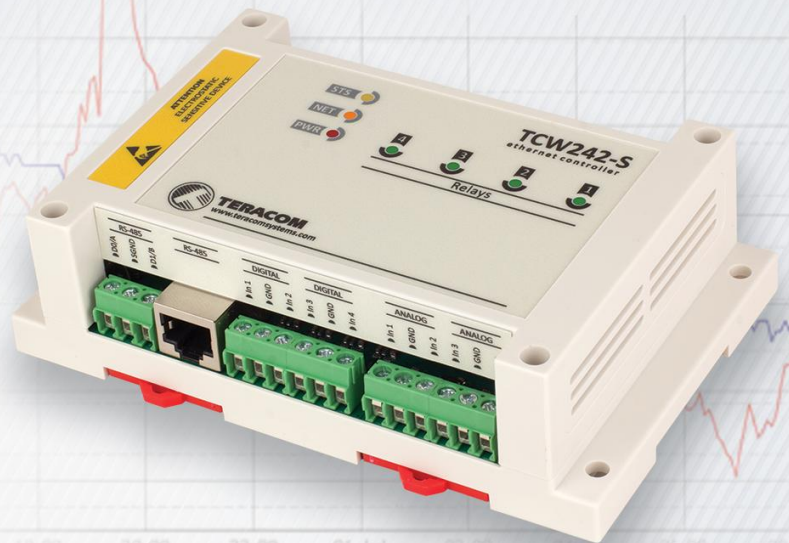




control solutions

TERACOM



TCW242-S Industrial IoT module

Version 1.0 / July 2021

1. Introduction

TCW242-S is an industrial IoT (Internet Of Things) module with Ethernet connectivity and data logging capability. All monitored parameters can be seen as numbers and as graphs.

The device has 4 digital inputs, 4 analog inputs, an RS-485 interface for MODBUS RTU sensors, and 4 relays. The analog inputs can be switched either in a current loop (0 to 20mA) or voltage (0 to 10V) modes. The relays can be activated remotely (WEB, SNMP, MODBUS TCP/IP, or HTTP API command) or locally - from the alarm conditions.

TCW242-S supports up to 24 Teracom and third-party MODBUS RTU sensors.

The user can arrange up to 24 independent monitoring channels from the inputs and sensor readings. Every channel can be set up using up to 2 input parameters and/or constants. There are two types of channels – discrete (for OPEN/CLOSED outputs monitoring) and general (for general monitoring).

The user can also arrange up to 24 independent alarms with 5 different user-selectable states. Every alarm can be set up using up to 2 limits and hysteresis. The alarm 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.

The device supports SNMP, HTTP API, and MODBUS TCP/IP protocols for machine-to-machine (M2M) communication.

2. Features

- 10/100 Mb Ethernet connectivity with Auto-MDIX;
- Password protected, web-based configuration and control;
- 4 digital (dry contact) inputs;
- 4 analog inputs with 0/10V and 0/20mA modes;
- 4 relays with NO and NC contacts;
- MODBUS RTU support for up to 24 sensors (registers);
- Up to 24 channels for monitoring;
- Up to 24 independent alarms;
- SNMPv2 and SNMPv3 support;
- SNMP traps sending for alert conditions;
- HTTP and SNMP port changing;
- HTTP API commands;
- Periodical HTTP/HTTPS Post of XML/JSON status files for client-server systems;
- MODBUS TCP/IP support;
- Dynamic DNS with DynDNS, No-IP and DNS-O-Matic support;
- NTP protocol support;
- Data logger for up to 70000 records;
- DIN Rail Mountable;
- Wide power supply voltage range;
- Backup/restore of settings;
- Remote firmware update.

3. Applications

The industrial IoT module is designed to solve a wide range of tasks in the light industry sector, remote monitoring and control in communication facilities, food and beverage storage, greenhouses, water stations, etc.

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 MODBUS 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: 10 bits

Mode: Voltage or current loop

Input Range: 0/10V or 0/20mA

Accuracy: $\pm 1\%$

Sampling Rate: 600mS per channel (averaged value of 100 samples)

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 relays' 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! Risk of explosion if the battery is replaced by an incorrect type.

5. LED indicators

The following indicators show the status of the controller:

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

6. Installation and setup

This device must be installed by qualified personnel.

This device must not be installed directly outdoors.

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

6.1. Mounting

TCW242-S should be mounted in a clean and dry location on a non-flammable surface. Ventilation is recommended for installations where the ambient air temperature is expected to be high.

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

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

TCW242-S can be mounted to a standard (35mm by 7.55mm) DIN rail. Attach the controller to the DIN rail by hooking the hook on the back of the enclosure to the DIN rail and then snap the bottom hook into place.

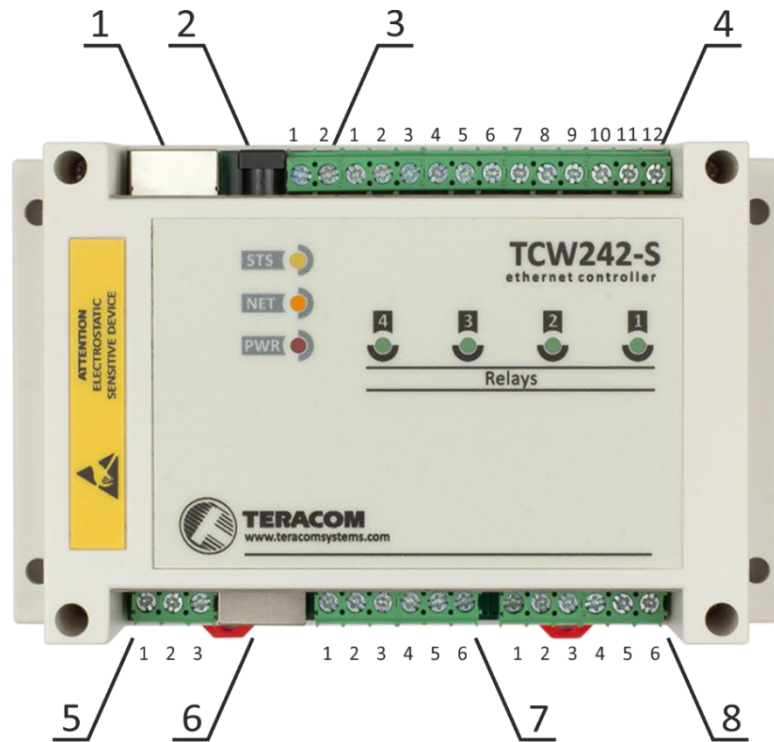
6.2. Connection

Attention! Disconnect power supply before wiring.

The correct wiring procedure is as follows:

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

Make sure that the wires are properly attached to the terminals and that the terminals are tightened. Not the proper wiring and configuration can cause permanent damage to TCW242-S or the equipment to which it is connected or both.



Connector 1	Ethernet - RJ45	Connector 6	Pin1 – not connected (most left) Pin2 – not connected Pin3 – not connected Pin4 – Line B- (D1) Pin5 – Line A+ (D0) Pin6 – not connected Pin7 – +VDD Pin8 – GND
Connector 2	Power - 2.1x5.5mm connector, central positive	Connector 7	Pin1 – Digital In 1 Pin2 – GND Pin3 – Digital In 2 Pin4 – Digital In 3 Pin5 – GND Pin6 – Digital In 4
Connector 3	Pin1 – Power positive Pin2 – Power negative	Connector 8	Pin1 – Analog In 1 Pin2 – GND Pin3 – Analog In 2 Pin4 – Analog In 3 Pin5 – GND Pin6 – Analog In 4
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 – Line A+ (D0) Pin2 – GND Pin3 – Line B- (D1)		

6.2.1. Power supply connection

TCW242-S is designed to be supplied by adapter SYS1308(N)-2412-W2E or similar, intended for use in the conditions of overvoltage category II, and prior assessed for compliance with safety requirements. The power supply equipment shall be resistant to short circuits and overload in a secondary circuit.

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

6.2.2. Digital inputs connection

All inputs are non-isolated from the power supply.

The digital inputs of TCW242-S in OPEN/CLOSED mode can be used for monitoring of devices with “dry contact” outputs – door contact switch, push-button, PIR detector, etc.

The following picture illustrates how a dry contact switch can be connected to the input of TCW242-S. One side of the contact is connected to “Digital In” and another side is connected to “GND” terminals.



The digital inputs of TCW242-S in COUNTER mode can be used for monitoring devices with a “dry contact” pulse output interface.

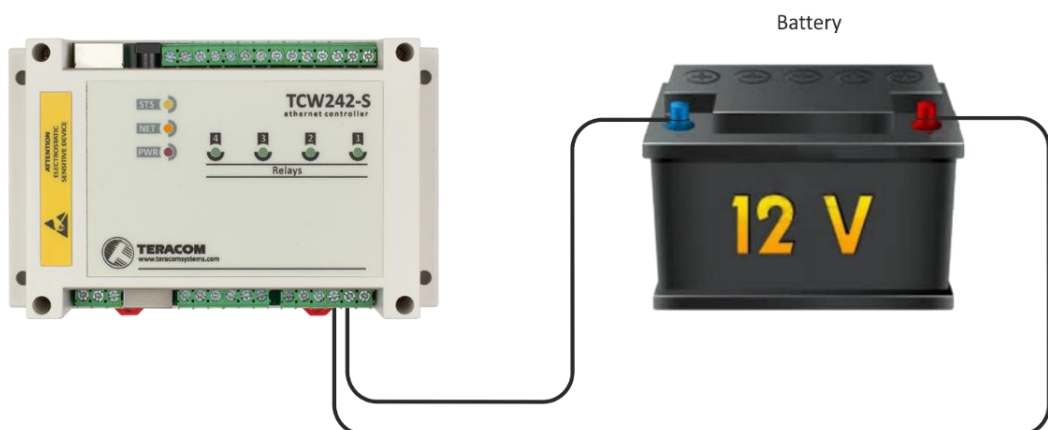
The maximum cable length should be up to 30 meters.

