>
    <alarm/>
    <assign>0</assign>
  </AL15>
  <AL16>
    <description>AL16</description>
    <alarmbin>0</alarmbin>
    <alarm/>
    <assign>0</assign>
  </AL16>
  <AL17>
    <description>AL17</description>
    <alarmbin>0</alarmbin>
    <alarm/>
    <assign>0</assign>
  </AL17>
  <AL18>
    <description>AL18</description>
    <alarmbin>0</alarmbin>
    <alarm/>
    <assign>0</assign>
  </AL18>
  <AL19>
    <description>AL19</description>
    <alarmbin>0</alarmbin>
    <alarm/>
    <assign>0</assign>
  </AL19>
  <AL20>
    <description>AL20</description>
    <alarmbin>0</alarmbin>
    <alarm/>
    <assign>0</assign>
  </AL20>
  <AL21>
    <description>AL21</description>
    <alarmbin>0</alarmbin>
    <alarm/>
    <assign>0</assign>
  </AL21>
  <AL22>
```

```
      <description>AL22</description>
      <alarmbin>0</alarmbin>
      <alarm/>
      <assign>0</assign>
    </AL22>
    <AL23>
      <description>AL23</description>
      <alarmbin>0</alarmbin>
      <alarm/>
      <assign>0</assign>
    </AL23>
    <AL24>
      <description>AL24</description>
      <alarmbin>0</alarmbin>
      <alarm/>
      <assign>0</assign>
    </AL24>
  </AL>
  <R>
    <R1>
      <description>Relay 1</description>
      <value>ON</value>
      <valuebin>1</valuebin>
    </R1>
    <R2>
      <description>Relay 2</description>
      <value>OFF</value>
      <valuebin>0</valuebin>
    </R2>
    <R3>
      <description>Relay 3</description>
      <value>OFF</value>
      <valuebin>0</valuebin>
    </R3>
    <R4>
      <description>Relay 4</description>
      <value>OFF</value>
      <valuebin>0</valuebin>
    </R4>
  </R>
  <HTTPPost>
    <Key/>
    <PostPeriod>300</PostPeriod>
  </HTTPPost>
  <MQTT>
    <Period>120</Period>
  </MQTT>
  <Sys>
    <hwerr/>
    <HighAlarmbin>4</HighAlarmbin>
    <HighAlarm>Minor</HighAlarm>
  </Sys>
  <Time>
    <Date>19.07.2021</Date>
    <Time>12:11:08</Time>
  </Time>
</Monitor>
```

Where:

<CH1>… <CH24> - channels;

<AL1> … <AL24> - alarms;

<R1> … <R4> - relays

<valuebin> - number values 0 or 1;

<alarmbin> - number values from 0 to 6;

<alarm> - Undefined, Normal, Indeterminate, Warning, Minor, Major, Critical;

<assign>0</assign> - alarm not assigned to any channel;

<selch>0</selch> - channel is not displayed on Monitoring -> Channels section

JSON file structure


```
{
  "Monitor": {
    "DeviceInfo": {
      "DeviceName": "TCW242",
      "HostName": "TCW242 ",
      "ID": "5c:32:c5:00:a8:0b ",
      "FwVer": "TCW242-v1.000",
      "MnfInfo": "www.teracomsystems.com",
      "SysContact": "info@teracomsystems.com",
      "SysName": "TCW242",
      "SysLocation": "Location"
    },
    "CH": {
      "CH1": {
        "type": "0",
        "description": "Temperature",
        "value": "24.386",
        "valuebin": "",
