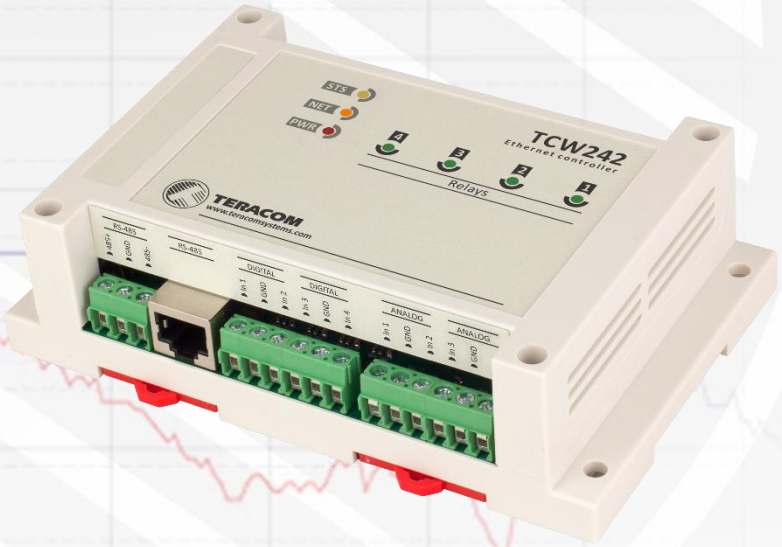
control solutions
TERACOM



Industrial IoT module TCW242

USER MANUAL

1. Introduction

TCW242 is an Industrial IoT module designed for monitoring and control applications over Ethernet networks.

The device provides four digital inputs, four analog inputs, four relay outputs, and an RS-485 interface for connecting MODBUS RTU sensors and remote I/O modules. The analog inputs support both 0–10 V and 0–20 mA signals, allowing direct connection of a wide range of industrial sensors and transmitters.

TCW242 supports up to 24 monitored channels and up to 24 independent alarms. Alarm notifications can be delivered through email, SNMP traps, HTTP/HTTPS POST, and MQTT messages. Relay outputs can be controlled remotely through HTTP API commands and MQTT, or locally based on monitored parameter values.

The module supports HTTP API, SNMP, MODBUS TCP/IP, and MQTT, enabling integration with SCADA systems, cloud platforms, Building Management Systems (BMS), and custom applications. MQTT functionality includes publisher and subscriber modes, multiple JSON payload formats, and support for server and client certificates for secure communication.

All monitored parameters and relay states can be stored in the built-in data logger and periodically uploaded to a remote server in CSV format. The device also supports periodic HTTP/HTTPS POST transmission of status data in XML or JSON format.

TCW242 is a versatile solution for industrial automation, environmental monitoring, energy management, and remote monitoring applications.

2. Features

- Ethernet 10/100 Mbps connectivity
- Web-based configuration and control
- Four digital inputs (S0 compatible, EN62053-31)
- Four analog inputs (0–10 V / 0–20 mA)
- Four relay outputs (NO/NC contacts)
- RS-485 interface for MODBUS RTU devices
- Support for up to 24 MODBUS RTU registers
- Up to 24 monitored channels
- Up to 24 independent alarms
- Four schedules for automatic relay control
- Custom logical functions
- HTTP/HTTPS API support
- MQTT 3.1.1 support (publisher/subscriber)
- Four MQTT JSON payload formats
- MQTT certificate support
- MODBUS TCP/IP support
- SNMP v2c and SNMPv3 support
- Alarm notifications via SNMP traps
- Periodic HTTP/HTTPS POST (XML/JSON)
- Data logger with CSV export and upload
- Email notifications with TLS support
- Configuration backup and restore
- Remote firmware update via web browser

3. Applications

- Industrial automation
  TCW242 can be used for monitoring and control of industrial equipment, production lines, pump stations, compressor systems, and other automation applications. The combination of digital inputs, analog inputs, relay outputs, and MODBUS RTU connectivity enables integration with a wide range of industrial sensors and actuators. MQTT support allows monitored data and alarms to be exchanged with IoT platforms and distributed monitoring systems.
- Environmental monitoring
  The device is suitable for monitoring temperature, humidity, air quality, water level, pressure, illumination, and other environmental parameters. Alarm notifications and data logging allow continuous supervision of critical conditions in indoor and outdoor installations.
- Building and facility management
  TCW242 can be integrated into Building Management Systems (BMS) for monitoring HVAC equipment, electrical systems, water systems, access control equipment, and other building infrastructure. Standard communication interfaces simplify integration with existing monitoring platforms.
- Energy and utility monitoring
  Using S0-compatible digital inputs and MODBUS RTU energy meters, the device can monitor electricity, water, gas, and heat consumption. Logged data can be periodically uploaded to remote servers for analysis, reporting, and energy management.
- Remote monitoring systems
  The built-in Ethernet interface enables deployment in remote monitoring applications where operational data must be collected and made available to central monitoring software. The device can gather information from local sensors and MODBUS RTU devices, generate alarms, store historical records, and provide remote access to monitored parameters and outputs.

4. Specifications

- Physical characteristics
  Dimensions: 145 x 90 x 40 mm
  Weight: 200 g
- Environmental conditions
  Operating temperature range: -20 to 55 °C
  Storage temperature range: -25 to 60 °C
  Operating relative humidity range: 10 to 80 %RH (non-condensing)
- Power supply
  Operating voltage range (including -15/+20% according to IEC 62368-1): 10 to 28 VDC
  Current consumption: 0.35A @ 12 VDC (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: +5.5 VDC
  Sampling period: 1ms

Digital filtering time interval: 5 to 60000ms
Minimum pulse width (COUNTER mode): 8 ms (min.) at 5 ms filtering time
Minimum pause between pulses (COUNTER mode): 8 ms (min.) at 5 ms filtering time
Maximum counting frequency: 65 Hz (at 50% duty cycle and 5 ms filtering time)

- Analog inputs
  Isolation: Non-isolated
  Type: Single-ended
  Resolution: 12 bits
  Mode: Voltage or current loop
  Input Range: 0/10 V or 0/20 mA
  Accuracy: ±1 %
  Input impedance in voltage mode: 1 MΩ (min.)
  Input impedance in current loop mode: 455 Ω
- 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 MΩ (min.) @ 500 VDC
  Mechanical endurance: 10 000 000 operations
  Electrical endurance: 100 000 operations @ 3 A resistive load
  Contact resistance: 50 mΩ max. (initial value)
  Minimum pulse output: 0.1 Hz at rated load

  Attention!
  The device does not contain any internal overcurrent protection facilities on the relay's contact lines. External fuses or short circuit current limiting circuit breakers, rated to 3 Amps, are to be used for overcurrent protection of the connecting lines.
- Internal FLASH memory
  Settings segment endurance: 100 000 cycles (Every setting change is a memory cycle).
  Data logger segment endurance: 100 000 full scrolls (70000 records) of the logger.
  Update segment endurance: 100 000 cycles (updates).
- Lithium battery
  Type: CR1220

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

5. LED indicators

The LED indicators provide a visual representation of the controller status:

- REL1–REL4 (green) – This LED turns on when the corresponding relay is activated, indicating that the NO contact is connected to the COM terminal.
- PWR (red) – Remains on during normal operation. In the event of a hardware error, it blinks together with the STS indicator.
- STS (yellow) – Flashes to indicate that the controller's main program is running.

- Indicates Ethernet network status. The indicator remains on when a network link is established and flashes during network activity.

6. Installation and configuration

The TCW242 must be installed by qualified personnel to ensure proper operation and safety. The installation process includes mounting the device, connecting the inputs and outputs, supplying power, and completing the configuration through a web browser.

The device is intended for indoor use only and must not be installed outdoors.

6.1. Mounting

The TCW242 must be installed in a clean, dry location on a non-flammable surface. If the installation environment has high ambient temperatures, ensure adequate ventilation around the device to maintain optimal operation.

To mount the device to a wall, use two 8 × 60 mm plastic dowels (e.g. Würth GmbH 0912 802 002) and two 6 × 70 mm screws (e.g. Würth GmbH 0157 06 70). Refer to Appendix D, Figure 1 for detailed mechanical specifications.

Ensure proper spacing from adjacent equipment to allow sufficient ventilation and electrical isolation. Maintain a minimum clearance of 50 mm on all sides, as shown in Appendix D, Figure 2.

The device can also be mounted on a standard 35 mm DIN rail.

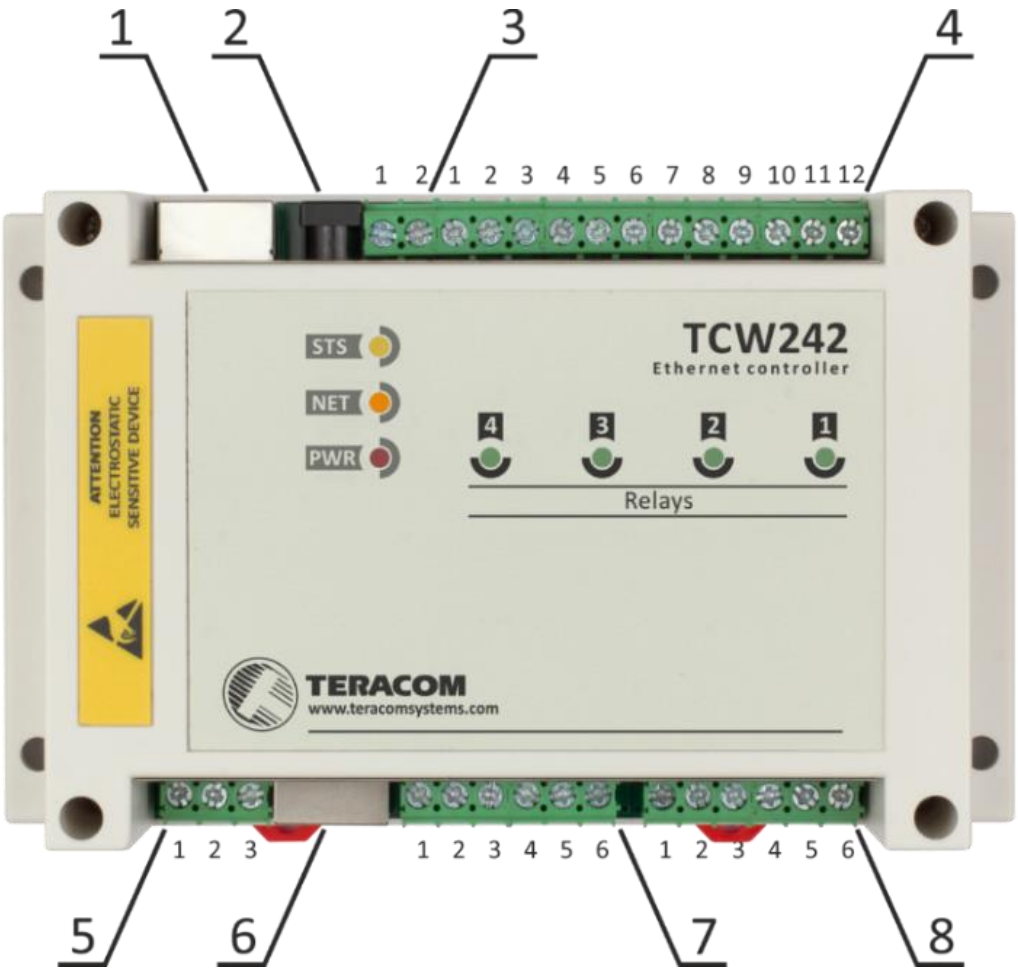
6.2. Connection

Attention! Always disconnect the power supply from the device before starting any wiring to avoid electrical hazards or equipment damage.

Follow these steps for correct wiring:

- Ensure the power supply to the device is turned off.
- Make the necessary wiring connections to the terminals.
- Turn on the power supply.

It is strongly recommended to test and configure the TCW242 without connecting it to any controlled devices. Ensure that all wires are securely attached to the terminals and that the terminal screws are properly tightened. Improper wiring or configuration may cause permanent damage to the device, the connected equipment, or both.



Connector 1	Ethernet - RJ45
Connector 2	Power - 2.1x5.5mm connector, central positive
Connector 3	**Pin1** – Power positive **Pin2** – Power negative
Connector 4	**Pin1** – NC 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** – 485+ **Pin2** – GND **Pin3** – 485-
Connector 6	**Pin1** – not connected (most left) **Pin2** – not connected **Pin3** – not connected **Pin4** – 485- **Pin5** – 485+ **Pin6** – not connected **Pin7** – +VDD **Pin8** – GND
Connector 7	**Pin1** – Digital In 1 **Pin2** – GND **Pin3** – Digital In 2 **Pin4** – Digital In 3 **Pin5** – GND **Pin6** – Digital In 4
Connector 8	**Pin1**– Analog In 1 **Pin2** – GND **Pin3** – Analog In 2 **Pin4** – Analog In 3 **Pin5** – GND **Pin6** – Analog In 4

6.2.1 Power supply connection

The TCW242 must be powered by the adapter SYS1308(N)-2412-W2E or an equivalent power supply suitable for overvoltage category II and certified for safety compliance. The power supply should be able to withstand short circuits and overloads on the secondary circuit. Ensure that the device can be easily disconnected from the power supply during operation.

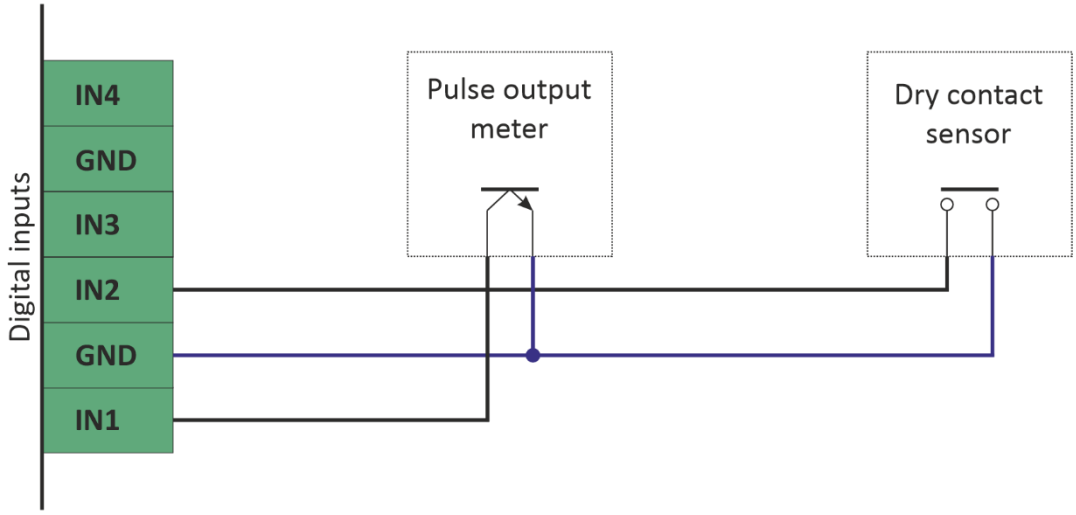
6.2.2 Digital inputs connection

The digital inputs are not isolated from the power supply ground.

The digital inputs can operate in two modes: Dry Contact and Counter.

In Dry Contact mode, the inputs can monitor devices with dry contact outputs, such as door contacts, push buttons, relay contacts, PIR detectors, and similar devices. To connect a dry contact device, connect one terminal of the contact to the selected input and the other terminal to the corresponding GND terminal.

In Counter mode, the inputs can count pulses generated by devices with pulse output interfaces, such as electricity, water, gas, and heat meters. Pulse counting is performed by monitoring transitions between the open and closed states of the input.

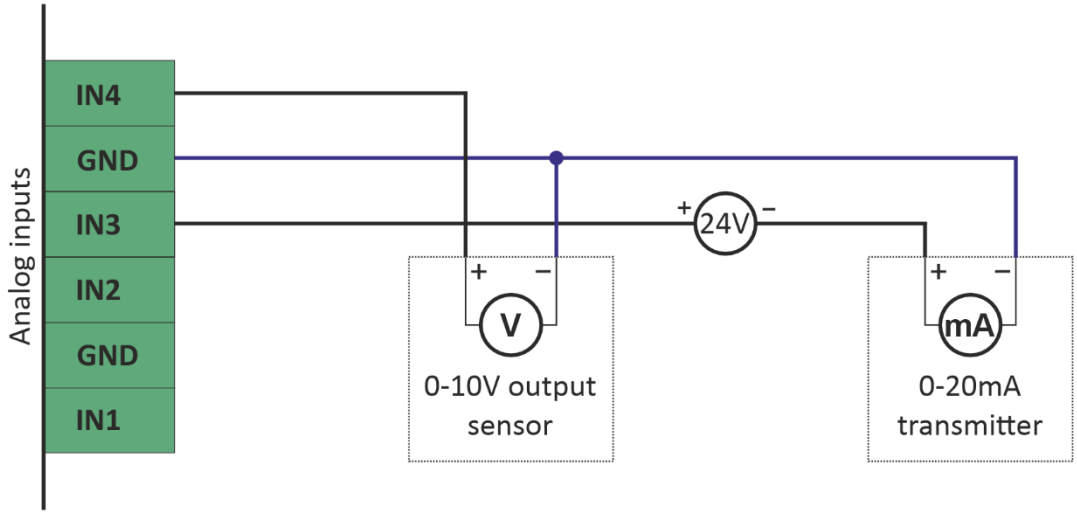


The maximum cable length between the device and the connected sensor or meter should not exceed 30 m

6.2.3 Analog inputs connection

The analog inputs are not isolated from the power supply ground.

The analog inputs can operate in either voltage mode (0–10 V) or current loop mode (0–20 mA). They can be connected directly to sensors, transmitters, and other devices that provide compatible analog outputs.



In voltage mode, connect the positive output of the sensor or transmitter to the selected analog input and the negative output to the corresponding GND terminal.

In current loop mode, connect the current loop transmitter to the selected analog input and provide an external power supply for the current loop. Connect the transmitter output to the analog input and the power supply return to the corresponding GND terminal.

The maximum cable length between the device and the connected sensor or transmitter should not exceed 30 m.