6.2.3. Analog inputs connection

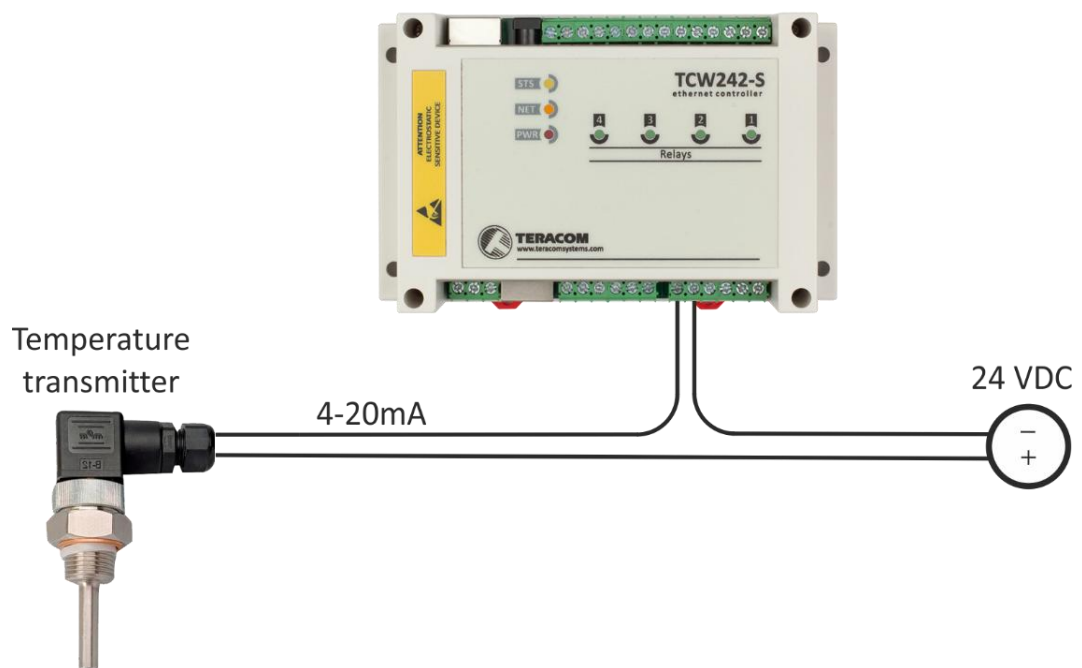
All inputs are non-isolated from the power supply.

Analog inputs of TCW242-S can be used for monitoring devices with voltage and current loop outputs. They can be connected directly to analog sensors for temperature, humidity, current/voltage transducers, etc.

The following picture illustrates how a battery can be connected to the analog input of TCW242-S in voltage mode. The positive terminal is connected to “Analog In” and the negative terminal to “GND”.



The following picture illustrates how an analog sensor for temperature with current loop output is connected to the analog input. The active terminal is connected to “Analog In” while the shield terminal is to “GND”.



The maximum cable length should be up to 30 meters.

6.2.4. RS-485 connection

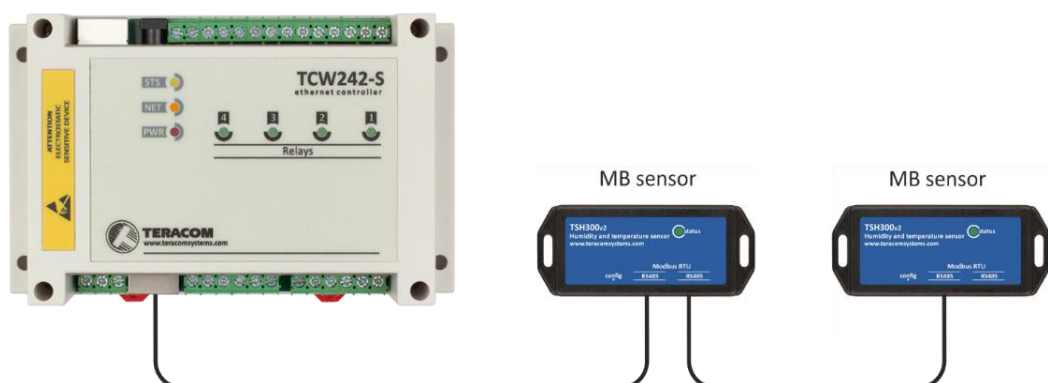
RS-485 interface is non-isolated from the power supply.

Up to 24 MODBUS RTU sensors can be connected to TCW242-S. The device supports Teracom and third-party sensors.

Connections can be realized by a standard RJ-45 connector. The used pinout is the one recommended in the document “MODBUS over Serial Line Specification and Implementation Guide” available on www.modbus.org.

It is mandatory to use 120ohms line terminators at both ends of the bus. TCW242-S incorporates one of the terminators and should be placed at one end. So the client should only take care of terminating at the other end of the line.

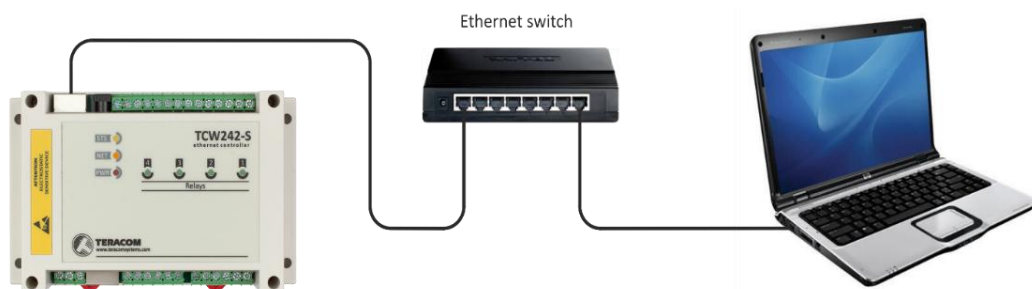
It is strongly recommended using “daisy-chained” (linear topology) for multiple sensors:



It is recommended to use only UTP/FTP cables and keep the total cable length up to 30 m, although functionality has been achieved on a longer distance.

6.2.5. Network connection

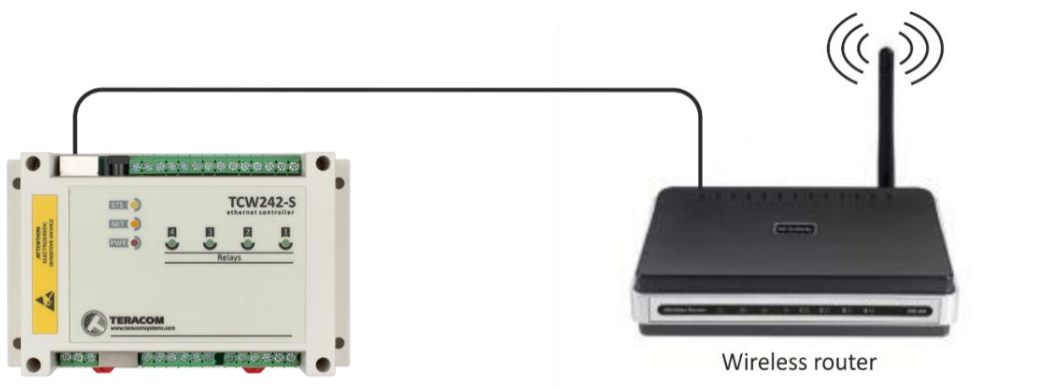
The Ethernet port of TCW242-S should be connected to 10/100 Base-T Ethernet hub, switch, or router.



For configuration, TCW242-S may be connected directly to the Ethernet port on a computer. The device support Auto-MDIX and it is not necessary to use a “crossover” cable, standard “straight-through” can be also used.



TCW242-S can be used in a wireless network by connecting through a wireless router.

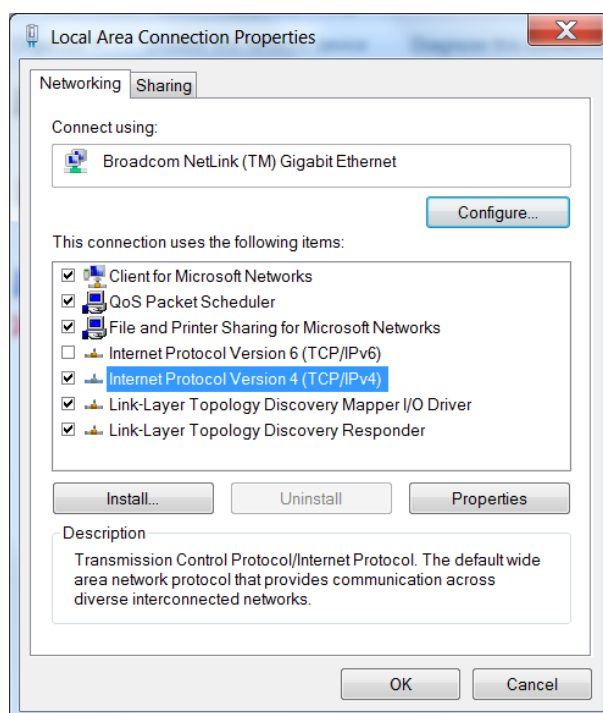


6.3. Communication setup

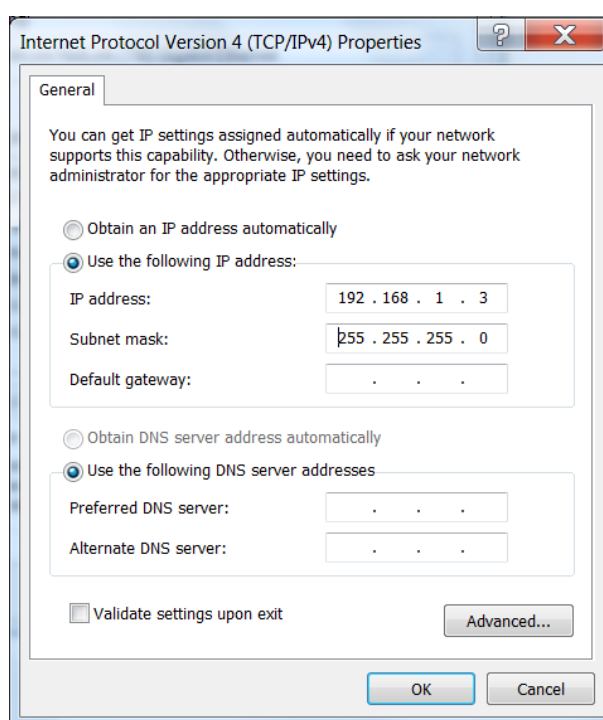
By default TCW242-S is delivered with the following network settings:

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

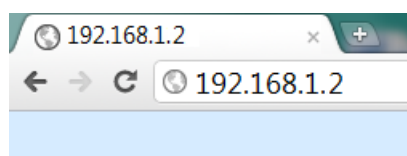
Communication with TCW242-S can be established by assigning a temporary IP address to the computer. For computers with Windows OS assigning of IP address is made in “Local area connection properties”:



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



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



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

A screenshot of a web browser's sign-in pop-up window. The window has a title bar and contains the following text: "Sign in", "http://192.168.1.2", and "Your connection to this site is not private". Below this, there are two input fields: "Username" and "Password". At the bottom right, there are two buttons: "Sign in" (highlighted in blue) and "Cancel".

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

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

All TCW controllers connected to LAN can be easily found by a free tool "TCW discoverer". It is available for Win and Mac operating systems and can be downloaded from www.teracomsystems.com.

7. Concept of setup

The main setup channel of the device is the web interface. It is recommended to use it, even more of the settings are available through SNMP and HTTP API commands.

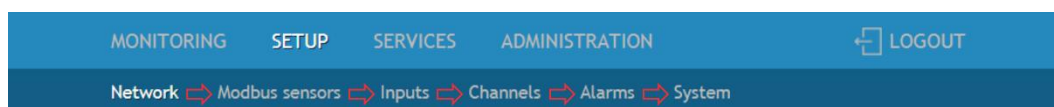
The device setup begins with network settings.