        "unit": "°C",
        “alarmbin”: "4",
        “alarm”: "Minor",
        "selch":"3"
      },
      "CH2": {
        "type": "0",
        "description": "Humidity",
        "value": "52.490",
        "valuebin": "",
        "unit": "RH",
        “alarmbin”: "1",
        “alarm”: "Normal",
        "selch":"4"
      },
      "CH3": {
        "type": "2",
        "description": "Digital Input 1",
        "value": "OPEN",
        "valuebin": "1",
        "unit": "",
        “alarmbin”: "1",
        “alarm”: "Normal",
        "selch":"31"
      },
      "CH4": {
        "type": "0",
        "description": "V04",
        "value": "---",
        "valuebin": "",
        "unit": "",
        “alarmbin”: "0",
        “alarm”: "",
        "selch":"0"
      },
      "CH5": {
        "type": "0",
        "description": "V05-Voltage",
        "value": "0.000",
        "valuebin": "",
        "unit": "V",
        “alarmbin”: "1",
        “alarm”: "Normal",
        "selch":"27"
      },
      "CH6": {
        "type": "0",
        "description": "V06-Current",
        "value": "0.000",
        "valuebin": "",
        "unit": "A",
        “alarmbin”: "1",
        “alarm”: "Normal",
        "selch":"28"
      },
```

```
"CH7": {
   "type": "1",
   "description": "V07-Energy",
   "value": "6587.396",
    "valuebin": "",
   "unit": "kWh",
   “alarmbin”: "0",
   “alarm”: "",
   "selch":"27"
},
"CH8": {
   "type": "0",
   "description": "V08",
   "value": "---",
    "valuebin": "",
   "unit": "",
   “alarmbin”: "0",
   “alarm”: "",
   "selch":"0"
},
"CH9": {
   "type": "0",
   "description": "V09",
   "value": "---",
    "valuebin": "",
   "unit": "",
   “alarmbin”: "0",
   “alarm”: "",
   "selch":"0"
},
"CH10": {
   "type": "0",
   "description": "V10",
   "value": "---",
    "valuebin": "",
   "unit": "",
   “alarmbin”: "0",
   “alarm”: "",
   "selch":"0"
},
"CH11": {
   "type": "0",
   "description": "V11",
   "value": "---",
    "valuebin": "",
   "unit": "",
   “alarmbin”: "0",
   “alarm”: "",
   "selch":"0"
},
"CH12": {
   "type": "0",
   "description": "V12",
   "value": "---",
    "valuebin": "",
   "unit": "",
   “alarmbin”: "0",
   “alarm”: "",
   "selch":"0"
},
"CH13": {
   "type": "0",
   "description": "V13",
   "value": "---",
    "valuebin": "",
   "unit": "",
   “alarmbin”: "0",
   “alarm”: "",
   "selch":"0"
},
"CH14": {
   "type": "0",
   "description": "V14",
   "value": "---",
    "valuebin": "",
   "unit": "",
   “alarmbin”: "0",
```

```
    “alarm”: "",
    "selch":"0"
  },
  "CH15": {
    "type": "0",
    "description": "V15",
    "value": "---",
     "valuebin": "",
    "unit": "",
    “alarmbin”: "0",
    “alarm”: "",
    "selch":"0"
  },
  "CH16": {
    "type": "0",
    "description": "V16",
    "value": "---",
     "valuebin": "",