6.2.4 RS-485 connection

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

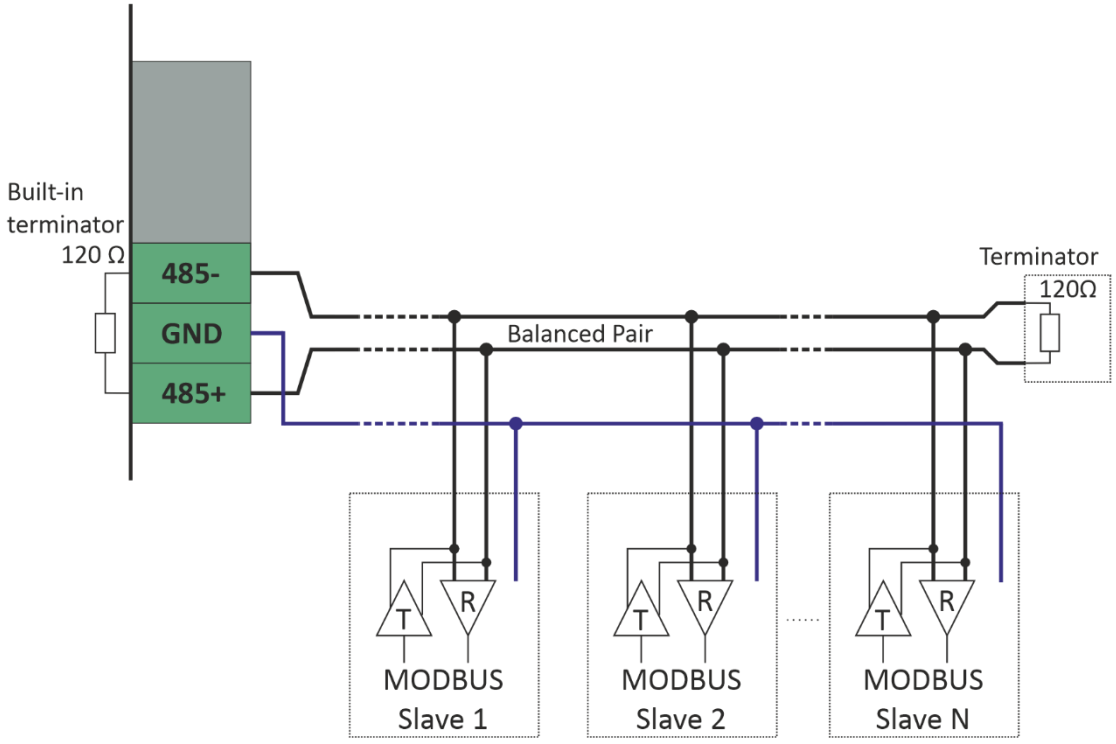
The TCW242 supports communication with MODBUS RTU slave devices over the RS-485 interface.

The RS-485 interface uses a balanced differential pair, providing reliable communication in electrically noisy industrial environments and over long cable distances.

The TCW242 includes a built-in 120 Ω termination resistor. For proper operation, the device should be installed at one end of the RS-485 line, while a second 120 Ω termination resistor should be installed at the opposite end of the bus.

For installations with multiple devices, a linear (daisy-chain) topology must be used. Reliable operation is guaranteed only when all devices are connected in a daisy-chain configuration with proper termination at both ends of the RS-485 bus.

Shielded twisted-pair cable is recommended for RS-485 installations. To minimize ground-loop currents and improve noise immunity, the cable shield should be connected to protective earth at a single point, preferably near the RS-485 master device.



Keep the total cable length within 30 m; however, reliable operation has been observed over longer distances under suitable installation conditions.

6.2.5 Relay connection

The TCW242 provides four relay outputs with SPDT (Single Pole Double Throw) contacts. Each relay is equipped with Normally Open (NO), Normally Closed (NC), and Common (COM) terminals.

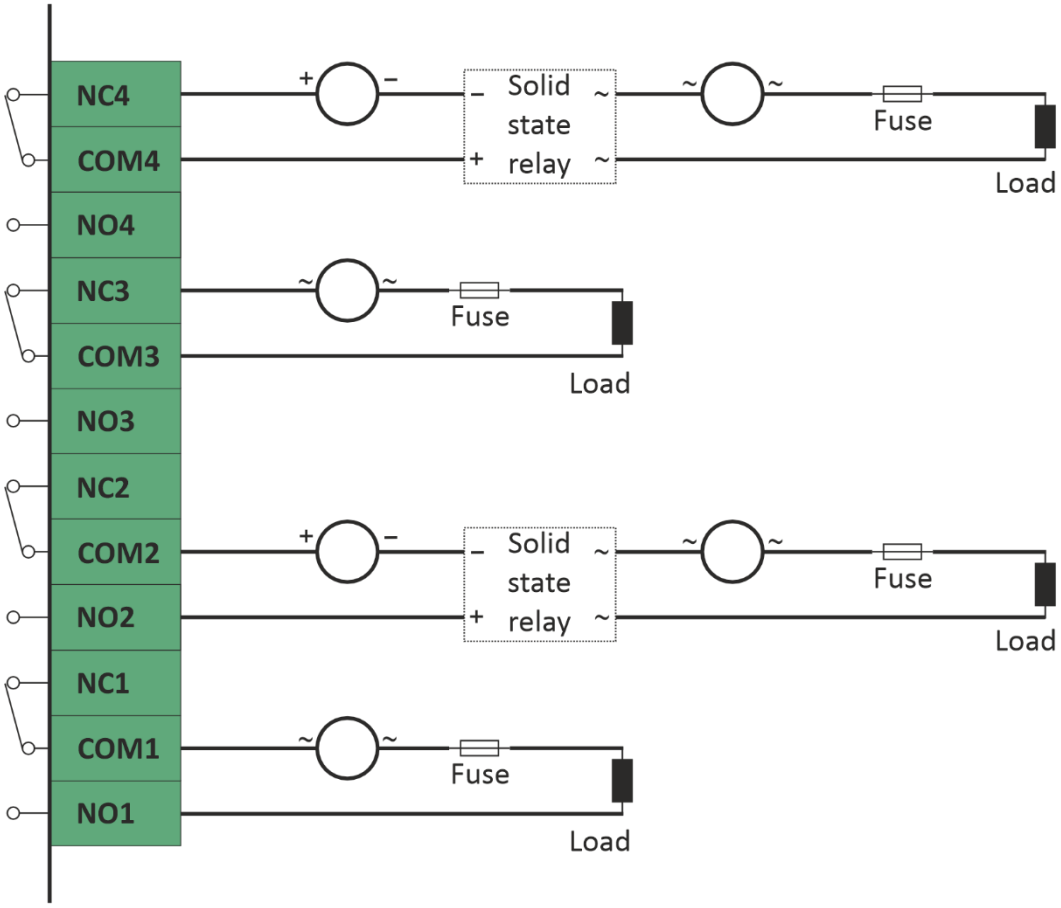
When the relay is de-energized, the COM terminal is connected to the NC terminal. When the relay is energized, the COM terminal is connected to the NO terminal.

The relay outputs can be used for switching AC/DC loads directly or for controlling external devices such as contactors and solid-state relays (SSRs).

The relay contacts are not protected by internal fuses or circuit breakers. To ensure reliable operation and protect the connected wiring and equipment, external overcurrent protection devices must be installed in accordance with the load requirements and local electrical regulations.

When switching inductive loads, appropriate suppression devices should be used to reduce contact wear and improve relay life.

The relay output wiring examples shown above illustrate typical connections for AC and DC loads, including control of external solid-state relays.



6.2.6 Ethernet connectivity

The TCW242 is equipped with a 10/100Base-T Ethernet interface for configuration, monitoring, and control over IP networks.

The Ethernet interface supports Auto-MDIX, allowing the use of standard Ethernet cables without regard to cable type.

The device can be connected directly to a computer for initial configuration or integrated into a local area network through an Ethernet switch, router, or other network infrastructure equipment.

Network parameters, including IP address, subnet mask, gateway address, and DNS servers, are configured through the web interface.

6.3. Communication setup

The TCW242 is shipped with the following factory network settings:

- IP address: 192.168.1.2
- Subnet mask: 255.255.255.0
- Default gateway: 192.168.1.1

To access the device for the first time, connect a computer to the same Ethernet network and configure the computer with an IP address from the same subnet, for example 192.168.1.3.

Open a web browser and enter the device IP address:

192.168.1.2

A login dialog will be displayed. Enter the default username and password:

Username: admin

Password: admin

After successful authentication, the web-based configuration interface will be displayed.

If the device is connected to an existing network, the IP settings can be modified through the web interface to match the network requirements.

All Teracom TCW controllers connected to the local network can be automatically discovered using the free TCW Discoverer utility. The software is available for both Windows and macOS and can be downloaded from the Teracom website.

7. Set up concept

The TCW242 is configured primarily through its web interface. The web interface provides access to all device settings and is the recommended method for initial setup and configuration.

The device configuration follows a logical sequence, where each step builds upon the previous one. Understanding this sequence simplifies commissioning and helps avoid configuration errors.

The recommended setup procedure is:

1. Network configuration

Configure the network parameters required for communication with the device.

2. Input/Output configuration

Configure the primary data sources, including digital inputs, analog inputs, relay outputs, and MODBUS RTU devices.

3. Channel configuration

Create channels based on the configured inputs and MODBUS registers. A channel may represent a single parameter or a calculated value derived from up to two parameters and constants using mathematical operations.

4. Alarm configuration

Configure alarms based on channel values. Each alarm can contain up to two independent conditions and can be associated with a specific channel if required.

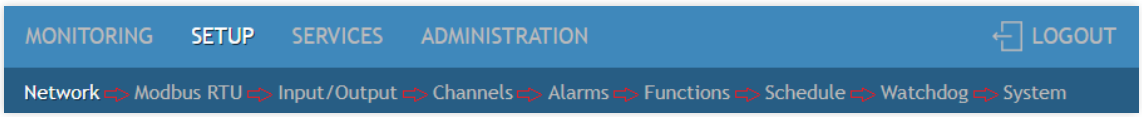
5. Functions, schedules, and watchdog

Configure logical functions, time schedules, and watchdog actions used for automation and control.

6. Services configuration

Enable and configure the required services, such as Data Logger, SNMP, HTTP POST, MQTT, Dynamic DNS, and other communication services.

In summary, the recommended configuration sequence is:



8. Web interface

The web interface provides access to device configuration, monitoring, and administration functions. All pages are encoded in UTF-8 and support multiple simultaneous user sessions.

The TCW242 web interface is accessible through both HTTP and HTTPS connections

8.1. Monitoring

The Monitoring section provides real-time information about channel values, alarms, relay outputs, and MODBUS RTU relays.

The displayed information is refreshed automatically according to the configured page refresh interval. The refresh interval can be configured in Setup → System → Refresh of Channels and Alarms Pages. A value of 0 disables automatic refresh. The default value is 1 second.

The Monitoring section consists of the following pages:

- Channels, relays and MODBUS RTU relays – Displays the current values and alarm status of all configured channels together with the status of relay outputs and MODBUS RTU relays.
- Alarms – Displays the current status of all configured alarms.
- Graphs – Provides graphical visualization of data stored in the internal data logger.

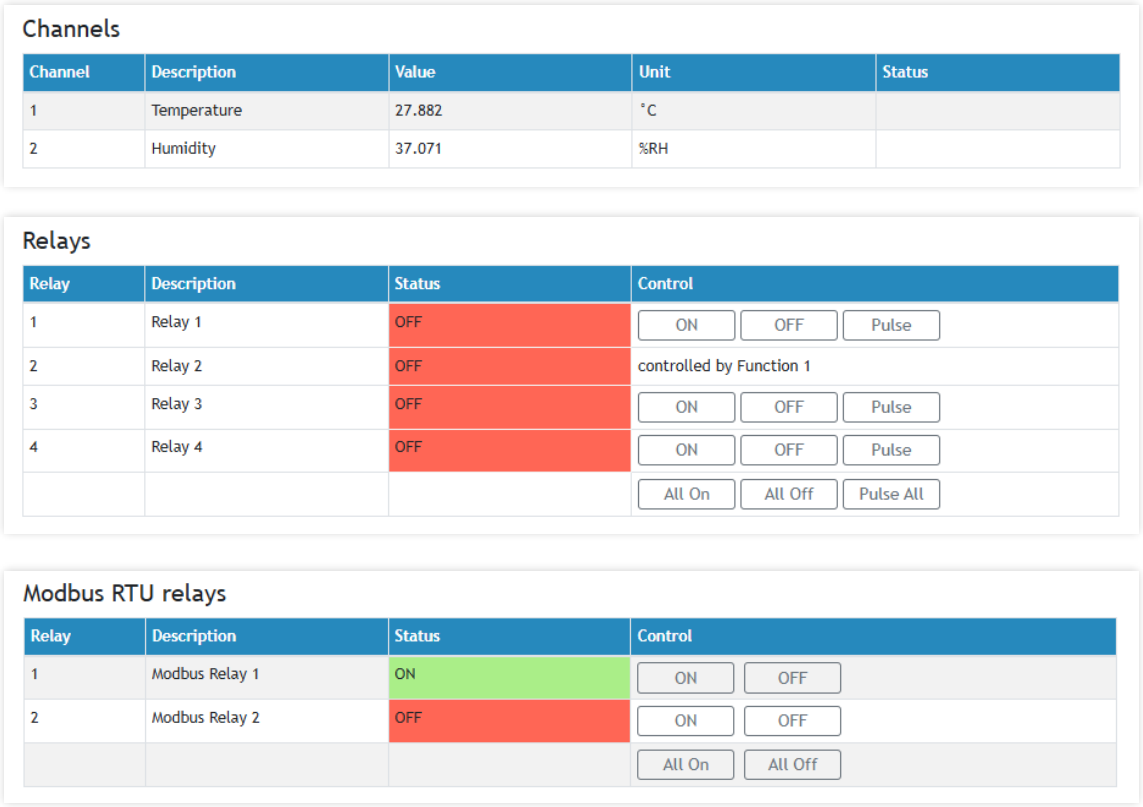
8.1.1 Channels, relays and Modbus RTU relays

This page displays the current values and alarm status of all configured channels together with the status of relay outputs and MODBUS RTU relays.

The channel table shows the channel description, current value, engineering unit, and alarm status. Depending on the configured alarm conditions, a channel may be displayed in different alarm states such as Normal, Warning, Minor, Major, or Critical.

The relay tables display the current state of all local and MODBUS RTU relay outputs. Relay outputs can be controlled remotely through the web interface, HTTP API, MQTT, or SNMP. Relay operation can also be controlled automatically by alarms, schedules, functions, and watchdog events.

When a relay is controlled automatically, the controlling source is displayed in the Control column instead of manual control buttons.



Channels

Channel	Description	Value	Unit	Status
1	Temperature	27.882	°C	
2	Humidity	37.071	%RH	

Relays

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

Modbus RTU relays

Relay	Description	Status	Control
1	Modbus Relay 1	ON	ON OFF
2	Modbus Relay 2	OFF	ON OFF
			All On All Off

The information on this page is refreshed automatically according to the configured page refresh interval.

For manual control through the web interface, each local relay provides ON, OFF, and Pulse buttons. Additional All ON, All OFF, and Pulse All buttons allow simultaneous control of all local relay outputs.

The pulse duration is configured individually for each relay in Setup → Input/Output → Relay Outputs.

MODBUS RTU relays provide ON and OFF control buttons, as well as All ON and All OFF buttons for group control. Pulse operation is not available for MODBUS RTU relays.

When a relay is controlled automatically by an alarm, Any Alarm, function, schedule, or watchdog, the controlling source is displayed in the Control column instead of manual control buttons. The corresponding control mode is configured in Setup → Input/Output → Relay Outputs or Setup → Input/Output → MODBUS Relay Outputs.

8.1.2 Alarms

This page displays the current status of all configured alarms.

Alarms	Description	Status
1	Alarm 1	Normal
2	Alarm 2	Warning
3	Alarm 3	Minor
4	Alarm 4	Major
5	Alarm 4	Critical

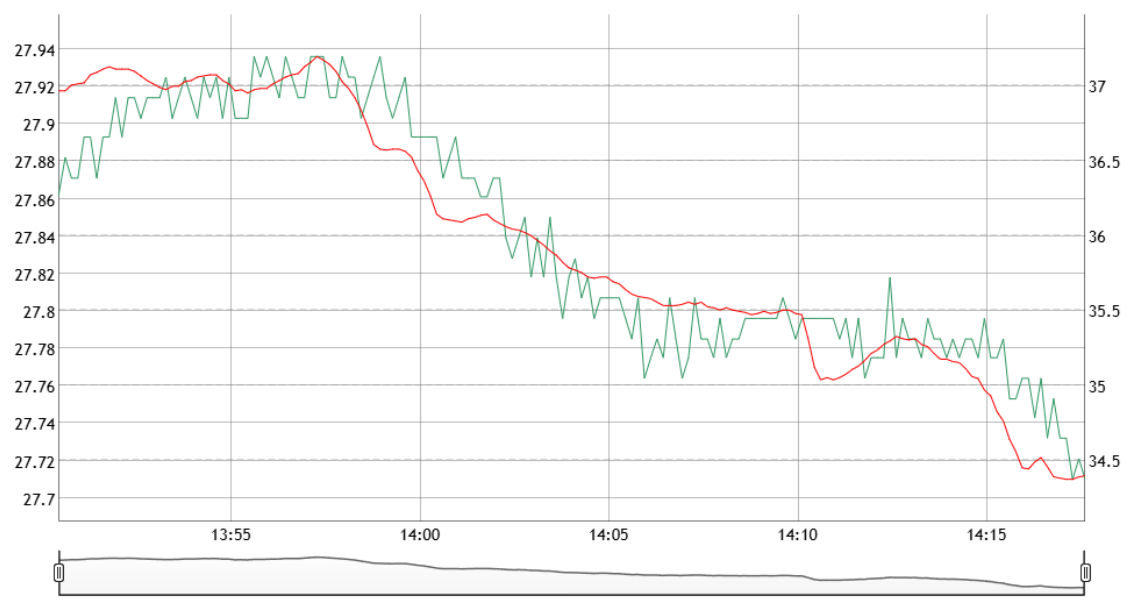
The alarm table shows the alarm description and current alarm state. Depending on the configured conditions, alarms may be displayed in states such as Normal, Warning, Minor, Major, or Critical.

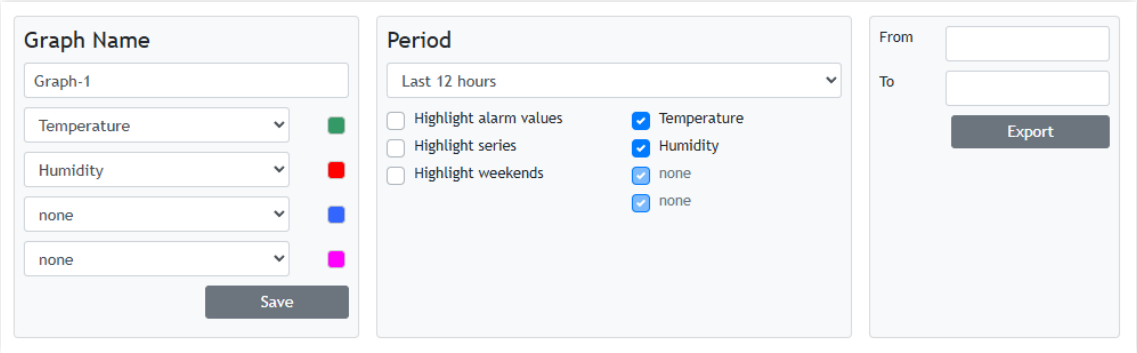
The information on this page is refreshed automatically according to the configured page refresh interval.

8.1.3 Graphs

The graph pages provide graphical visualization of data stored in the internal data logger.

Up to six independent graphs can be configured. Each graph can display up to four items simultaneously. Items may represent channels, alarms, relay outputs, or MODBUS RTU relays.