Then the so-called primary parameters (MODBUS RTU sensors/registers, analog inputs, and digital inputs) are set up.

The channels are formed from already set up primary parameters. Every channel can combine up to two primary parameters and constants with math operations. It is possible that a channel is formed by only one primary parameter.

Once all channels have been tuned, the alarm can be set up. It is important to remember that alarms work with channels, not with primary parameters. Each alarm can have up to two conditions. Different channels may be involved in the conditions. Alarms are independent of the channels but can also be assigned to a channel.

In other words, the right setup should follow the following order:



Once everything above is set correctly, the wanted services (data logger, SNMP, HTTP API, etc.) can be turned on.

8. Web interface

The web interface allows configuration and monitoring. All pages are UTF-8 encoded.

The controller supports a few active sessions. For the WEB interface, the device supports HTTP only (HTTPS is not supported).

8.1. Monitoring

This section displays the status of all channels, relays, and alarms – textually and graphically.

The pages "Data" and "Alarms" are refreshed automatically on at interval of 0 to 253 seconds. Zero means no automatic refresh. This parameter is set in section "Setup-System- Refresh of channels and alarms pages". By default, the refresh interval is 1 second.

8.1.1. Channels and relays

This page displays the status of relays and all monitored channels – their values and alarm status. The information is updated on 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	<input type="button" value="ON"/> <input type="button" value="OFF"/> <input type="button" value="Pulse"/>
2	Relay 2	OFF	<input type="button" value="ON"/> <input type="button" value="OFF"/> <input type="button" value="Pulse"/>
3	Relay 3	OFF	<input type="button" value="ON"/> <input type="button" value="OFF"/> <input type="button" value="Pulse"/>
4	Relay 4	OFF	<input type="button" value="ON"/> <input type="button" value="OFF"/> <input type="button" value="Pulse"/>
			<input type="button" value="All On"/> <input type="button" value="All Off"/> <input type="button" value="Pulse All"/>

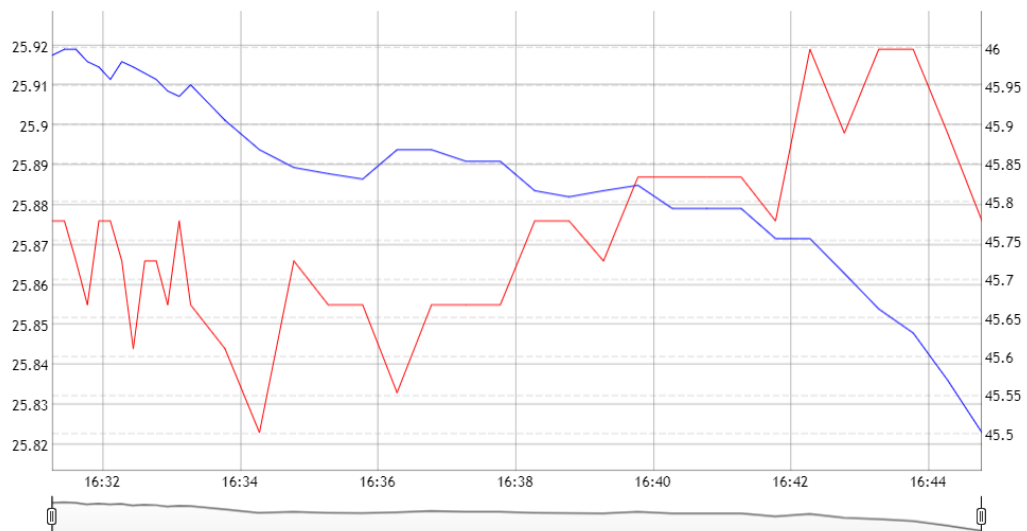
8.1.2. Alarms

This page displays the status of all alarms. The information is updated on 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

All channels together with their alarm statuses can be monitored graphically in tabs Graph-1 to Graph-6. Every graph supports up to 4 channels with up to 2 different dimensions. The curve's color for every channel is selectable. The alarms status colors are fixed. There are a few checkboxes for display modification.



Graphs show information from the past, so it is mandatory that the logger is active.

It is important to know that the information on graphs is static, it is not updated with the newest values. If you want to see the last information, the page should be reloaded. The information can be exported in a CSV file.

8.2. Setup

8.2.1. Network

The network parameters can be set on this page.

The module supports static and dynamic IP addresses.

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

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

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

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

8.2.2. Modbus sensors

8.2.2.1. Modbus RTU communication setup

Modbus RTU communication setup

This section allows you to set the communication parameters of the RS-485 interface – bit rate, parity, and a number of stop bits. By default, the settings are 19200, even parity, and 1 stop bit. It is mandatory that all sensors on the bus use the same bit rate, parity, and number of stop bits. Before adding any sensor to the interface its parameters should be set up properly.

In the right part of the section, there is a tool that scans the bus and reports the number of the found sensor together with their addresses. It is very useful at a time when you add new sensors. It is recommended to use a small address segment to speed up the scan process.

Playing with “Scan time-out for sensor answer” it is possible to find this parameter for an unknown sensor. The test starts with a large time-out (for example 500ms) and gradually

decreases the time until the sensor stops responding. In order for the operation sustainability, the found time should be increased with for example 20%.



Modbus RTU communication setup

Bit rate: 19200 Scan time-out for sensor answer, ms: 100 Max scan time: 24700

Parity: even First address: 1

Stop bits: 1 Last address: 247

Scan

Found: 1

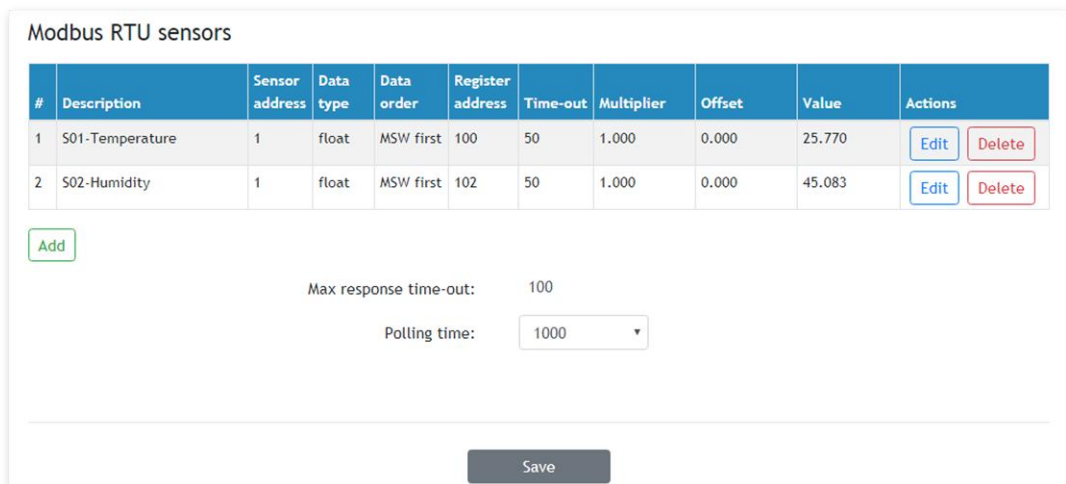
sensors with following addresses: 1

8.2.2.2. Modbus RTU sensors

This section allows you to add, delete or edit MODBUS RTU sensors/registers. All these are primary parameters and can be used in forming channels.

It is recommended to add sensors/registers one by one using the scan tool described in 8.2.2.1.

Up to 24 sensors/registers can be added. All they are shown in the table.



Modbus RTU sensors

#	Description	Sensor address	Data type	Data order	Register address	Time-out	Multiplier	Offset	Value	Actions
1	S01-Temperature	1	float	MSW first	100	50	1.000	0.000	25.770	Edit Delete
2	S02-Humidity	1	float	MSW first	102	50	1.000	0.000	45.083	Edit Delete

Add

Max response time-out: 100

Polling time: 1000

Save

According to the MODBUS convention, possible addresses for sensors are in the range of 1 to 247.

The Multiplier and Offset works as follows:

$$\text{Value} = (\text{Raw_Value} * \text{Multiplier}) + \text{Offset}.$$

If you want to see the raw value of the sensor/register set Multiplier=1 and Offset=0.

During operation, all sensors are polled consecutively. The controller expects an answer in "response time-out". If the sensor does not respond for that time, the controller addresses the next sensor. If the same sensor does not respond in series 3 times, the device assumes that it is not present in the system but continues to polls it.

According to the above paragraph, special attention should be paid to response time-out. On the one hand, the time-out should be large enough for the sensor to respond, but on the other hand, the sum of all sensor time-outs forms the "maximum response time-out" of the whole system. The "maximum response time-out" determines the response of the system.

For the sustainable system operation, polling of every sensor is made at a fixed time - "Polling time". It is user-selectable and can be 1, 2, 3, or 4 seconds. By default, it is 1 second.

It is recommended that "Polling time" ≥ "Maximum response time-out".

8.2.2.3. Sensor setup tool

The device has also a simple but useful tool for configuration and control of MODBUS RTU sensors. It can change the addresses and communication parameters of sensors from different manufacturers.

Communication setup

Bit rate

19200

Parity

even

Stop bits

1

Time-out

100

First address

1

Last address

247

Scan

Found: 1
sensors with following addresses: 1

MB Address

1

Sensor communication register setup

Bit rate register #

11

Parity, stop register #

12

Address register #

10

Value

19200

Value

1

Value

1

Read

Write

Transfer successful.

Sensor registers check/update

Start address

100

Data type

float

Number of registers

2

Data order

MSW first

Value

30.800

Read

Write

Transfer successful.

8.2.3. Input/Output

This page is used for the parameterization of relays, analog, and digital inputs.

8.2.3.1. Analog inputs

TCW242-S has 4 analog inputs. All they are isolated from the power supply but use the same ground. Every analog input can work in voltage (0-10V) or current loop mode (0-20mA).

For every analog input Multiplier and Offset can be set. They work as follows:

$$\text{Value} = (\text{Raw_Value} * \text{Multiplier}) + \text{Offset}.$$

If you want to see the raw value of the sensor/register set Multiplier=1 and Offset=0.