    "unit": "",
    “alarmbin”: "0",
    “alarm”: "",
    "selch":"0"
  },
  "CH17": {
    "type": "0",
    "description": "V17",
    "value": "---",
     "valuebin": "",
    "unit": "",
    “alarmbin”: "0",
    “alarm”: "",
    "selch":"0"
  },
  "CH18": {
    "type": "0",
    "description": "V18",
    "value": "---",
     "valuebin": "",
    "unit": "",
    “alarmbin”: "0",
    “alarm”: "",
    "selch":"0"
  },
  "CH19": {
    "type": "0",
    "description": "V19",
    "value": "---",
     "valuebin": "",
    "unit": "",
    “alarmbin”: "0",
    “alarm”: "",
    "selch":"0"
  },
  "CH20": {
    "type": "0",
    "description": "V20",
    "value": "---",
     "valuebin": "",
    "unit": "",
    “alarmbin”: "0",
    “alarm”: "",
    "selch":"0"
  },
  "CH21": {
    "type": "0",
    "description": "V21",
    "value": "---",
     "valuebin": "",
    "unit": "",
    “alarmbin”: "0",
    “alarm”: "",
    "selch":"0"
  },
  "CH22": {
    "type": "0",
    "description": "V22",
    "value": "---",
```

```
      "valuebin": "",
      "unit": "",
      “alarmbin”: "0",
      “alarm”: "",
      "selch":"0"
    },
    "CH23": {
      "type": "0",
      "description": "V23",
      "value": "---",
      "valuebin": "",
      "unit": "",
      “alarmbin”: "0",
      “alarm”: "",
      "selch":"0"
    },
    "CH24": {
      "type": "0",
      "description": "V24",
      "value": "---",
      "valuebin": "",
      "unit": "",
      “alarmbin”: "0",
      “alarm”: "",
      "selch":"0"
    }
  },
  "AL": {
    "AL1": {
      "description": "AL01-T",
      “alarmbin”: "4",
      “alarm”: " Minor ",
      "assign":"1"
    },
    "AL2": {
      "description": "AL02-H",
      “alarmbin”: "1",
      “alarm”: "Normal",
      "assign":"2"
    },
    "AL3": {
      "description": "AL03",
      “alarmbin”: "0",
      “alarm”: "",
      "assign":"0"
    },
    "AL4": {
      "description": "AL04",
      “alarmbin”: "0",
      “alarm”: "",
      "assign":"0"
    },
    "AL5": {
      "description": "AL05",
      “alarmbin”: "0",
      “alarm”: "",
      "assign":"0"
    },
    "AL6": {
      "description": "AL06",
      “alarmbin”: "0",
      “alarm”: "",
      "assign":"0"
    },
    "AL7": {
      "description": "AL07",
      “alarmbin”: "0",
      “alarm”: "",
      "assign":"0"
    },
    "AL8": {
      "description": "AL08",
      “alarmbin”: "0",
      “alarm”: "",
      "assign":"0"
    },
    "AL9": {
```

```
    "description": "AL09",
    “alarmbin”: "0",
    “alarm”: "",
    "assign":"0"
  },
  "AL10": {
    "description": "AL10",
    “alarmbin”: "0",
    “alarm”: "",
    "assign":"0"
  },
  "AL11": {
    "description": "AL11",
    “alarmbin”: "0",
    “alarm”: "",
    "assign":"0"
  },
  "AL12": {
    "description": "AL12",
    “alarmbin”: "0",