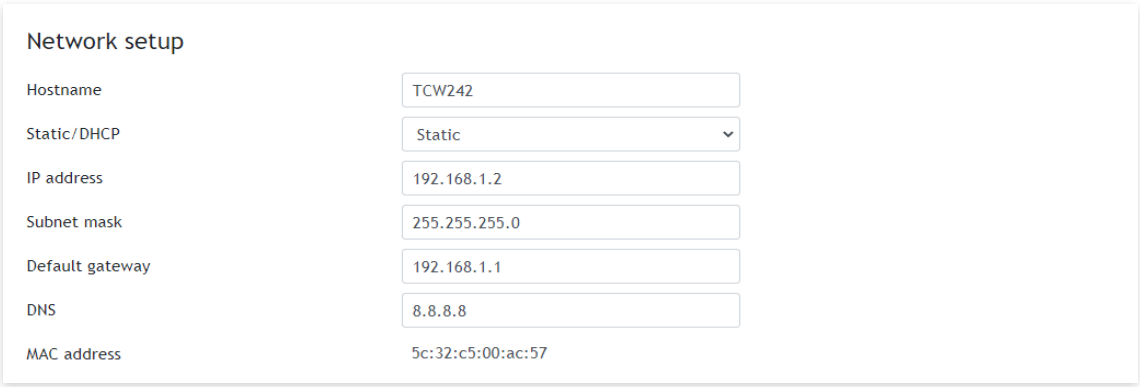
Each graph item can be assigned to either the left or right Y-axis, allowing parameters with different units to be displayed on the same graph.

Graph data is loaded from the internal data logger and is not updated in real time. To display newly recorded values, reload the page.

8.2. Setup

8.2.1 Network

This page is used to configure the network parameters of the device.



The TCW242 supports both static IP addressing and automatic network configuration through DHCP.

It is recommended to change the factory default IP address before connecting multiple TCW242 controllers to the same network to avoid IP address conflicts.

The Hostname field defines the device name used for identification on the network. The hostname can contain up to 15 characters and is displayed by the TCW Discoverer utility.

The DNS server setting is used when the device accesses network resources by domain name, such as NTP servers, SMTP servers, MQTT brokers, Dynamic DNS providers, and HTTP/HTTPS servers.

The MAC address field displays the unique hardware address of the Ethernet interface and is read-only.

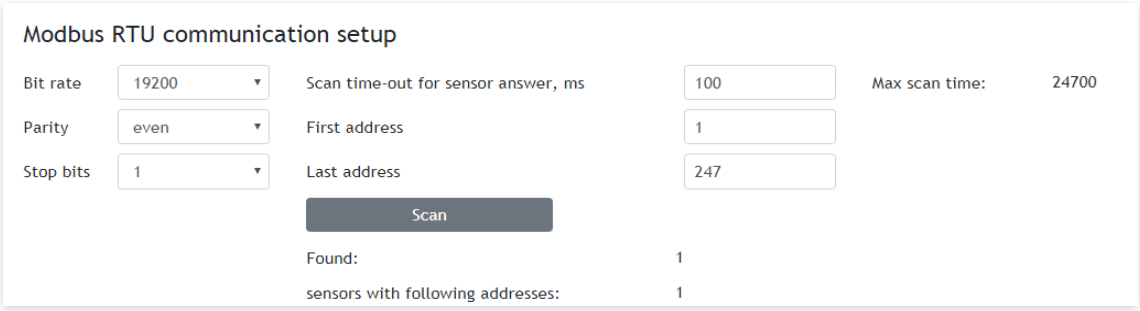
8.2.2 Modbus RTU

8.2.2.1 Modbus RTU communication setup

This section is used to configure the communication parameters of the RS-485 interface.

The Bit rate, Parity, and Stop bits settings must match the communication parameters of all MODBUS RTU devices connected to the bus. By default, the TCW242 uses 19200 bps, Even parity, and 1 stop bit.

The Scan function can be used to discover MODBUS RTU devices connected to the RS-485 bus. The scan process checks all addresses within the specified address range and displays the addresses of the detected devices.

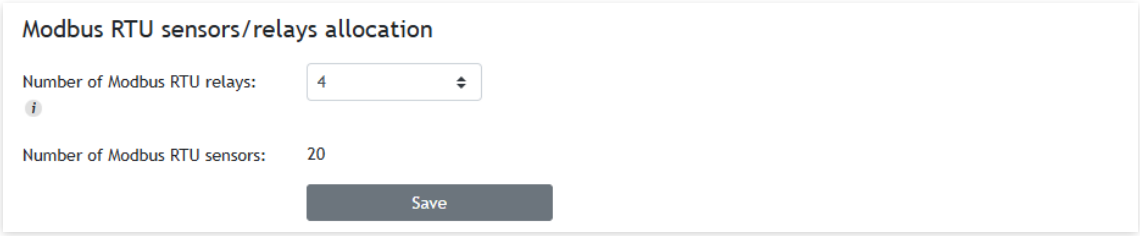


The Scan time-out for sensor answer parameter defines how long the controller waits for a response from a MODBUS RTU device before continuing with the next address. Increasing this value may improve detection of slower devices but will increase the overall scan duration.

The Max scan time field displays the estimated maximum duration of the scan based on the selected address range and timeout value.

8.2.2.2 Modbus RTU sensors/relays allocation

The TCW242 supports up to 24 MODBUS RTU resources. These resources can be allocated either to MODBUS RTU sensors or to MODBUS RTU relays.



The Number of MODBUS RTU relays parameter defines how many resources are reserved for relay outputs. The remaining resources are automatically available for MODBUS RTU sensors.

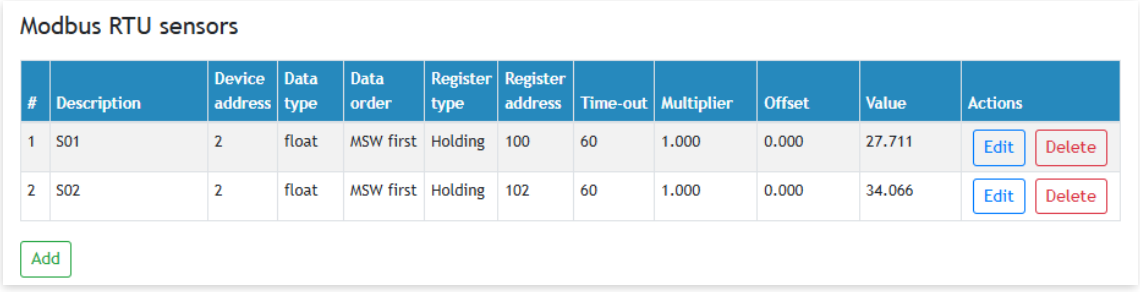
The displayed Number of MODBUS RTU sensors value indicates the number of resources currently available for sensor allocation.

The total number of allocated MODBUS RTU sensors and relays cannot exceed 24.

Press Save to apply the changes.

8.2.2.3 Modbus RTU sensors

This section is used to add, edit, and delete MODBUS RTU sensors.



All configured MODBUS RTU sensors are treated as primary parameters and can be used for channel creation.

The scan function described in section 8.2.2.1 can be used to verify communication settings and detect device addresses before adding sensors.

The maximum number of sensors that can be configured depends on the allocation defined in section 8.2.2.2 MODBUS RTU sensors/relays allocation.

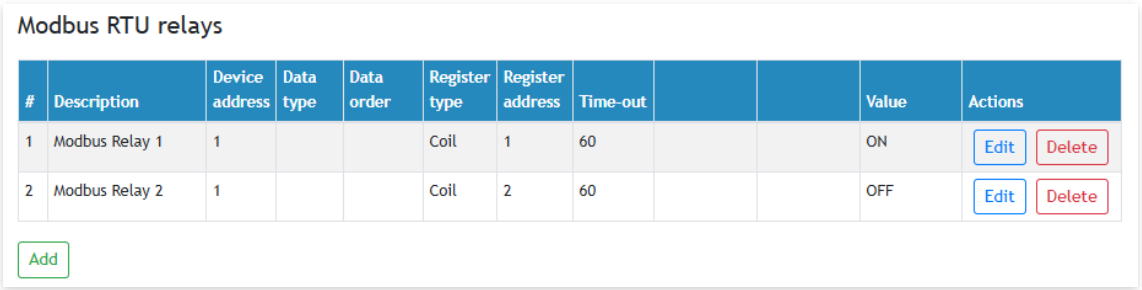
The Multiplier and Offset parameters are applied according to the following formula:

Value = (Raw Value × Multiplier) + Offset

The default values are Multiplier = 1.000 and Offset = 0.000, so the calculated value is equal to the raw value.

8.2.2.4 Modbus RTU relays

This section is used to add, edit, and delete MODBUS RTU relays.



MODBUS RTU relays extend the output capabilities of the TCW242 by allowing relay control through external MODBUS RTU devices. The TCW242 supports both Teracom MODBUS RTU relay modules and compatible third-party devices.

The maximum number of MODBUS RTU relays that can be configured depends on the allocation defined in section 8.2.2.2 MODBUS RTU sensors/relays allocation.

The scan function described in section 8.2.2.1 can be used to verify communication settings and detect device addresses before adding relays.

8.2.2.5 Polling

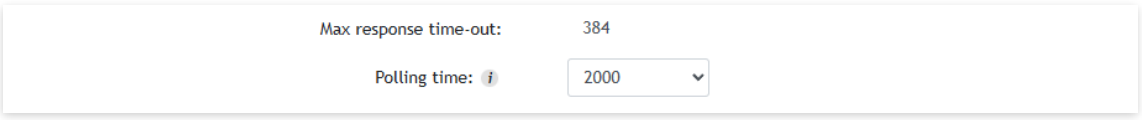
During normal operation, all configured MODBUS RTU sensors and relays are polled sequentially.

For each device, the controller waits for a response during the configured response timeout period. If no response is received within this period, polling continues with the next device. After three consecutive communication failures, the device is considered unavailable; however, the controller continues attempting communication during subsequent polling cycles.

The response timeout should be configured carefully. It must be long enough to accommodate the response time of the connected device, while unnecessarily large values increase the overall polling cycle duration.

The Max response time-out value is calculated as the sum of the response timeout values configured for all MODBUS RTU sensors and relays.

The Polling time parameter defines how often a complete polling cycle is started. Available values are 1, 2, 3, 4, 5, 10, 15, 30, 60, 120, and 180 seconds. The default value is 1 second.

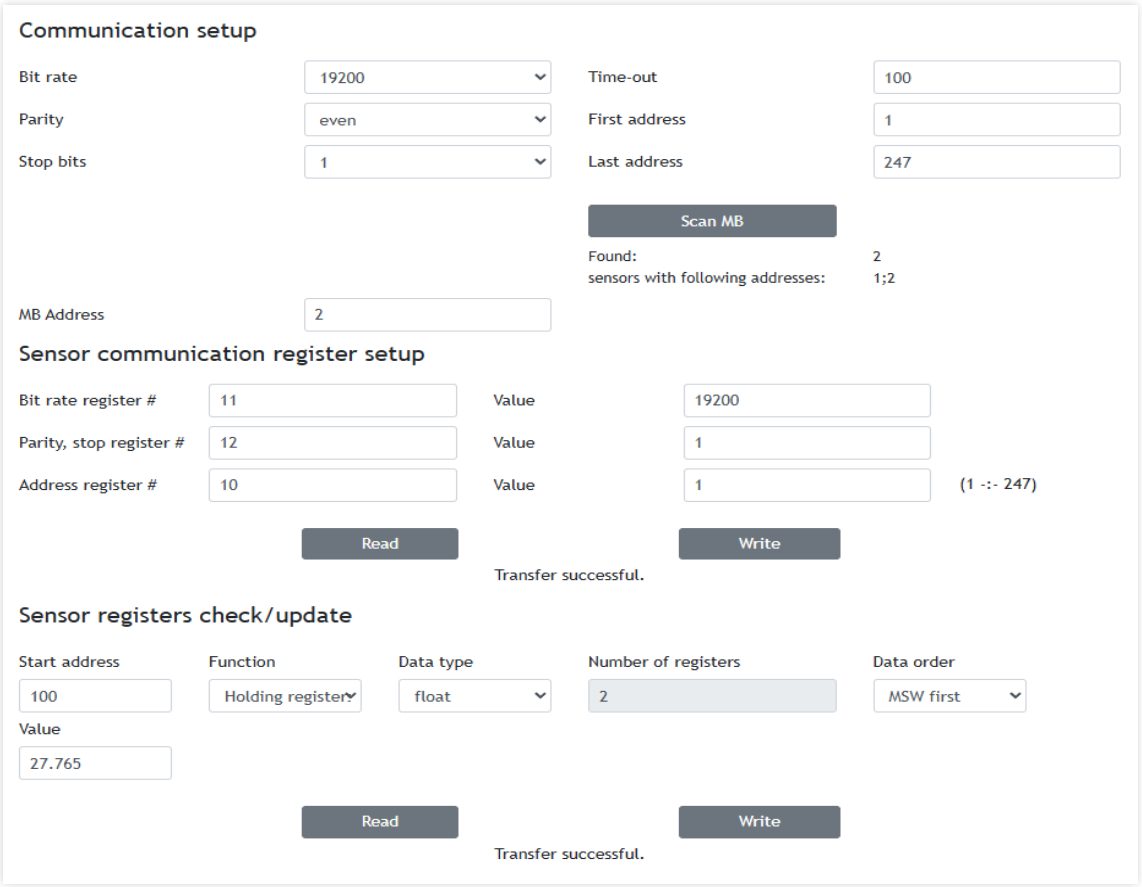


For proper operation, the configured Polling time should be greater than or equal to the displayed Max response time-out value.

8.2.2.6 Sensor setup tool

The Sensor Setup Tool provides direct access to MODBUS RTU device registers and can be used for commissioning, diagnostics, and configuration of MODBUS RTU sensors and relay modules.

The tool allows reading and writing of MODBUS registers using different register types, data formats, and byte orders. This makes it possible to verify device communication, check register contents, and configure devices from various manufacturers without the need for additional software.



For supported devices, the tool can also be used to change communication parameters such as device address, bit rate, parity, and stop bits.

The Sensor communication register setup section provides quick access to registers commonly used for configuring device communication parameters.

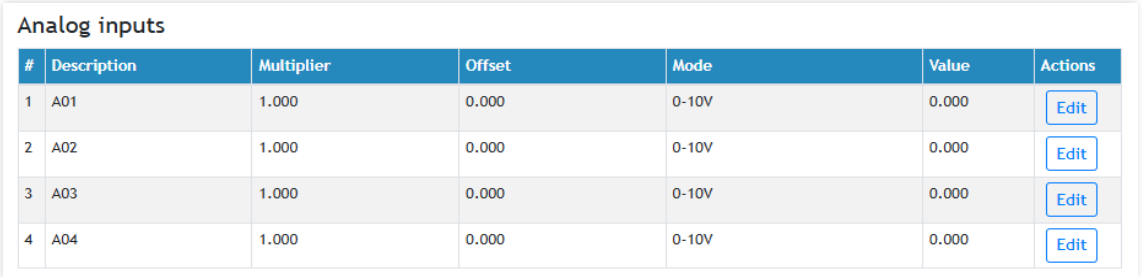
The Sensor registers check/update section allows reading and writing arbitrary MODBUS registers. This can be useful for verifying register maps, troubleshooting communication issues, and testing device operation during installation and maintenance.

8.2.3 Input/Output

This page enables the configuration of relays, Modbus RTU relays, as well as analog and digital inputs.

8.2.3.1 Analog inputs

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



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

For every analog input, the "Unit" "Multiplier" and "Offset" fields are available to convert the raw voltage or current into meaningful engineering units.

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

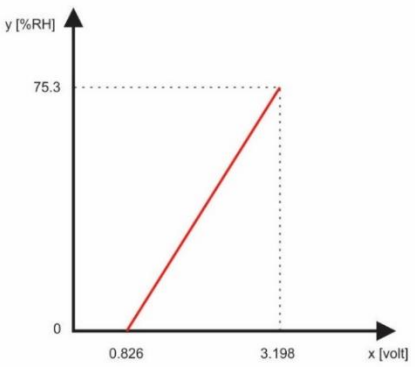
Example:

For the humidity sensor HIH-4000-003, the following data is available from the datasheet:

VOUT = 0.826 V at 0 %RH

VOUT = 3.198 V at 75.3 %RH

The sensor outputs raw voltage values, but the goal is to convert these into the corresponding relative humidity (RH) values. To achieve this, a Multiplier and an Offset are used. These parameters allow calculation of the relative humidity for any voltage within the calibrated range.



Calculation of the multiplier (MU)

The multiplier (MU) is determined by the ratio of the change in relative humidity (ΔRH) to the change in voltage (ΔV). Geometrically, this corresponds to the slope of a straight line.

The multiplier is calculated as follows:

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

Calculation of the offset (OF)

The offset (OF) is calculated using the multiplier and one of the known points. By substituting the scaled value (SV) and the corresponding raw value (RV) into the equation:

SV = RV × MU + OF

the offset can be calculated as:

OF = SV − (RV × MU)

Using the point where SV = 0 and RV = 0.826, the offset is:

OF = 0 − (0.826 × 31.745) = 0 − 26.22 = −26.22

The offset can also be verified using the second known point, where SV = 75.3 and RV = 3.198:

OF = 75.3 − (3.198 × 31.745) = 75.3 − 101.52 = −26.22

Final formula

The final conversion formula for this sensor is:

SV = RV × 31.745 − 26.22

Verification

To verify the correctness of the formula, we can check the case where VOUT = 0.826 V (0 %RH):

SV = 0.826 × 31.745 − 26.22 = 26.22 − 26.22 = 0 %RH

This verifies that the formula correctly converts the sensor voltage into the corresponding relative humidity value.

The scaled value is calculated using the following formula:

SV = RV * MU + OF

Where:

SV – scaled (displayed) value;

RV – raw voltage from the source;

MU – multiplier;

OF – offset.

8.2.3.2 Digital inputs

The TCW242 features four digital inputs, all of which are non-isolated from the power supply ground. Each digital input can operate in either OPEN/CLOSE mode or COUNTER mode. In COUNTER mode, counting can be configured to occur on rising edges, falling edges, or both.