By default and after “Factory default settings” procedure: Multiplier=1.00, Offset=0.00 and Mode=0-10V

Analog inputs						
#	Description	Multiplier	Offset	Mode	Value	Actions
1	A01	1.000	0.000	0-10V	0.000	Edit
2	A02	1.000	0.000	0-10V	0.000	Edit
3	A03	1.000	0.000	0-10V	0.000	Edit
4	A04	1.000	0.000	0-10V	0.000	Edit

Example:

For humidity sensor HIH-4000-003 the following parameter (coming from datasheet) should be set for fine work:

Multiplier - 31.74
Offset - 0.826
Mode - 0-10V
Unit - %RH (is set in the channels web page)

The value of the multiplier is inverted of slope parameter (1/0.0315);

If the output voltage of this sensor is 3.198V the value of analog input will be 75.28% RH:

$$75.28 = (3.198 - 0.826) * 31.74$$

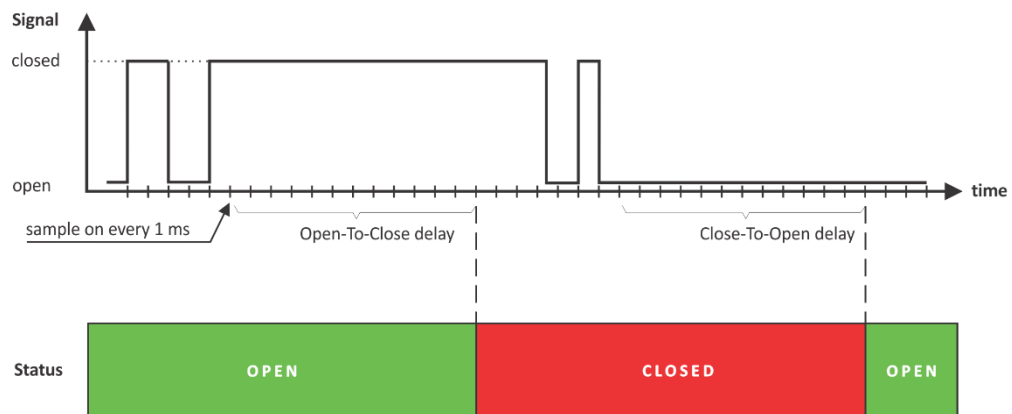
8.2.3.2. Digital inputs

TCW242-S has 4 digital inputs. All they are isolated from the power supply but use the same ground. Every digital input can work in OPEN/CLOSE or COUNTER mode. In COUNTER mode, counting can be made on rising, falling, or both edges.

For the counter mode, there is a possibility for setting up the counter initial 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. The delays are in the range of 5 to 60000mS. These delays can be used for additional digital filtering. The delays are applied in both modes.



In the picture above Open-to Close and Close-to-Open delays are set to 13mS.

8.2.3.3. Relay outputs

Each relay can be activated either remotely by the WEB interface, HTTP API, and SNMP or locally, from the alarm conditions.

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

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

8.2.4. Channels

This section allows you to add, delete or edit channels. All they can be monitored on monitoring pages and their values can be recorded periodically by the data logger.

Up to 24 channels can be set up.

#	Description	Parameter 1	OP 1	Parameter 2	OP 2	Coefficient 1	OP 3	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](#)

There are two types of channels – discrete (it forms by one digital input in OPEN/CLOSE mode) and general (it forms by up to 2 primary parameters and constants)

It is important to know that for general channels, the sequence of operations is OP1, then OP2, and finally OP3.

It is possible to use a digital input in OPEN/CLOSE mode to form a general channel. In this case, its values for operations are 0 (for CLOSE) and 1 (for OPEN).

8.2.5. Alarms

This section allows you to add, delete or edit alarms. Up to 24 alarms can be set up.

There are four types of alarms – Warning, Minor, Major and Critical.

Only channels can be used to form the alarms.

The alarms are independent but can be assigned to a channel. There isn't a restriction to which channel to assign the alarm.

For every alarm up to 2 conditions can be used. Both conditions can be logically joined using the logical operators AND and OR.

It is not mandatory to use the same channel in both conditions. In this case, conditions for two different channels can be combined 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](#)

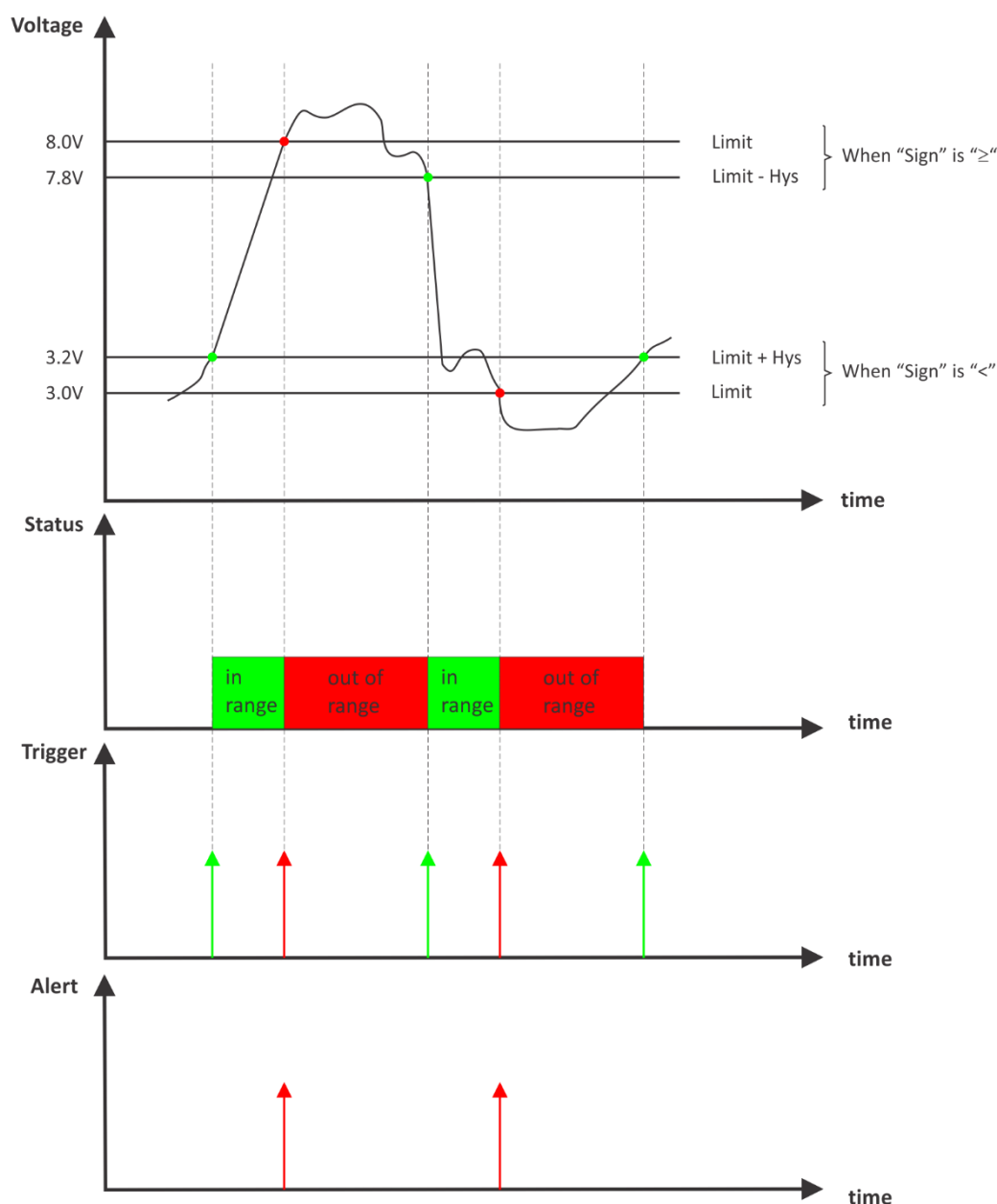
The “Limit” indicates the border of the working range for the observed channel.

A trigger condition occurs when the value goes higher (“Sign” is “≥”) and lower (“Sign” is “<”) than the limit. In both cases, the monitored parameter goes out of range.

Coming back in range for the observed channel is considered when the value goes higher than Limit + Hysteresis (“Sign” is “<”) or lower than Limit – Hysteresis (“Sign” is “≥”).

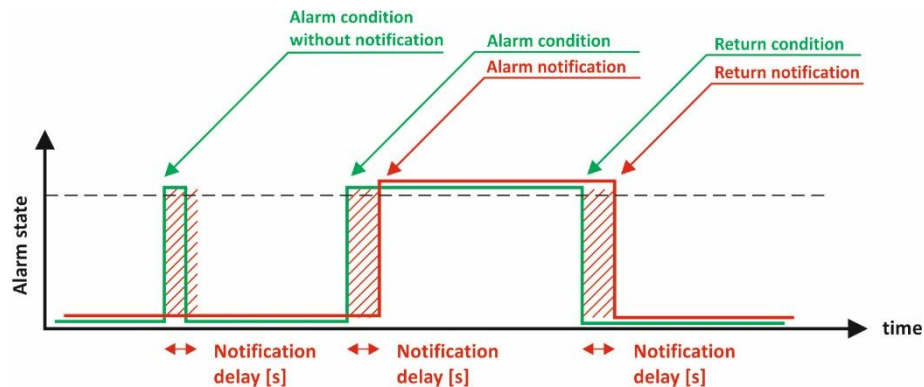
The hysteresis is used to prevent excessive triggering when the value vacillates around the trigger point.

It is strongly recommended not to use Hysteresis=0.0.



For every alarm “Return notification” option is available. If this option is chosen there will be a notification also when the parameter returns in range.

For every alarm, there is a “Notification delay” parameter. It is very useful as a filter for short alarm conditions.



8.2.6. System

On this page, some general system parameters can be set up.

General

System name

TCW242S

System location

Location

System contact

info@teracomsystems.com

Web access

Authentication

Enable

HTTP port

80

HTTP API

Authentication

Enable

Monitoring page automatic refresh

Interval (0-253), seconds

1

Display

Channels

☒

Relays

☒

Alarm colors

Warning

Minor

Major

Critical

Save

System name, system location, and system contact are used for device identification. They are presented in SNMP three and XML/JSON status files.

WEB access authentication by default is active with admin/admin details.

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

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

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

Alarms colors are fixed and are presented here just for information.

8.3. Services

8.3.1. NTP

The internal real-time clock of the controller can be set either manually or automatically. For automatic clock synchronization, the controller supports NTP (Network Time Protocol).

The screenshot displays a web-based configuration page for NTP. The 'Time setup' section includes a 'Time configuration' dropdown set to 'Manual'. Below it are input fields for 'NTP server IP/URL' (time.google.com), 'Time zone' (+00:00), 'Interval (h)' (12), and 'If not found (h)' (1). A 'Set time' field shows '15.05.2019, 13:54:13'. The 'Uptime' section shows '0days, 00:03:31'. A 'Save' button is at the bottom. A blue footer bar contains status information: 'Current time' (15.05.2019, 13:54:26), 'Last updated' (---), 'Status' (Undefined), and 'Stratum' (0).

Time setup	
Time configuration	Manual
NTP server IP/URL	time.google.com
Time zone	+00:00
Interval (h)	12
If not found (h)	1
Set time	15.05.2019, 13:54:13
Uptime	
Uptime	0days, 00:03:31
Save	
Current time 15.05.2019, 13:54:26	
Last updated ---	
Status Undefined	
Stratum 0	

By default NTP synchronization is disabled, server – time.google.com, Time zone +00:00, and interval of 12 hours.