    “alarm”: "",
    "assign":"0"
  },
  "AL13": {
    "description": "AL13",
    “alarmbin”: "0",
    “alarm”: "",
    "assign":"0"
  },
  "AL14": {
    "description": "AL14",
    “alarmbin”: "0",
    “alarm”: "",
    "assign":"0"
  },
  "AL15": {
    "description": "AL15",
    “alarmbin”: "0",
    “alarm”: "",
    "assign":"0"
  },
  "AL16": {
    "description": "AL16",
    “alarmbin”: "0",
    “alarm”: "",
    "assign":"0"
  },
  "AL17": {
    "description": "AL17",
    “alarmbin”: "0",
    “alarm”: "",
    "assign":"0"
  },
  "AL18": {
    "description": "AL18",
    “alarmbin”: "0",
    “alarm”: "",
    "assign":"0"
  },
  "AL19": {
    "description": "AL19",
    “alarmbin”: "0",
    “alarm”: "",
    "assign":"0"
  },
  "AL20": {
    "description": "AL20",
    “alarmbin”: "0",
    “alarm”: "",
    "assign":"0"
  },
  "AL21": {
    "description": "AL21",
    “alarmbin”: "0",
    “alarm”: "",
    "assign":"0"
  },
```

```
      "AL22": {
        "description": "AL22",
        “alarmbin”: "0",
        “alarm”: "",
        "assign":"0"
      },
      "AL23": {
        "description": "AL23",
        “alarmbin”: "0",
        “alarm”: "",
        "assign":"0"
      },
      "AL24": {
        "description": "AL24",
        “alarmbin”: "0",
        “alarm”: "",
        "assign":"0"
      }
    },
    "R": {
      "R1": {
        "description": "Relay 1",
        "value": "ON",
        "valuebin": "1"
      },
      "R2": {
        "description": "Relay 2",
        "value": "OFF",
        "valuebin": "0"
       },
       "R3": {
         "description": "Relay 3",
         "value": "OFF",
         "valuebin": "0"
       },
       "R4": {
         "description": "Relay 4",
         "value": "OFF",
         "valuebin": "0"
        }
    },
    "HTTPPost": {
      "Key": "",
      "PostPeriod": "300"
    },
    "MQTT": {
      "Period": "120"
    },
    "Sys": {
      "hwerr": "",
      "HighAlarmbin": "4",
      "HighAlarm": "Major"
    },
    "Time": {
      "Date": "19.07.2021",
      "Time": "10:30:00"
    }
  }
}
```

MODBUS TCP/IP full address table

Parameter	FC	PDU decimal address	Data size	Data
Relay 1÷4	01,05,15	100	Discrete	
Digital input 1÷4	02	100	Discrete	
Relay 1÷4 description	03,06,16	10000	16 bytes UTF-8	
Relay 1÷4 pulse width	03,06,16	10100	32-bit Float	
Relay 1÷4 activated from	03,06,16	10200	16-bit unsign int	webHttpApi(0), al01(1), al02(2), al03(3), al04(4), al05(5), al06(6), al07(7), al08(8), al09(9), al10(10), al11(11), al12(12), al13(13), al14(14), al15(15), al16(16), al17(17), al18(18), al19(19), al20(20), al21(21), al22(22), al23(23), al24(24), anyAlarm(25), schedule1(26), schedule2(27), schedule3(28), schedule4(29)
Relay 1÷4 action on alarm	03,06,16	10300	16-bit unsign int	on(0), pulse(2)