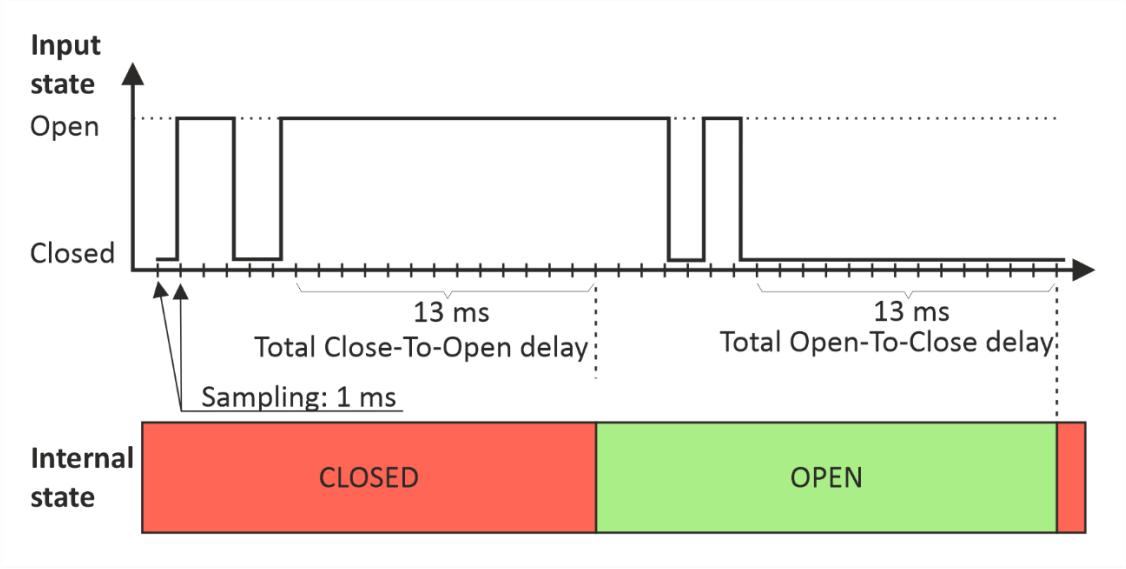
Additionally, the COUNTER mode allows for the setup of an initial counter value.

Digital inputs

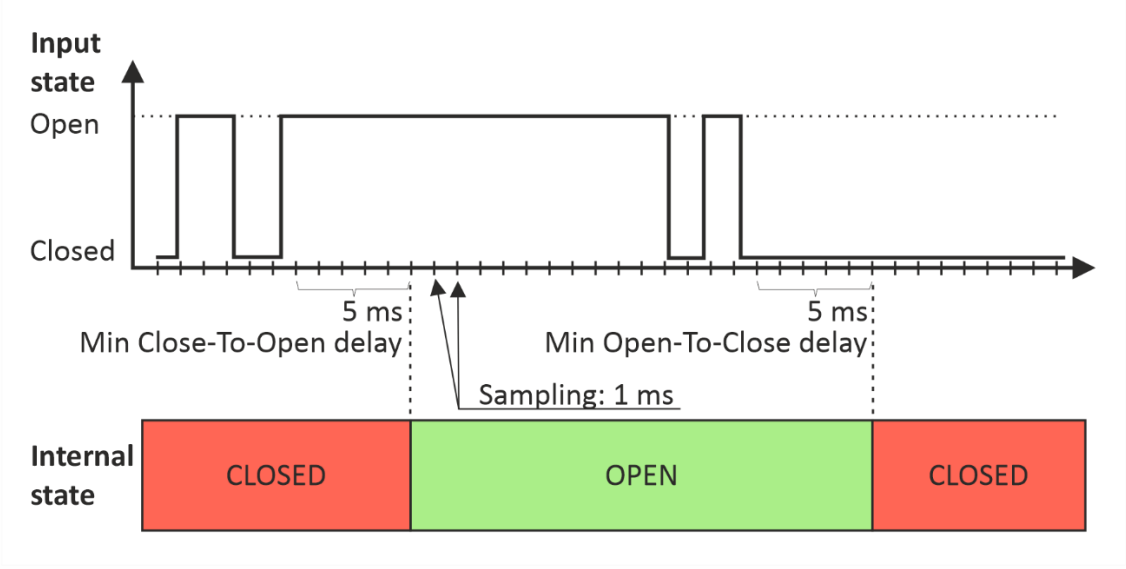
#	Description	Closed state	Open state	C/O delay	O/C delay	Mode	Value	Actions
1	D1	CLOSED	OPEN	5	5	Discrete(Open/Closed)	OPEN	Edit
2	D2	CLOSED	OPEN	5	5	Counter(Rising edge)	15251	Edit
3	D3	CLOSED	OPEN	5	5	Counter(Falling edge)	0	Edit
4	D4	CLOSED	OPEN	5	5	Counter(Both edges)	0	Edit

There are two configurable delays: Open-to-Close and Close-to-Open, each ranging from 5 to 60,000 milliseconds. These delays can be utilized for additional digital filtering and are applicable in both modes.

In the picture below, the Open-to-Close and Close-to-Open delays are both set to 13 milliseconds.

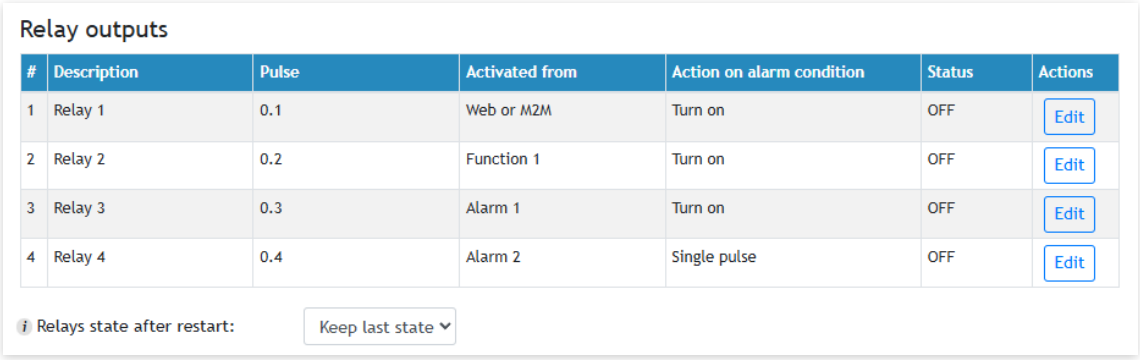


In the picture below, the Open-to-Close and Close-to-Open delays are both set to 5 milliseconds, which is the minimum delay.



8.2.3.3 Relay outputs

The TCW242 features 4 relay outputs that can be controlled through the web interface, HTTP API, SNMP, Functions, Schedules, Watchdogs, or locally using alarm conditions. The web control for each relay provides "On", "Off", and "Pulse" buttons, along with "All On", "All Off", and "Pulse All" for collective control. The pulse duration can be set individually for each relay.

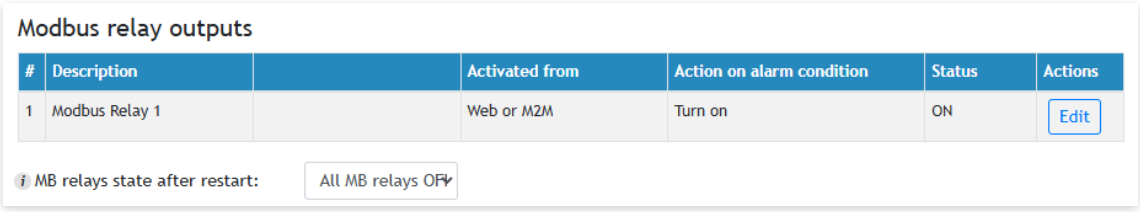


The "Relay state after restart" can be configured to “On”, “Off”, or “Last state”. Selecting “Off” may be a good safety option.

8.2.3.4 Modbus RTU relay outputs

The TCW242 supports up to 24 Modbus RTU relay outputs. They can be controlled in the same way as built-in relay outputs—via the web interface, HTTP API, SNMP, Functions, Schedules, Watchdogs, or locally using alarm conditions.

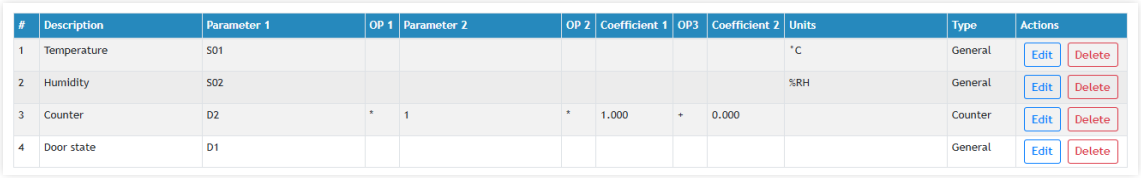
The web control for each relay provides "On" and "Off" buttons, along with "All On" and "All Off" for collective control. Pulse activation of Modbus RTU relays is not supported.



The "Relay state after restart" can be configured to "On", "Off", or "Last state". Selecting "Off" may be a good safety option.

8.2.4 Channels

This section allows you to add, delete, or edit channels. All channels can be monitored on the monitoring pages, and their values can be recorded periodically by the data logger. Up to 24 channels can be configured.



There are two types of channels:

- Discrete: Formed by a single digital input in OPEN/CLOSE mode.
- General: Formed by up to two primary parameters and constants. It's important to note that, the sequence of operations is OP1, followed by OP2, and finally OP3.

8.2.5 Alarms

This section allows you to add, delete, or edit alarms, with a maximum of 24 alarms available for configuration.

There are four types of alarms: Warning, Minor, Major, and Critical. Alarms can only be formed using channels. While alarms are independent, they can be assigned to any channel, with no restrictions on which channel can receive the alarm.

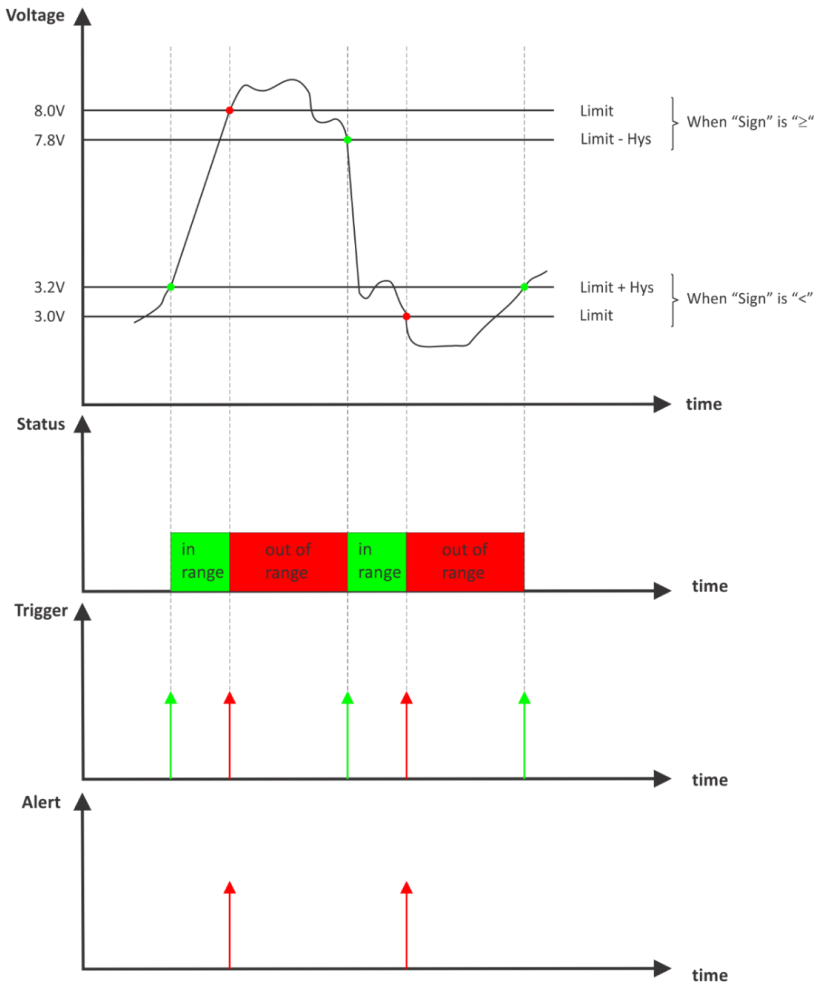
Each alarm can utilize up to two conditions, which can be logically combined using the operators AND and OR. It is not mandatory for both conditions to use the same channel; conditions from different channels can be combined within a single alarm.

#	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	Alarm 1	Temperature	≥	50.000					Warning	Temperature	Trap C1	None	None	Edit Delete
2	Alarm 2	Humidity	<	50.000					Warning	Humidity	Send mail	None	None	Edit Delete
3	Alarm 3	Counter	<	60.000					Minor	Counter	MQTT Publish	None	None	Edit Delete
4	Alarm 4	Door state	≥	1.000					Major	Door state	None	None	None	Edit Delete

The “Limit” indicates the boundary of the working range for the monitored channel. A trigger condition occurs when the value exceeds the limit (“Sign” is “≥”) or falls below the limit (“Sign” is “<”). In both scenarios, the monitored parameter goes out of range.

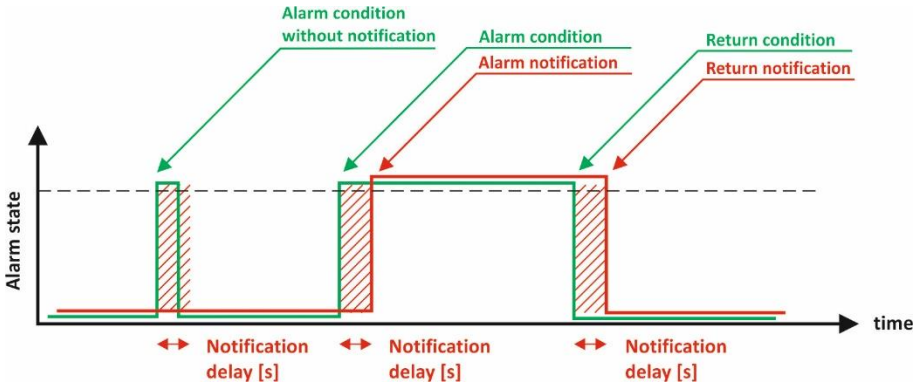
Returning to the acceptable range for the observed channel is considered when the value exceeds the Limit + Hysteresis (“Sign” is “<”) or falls below the Limit – Hysteresis (“Sign” is “≥”). Hysteresis is employed to prevent excessive triggering when the value fluctuates around the trigger point.

It is strongly recommended not to set Hysteresis to 0.0.



For each alarm, the “Return Notification” option is available. When this option is selected, a notification will also be sent when the parameter returns to the acceptable range.

Additionally, each alarm has a “Notification Delay” parameter, which serves as a useful filter for brief alarm conditions.



8.2.6 Functions

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

#	Condition 1	Func	Condition 2	Func	Condition 3	Func	Condition 4	Action		Actions
								Action 1	Action 2	
1	Alarm 1	OR	Schedule 1					HTTP Post	None	Edit Delete
2	Alarm 1	AND	Alarm 2					SNMP Trap	None	Edit Delete

Create

8.2.7 Schedule

The TCW242 features 4 schedules, each with up to 4 tasks. The schedules allow you to create tasks based on calendar dates and can be combined with alarms for more advanced control. Two relays can control a single device, with one following the alarm and the other following the schedule.

Schedule setup

Schedule 1 Schedule 1

Mode	Date	Mon	Tue	Wed	Thu	Fri	Sat	Sun	ON	OFF
Weekly	01.01.2016	☑	☑	☑	☑	☑	☐	☐	00:00:00	00:10:00.0
Weekly	01.01.2016	☐	☐	☐	☐	☐	☑	☑	00:00:00	08:00:00.0
Disable	01.01.2016	☐	☐	☐	☐	☐	☐	☐	00:00:00	23:59:59.9
Once	01.01.2027	☐	☐	☐	☐	☐	☐	☐	07:00:00	08:00:00.0

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

A resolution of 0.1 seconds for "OFF time" allows for short pulse support.

Here are some examples:

- A single task for a time period:

Once	01.01.2027	☐	☐	☐	☐	☐	☐	☐	07:00:00	08:00:00.0

With the above setting, an event will occur on 1.01.2027 from 07:00 to 08:00.

- A weekly task for a time period:



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

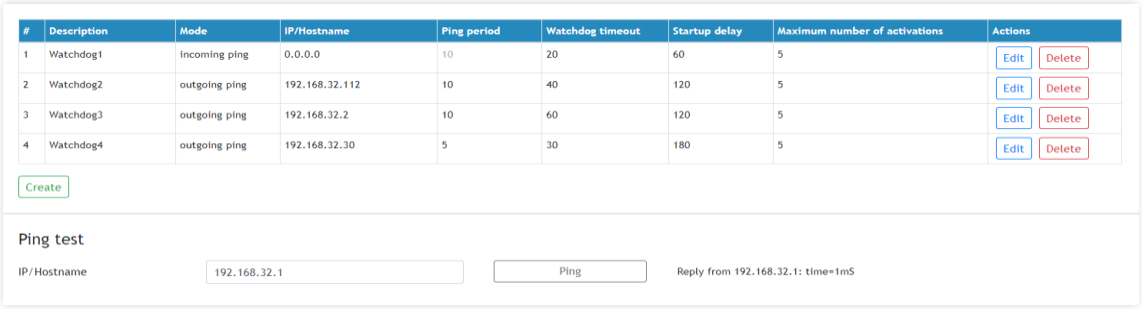
- A weekly task for a time period which includes midnight:



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

8.2.8 Watchdog

The Watchdog feature enables the remote supervision of network devices using the ICMP protocol. In the absence of an ICMP echo reply (outgoing ping) or an ICMP echo request (incoming ping), it activates a relay, which can trigger a reboot for certain devices. The TCW242 provides up to four independent Watchdogs.



Each Watchdog can be configured to operate in either ICMP echo reply (outgoing ping) or ICMP echo request (incoming ping) mode, or it can be disabled. Disabling the Watchdog will retain the previous settings.

The Ping reply timeout establishes the duration within which a ping reply must be received and is applicable solely for outgoing pings.

The Ping period sets the time gap between individual pings, which is exclusively applied for outgoing pings.

The Watchdog timeout dictates the duration within which an echo reply (outgoing ping) or echo request (incoming ping) must occur; beyond this period, the Watchdog is triggered.

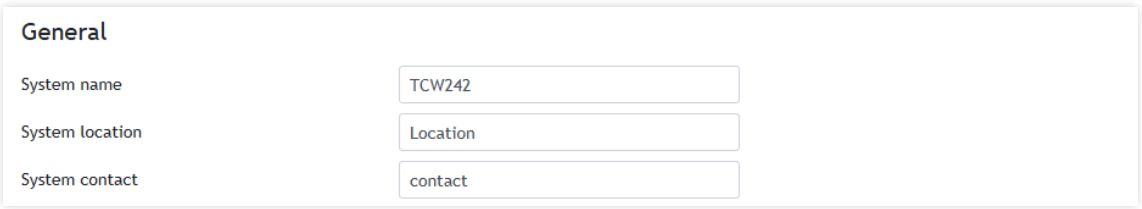
The startup delay specifies the waiting period after a reboot before the Watchdog reactivates.

The Maximum number of activations determines the consecutive instances of relay activations during sequential Watchdog triggers.

To activate a given relay, the corresponding Watchdog must be connected to this relay in the "Setup -> Input/Output -> Relay outputs" or "Setup -> Input/Output -> Modbus relay outputs" section.