8.3.2. SNMP

TCW242-S supports SNMPv2 and SNMPv3.

The default parameters 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

The more advanced SNMPv3 offers security management for administration, authentication, and privacy.

SNMPv3 provides the following configuration possibilities:

- No authentication and no privacy (noAuthNoPriv) - usually for monitoring;
- Authentication and no privacy (authNoPriv) - usually for control;

- Authentication and privacy (authPriv) - usually for downloading secrets.

User-based Authentication Mechanism is based on the following:

- MD5 message-digest algorithm in HMAC;
- SHA, an optional alternative algorithm;
- None authentication.

User-based Privacy Mechanism is based on the following:

- Data Encryption Standard (DES);
- Advanced Encryption Standard (AES);
- None encryption.

For alarm notification, an SNMP trap can be sent to up to 5 independent recipients. Different ports and communities can be used for every trap.

SNMP trap is sent also after reset.

The actual MIB file can be downloaded from the link at the bottom of the web page.

SNMP setup

SNMP

Enable

SNMP port

161

SNMPv3

Enable

Read community

public

Write community

private

Security User Name

teracom

Security Level

authNoPriv

Authentication Protocol

MD5

Authentication Password

Trc:Auth#135

Privacy Protocol

none

Privacy Password

Trc:Priv&246

SNMP traps

IP	Port	Community	Status	Action
192.168.32.30	162	public	Enable	Test
0.0.0.0	162	public	Disable	Test
0.0.0.0	162	public	Disable	Test
0.0.0.0	162	public	Disable	Test
0.0.0.0	162	public	Disable	Test

Save

[Download MIB File](#)

8.3.3. Logger

Logger setup

Logger: Enable

Logger mode: Time mode

Logger record sync: Disable

Log interval (10-3600), seconds: 120

Sync to the minute, (00-59): 0

Log interval, minutes: 15

HTTP upload setup

HTTP upload: Disable

Server:

Upload interval (h): 1h

Sync time: 00:00:00

Upload test log Force upload Download full log

Save

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

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

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

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

Example:

The current settings are:

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

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

The device is powered up.

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

There are two ways to reach the logger records:

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

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

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

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

Example:

The current time is 19:31, the Upload period is 3 hours and the Sync time is 9:00.

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

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

By default, the logger is disabled.

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

8.3.4. HTTP Post

TCW242-S can periodically upload an XML/JSON file to a dedicated server, using HTTP or HTTPS Post. The HTTPS is over TLS 1.0, TLS 1.1, and TLS 1.2 with RSA as a key exchange/agreement and authentication.

The XML/JSON file contains the current status of all monitored parameters and extra system information. The file format is chosen from the drop-down menu.

The screenshot shows the 'HTTP post setup' configuration window. It contains the following fields and options:

- HTTP post:** A dropdown menu set to 'Enable'.
- Data format:** A dropdown menu set to 'XML'.
- Protocol:** A dropdown menu set to 'https'.
- Server 1:** A text field containing 'http:// www.teracomsystems.com:443/temp/post/' with a 'Test' button (checked).
- Server 2:** A text field containing 'http:// www.teracomsystems.com:443/posttestlog.' with a 'Test' button (checked).
- Server 3:** A text field containing 'http://' with a 'Test' button (unchecked).
- Period (10-14400), seconds:** A text field containing '10'.
- Connect on any alarm:** An unchecked checkbox.
- Key:** An empty text field.
- Process answer:** A dropdown menu set to 'No'.
- Save:** A button at the bottom center.

The HTTP/HTTPS Post can be sent up to 3 independent servers. The HTTP servers can be addressed either by domain name or IP address.

The “Period” can be set between 10 and 14400 seconds. This parameter can be changed remotely also by HTTP API. The "Period" determines at what time the control software receives up-to-date information from TCW242-S and can, therefore, make changes to some of the parameters. The shorter is "Period", the closer to the real-time operation is the system. On the other hand, as shorter is the "Period" as higher is the data traffic through the network.

If the checkbox “Connect on any alarm” is selected, the HTTP/HTTPS Post request will be sent in an alarm condition.

The “Key” field is user-defined. Its value is sent in an XML/JSON file and can be used for device identification.

If the “Process Answer” option is enabled, TCW242-S will execute the commands, sent by the remote server as an answer to HTTP/HTTPS Post.

More about HTTP/HTTPS Post can be read in the HTTP API section.

8.3.5. Schedule

TCW242-S supports four schedules. In every schedule, up to four different tasks can be set.

The schedules are useful for creating tasks that vary with calendar dates. It is possible to combine two relays in control of one device - one relay follows the alarm condition and the other follows the schedule. In this case, more complex control can be arranged.

Schedule setup

Schedule 1

Mode	Date	Mon	Tue	Wed	Thu	Fri	Sat	Sun	ON	OFF
Weekly ▾	01.01.2016	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	08:00:00	17:00:00.0
Weekly ▾	01.01.2016	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	12:00:00	12:00:03.0
Disable ▾	01.01.2016	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	00:00:00	23:59:59.9
Disable ▾	01.01.2016	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	00:00:00	23:59:59.9

Schedule 2

Mode	Date	Mon	Tue	Wed	Thu	Fri	Sat	Sun	ON	OFF
Once ▾	19.07.2020	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	08:00:00	08:59:59.9
Disable ▾	01.01.2016	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	00:00:00	23:59:59.9
Disable ▾	01.01.2016	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	00:00:00	23:59:59.9
Disable ▾	01.01.2016	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	00:00:00	23:59:59.9

Schedule 3

Mode	Date	Mon	Tue	Wed	Thu	Fri	Sat	Sun	ON	OFF
Disable ▾	01.01.2016	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	00:00:00	23:59:59.9
Disable ▾	01.01.2016	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	00:00:00	23:59:59.9
Disable ▾	01.01.2016	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	00:00:00	23:59:59.9
Disable ▾	01.01.2016	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	00:00:00	23:59:59.9

Schedule 4

Mode	Date	Mon	Tue	Wed	Thu	Fri	Sat	Sun	ON	OFF
Disable ▾	01.01.2016	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	00:00:00	23:59:59.9
Disable ▾	01.01.2016	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	00:00:00	23:59:59.9
Disable ▾	01.01.2016	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	00:00:00	23:59:59.9
Disable ▾	01.01.2016	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	00:00:00	23:59:59.9

There are two types of schedule depending on repetition and duration – single and weekly tasks. Here are some examples:

- A single task for a time period:

Once ▾

19.07.2020

☐

☐

☐

☐

☐

☐

☐

08:00:00

09:00:00.0

With the above setting, there will be an event on 06.04.2020 starts at 08:00 and ends at 09:00. The resolution for “OFF time” is 0.1 seconds, which gives a possibility for very short pulses support.

- A weekly task for a time period:

Weekly	01.01.2016	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	08:00:00	17:00:00.0
--------	------------	-------------------------------------	-------------------------------------	-------------------------------------	-------------------------------------	-------------------------------------	--------------------------	--------------------------	----------	------------

With the above setting, there will be an event every working day of the week that starts at 08:00 and ends at 17:00.

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

Weekly	01.01.2016	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	23:00:00	24:00:00.0
Weekly	01.01.2016	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	00:00:00	01:00:00.0

With the above setting, there will be an event for 2 hours between Monday 23:00:00 and Tuesday 01:00:00.

8.3.6. Dynamic DNS

TCW242-S supports the following DNS services – DynDNS, No-IP, and DNS-O-Matric.

Dynamic DNS setup

Dynamic DNS

Enable

Service

DynDNS

Hostname

User

Password

Maintainer e-mail

DDNS last status

The current configuration is not valid.

The email is required of some providers for client's identification

Save

With dynamic DNS, TCW242-S can be accessed from the Internet having only a dynamic public IP address.

8.3.7. MODBUS

TCW242-S supports MODBUS TCP/IP.

Modbus TCP setup

Modbus TCP

Enable

Port

502

Save

The standard port for this protocol is 502.

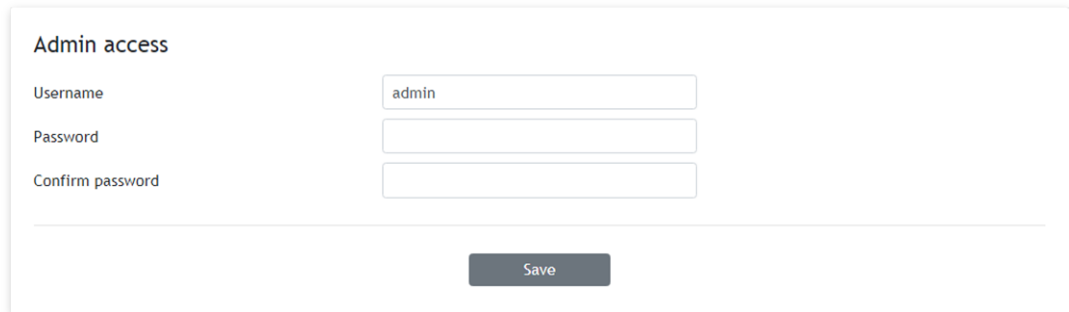
By default, Modbus is disabled.

More about this functionality can be read in the MODBUS section.

8.4. Administration

8.4.1. User/Pass

The TCW242-S supports one user only. It has administrative rights.



Admin access

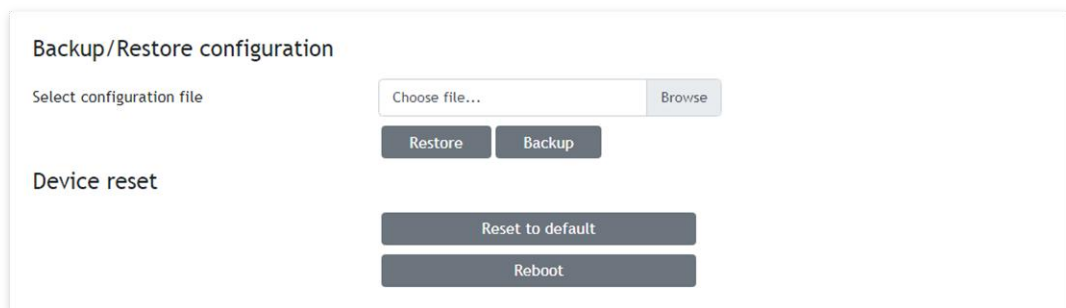
Username

Password

Confirm password

8.4.2. Backup/Restore

The TCW242-S supports backup and restoration of all user settings. All settings are saved in an XML backup file. This file can be used after this to restore on many devices. This is very useful for multiplying similar settings to a batch of controllers.