Relays state after restart	03,06	10400	16-bit unsign int	off(0), on(1), laststate(2)
Digital input 1÷4 Description	03,06,16	12000	16 bytes UTF-8	
Digital input 1÷4 closed state description	03,06,16	12100	16 bytes UTF-8	
Digital input 1÷4 open state description	03,06,16	12200	16 bytes UTF-8	
Digital input 1÷4 mode	03,06,16	12300	16-bit unsign int	openclosed(0), risingEdge(1), fallingEdge(2), bothEdges(3)
Digital input 1÷4 close to open delay	03,06,16	12400	32-bit unsign int	
Digital input 1÷4 open to close delay	03,06,16	12500	32-bit unsign int	
Digital input 1÷4 counter init value	03,06,16	12600	32-bit unsign int	
Digital input 1÷4 counter value	03	12700	32-bit unsign int	
Analog input 1÷4 description	03,06,16	14000	16 bytes UTF-8	
Analog input 1÷4 multiplier	03,06,16	14100	32-bit Float	
Analog input 1÷4 offset	03,06,16	14200	32-bit Float	
Analog input 1÷4 mode(V/mA)	03,06,16	14300	16-bit unsign int	0-10V(0), 4-20mA(1)
Analog input 1÷4 value	03	14400	32-bit Float	
MB sensor 1÷24 description	03,06,16	16000	16 bytes UTF-8	
MB sensor 1÷24 multiplier	03,06,16	16200	32-bit Float	

MB sensor 1÷24 offset	03,06,16	16300	32-bit Float	
MB sensor 1÷24 value	03	16400	32-bit Float	
MB sensor 1÷24 counter	03	16500	32-bit unsign int	
Channel 1÷24 type	03,06,16	18000	16-bit unsign int	general(0), discrete(2) , counter(3)
Channel 1÷24 description	03,06,16	18100	16 bytes UTF-8	
Channel 1÷24 parameter 1	03,06,16	18300	16-bit unsign int	none(0), s01(3), s02(4), s03(5), s04(6), s05(7), s06(8), s07(9), s08(10), s09(11), s10(12), s11(13), s12(14), s13(15), s14(16), s15(17), s16(18), s17(19), s18(20), s19(21), s20(22), s21(23), s22(24), s23(25), s24(26), a01(27), a02(28), a03(29), a04(30), d01(31), d02(32), d03(33), d04(34)
Channel 1÷24 op 1	03,06,16	18400	16-bit unsign int	none(0), multiplication(1), division(2), sum(3), subtract(4)
Channel 1÷24 parameter 2	03,06,16	18500	16-bit unsign int	none(0), s01(3), s02(4), s03(5), s04(6), s05(7), s06(8), s07(9), s08(10), s09(11), s10(12), s11(13), s12(14), s13(15), s14(16), s15(17), s16(18), s17(19), s18(20), s19(21), s20(22), s21(23), s22(24), s23(25), s24(26), a01(27), a02(28), a03(29), a04(30), d01(31), d02(32), d03(33), d04(34)
Channel 1÷24 op 2	03,06,16	18600	16-bit unsign int	none(0), multiplication(1), division(2), sum(3), subtract(4)
Channel 1÷24 coeff 1	03,06,16	18700	32-bit Float	
Channel 1÷24 op 3	03,06,16	18800	16-bit unsign int	none(0), multiplication(1), division(2), sum(3), subtract(4)
Channel 1÷24 coeff 2	03,06,16	18900	32-bit Float	
Channel 1÷24 unit	03,06,16	19000	16 bytes UTF-8	
Channel 1÷24 value	03	19200	32-bit Float	
Channel 1÷24 counter	03	19300	32-bit unsign int	
Channel 1÷24 alarm status	03	19400	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 1÷24 description	03,06,16	20000	16 bytes UTF-8	