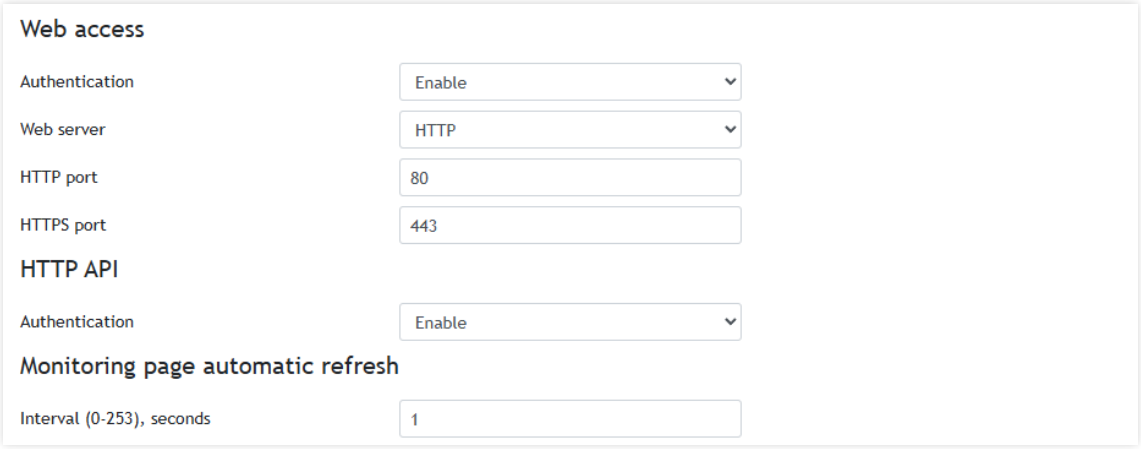
8.2.9 System

This page allows you to configure various general system parameters.



The system name, system location, and system contact information are used for device identification and are presented in SNMPv3 and XML/JSON status files.

By default, web access authentication is enabled with the credentials admin/admin. The web server typically operates over the HTTP protocol through port 80, which is customizable. This flexibility is particularly useful for routers that do not support separate outside/inside ports for port forwarding.

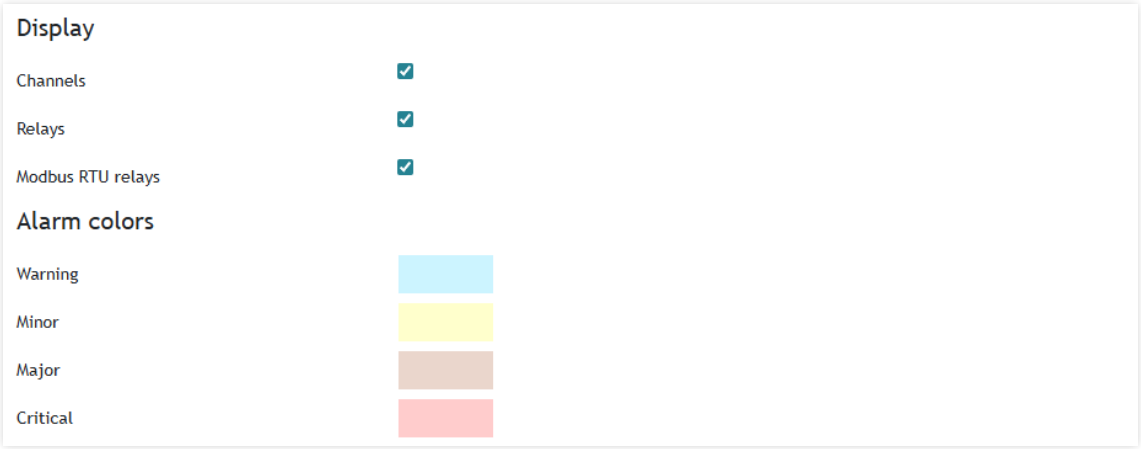


To enhance security, the web server can transition to HTTPS, defaulting to port 443, also customizable. The implementation of HTTPS utilizes TLS 1.0, TLS 1.1, and TLS 1.2, with RSA serving as the key exchange, agreement, and authentication mechanism.

It's important to note that the TCW242 uses a self-signed certificate for HTTPS, which may trigger a "Your connection is not private" warning in some web browsers. Additionally, the TCW242 operates with limited resources, meaning that running in HTTPS mode may be slower than in HTTP mode. For optimal performance in HTTPS mode, we recommend using Firefox due to its lower resource requirements compared to other browsers.

HTTP API access authentication is also enabled by default, with the same credentials as web access. The controller supports two types of authentication; further details can be found in the explanation for the HTTP API below.

The refresh interval can be set between 0 and 253 seconds, where a setting of zero means no automatic refresh.

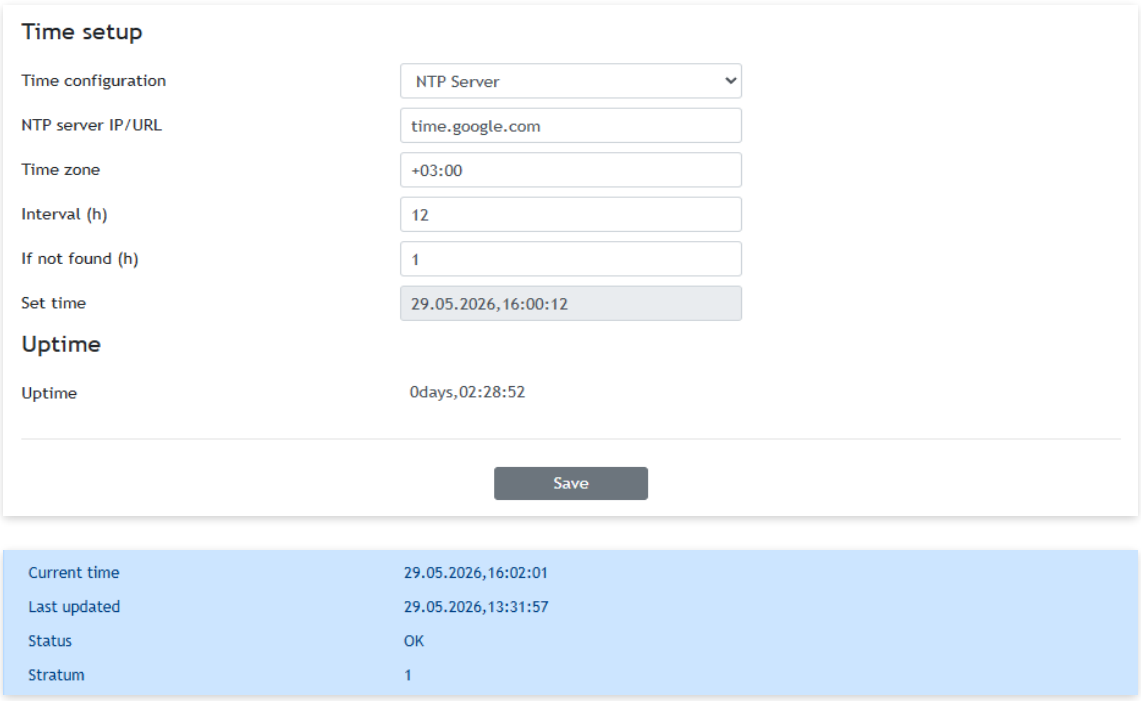


Alarm colors are fixed and provided here for informational purposes only.

8.3. Services

8.3.1 NTP

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



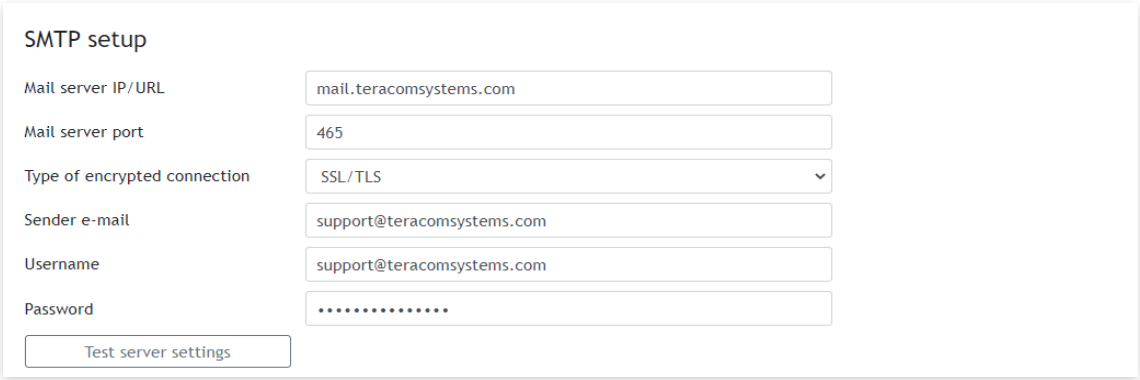
For automatic clock synchronization, the controller supports NTP (Network Time Protocol), and all necessary parameters for this synchronization are available in this section.

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

8.3.2 SMTP

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

8.3.2.1 SMTP setup



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

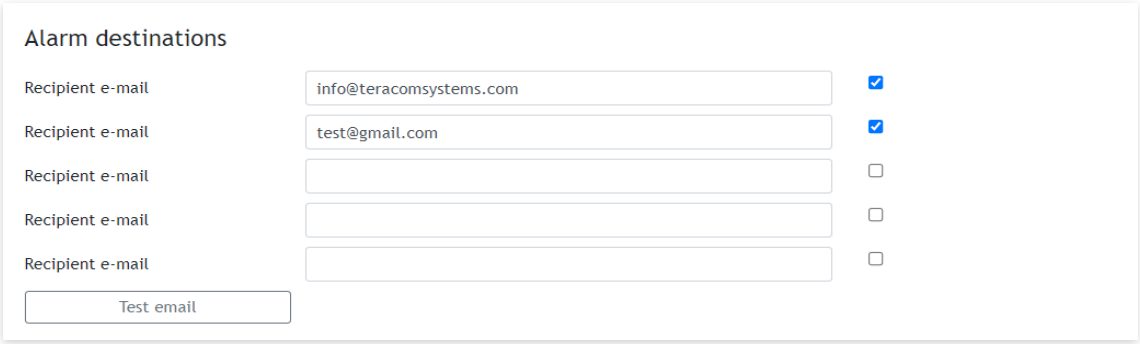
The sender's email, username, and password are standard authentication details. In the TCW242, each of these fields can accommodate up to 127 characters in length.

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

For secure communication with mail servers, the Transport Layer Security (TLS) protocol is employed. The TCW242 supports TLS versions 1.0, 1.1, and 1.2, using RSA for key exchange, agreement, and authentication. STARTTLS is also supported.

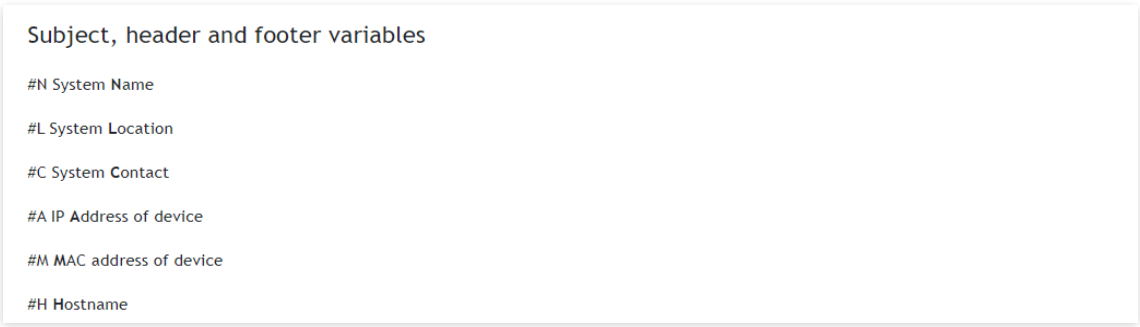
8.3.2.2 Alarm destination

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



8.3.2.3 E-mail details

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



8.3.3 SNMP

TCW242 supports SNMPv2 and SNMPv3. The default parameters are as follows*:*

- 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 enhanced security management for administration, authentication, and privacy. SNMPv3 provides the following configuration options:

- No authentication and no privacy (noAuthNoPriv) - Typically used for monitoring;
- Authentication and no privacy (authNoPriv) - Usually employed for control;

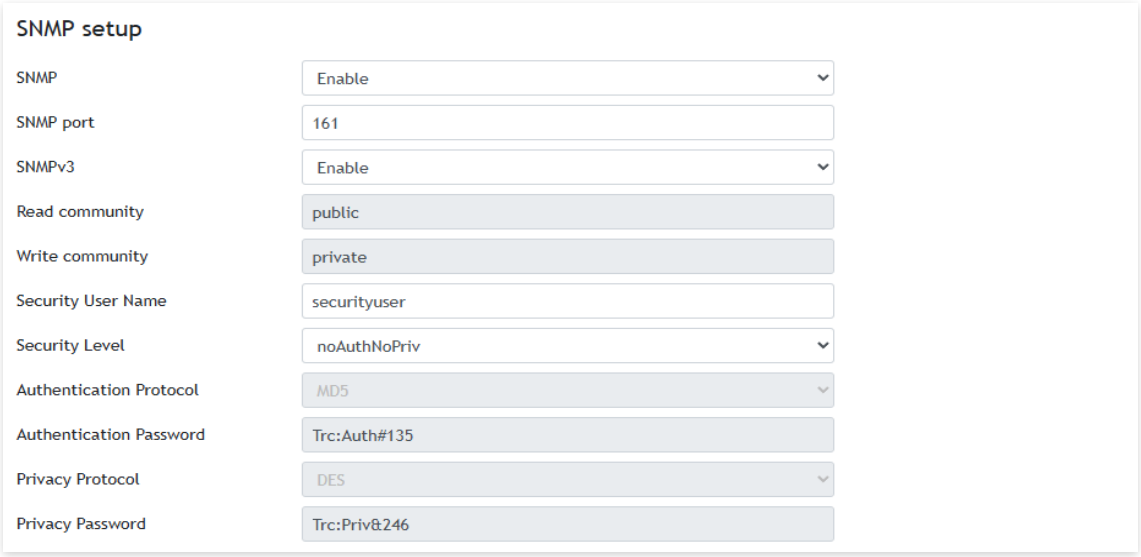
- Authentication and privacy (authPriv) - Commonly used for handling sensitive data.

User-based Authentication Mechanism is based on the following:

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

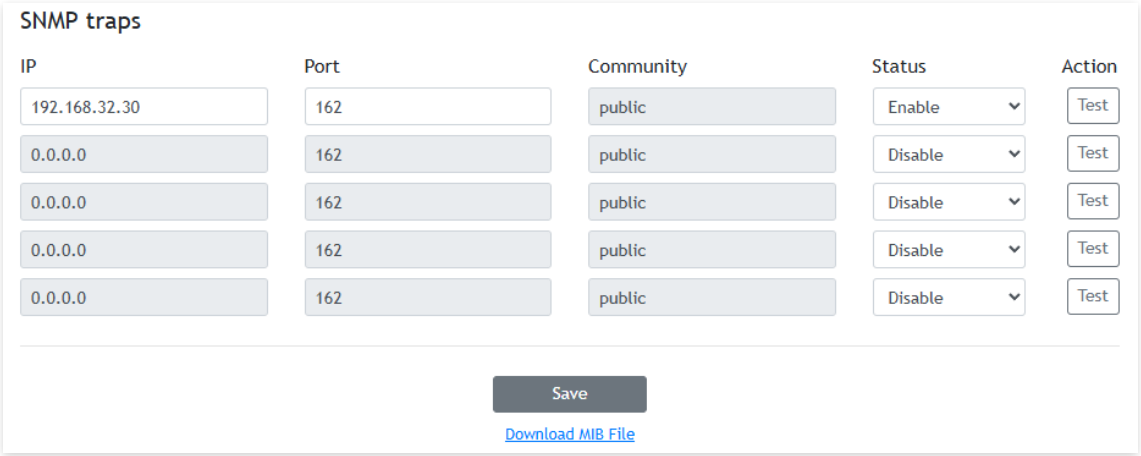
User-based Privacy Mechanism is based on the following:

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



For alarm notifications, an SNMP trap can be sent to up to five independent recipients. Each trap can use a different port and community. An SNMP trap is also sent after a reset.

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



8.3.4 Logger

The TCW242 supports data logging for all monitored channels and alarm events, allowing historical records to be stored locally and periodically uploaded to a remote server.

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

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

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

For synchronized logging, the device supports Logger record sync, which aligns log records to a specific time reference. When synchronization is enabled, the logger creates records at exact intervals based on the configured synchronization point instead of counting from the device startup time. This is useful for applications requiring precise time-based measurements or synchronization with external systems.

When synchronized logging is enabled, the Sync to the minute parameter defines the exact minute within each hour (00–59) when synchronized logging begins. The Log interval, minutes parameter defines the interval between synchronized records.

Example:

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

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

By default, the logger is disabled.

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

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

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

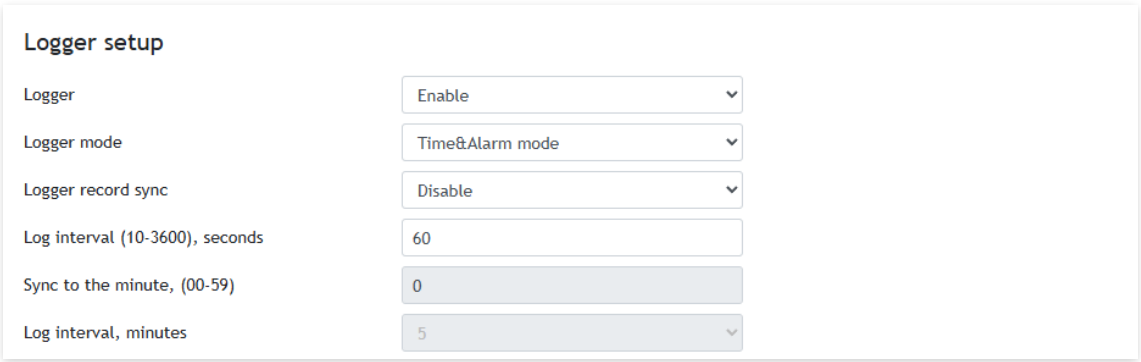
The Sync time parameter defines the exact time of day used as a reference point for periodic uploads. This ensures uploads occur at predictable intervals aligned to the configured schedule.