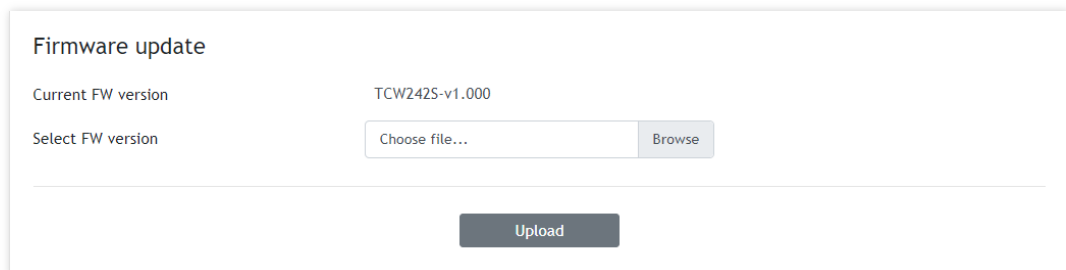
Backup/Restore configuration

Select configuration file

Device reset

8.4.3. FW update

The TCW242-S can be updated via a WEB interface.



Firmware update

Current FW version TCW242S-v1.000

Select FW version

To update the device follow the steps below:

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

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

8.5. Logout

The TCW242-S support a few session, but the good practice is to log out after finishing the work.

9. Protocols and API

9.1. SNMP

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

The TCW242-S can be configured and monitored through SNMP.

This could be done using every SNMP v.2 or v.3 compatible program. Parameters that can be changed, are grouped according to their functions in the tables below. To obtain a valid OID number, replace the symbol “!” with “1.3.6.1.4.1.38783”.

To save the changes **configurationSaved** (OID !.8.6.3.0) should be set 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), schedule1(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)}
l.8.3.1.1.1.7.?0	chOP2.?	read-write	Channel operand 2	INTEGER {none(0), multiplication(1), division(2), sum(3), subtract(4)}
l.8.3.1.1.1.8.?0	chCoef1.?	read-write	Channel coefficient 1 x1000 in Integer format	Integer32
l.8.3.1.1.1.9.?0	chOP3.?	read-write	Channel operand 3	INTEGER {none(0), multiplication(1), division(2), sum(3), subtract(4)}
l.8.3.1.1.1.10.?0	chCoef2.?	read-write	Channel coefficient 2 x1000 in Integer format	Integer32
l.8.3.1.1.1.11.?0	chUnit.?	read-write	Channel unit	DisplayString
l.8.3.1.1.1.12.?0	chCumulInitValue.?	read-write	Channel cumulative initial value	Integer32
l.8.3.1.1.1.13.?0	chValue.?	read-only	Channel value x1000 in Integer format	Integer32
l.8.3.1.1.1.14.?0	chCounter.?	read-only	Channel as 32-bit counter	Counter32
l.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
l.8.3.2.1.1.2.?0	relayState.?	read-write	Relay state	INTEGER {off(0), on(1)}
l.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
l.8.3.3.1.1.2.?0	alDescription.?	read-write	Alarm description	DisplayString
l.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)}
l.8.3.3.1.1.4.?0	alCond1Operand.?	read-write	Alarm condition 1 operand	INTEGER {larger(1), less(2)}
l.8.3.3.1.1.5.?0	alCond1Limit.?	read-write	Alarm condition 1 limit x1000 in Integer format	Integer32
l.8.3.3.1.1.6.?0	alCond1Hys.?	read-write	Alarm condition 1 hysteresis x1000 in Integer format	Integer32
l.8.3.3.1.1.7.?0	alCond1AlarmState.?	read-write	Alarm condition 1 discrete alarm state	INTEGER {open(0), closed(1)}
l.8.3.3.1.1.8.?0	alCondLogic.?	read-write	Alarm conditions logic	INTEGER {none(0), and(1), or(2)}
l.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) }

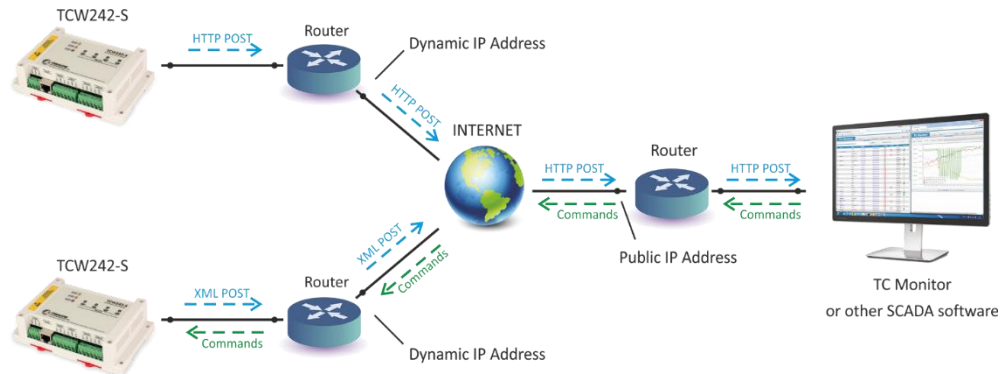
9.2. HTTP API

9.2.1. HTTP Post

TCW242-S can execute HTTP/HTTPS Post to upload XML/JSON files to a dedicated server.

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

The typical monitoring application is shown in the picture below:



HTTP/HTTPS post can be sent periodically or periodically plus on an alarm condition. As an answer, the server can send HTTP Get with the appropriate command - **see 9.2.3. HTTP commands**

To test HTTP/HTTPS Post follow the steps below:

- Save following code like post.php:

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

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

- Copy the post.php file on a public webserver with PHP support. To verify that the script is working properly, you can type the URL (for example www.yourserverURL.com/post.php) in your web browser. If all is OK, a web page with “The PHP script is working!” will be shown.
- Set the controller to send an HTTP/HTTPS POST to your web server. Enter the address (yourserverURL.com/post.php) in the URL field. Click on the “Test HTTP Post” button.
- If the HTTP/HTTPS POST is received and processed, “OK” will be shown close to the button. Along with this, an XML file will be created in the same directory, where post.php is located. The file name will contain time information and looks like 20190420103318_status.xml.

9.2.2. HTTP Get

HTTP Get can be used to monitor TCW242-S via XML or JSON files. The format is as follows:

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

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

See sections **9.2.4 XML file structure** and **9.2.5 JSON file structure** for details of files.

HTTP Get can be sent at any time to TCW242-S if it is on the same network or it has appropriate routing.

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

9.2.2.1. Commands

All commands used with HTTP/HTTPS Post can be used also with HTTP Get. The right format is:

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

Where:

yyy is the command;

xxx is the parameter.

Example:

`http://device.ip.address/status.xml?pper=300` will set Post period = 300 sec.

9.2.2.2. HTTP GET authentication

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

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

Example:

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

9.2.3. List of HTTP API commands

Command	Description
dataf=x	Data format XML/JSON for HTTP 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 m ON (m is 1,2,3 or 4) (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 m OFF (m is 1,2,3 or 4) (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
rtg=n	Toggle relay m state (m is 1,2,3 or 4) (n is 1,2,4 or 8) rtg=1 - will toggle relay 1 state rtg=2 - will toggle relay 2 state rtg=4 - will toggle relay 3 state rtg=8 - will toggle relay 4 state
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.)

9.2.4. XML file structure

```
<Monitor>
  <DeviceInfo>
    <DeviceName>TCW242-S</DeviceName>
    <HostName>TCW242S </HostName>
    <ID>5c:32:c5:00:a8:0b</ID>
    <FwVer>TCW242S-v1.000</FwVer>
    <MnflInfo>www.teracomsystems.com</MnflInfo>
    <SysContact>info@teracomsystems.com</SysContact>
    <SysName>TCW242S</SysName>
    <SysLocation>Location</SysLocation>
  </DeviceInfo>
  <CH>
    <CH1>
      <type>0</type>
      <description>Temperature</description>
      <value>24.386</value>
      <unit>°C</unit>
      <alarmbin>4</alarmbin>
      <alarm>Minor</alarm>
      <selch>3</selch>
    </CH1>
    <CH2>
      <type>0</type>
      <description>Humidity</description>
      <value>51.323</value>
      <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>
      <unit/>
      <alarmbin>1</alarmbin>
      <alarm>Normal</alarm>
      <selch>31</selch>
    </CH3>
    <CH4>
      <type>0</type>
      <description>V04</description>
      <value>----</value>
      <unit/>
      <alarmbin>0</alarmbin>
      <alarm/>
      <selch>0</selch>
    </CH4>
    <CH5>
      <type>0</type>
      <description>V05-Voltage</description>
      <value>0.000</value>
      <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>
      <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>
      <unit>kWh</unit>
      <alarmbin>0</alarmbin>
      <alarm/>
      <selch>27</selch>
  </CH>
</Monitor>
```

```

</CH7>
<CH8>
  <type>0</type>
  <description>V08</description>
  <value>---</value>
  <unit/>
  <alarmbin>0</alarmbin>
  <alarm/>
  <selch>0</selch>
</CH8>
<CH9>
  <type>0</type>
  <description>V09</description>
  <value>---</value>
  <unit/>
  <alarmbin>0</alarmbin>
  <alarm/>
  <selch>0</selch>
</CH9>
<CH10>
  <type>0</type>
  <description>V10</description>
  <value>---</value>
  <unit/>
  <alarmbin>0</alarmbin>
  <alarm/>
  <selch>0</selch>
</CH10>
<CH11>
  <type>0</type>
  <description>V11</description>
  <value>---</value>
  <unit/>
  <alarmbin>0</alarmbin>
  <alarm/>
  <selch>0</selch>
</CH11>
<CH12>
  <type>0</type>
  <description>V12</description>
  <value>---</value>
  <unit/>
  <alarmbin>0</alarmbin>
  <alarm/>
  <selch>0</selch>
</CH12>
<CH13>
  <type>0</type>
  <description>V13</description>
  <value>---</value>
  <unit/>
  <alarmbin>0</alarmbin>
  <alarm/>
  <selch>0</selch>
</CH13>
<CH14>
  <type>0</type>
  <description>V14</description>
  <value>---</value>
  <unit/>
  <alarmbin>0</alarmbin>
  <alarm/>
  <selch>0</selch>
</CH14>
<CH15>
  <type>0</type>
  <description>V15</description>
  <value>---</value>
  <unit/>
  <alarmbin>0</alarmbin>
  <alarm/>
  <selch>0</selch>
</CH15>
<CH16>
  <type>0</type>
  <description>V16</description>
  <value>---</value>