Alarm 1÷24 condition 1 channel	03,06,16	20200	16-bit unsign int	none(0), v01(1), v02(2), v03(3), v04(4), v05(5), v06(6), v07(7), v08(8),

				v09(9), v10(10), v11(11), v12(12), v13(13), v14(14), v15(15), v16(16), v17(17), v18(18), v19(19), v20(20), v21(21), v22(22), v23(23), v24(24)
Alarm 1÷24 condition 1 operand	03,06,16	20300	16-bit unsign int	larger(0), less(1)
Alarm 1÷24 condition 1 limit	03,06,16	20400	32-bit Float	
Alarm 1÷24 condition 1 hysteresis	03,06,16	20500	32-bit Float	
Alarm 1÷24 condition 1 discrete al. state	03,06,16	20600	16-bit unsign int	open(0), closed(1)
Alarm 1÷24 function	03,06,16	20700	16-bit unsign int	none(0}, and(1), or(2)
Alarm 1÷24 condition 2 channel	03,06,16	20800	16-bit unsign int	none(0), v01(1), v02(2), v03(3), v04(4), v05(5), v06(6), v07(7), v08(8), v09(9), v10(10), v11(11), v12(12), v13(13), v14(14), v15(15), v16(16), v17(17), v18(18), v19(19), v20(20), v21(21), v22(22), v23(23), v24(24)
Alarm 1÷24 condition 2 operand	03,06,16	20900	16-bit unsign int	larger(0), less(1)
Alarm 1÷24 condition 2 limit	03,06,16	21000	32-bit Float	
Alarm 1÷24 condition 2 hysteresis	03,06,16	21100	32-bit Float	
Alarm 1÷24 condition 2 discrete al. state	03,06,16	21200	16-bit unsign int	open(0), closed(1)
Alarm 1÷24 type	03,06,16	21300	16-bit unsign int	warning(3), minor(4), major(5), critical(6)
Alarm 1÷24 assigned to channel	03,06,16	21400	16-bit unsign int	none(0), v01(1), v02(2), v03(3), v04(4), v05(5), v06(6), v07(7), v08(8),v09(9), v10(10), v11(11), v12(12), v13(13), v14(14), v15(15), v16(16),v17(17), v18(18), v19(19), v20(20), v21(21), v22(22), v23(23), v24(24)
Alarm 1÷24 delay	03,06,16	21500	32-bit Float	
Alarm 1÷24 action on return	03,06,16	21600	16-bit unsign int	no(0), yes(1)
Alarm 1÷24 action 1	03,06,16	21700	16-bit unsign int	none(0), trapcond1(1), trapcond2(2), trapcond1and2(3), postiostate(4), mqttpublish(6)
Alarm 1÷24 action 2	03,06,16	21800	16-bit unsign int	none(0), trapcond1(1), trapcond2(2), trapcond1and2(3), postiostate(4), mqttpublish(6)
Alarm 1÷24 action 3	03,06,16	21900	16-bit unsign int	none(0), trapcond1(1), trapcond2(2),

				trapcond1and2(3), postiostate(4), mqttpublish(6)
Alarm 1÷24 status	03	22000	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Function 1÷4 state	03	23000	16-bit unsign int	false(0), true(1)
Function 1÷4 alarm state	03	23100	16-bit unsign int	normal(0), alarm(1)
Save configuration	03,06	24000	16-bit unsign int	unsaved(0), saved(1)
Restart device	03,06	24001	16-bit unsign int	cancel(0), restart(1)
HW error	03	24002	16-bit unsign int	noErr(0), hwErr(1)
Device ID	03	24100	18 bytes UTF-8	Example: 5c:32:c5:00:ac:52
Hostname	03	24200	16 bytes UTF-8	
Device IP	03	24300	16 bytes UTF-8	Example: 192.168.1.2

Appendix D

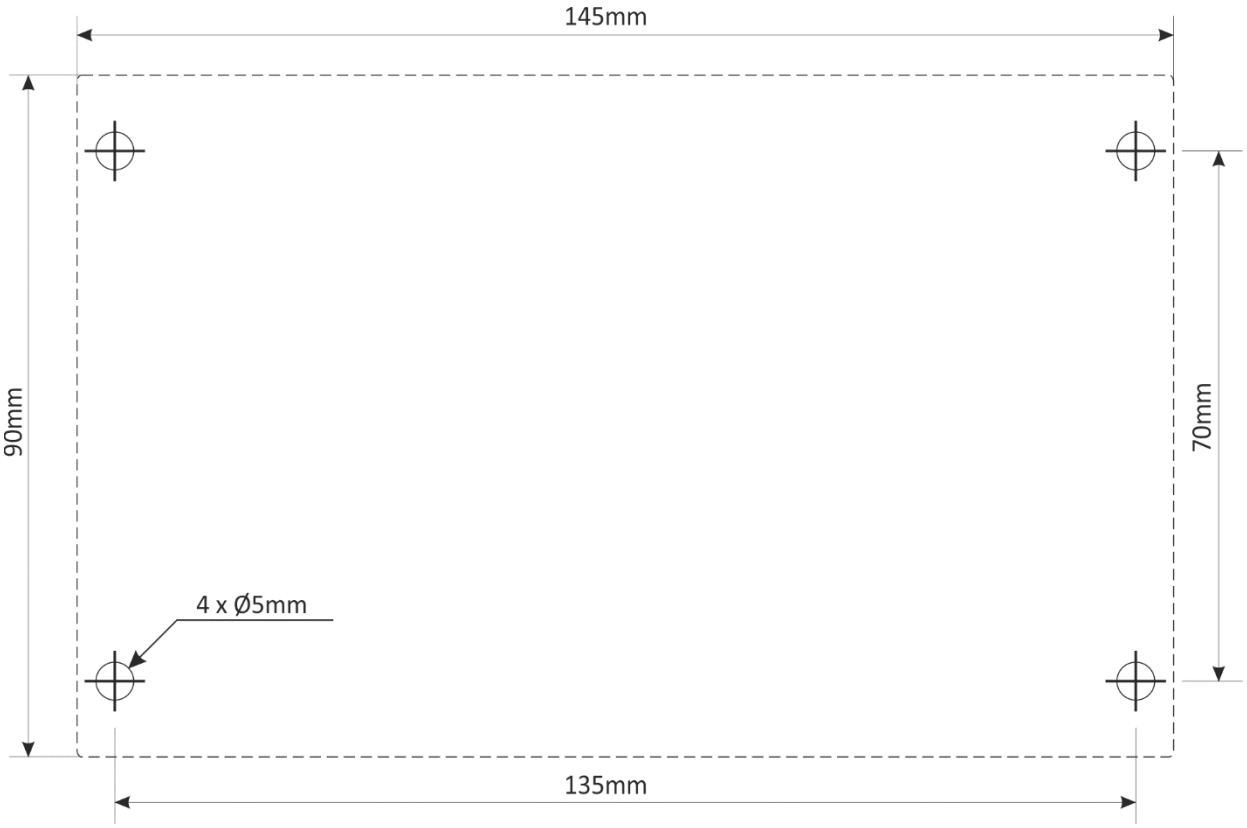


Fig.1

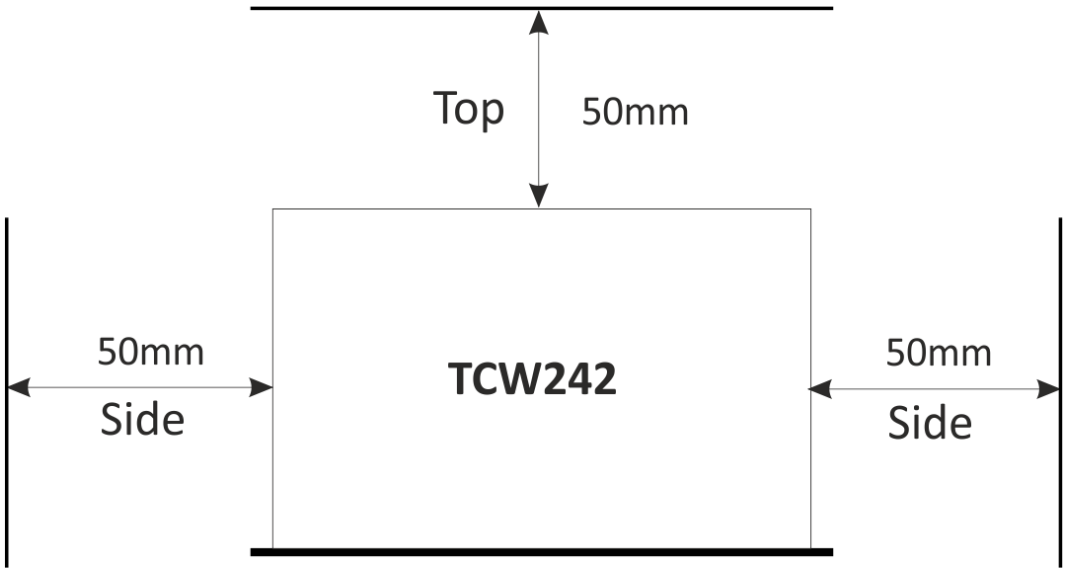


Fig.2