Example:

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

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

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



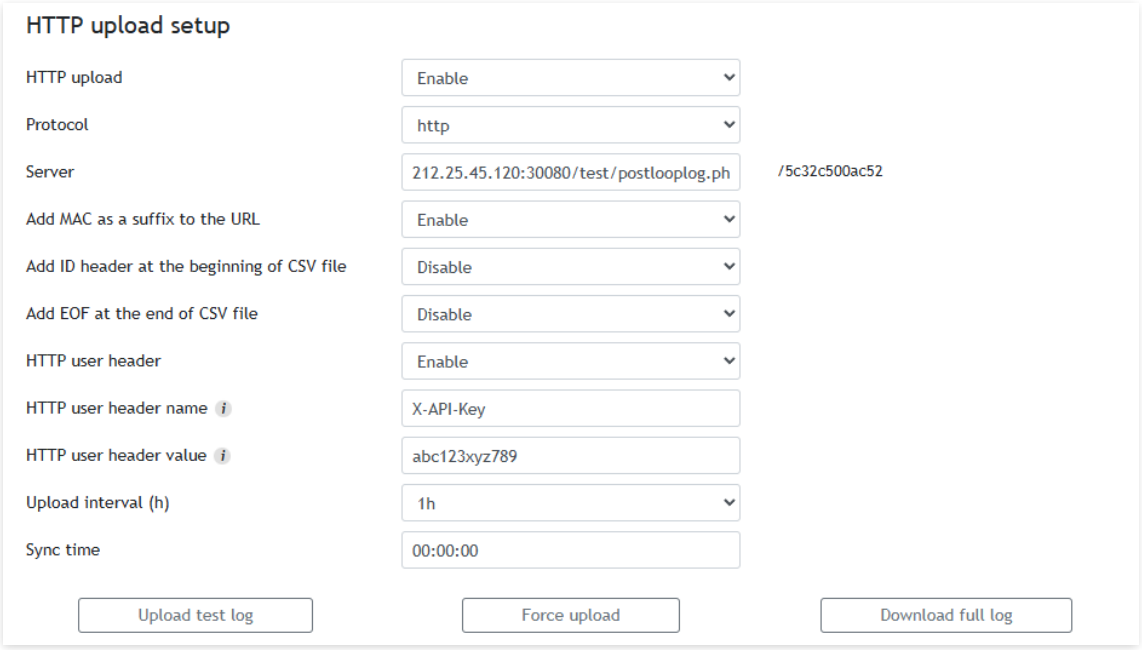
Configuration parameters:

- Logger – Enables or disables the logger functionality.
- Logger mode – Defines when data is recorded:
  Time – Records data at regular intervals.

Alarm – Records data only when an alarm event occurs.

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

- Logger record sync – Enables synchronization of log records to the configured time reference.
- Log interval (seconds) – Defines the interval between log records. Adjustable from 10 to 3600 seconds.



HTTP Upload configuration parameters:

- Sync to the minute – Defines the minute (00–59) used for synchronized logging.
- Log interval, minutes – Defines the interval between synchronized records when logger record sync is enabled.
- HTTP upload – Enables or disables periodic HTTP/HTTPS upload of logger records.
- Protocol – Selects the communication protocol for uploads (HTTP/HTTPS).
- Server – Specifies the destination server address (IP address, domain name, and optional port) where CSV log files will be uploaded.
- Add MAC as a suffix to the URL – Appends the device MAC address to the configured server URL for unique identification.
- Add ID header at the beginning of CSV file – Adds a header row containing parameter identifiers for easier automated processing.
- Add EOF at the end of CSV file – Appends an end-of-file marker for compatibility with server-side parsers.
- HTTP user header – Enables or disables sending a custom HTTP header with the upload request.
- HTTP user header name – Defines the custom HTTP header field name.
- HTTP user header value – Defines the custom HTTP header field value.
- Upload interval (h) – Defines how often stored records are uploaded to the server.
- Sync time – Defines the daily reference time for upload synchronization.
- Upload test log – Sends a test CSV file to verify server connectivity and upload configuration.
- Force upload – Immediately uploads all unsent logger records to the configured server.

- Download full log – Downloads the complete logger file from the device for local storage or analysis.

8.3.5 HTTP POST

The TCW242 supports automatic data transmission to remote servers through HTTP or HTTPS POST requests, enabling integration with cloud platforms, SCADA systems, monitoring software, and custom backend applications.

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

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

The device supports up to three independent destination servers, allowing the same data to be sent simultaneously to multiple systems for redundancy, monitoring, or parallel processing. Each server can be configured using either an IP address or a domain name.

The Period parameter defines the interval between periodic transmissions and can be configured from 1 minute to 48 hours. A shorter period provides more frequent updates and near real-time operation but increases network traffic.

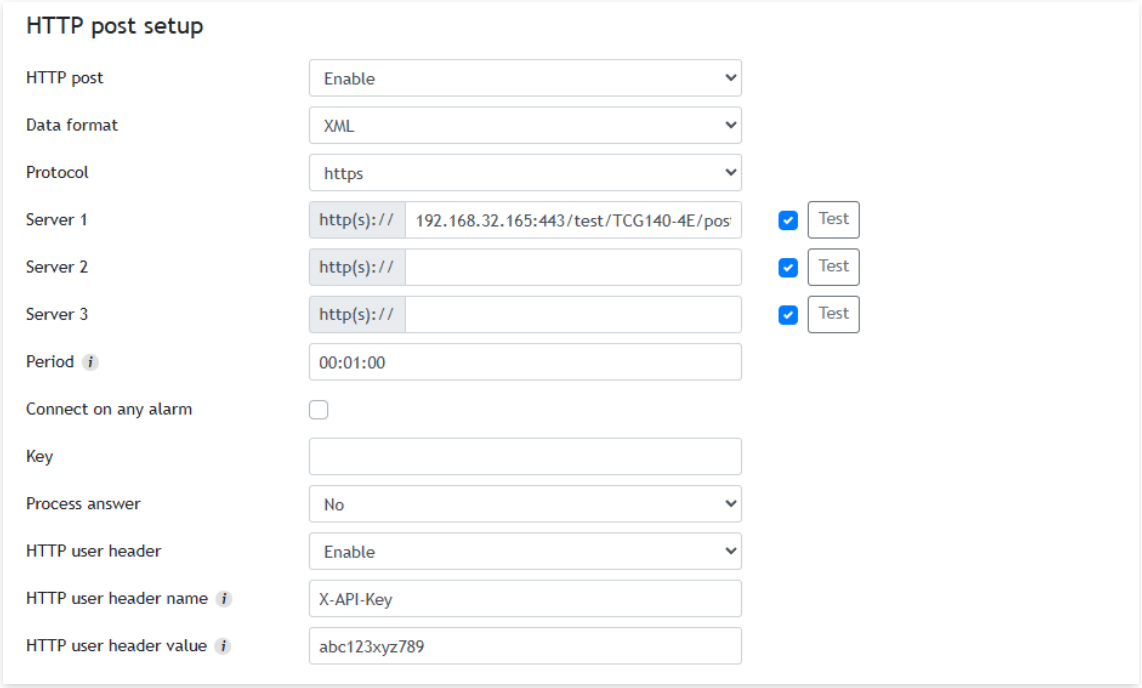
If the Connect on any alarm option is enabled, the device will immediately send an HTTP/HTTPS POST request whenever an alarm condition occurs, regardless of the configured periodic transmission interval.

The Key parameter is a user-defined identifier included in every XML/JSON payload. It can be used by the receiving server for device identification, installation association, or request authentication.

If the Process answer option is enabled, the device processes the HTTP response from the remote server. This allows the server to return supported API commands for remote control or configuration updates (see HTTP API Commands section).

The device also supports custom HTTP headers, enabling compatibility with authenticated APIs, cloud services, and specialized server-side applications.

Each configured server can be tested individually using the Test button to verify connectivity and server response.



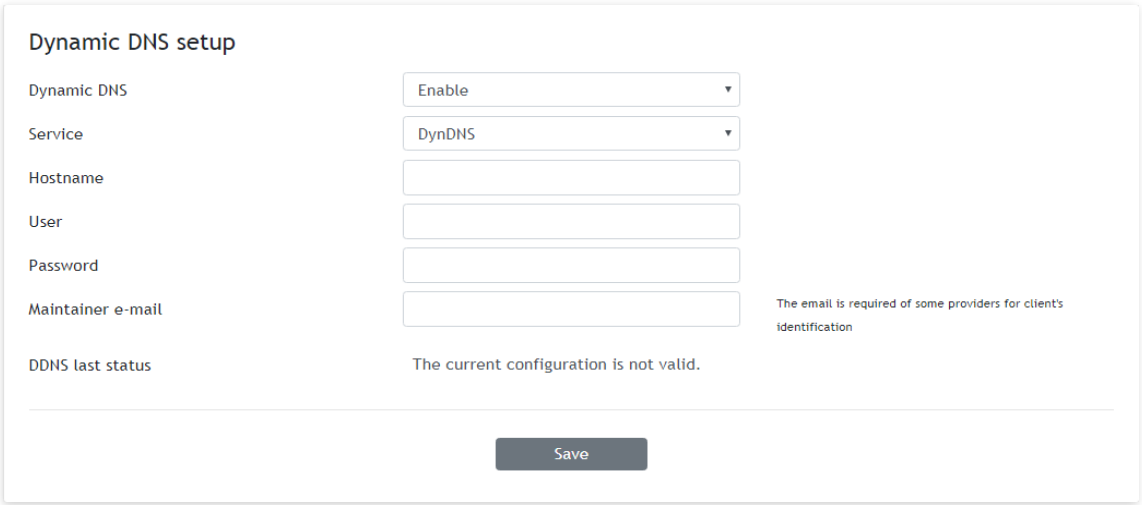
Configuration parameters:

- HTTP POST – Enables or disables the HTTP POST functionality.
- Data format – Defines the payload structure sent to the server (XML or JSON).
- Protocol – Selects the communication protocol (HTTP or HTTPS).
- Server 1 – Specifies the primary destination server address (IP address, domain name, and optional port).
- Server 2 – Specifies the secondary destination server address.
- Server 3 – Specifies the third destination server address.
- Period – Defines the interval between periodic transmissions. Adjustable from 1 minute to 48 hours.
- Connect on any alarm – Sends data immediately when an alarm condition occurs.
- Key – A user-defined identifier included in every XML/JSON payload for source identification or authentication.
- Process answer – Enables processing of the server response, allowing supported API commands to be executed by the device.
- HTTP user header – Enables or disables sending a custom HTTP header.
- HTTP user header name – Defines the custom HTTP header field name (for example: Authorization, X-API-Key, Content-Type).
- HTTP user header value – Defines the value associated with the custom HTTP header field.
- Test – Sends a test HTTP/HTTPS POST request to the selected server to verify connectivity and configuration.

8.3.6 Dynamic DNS

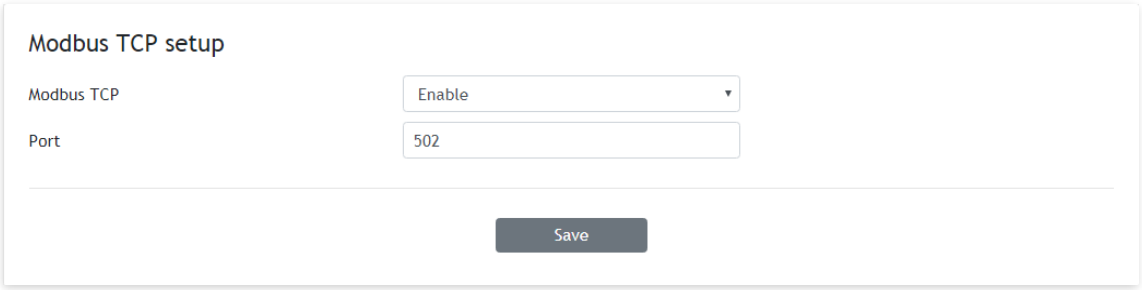
With dynamic DNS, the TCW242 can be accessed from the public Internet without the need for a broadband account with a static IP address.

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



8.3.7 Modbus TCP/IP

TCW242 supports Modbus TCP/IP via the Ethernet interface.



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

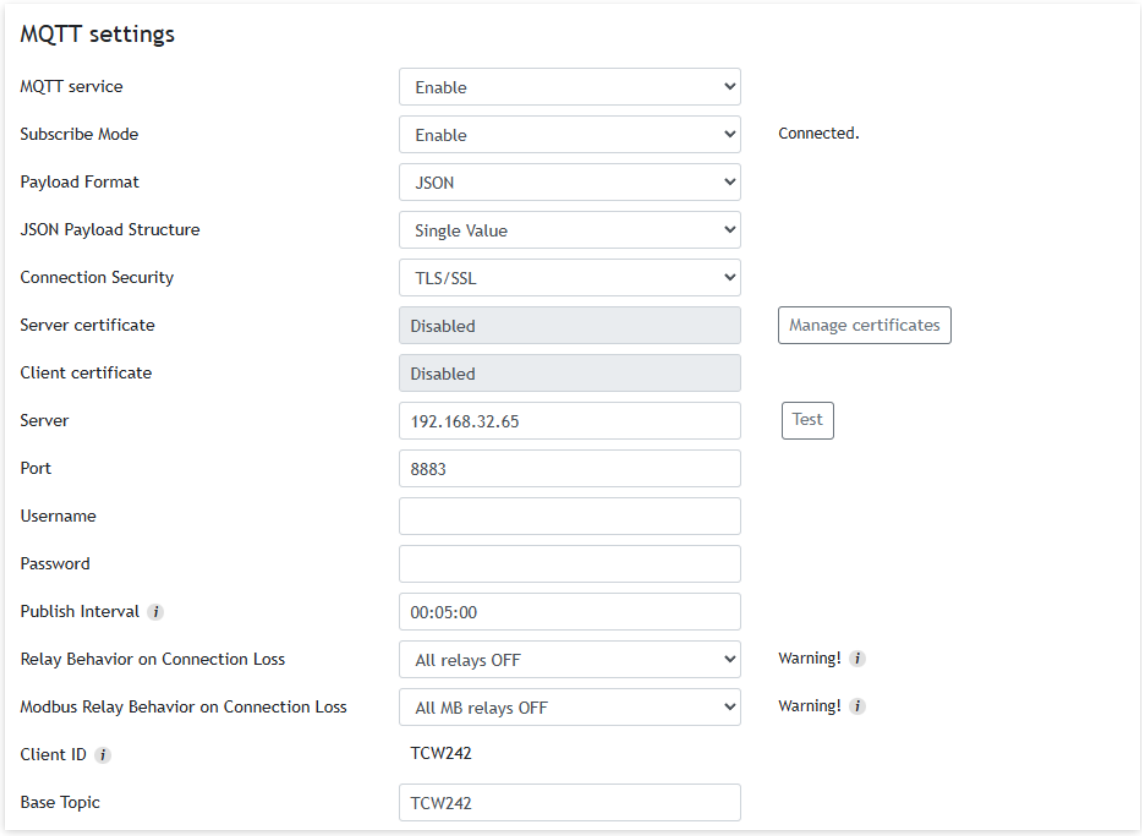
A table containing the register addresses can be found in section " Modbus TCP/IP".

8.3.8 MQTT

This page is used to configure MQTT communication. The TCW242 can publish monitored values and states to an MQTT broker and subscribe to MQTT topics for remote relay control and device configuration through API commands.

8.3.8.1 MQTT connection settings

This section is used to configure the general MQTT communication parameters.



The TCW242 supports both MQTT publishing (sending data to the broker) and MQTT subscribing (receiving commands from the broker). By default, both functions are disabled.

After enabling the MQTT service, the device establishes a connection to the MQTT broker using the parameters configured in the Server, Port, Username, and Password fields. The Server field can contain either a hostname or an IP address. The Server, Username, and Password fields support up to 127 characters.

The Port field defines the MQTT broker communication port. The default port is 1883 for unencrypted communication and 8883 for MQTT over TLS/SSL.

The Connection Security parameter defines whether the connection to the broker is unencrypted or protected by TLS/SSL. For secure MQTT connections, server certificate validation and client certificate authentication can be enabled independently. Certificates and private keys are managed through the Administration → Certificates page.

The Payload Format parameter defines the format of the published MQTT messages. The supported formats are Plain Text and JSON. When JSON is selected, the JSON Payload Structure parameter defines how published data is organized within the MQTT payload.

The Publish Interval parameter defines how often the device publishes data to the MQTT broker. The allowed interval range is from 1 minute to 48 hours.

The Relay Behavior on Connection Loss parameter defines the state of the local relay outputs when the connection to the MQTT broker is lost. Available options are All relays OFF, All relays ON, and Keep last state.

The Modbus Relay Behavior on Connection Loss parameter defines the state of configured MODBUS RTU relays when the connection to the MQTT broker is lost. Available options are All relays OFF, All relays ON, and Keep last state.

The Base Topic field defines the root MQTT topic used for all published and subscribed messages. All MQTT topics configured on this page are formed relative to the configured Base Topic.

The Client ID uniquely identifies the device on the MQTT broker. By default, the System ID is used as the Client ID.

The Test button can be used to verify the connectivity, authentication, and security settings of the configured MQTT broker.

8.3.8.2 MQTT topics and data publishing

This section is used to configure MQTT topics and publishing settings for channels, alarms, relays, and MODBUS RTU relays.

Each topic section (Channels, Alarms, Relays, and MODBUS RTU Relays) has a configurable Section Topic. Each item within a section has a configurable Item Topic.

The complete MQTT topic is formed as follows:

Base Topic / Section Topic / Item Topic / Parameter

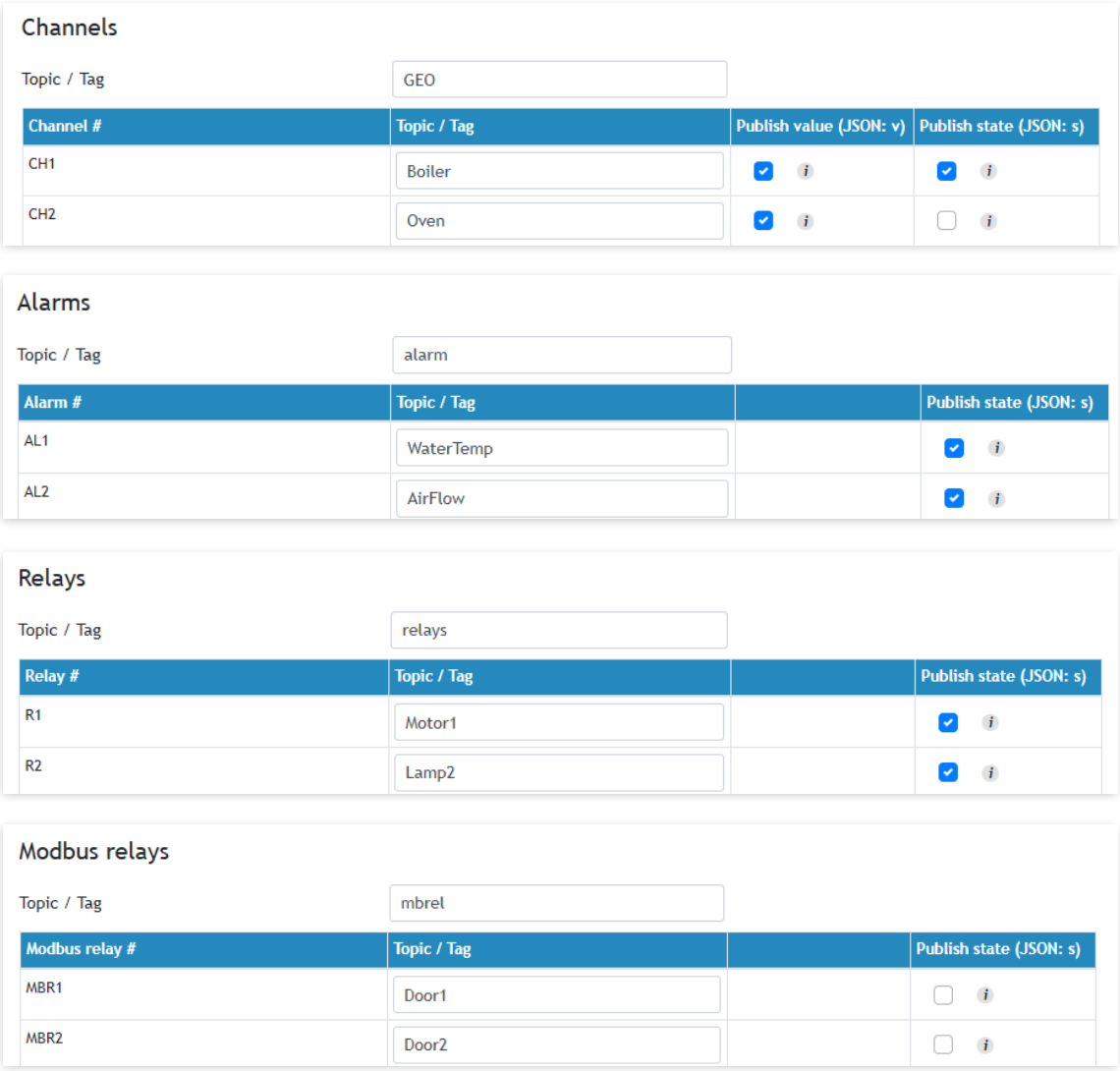
where:

- Base Topic – the root MQTT topic configured in the MQTT connection settings;
- Section Topic – the user-defined topic name assigned to a topic section;
- Item Topic – the user-defined name assigned to a channel, alarm, relay, or MODBUS RTU relay;
- Parameter – the final topic level identifying the published parameter, such as value (v) or state (s).