```

```

<unit/>
<alarmbin>0</alarmbin>
<alarm/>
<selch>0</selch>
</CH16>
<CH17>
  <type>0</type>
  <description>V17</description>
  <value>---</value>
  <unit/>
  <alarmbin>0</alarmbin>
  <alarm/>
  <selch>0</selch>
</CH17>
<CH18>
  <type>0</type>
  <description>V18</description>
  <value>---</value>
  <unit/>
  <alarmbin>0</alarmbin>
  <alarm/>
  <selch>0</selch>
</CH18>
<CH19>
  <type>0</type>
  <description>V19</description>
  <value>---</value>
  <unit/>
  <alarmbin>0</alarmbin>
  <alarm/>
  <selch>0</selch>
</CH19>
<CH20>
  <type>0</type>
  <description>V20</description>
  <value>---</value>
  <unit/>
  <alarmbin>0</alarmbin>
  <alarm/>
  <selch>0</selch>
</CH20>
<CH21>
  <type>0</type>
  <description>V21</description>
  <value>---</value>
  <unit/>
  <alarmbin>0</alarmbin>
  <alarm/>
  <selch>0</selch>
</CH21>
<CH22>
  <type>0</type>
  <description>V22</description>
  <value>---</value>
  <unit/>
  <alarmbin>0</alarmbin>
  <alarm/>
  <selch>0</selch>
</CH22>
<CH23>
  <type>0</type>
  <description>V23</description>
  <value>---</value>
  <unit/>
  <alarmbin>0</alarmbin>
  <alarm/>
  <selch>0</selch>
</CH23>
<CH24>
  <type>0</type>
  <description>V24</description>
  <value>---</value>
  <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>
<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

<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

9.2.5. JSON file structure

```

{
  "Monitor": {
    "DeviceInfo": {
      "DeviceName": "TCW242-S",
      "HostName": "TCW242S ",
      "ID": "5c:32:c5:00:a8:0b ",
      "FwVer": "TCW242S-v1.000",
      "MnfInfo": "www.teracomsystems.com",
      "SysContact": "info@teracomsystems.com",
      "SysName": "TCW242S",
      "SysLocation": "Location"
    },
    "CH": {
      "CH1": {
        "type": "0",
        "description": "Temperature",
        "value": "24.386",
        "unit": "°C",
        "alarmbin": "4",
        "alarm": "Minor",
        "selch": "3"
      },
      "CH2": {
        "type": "0",
        "description": "Humidity",
        "value": "52.490",
        "unit": "RH",
        "alarmbin": "1",
        "alarm": "Normal",
        "selch": "4"
      }
    }
  }
}

```

```

"CH3": {
  "type": "2",
  "description": "Digital Input 1",
  "value": "OPEN",
  "unit": "",
  "alarmbin": "1",
  "alarm": "Normal",
  "selch": "31"
},
"CH4": {
  "type": "0",
  "description": "V04",
  "value": "---",
  "unit": "",
  "alarmbin": "0",
  "alarm": "",
  "selch": "0"
},
"CH5": {
  "type": "0",
  "description": "V05-Voltage",
  "value": "0.000",
  "unit": "V",
  "alarmbin": "1",
  "alarm": "Normal",
  "selch": "27"
},
"CH6": {
  "type": "0",
  "description": "V06-Current",
  "value": "0.000",
  "unit": "A",
  "alarmbin": "1",
  "alarm": "Normal",
  "selch": "28"
},
"CH7": {
  "type": "1",
  "description": "V07-Energy",
  "value": "6587.396",
  "unit": "kWh",
  "alarmbin": "0",
  "alarm": "",
  "selch": "27"
},
"CH8": {
  "type": "0",
  "description": "V08",
  "value": "---",
  "unit": "",
  "alarmbin": "0",
  "alarm": "",
  "selch": "0"
},
"CH9": {
  "type": "0",
  "description": "V09",
  "value": "---",
  "unit": "",
  "alarmbin": "0",
  "alarm": "",
  "selch": "0"
},
"CH10": {
  "type": "0",
  "description": "V10",
  "value": "---",
  "unit": "",
  "alarmbin": "0",
  "alarm": "",
  "selch": "0"
},
"CH11": {
  "type": "0",
  "description": "V11",
  "value": "---",
  "unit": "",

```

```

    "alarmbin": "0",
    "alarm": "",
    "selch": "0"
  },
  "CH12": {
    "type": "0",
    "description": "V12",
    "value": "---",
    "unit": "",
    "alarmbin": "0",
    "alarm": "",
    "selch": "0"
  },
  "CH13": {
    "type": "0",
    "description": "V13",
    "value": "---",
    "unit": "",
    "alarmbin": "0",
    "alarm": "",
    "selch": "0"
  },
  "CH14": {
    "type": "0",
    "description": "V14",
    "value": "---",
    "unit": "",
    "alarmbin": "0",
    "alarm": "",
    "selch": "0"
  },
  "CH15": {
    "type": "0",
    "description": "V15",
    "value": "---",
    "unit": "",
    "alarmbin": "0",
    "alarm": "",
    "selch": "0"
  },
  "CH16": {
    "type": "0",
    "description": "V16",
    "value": "---",
    "unit": "",
    "alarmbin": "0",
    "alarm": "",
    "selch": "0"
  },
  "CH17": {
    "type": "0",
    "description": "V17",
    "value": "---",
    "unit": "",
    "alarmbin": "0",
    "alarm": "",
    "selch": "0"
  },
  "CH18": {
    "type": "0",
    "description": "V18",
    "value": "---",
    "unit": "",
    "alarmbin": "0",
    "alarm": "",
    "selch": "0"
  },
  "CH19": {
    "type": "0",
    "description": "V19",
    "value": "---",
    "unit": "",
    "alarmbin": "0",
    "alarm": "",
    "selch": "0"
  },
  "CH20": {

```

```

    "type": "0",
    "description": "V20",
    "value": "---",
    "unit": "",
    "alarmbin": "0",
    "alarm": "",
    "selch": "0"
  },
  "CH21": {
    "type": "0",
    "description": "V21",
    "value": "---",
    "unit": "",
    "alarmbin": "0",
    "alarm": "",
    "selch": "0"
  },
  "CH22": {
    "type": "0",
    "description": "V22",
    "value": "---",
    "unit": "",
    "alarmbin": "0",
    "alarm": "",
    "selch": "0"
  },
  "CH23": {
    "type": "0",
    "description": "V23",
    "value": "---",
    "unit": "",
    "alarmbin": "0",
    "alarm": "",
    "selch": "0"
  },
  "CH24": {
    "type": "0",
    "description": "V24",
    "value": "---",
    "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"
},
"Sys": {
  "hwerr": "",
  "HighAlarmbin": "4",
  "HighAlarm": "Major"
},
"Time": {
  "Date": "19.07.2021",
  "Time": "10:30:00"
}
}
}

```

9.3. MODBUS TCP/IP

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

9.3.1. Codes and answers

9.3.1.1. Read Coil Status (FC=01)

Request

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

01 0064 0001

01: The Function Code 1 (read Coil Status)

0064: The Data Address of the coil to read (0064 hex = 100)

0001: The total number of coils requested. (01 hex = 1)

Response

01 01 01

01: The Function Code 1 (read Coil Status)

01: The number of data bytes to follow

01: 7 space holders & Coils 1 (0000 0001)

Due to the number of coils requested, the last data field **01** contains the status of only 1 coil. The 7 most significant bits in this data field are filled in with zeroes. The activated relay is 1.

9.3.1.2. Force Single Coil (FC=05)

Request

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

05 0064 FF00

05: The Function Code 5 (Force Single Coil)

0064: The Data Address of the coil. (0064 hex = 100)

FF00: The status to write (FF00 = ON, 0000 = OFF)

Response

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

05 0064 FF00

05: The Function Code 5 (Force Single Coil)

0064: The Data Address of the coil. (0064 hex = 100)

FF00: The status written (FF00 = ON, 0000 = OFF)

9.3.1.3. Read Holding Registers (FC=03)

Request

This command is requesting the content of holding registers 400 – Channel 1 value.

03 0190 0002

03: The Function Code 3 (read Holding Registers)

0064: The Data Address of the first register requested (0190 hex = 400)

0002: The total number of registers requested. (read 2 registers each 2 bytes = 4 bytes)

Response

03 04 41 C3 B0 21

03: The Function Code 3 (read Channel 1 Holding Registers)
 04: The number of data bytes to follow (2 registers x 2 bytes each = 4 bytes)
 41C3 B021: 4 bytes value
 All holding registers with float values are sent in big-endian.
 In the example above, the value of 24.4610004 is sent.

Request

This command is requesting the content of holding registers 5000 – Channel 1 description.

03 1388 0008

03: The Function Code 3 (read Holding Registers)
 1388: The Data Address of the first register requested (1388 hex = 5000)
 0008: The total number of registers requested (read 8 registers each 2 bytes = 16 bytes)

Response

03 10 54 65 6D 70 65 72 61 74 75 72 65 00 00 00 00 00

03: The Function Code 3 (read Analog Output Holding Registers)
 10: The number of data bytes to follow (8 registers x 2 bytes each = 16 bytes)
 54 65 6D 70 65 72 61 74 75 72 65 00 00 00 00 00: 16 bytes value
 All holding registers with strings are sent in big-endian.
 The answer is padded with 0.
 In the example above, the string “Temperature” is sent.

9.3.1.4. Exception codes

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

01 Illegal function

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

02 Illegal data address

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

9.3.2. Address table

Parameter	FC	Address (Decimal)	Data size
Read Relay 1	01	100	Discrete
Read Relay 2	01	101	Discrete
Read Relay 3	01	102	Discrete
Read Relay 4	01	103	Discrete
Write Relay 1	05	100	Discrete
Write Relay 2	05	101	Discrete
Write Relay 3	05	102	Discrete

Write Relay 4	05	103	Discrete
Read Channel 1	03	400	32-bit Float
Read Channel 2	03	402	32-bit Float
Read Channel 3	03	404	32-bit Float
Read Channel 4	03	406	32-bit Float
Read Channel 5	03	408	32-bit Float
Read Channel 6	03	410	32-bit Float
Read Channel 8	03	412	32-bit Float
Read Channel 9	03	414	32-bit Float
Read Channel 10	03	416	32-bit Float
Read Channel 11	03	418	32-bit Float
Read Channel 12	03	420	32-bit Float
Read Channel 13	03	422	32-bit Float
Read Channel 14	03	424	32-bit Float
Read Channel 15	03	426	32-bit Float
Read Channel 16	03	428	32-bit Float
Read Channel 17	03	430	32-bit Float
Read Channel 18	03	432	32-bit Float
Read Channel 19	03	434	32-bit Float
Read Channel 20	03	436	32-bit Float
Read Channel 21	03	438	32-bit Float
Read Channel 22	03	440	32-bit Float
Read Channel 23	03	442	32-bit Float
Read Channel 24	03	444	32-bit Float
Channels 1:-24, type Counter			
Read Channel 1	03	500	32-bit unsign int
Read Channel 2	03	502	32-bit unsign int
Read Channel 3	03	504	32-bit unsign int
Read Channel 4	03	506	32-bit unsign int
Read Channel 5	03	508	32-bit unsign int
Read Channel 6	03	510	32-bit unsign int
Read Channel 8	03	512	32-bit unsign int
Read Channel 9	03	514	32-bit unsign int
Read Channel 10	03	516	32-bit unsign int
Read Channel 11	03	518	32-bit unsign int
Read Channel 12	03	520	32-bit unsign int
Read Channel 13	03	522	32-bit unsign int
Read Channel 14	03	524	32-bit unsign int
Read Channel 15	03	526	32-bit unsign int
Read Channel 16	03	528	32-bit unsign int
Read Channel 17	03	530	32-bit unsign int
Read Channel 18	03	532	32-bit unsign int
Read Channel 19	03	534	32-bit unsign int
Read Channel 20	03	536	32-bit unsign int
Read Channel 21	03	538	32-bit unsign int
Read Channel 22	03	540	32-bit unsign int

Read Channel 23	03	542	32-bit unsign int
Read Channel 24	03	544	32-bit unsign int
Read Channel 1 Dimension	03	1000	16 bytes UTF-8
Read Channel 2 Dimension	03	1008	16 bytes UTF-8
Read Channel 3 Dimension	03	1016	16 bytes UTF-8
Read Channel 4 Dimension	03	1024	16 bytes UTF-8
Read Channel 5 Dimension	03	1032	16 bytes UTF-8
Read Channel 6 Dimension	03	1040	16 bytes UTF-8
Read Channel 7 Dimension	03	1048	16 bytes UTF-8
Read Channel 8 Dimension	03	1056	16 bytes UTF-8
Read Channel 9 Dimension	03	1064	16 bytes UTF-8
Read Channel 10 Dimension	03	1072	16 bytes UTF-8
Read Channel 11 Dimension	03	1080	16 bytes UTF-8
Read Channel 12 Dimension	03	1088	16 bytes UTF-8
Read Channel 13 Dimension	03	1096	16 bytes UTF-8
Read Channel 14 Dimension	03	1104	16 bytes UTF-8
Read Channel 15 Dimension	03	1112	16 bytes UTF-8
Read Channel 16 Dimension	03	1120	16 bytes UTF-8
Read Channel 17 Dimension	03	1128	16 bytes UTF-8
Read Channel 18 Dimension	03	1136	16 bytes UTF-8
Read Channel 19 Dimension	03	1144	16 bytes UTF-8
Read Channel 20 Dimension	03	1152	16 bytes UTF-8
Read Channel 21 Dimension	03	1160	16 bytes UTF-8
Read Channel 22 Dimension	03	1168	16 bytes UTF-8
Read Channel 23 Dimension	03	1176	16 bytes UTF-8
Read Channel 24 Dimension	03	1184	16 bytes UTF-8
Read Channel 1 Description	03	5000	16 bytes UTF-8
Read Channel 2 Description	03	5008	16 bytes UTF-8
Read Channel 3 Description	03	5016	16 bytes UTF-8
Read Channel 4 Description	03	5024	16 bytes UTF-8
Read Channel 5 Description	03	5032	16 bytes UTF-8
Read Channel 6 Description	03	5040	16 bytes UTF-8
Read Channel 7 Description	03	5048	16 bytes UTF-8
Read Channel 8 Description	03	5056	16 bytes UTF-8
Read Channel 9 Description	03	5064	16 bytes UTF-8
Read Channel 10 Description	03	5072	16 bytes UTF-8
Read Channel 11 Description	03	5080	16 bytes UTF-8
Read Channel 12 Description	03	5088	16 bytes UTF-8
Read Channel 13 Description	03	5096	16 bytes UTF-8
Read Channel 14 Description	03	5104	16 bytes UTF-8
Read Channel 15 Description	03	5112	16 bytes UTF-8
Read Channel 16 Description	03	5120	16 bytes UTF-8
Read Channel 17 Description	03	5128	16 bytes UTF-8
Read Channel 18 Description	03	5136	16 bytes UTF-8
Read Channel 19 Description	03	5144	16 bytes UTF-8
Read Channel 20 Description	03	5152	16 bytes UTF-8

Read Channel 21 Description	03	5160	16 bytes UTF-8
Read Channel 22 Description	03	5168	16 bytes UTF-8
Read Channel 23 Description	03	5176	16 bytes UTF-8
Read Channel 24 Description	03	5184	16 bytes UTF-8

10. Data Logger

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

The number of records depends on how long descriptions and what kind of characters are used. In the worst case (15 bytes description with characters from the highest part of UTF-8) the number of records is about 52371. This number is enough for 36 days with records every 1 minute.

In most cases, the data logger can keep 71400 records, which is enough for 49 days with records every 1 minute.

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

The first row of the log file is always the header. All rows, including the header, start with a 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 channel 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

Following 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;19.07.2021,16:49:49; 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;19.07.2021,16:49:49; 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;19.07.2021,16:49:49;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;19.07.2021,16:49:49;Time;...3;5;0;0;0;0;;;;;;;;;;;;;;...
```