When Plain Text payload format is selected, only the parameter value is published.

When JSON payload format is selected, the payload content depends on the selected JSON Payload Structure setting (Single Value, Item, Section, or All).

For the following examples, assume that the MQTT topic configuration shown below is used:



Channels

Topic / Tag: GEO

Channel #	Topic / Tag	Publish value (JSON: v)	Publish state (JSON: s)
CH1	Boiler	☑	☑
CH2	Oven	☑	☐

Alarms

Topic / Tag: alarm

Alarm #	Topic / Tag		Publish state (JSON: s)
AL1	WaterTemp		☑
AL2	AirFlow		☑

Relays

Topic / Tag: relays

Relay #	Topic / Tag		Publish state (JSON: s)
R1	Motor1		☑
R2	Lamp2		☑

Modbus relays

Topic / Tag: mbrel

Modbus relay #	Topic / Tag		Publish state (JSON: s)
MBR1	Door1		☐
MBR2	Door2		☐

Base Topic = TCW242
Channels section topic = GEO
CH1 item topic = Boiler
CH2 item topic = Oven
Alarms section topic = alarm
AL1 item topic = WaterTemp
AL2 item topic = AirFlow
Relays section topic = relays
R1 item topic = Motor1
R2 item topic = Lamp2
MB relays section topic = mbrel
MBR1 item topic = Door1
MBR2 item topic = Door2

Examples:

- Plain text:
  Topic: TCW242/GEO/Boiler/v
  Payload: 28.5

Topic: TCW242/GEO/Boiler/s

Payload: 1

Topic: TCW242/alarm/WaterTemp/s

Payload: 1

A separate MQTT topic is published for each parameter. The payload contains only the parameter value or state.

- JSON - Single Value

  Topic: TCW242/GEO/Boiler/v

  Payload: {"v":"28.5"}

  Topic: TCW242/GEO/Boiler/s

  Payload: {"s":"1"}

  Topic: TCW242/alarm/WaterTemp/s

  Payload: {"s":"1"}

  A separate MQTT topic is published for each parameter.

- JSON - Item

  Topic: TCW242/GEO/Boiler

  Payload: {"v":"28.5","s":"0"}

  Topic: TCW242/GEO/Oven

  Payload: {"v":"22.1"}

  A separate MQTT topic is published for each item. The payload contains all enabled parameters configured for that item.

- JSON - Section

  Topic: TCW242/GEO

  Payload: {

  ```
  "Boiler":{"v":"28.5","s":"0"},
  "Oven":{"v":"22.1"}
  }
  ```

  Topic: TCW242/alarm

  Payload: {

  ```
  "WaterTemp":{"s":"0"},
  " AirFlow":{"s":"1"}
  }
  ```

  A separate MQTT topic is published for each section.

- JSON – All

  Topic: TCW242

  Payload: {

  ```
  "GEO":{"Boiler":{"v": "28.5","s": "0"},
           "Oven":{"v": "22.1"}
         },
  "alarm":{"WaterTemp":{"s": "0"},
             "AirFlow":{"s": "1"}
           },
  "relays":{"Motor1":{"s": "1"},
            "Lamp2":{"s": "0"}
           }
  }
  ```

A single MQTT topic is published. The payload contains all enabled topic sections, items, and parameters.

Sections with no enabled items are omitted from the published JSON payload. In this example, the MODBUS RTU Relays section is omitted because no MODBUS RTU relays are enabled for publishing.

The selected JSON Payload Structure affects both the MQTT topic level used for publishing and the amount of information included in the payload.

8.3.8.3 MQTT subscribe mode

MQTT subscribe mode allows remote control of relay outputs, execution of device API commands, and on-demand publishing of the current device status through MQTT messages.

When MQTT subscribe mode is enabled, the TCW242 subscribes to MQTT topics and processes the received messages. Four subscription topics are used:

- Relay control topic – used for direct control of built-in relays;
- MODBUS RTU relay control topic – used for direct control of MODBUS RTU relays;
- API command topic – used for remote device configuration and control through API commands;
- Publish topic – used to trigger immediate publishing of the current device status.

Both plain text and JSON payloads are accepted for all MQTT subscribe functions, regardless of the globally selected MQTT Payload Format setting.

The subscribed topics are formed as follows:

Base Topic / Relays Section Topic / + / command

Base Topic / MODBUS Relays Section Topic / + / command

Base Topic / api

Base Topic / publish

where:

Base Topic – the root MQTT topic configured in the MQTT connection settings;

Relays Section Topic – the user-defined topic assigned to the Relays section;

MODBUS Relays Section Topic – the user-defined topic assigned to the MODBUS RTU Relays section;

"+" – MQTT single-level wildcard used to match all configured relay item topics.

The keywords command, api, and publish are fixed and cannot be changed.

Relay control using MQTT subscribe

Built-in and MODBUS RTU relays can be controlled through their dedicated command topics.

Example:

Base Topic = TCW242

Relays Section Topic = relays

Relay Item Topic = Motor1

Command topic:

TCW242/relays/Motor1/command

The following payloads are accepted:

- To turn ON the relay assigned to Motor1:
  Plain text payload : 1
  JSON payload : {"command":"1"}
- To turn OFF the relay assigned to Motor1:

Plain text payload : 0

JSON payload : {"command":"0"}

Device configuration and control using MQTT subscribe

The TCW242 supports remote configuration and control through the API topic:

TCW242/api

Commands received through this topic are processed in the same way as commands received through the HTTP API interface.

Examples:

- Plain text payload : ron=1
  Turns Relay 1 ON.
- Plain text payload : pper=15
  Changes the HTTP POST period to 15 seconds.
- JSON payload : {"api":"rpl=2"}
  Will pulse Relay 2.

A complete list of supported commands is available in the HTTP API section.

Publish command

The publish topic can be used to request immediate publication of the current device status.

Example:

Topic:
TCW242/publish

Any payload value can be used. The payload content is ignored.

Upon receiving a message on this topic, the controller immediately publishes its current data using the configured MQTT publishing settings.

8.3.8.4 MQTT Certificates

MQTT TLS/SSL communication supports server certificate validation and client certificate authentication.

Certificate and private key management is described in the Administration → Certificates section.

8.4. Administration

The Administration menu provides access to user management, certificate management, configuration backup and restore, firmware updates, and other maintenance functions

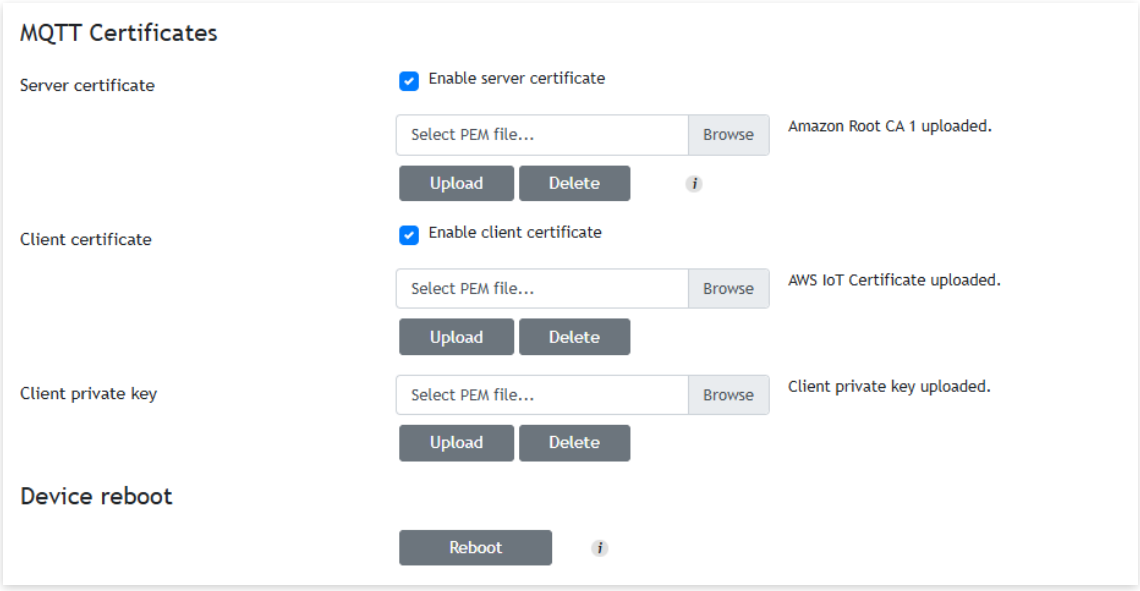
8.4.1 MQTT Certificates

The MQTT Certificates page is used to manage the certificates and private keys required for secure MQTT communication over TLS/SSL.

The controller supports three certificate-related files:

- Server certificate – used to validate the MQTT broker's identity.
- Client certificate – used for client authentication when required by the MQTT broker.
- Client private key – the private key associated with the client certificate.

All uploaded files must be in PEM format.



To upload a certificate or private key, select the corresponding PEM file and press Upload. Existing files can be removed using the Delete button.

The Enable server certificate and Enable client certificate options determine whether the uploaded certificates are used during the TLS authentication process.

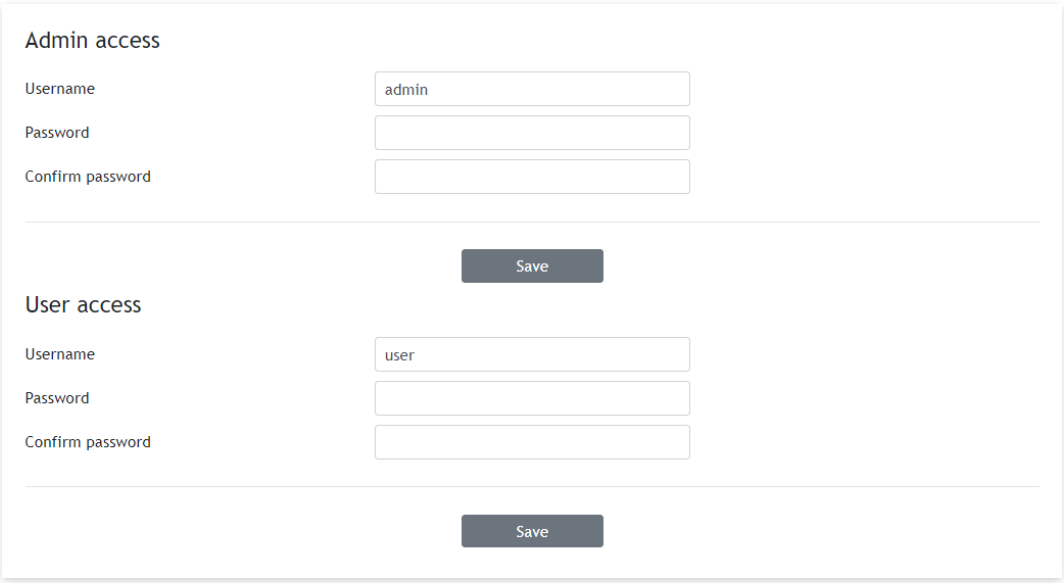
For client certificate authentication, both a valid client certificate and the corresponding client private key must be uploaded. The client certificate cannot be used without a matching private key.

The current status of each certificate or key is displayed next to the corresponding upload control.

Note: Changes to the server certificate, as well as enabling or disabling the Enable server certificate option, require a device reboot to take effect. The device can be restarted using the Reboot button located at the bottom of the page.

8.4.2 User/Pass

The TCW242 supports two user roles: "Admin" and "User".

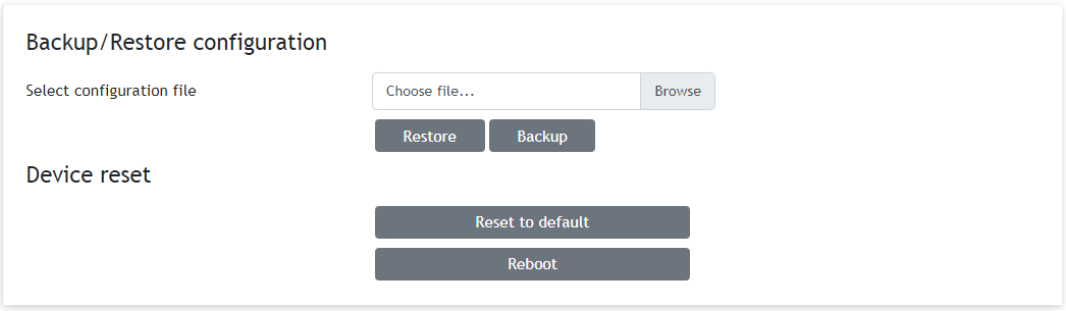


The Admin role has full administrative rights and can modify all settings.

The User role has limited access, allowing only viewing of the Monitoring page and preventing any modifications to settings.Both usernames and passwords can be up to 31 characters in length.

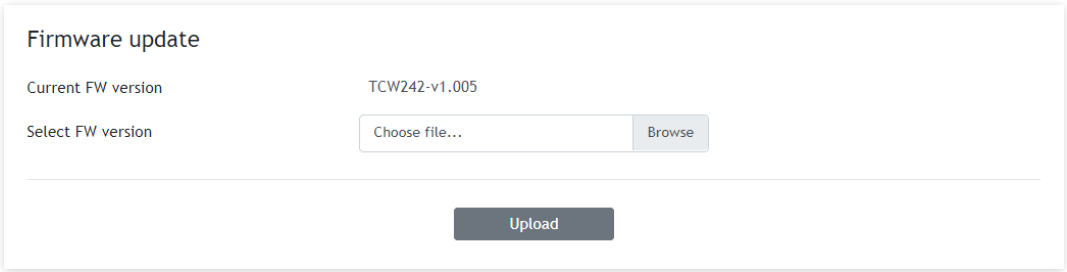
8.4.3 Backup/Restore

The TCW242 allows for the backup and restoration of all user settings. These settings are saved in an XML backup file, which can be utilized for restoring configurations on multiple devices. This feature is particularly useful for applying similar settings across a batch of controllers.



8.4.4 FW update

The TCW242 can be updated through its web interface.



To update the device, follow these steps:

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

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

8.5. Logout

The TCW242 supports multi-session access; however, it is recommended to log out after completing your tasks for security and best practices.

9. Protocols and API

9.1. SNMP

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

product

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

setup -> network

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 in the table below

OID	Name	Access	Description	Syntax
!.8.2.2.1.1.1.2.?	mbSenDescription.?	read-write	Sensor description	DisplayString
!.8.2.2.1.1.1.3.?	mbSenMult.?	read-write	Sensor multiplier x1000 in Integer format	Integer32
!.8.2.2.1.1.1.4.?	mbSenOffset.?	read-write	Sensor offset x1000 in Integer format	Integer32
!.8.2.2.1.1.1.5.?	mbSenVal.?	read-only	Sensor value x1000 in Integer format	Integer32
!.8.2.2.1.1.1.6.?	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 in the table below

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

OID	Name	Access	Description	Syntax
!.8.2.2.3.1.1.2.?	digInpDescription.?	read-write	Digital Input description	DisplayString
!.8.2.2.3.1.1.3.?	digInplowLevel.?	read-write	Digital Input closed state	DisplayString
!.8.2.2.3.1.1.4.?	digInphighLevel.?	read-write	Digital Input open state	DisplayString
!.8.2.2.3.1.1.5.?	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.?	digInpCloseToOpenDelay.?	read-write	Digital input Close To Open delay	Integer32(0..60000)
!.8.2.2.3.1.1.7.?	digInpOpenToCloseDelay.?	read-write	Digital input Open To Close delay	Integer32(0..60000)
!.8.2.2.3.1.1.8.?	digInpCounterInitValue.?	read-only	Digital input counter initial value	Integer32
!.8.2.2.3.1.1.9.?	digInpValue.?	read-only	Digital input value	Unsigned32

setup -> parameters -> relaySetup -> relOutSetupTable -> relOutSetupEntry -> relOutSetupIndex 1 to 4

replace "?" with a number from 1 to 4 in the table below

OID	Name	Access	Description	Syntax
!.8.2.2.4.1.1.2.**?**	relOutDescription.**?**	read-write	Relay description	DisplayString
!.8.2.2.4.1.1.3.**?**	relOutPulseWidth.**?**	read-write	Relay Pulse x100ms	Integer32
!.8.2.2.4.1.1.4.**?**	relOutActivation.**?**	read-write	Relay activated from	INTEGER {webHttpApi(0), al01(1), al02(2), al03(3), al04(4), al05(5), al06(6), al07(7), al08(8), al09(9), al10(10), al11(11), al12(12), al13(13), al14(14), al15(15), al16(16), al17(17), al18(18), al19(19), al20(20), al21(21), al22(22), al23(23), al24(24), anyAlarm(25), schedule1(26), schedule2(27), schedule3(28), schedule4(29), func1(30), func2(31), func3(32), func4(33), watchdog1(34), watchdog2(35), watchdog3(36), watchdog4(37) }
!.8.2.2.4.1.1.5.**?**	relOutAction.**?**	read-write	Relay action on alarm condition	INTEGER {on(0), pulse(2)}
!.8.2.2.4.2.0	relOutAfterRst	read-write	Relay state after restart	INTEGER {off(0), on(1), laststate(2)}

setup -> parameters -> modbusRelaySetup -> modbusRelOutSetupTable -> modbusRelOutSetupEntry -> modbusRelOutSetupIndex 1 to 24

replace "?" with a number from 1 to 24 in the table below

OID	Name	Access	Description	Syntax
!.8.2.2.5.1.1.2.?	relOutDescription.?	read-write	Relay description	DisplayString
!.8.2.2.5.1.1.4.?	relOutActivation.?	read-write	Relay activated from	INTEGER {webHttpApi(0), al01(1), al02(2), al03(3), al04(4), al05(5), al06(6), al07(7), al08(8), al09(9), al10(10), al11(11), al12(12), al13(13), al14(14), al15(15), al16(16), al17(17), al18(18), al19(19), al20(20), al21(21), al22(22), al23(23), al24(24), anyAlarm(25), schedule1(26), schedule2(27), schedule3(28), schedule4(29), func1(30), func2(31), func3(32), func4(33), watchdog1(34), watchdog2(35), watchdog3(36), watchdog4(37) }
!.8.2.2.5.2.0	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 in the table below