11. Factory default settings

TCW242-S can be restored to its original factory default settings in 3 different ways.

11.1. Factory default from the WEB interface

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

11.2. Factory default with the reset button

If the reset button is pressed for more than 5 seconds, while the device is working, all Network settings go to factory default.

11.3. General factory default with the reset button

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

- Press and hold the RESET button, then turn on the power supply;
- Yellow LED shines and red LED blinks about 5 times on a second;
- After about 5 seconds red LED will turn off, the button can be released;
- Yellow LED flashes on 1 second and red LED shines – the device is in working mode, with factory default settings.



The factory default settings are:

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

12. Environment information

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

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

13. Safety

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

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

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

There is a risk of overheating (damage) of the controller if recommended free spaces to adjacent devices are not ensured. The joint part with the external component shall have space for attachment/removal of the cable after installation.

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

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

- Ensure that the device is installed correctly, refer to this document;
- Log in to the devices via browser program;
- Make proper setup;
- Short the “Din1” and “GND”;
- Go to “Input/Output page” of WEB interface - proper parameter value should be displayed in section Digital inputs;
- Go to “Modbus sensors page” of WEB interface and Install Modbus RTU (TSH3XX/TST3XX or third-party) sensor - proper parameters value should be displayed;
- At the same time flashing “STS” led should indicate the proper operation;

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

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

14. 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.

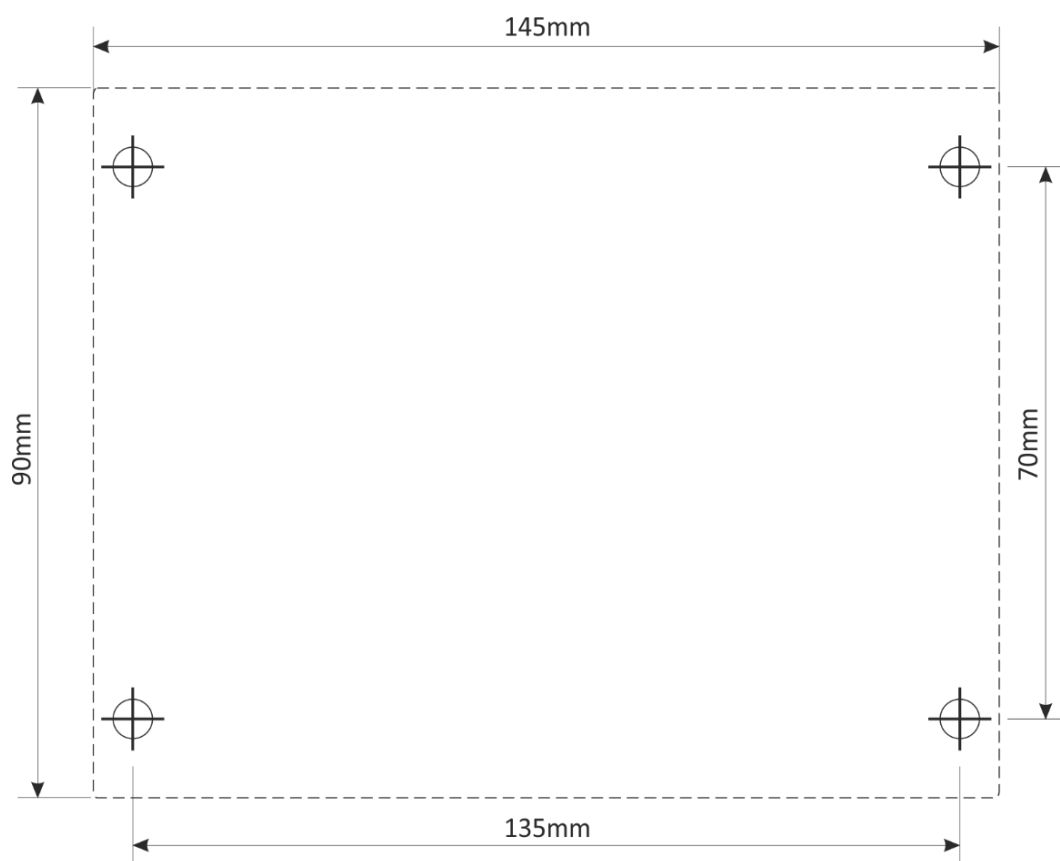


Fig.1

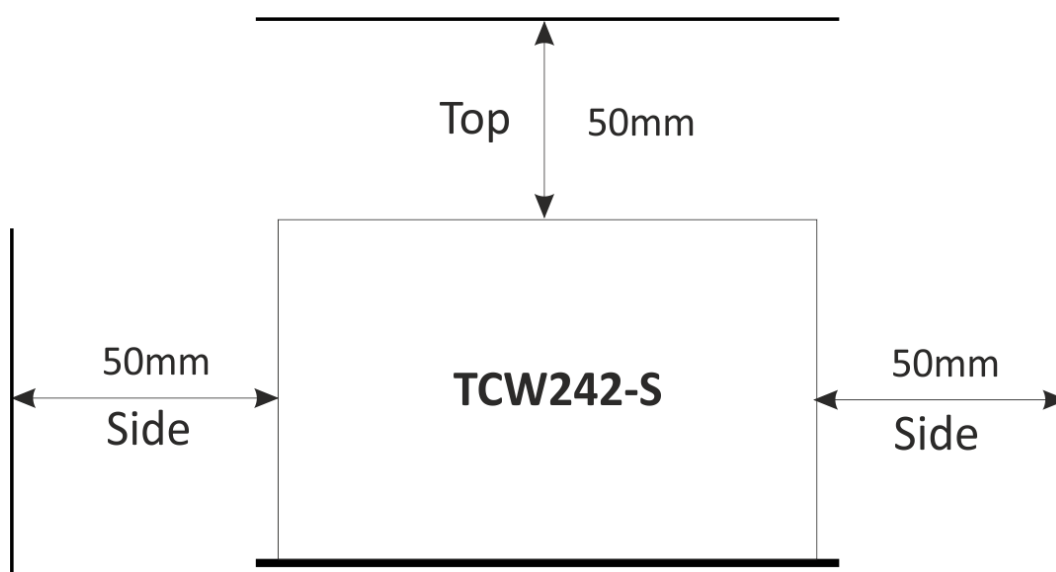


Fig.2