OID	Name	Access	Description	Syntax
!.8.3.1.1.1.2.**?**	chType.**?**	read-write	Channel type	INTEGER {general(0), cumulative(1), discrete(2), counter(3)}
!.8.3.1.1.1.3.**?**	chdescription.**?**	read-write	Channel description	DisplayString
!.8.3.1.1.1.4.**?**	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.**?**	chOP1.**?**	read-write	Channel operand 1	INTEGER{none(0), multiplication(1), division(2), sum(3), subtract(4)}
!.8.3.1.1.1.6.**?**	chParam2.**?**	read-write	Channel parameter 2	INTEGER {none(0), one(1), null(2), s01(3), s02(4), s03(5), s04(6), s05(7), s06(8), s07(9), s08(10),s09(11), s10(12), s11(13), s12(14), s13(15), s14(16), s15(17), s16(18), s17(19), s18(20), s19(21), s20(22), s21(23), s22(24), s23(25), s24(26),a01(27), a02(28), a03(29), a04(30), a05(31), a06(32),d01(33), d02(34), d03(35), d04(36)}
!.8.3.1.1.1.7.**?**	chOP2.**?**	read-write	Channel operand 2	INTEGER {none(0), multiplication(1), division(2), sum(3), subtract(4)}
!.8.3.1.1.1.8.**?**	chCoef1.**?**	read-write	Channel coefficient 1 x1000 in Integer format	Integer32
!.8.3.1.1.1.9.**?**	chOP3.**?**	read-write	Channel operand 3	INTEGER {none(0), multiplication(1), division(2), sum(3), subtract(4)}
!.8.3.1.1.1.10.**?**	chCoef2.**?**	read-write	Channel coefficient 2 x1000 in Integer format	Integer32
!.8.3.1.1.1.11.**?**	chUnit.**?**	read-write	Channel unit	DisplayString
!.8.3.1.1.1.13.**?**	chValue.**?**	read-only	Channel value x1000 in Integer format	Integer32
!.8.3.1.1.1.14.**?**	chCounter.**?**	read-only	Channel as 32-bit counter	Counter32
!.8.3.1.1.1.15.**?**	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 in the table below

OID	Name	Access	Description	Syntax
!.8.3.2.1.1.2.**?**	relayState.**?**	read-write	Relay state	INTEGER {off(0), on(1)}
!.8.3.2.1.1.3.**?**	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 in the table below

OID	Name	Access	Description	Syntax
!.8.3.3.1.1.2.**?**	alDescription.**?**	read-write	Alarm description	DisplayString
!.8.3.3.1.1.3.**?**	alCond1Channel.**?**	read-write	Alarm condition 1 channel	INTEGER {none(0), v01(1), v02(2), v03(3), v04(4), v05(5), v06(6), v07(7), v08(8), v09(9), v10(10), v11(11), v12(12), v13(13), v14(14), v15(15), v16(16), v17(17), v18(18), v19(19), v20(20), v21(21), v22(22), v23(23), v24(24)}
!.8.3.3.1.1.4.**?**	alCond1Operand.**?**	read-write	Alarm condition 1 operand	INTEGER{larger(1), less(2)}
!.8.3.3.1.1.5.**?**	alCond1Limit.**?**	read-write	Alarm condition 1 limit x1000 in Integer format	Integer32
!.8.3.3.1.1.6.**?**	alCond1Hys.**?**	read-write	Alarm condition 1 hysteresis x1000 in Integer format	Integer32
!.8.3.3.1.1.7.**?**	alCond1AlarmState.**?**	read-write	Alarm condition 1 discrete alarm state	INTEGER {open(0), closed(1)}
!.8.3.3.1.1.8.**?**	alCondLogic.**?**	read-write	Alarm conditions logic	INTEGER{none(0), and(1), or(2)}
!.8.3.3.1.1.9.**?**	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.**?**	alCond2Operand.**?**	read-write	Alarm condition 2 operand	INTEGER{larger(1), less(2)}
!.8.3.3.1.1.11.**?**	alCond2Limit.**?**	read-write	Alarm condition 2 limit x1000 in Integer format	Integer32
!.8.3.3.1.1.12.**?**	alCond2Hys.**?**	read-write	Alarm condition 2 hysteresis x1000 in Integer format	Integer32
!.8.3.3.1.1.13.**?**	alCond2AlarmState.**?**	read-write	Alarm condition 2 discrete alarm state	INTEGER {open(0), closed(1)}
!.8.3.3.1.1.14.**?**	alType.**?**	read-write	Alarm type	INTEGER {warning(3), minor(4), major(5), critical(6)}
!.8.3.3.1.1.15.**?**	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.**?**	alActionDelay.**?**	read-write	Alarm action delay	Integer32
!.8.3.3.1.1.17.**?**	alActionOnReturn.**?**	read-write	Alarm action on return	INTEGER {no(0), yes(1)}
!.8.3.3.1.1.18.**?**	alAction1.**?**	read-write	Alarm action 1	INTEGER {none(0), trapcond1(1), trapcond2(2), trapcond1and2(3), postiostate(4)}
!.8.3.3.1.1.19.**?**	alAction2.**?**	read-write	Alarm action 2	INTEGER {none(0), trapcond1(1), trapcond2(2), trapcond1and2(3), postiostate(4)}

!.8.3.3.1.1.20.**?**	alAction3.**?**	read-write	Alarm action 3	INTEGER {none(0), trapcond1(1), trapcond2(2), trapcond1and2(3), postiostate(4)}
!.8.3.3.1.1.21.**?**	alStatus.**?**	read-write	Alarm status	INTEGER {undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)}

monitorNcontrol

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

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

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

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

monitorNcontrol -> modbusrelays -> modbusRelOutTable -> modbusRelOutEntry -> modbusRelayIndex 1 to 24

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

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

monitorNcontrol-> wdTable -> wdEntry -> wdIndex 1 to 4

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

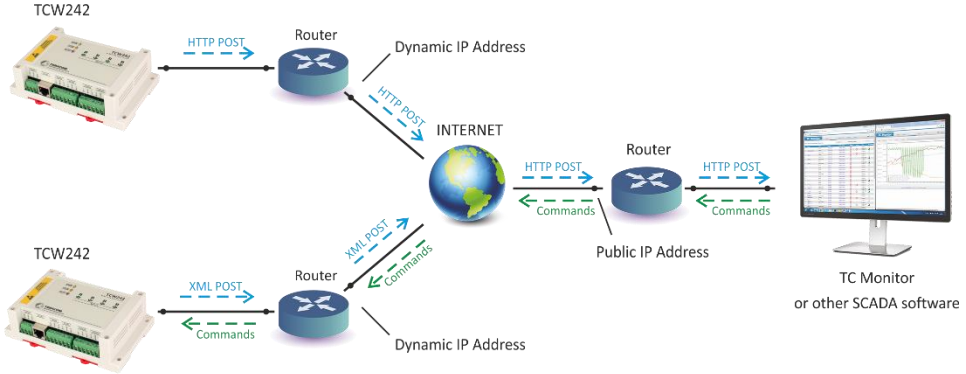
OID	Name	Access	Description	Syntax
!.8.3.9.1.1.2.?	wdState.?	read only	Watchdog state	INTEGER { false(0), true(1) }

9.2. HTTP API

9.2.1 HTTP POST

The TCW242 can execute HTTP/HTTPS POST requests to upload XML or JSON files to a dedicated server. This functionality is particularly useful when the controller is situated behind a router without a public IP address or when the user lacks access to the router's configuration. It is important to note that the server should possess a public IP address.

The typical monitoring application is shown in the picture below:



HTTP/HTTPS POSTs can be sent either periodically or periodically in conjunction with alarm conditions. In response, the server can send an HTTP GET request with the appropriate command - refer to section 9.2.3 for details on HTTP API commands

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

- Save the following code as post.php:

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

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

- Upload the post.php file to a public web server that supports PHP. To ensure the script is functioning correctly, enter the URL (for example, www.yourserverURL.com/post.php) in your web browser. If successful, a webpage displaying “The PHP script is working!” will appear.
- Configure the TCW242 controller to send HTTP/HTTPS POST requests to your web server. Input the URL (yourserverURL.com/post.php) in the URL field, and click the “Test HTTP POST” button.
- If the HTTP/HTTPS POST request is received and processed successfully, “OK” will be displayed next to the button. Additionally, an XML file will be created in the same directory where post.php is located. The filename will include a timestamp, formatted as 20190420103318_status.xml.

9.2.2 HTTP/HTTPS GET

HTTP/HTTPS GET can be utilized to monitor the TCW242 by retrieving XML or JSON files. The format for accessing the status is as follows:

http(s)://device.ip.address/status.xml

http(s)://device.ip.address/status.json

For further details regarding the structure of these files, refer to Appendix A XML file structure and Appendix B JSON file structure.

HTTP/HTTPS GET requests can be sent at any time to the TCW242 if it is on the same network or has appropriate routing.

If there is no direct access to the device, an HTTP/HTTPS GET can be sent immediately after receiving an HTTP/HTTPS POST from the same device.

9.2.2.1 Commands

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

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

Where:

yyy is the command;
xxx is the parameter.

Example:

To set the POST period to 300 seconds, the following command can be used

http(s)://device.ip.address/status.xml?pper=300

9.2.2.2 HTTP/HTTPS GET authentication

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

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

Example:

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

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

9.2.3 List of HTTP API commands

Command	Description
dataf=x	Data format XML/JSON for HHTP POST – 0 XML, 1 JSON
pushtls=x	HTTP/HTTPS 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 127 characters
ron=**n**	Turn relay ON (**n** is 1,2,4 or 8) ron=1 - will turn ON relay 1 ron=2 - will turn ON relay 2 ron=4 - will turn ON relay 3 ron=8 - will turn ON relay 4
ron=1&ron=2&ron=4&ron=8	Turn four relays ON

rof=**n**	Turn relay OFF (**n** is 1,2,4 or 8) rof=1 - will turn OFF relay 1 rof=2 - will turn OFF relay 2 rof=4 - will turn OFF relay 3 rof=8 - will turn OFF relay 4
rof=1&rof=2&rof=4&rof=8	Turn four relays OFF
rpl=**n**	Pulse relay **n** (**n** is 1,2,4 or 8) rpl=1 – will pulse relay 1 rpl=2 – will pulse relay 2 rpl=4 – will pulse relay 3 rpl=8 – will pulse relay 4
mdata=x	Data format JSON/Plain text for MQTT Publish – 0 JSON, 1 Plain text
mmode=x	Publish protocol, where x is 0 for unsecure and 1 for TLS/SSL
muser=xxxx	Username authentication for MQTT, where xxxx is a username
mpass=xxxx	Password authentication for MQTT, where xxxx is a password
murl=yyy	URL for MQTT publish, where yyy is a path murl=212.25.45.120
mport=yyyy	Port for MQTT publish, where yyyy is a port mport=1883
mper=x	MQTT publish period in seconds (x is between 60 and 172800) mper=600 – will set MQTT publish period to 600 seconds
save	Save all previous changes (except relays' one) in the FLASH memory. **As every save reflects the FLASH cycles (endurance), this command should be used very carefully.** pper=120&save – will set POST period to 120 seconds and save it
FIN	Terminate session (it works with HTTP/HTTPS POST, but not with HTTP GET)

9.3. Modbus TCP/IP

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

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

9.3.1 Codes and answers

9.3.1.1 Read Coil Status (FC=01)

Request

Command to obtain ON/OFF status of discrete coils at address 100.

01 0064 0001

Response

01 01 01

9.3.1.2 Force Single Coil (FC=05)

Request

This command sets the state of discrete address 100 to "ON".

05 0064 FF00

Response

The typical response is a copy of the request, sent back after writing to the coil.

05 0064 FF00

9.3.1.3 Read Discrete Inputs (FC=02)

Request

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

02 0064 0004

Response

02 01 0C

9.3.1.4 Read Holding Registers (FC=03)

Request

This command requests the value of holding register 19300.

03 4B64 0002

Response

03 04 41DD 4210

Request

This command requests the value of holding register 18100.

03 46B4 0008

Response

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

9.3.1.5 Write Single Register (FC=06)

Request

This command writes a value to the single register at address 18300.

06 477C 0003

Response

06 477C 0003

9.3.1.6 Write Multiple Registers (FC=16)

Request

This command writes a value to the single register at address 18300.

10 477C 0002 04 0003 0004

Response

10 477C 0002

9.3.1.7 Exception codes

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

01 Illegal function:

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

02 Illegal data address:

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

9.3.2 Main registers address table

The table below lists the most commonly used registers. A complete list of Modbus TCP/IP registers can be found in Appendix C.

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

Parameter	FC	PDU decimal address	Data size	Data
Relay 1	01,05,15	100	Discrete	
Relay 2	01,05,15	101	Discrete	
Relay 3	01,05,15	102	Discrete	
Relay 4	01,05,15	103	Discrete	
Digital input 1	02	100	Discrete	
Digital input 2	02	101	Discrete	
Digital input 3	02	102	Discrete	
Digital input 4	02	103	Discrete	
Analog input 1 value	03	14400	32-bit Float	
Analog input 2 value	03	14402	32-bit Float	
Analog input 3 value	03	14404	32-bit Float	
Analog input 4 value	03	14406	32-bit Float	
Channel 1 value	03	19300	32-bit Float	
Channel 2 value	03	19302	32-bit Float	
Channel 3 value	03	19304	32-bit Float	
Channel 4 value	03	19306	32-bit Float	
Channel 5 value	03	19308	32-bit Float	
Channel 6 value	03	19310	32-bit Float	
Channel 7 value	03	19312	32-bit Float	
Channel 8 value	03	19314	32-bit Float	
Channel 9 value	03	19316	32-bit Float	
Channel 10 value	03	19318	32-bit Float	
Channel 11 value	03	19320	32-bit Float	
Channel 12 value	03	19322	32-bit Float	

Channel 13 value	03	19324	32-bit Float	
Channel 14 value	03	19326	32-bit Float	
Channel 15 value	03	19328	32-bit Float	
Channel 16 value	03	19330	32-bit Float	
Channel 17 value	03	19332	32-bit Float	
Channel 18 value	03	19334	32-bit Float	
Channel 19 value	03	19336	32-bit Float	
Channel 20 value	03	19338	32-bit Float	
Channel 21 value	03	19340	32-bit Float	
Channel 22 value	03	19342	32-bit Float	
Channel 23 value	03	19344	32-bit Float	
Channel 24 value	03	19346	32-bit Float	
Alarm 1 status	03	22000	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 2 status	03	22001	16-bit unsign int	—"—
Alarm 3 status	03	22002	16-bit unsign int	—"—
Alarm 4 status	03	22003	16-bit unsign int	—"—
Alarm 5 status	03	22004	16-bit unsign int	—"—
Alarm 6 status	03	22005	16-bit unsign int	—"—
Alarm 7 status	03	22006	16-bit unsign int	—"—
Alarm 8 status	03	22007	16-bit unsign int	—"—
Alarm 9 status	03	22008	16-bit unsign int	—"—
Alarm 10 status	03	22009	16-bit unsign int	—"—
Alarm 11 status	03	22010	16-bit unsign int	—"—
Alarm 12 status	03	22011	16-bit unsign int	—"—
Alarm 13 status	03	22012	16-bit unsign int	—"—
Alarm 14 status	03	22013	16-bit unsign int	—"—
Alarm 15 status	03	22014	16-bit unsign int	—"—
Alarm 16 status	03	22015	16-bit unsign int	—"—
Alarm 17 status	03	22016	16-bit unsign int	—"—
Alarm 18 status	03	22017	16-bit unsign int	—"—
Alarm 19 status	03	22018	16-bit unsign int	—"—
Alarm 20 status	03	22019	16-bit unsign int	—"—
Alarm 21 status	03	22020	16-bit unsign int	—"—
Alarm 22 status	03	22021	16-bit unsign int	—"—
Alarm 23 status	03	22022	16-bit unsign int	—"—
Alarm 24 status	03	22023	16-bit unsign int	—"—
Save configuration	03,06	24000	16-bit unsign int	unsaved(0), saved(1)

9.4. Modbus RTU

9.4.1 Communication parameters

For Modbus RTU, TCW242 supports the following communication parameters:

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

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

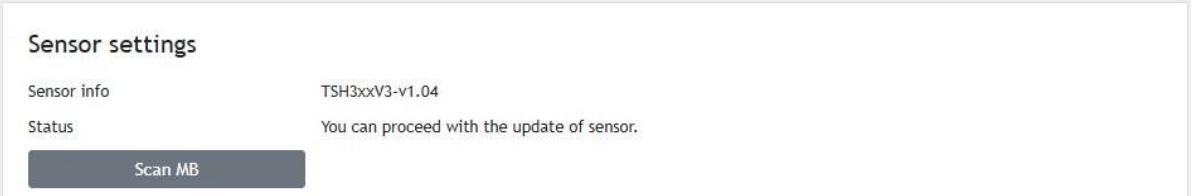
- Baud rate – 19200;
- Data bits – 8;
- Stop bits – 1;
- Parity – Even.

9.4.2 Teracom sensors update tool

The TCW242 supports the Teracom sensor firmware update tool, which is available at http://device.ip.address/teracom485.htm.

Attention! To make any changes to a Modbus RTU sensor, it should be the only device on the RS-485 bus.

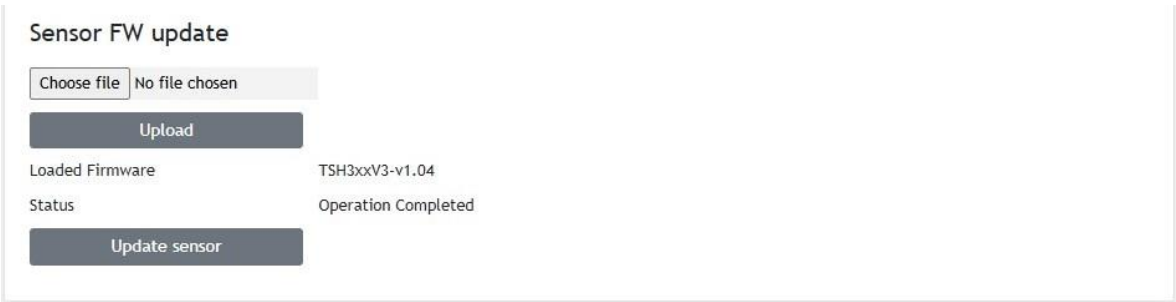
9.4.2.1 Sensor settings



The tool operates using the current Modbus RTU communication parameters. To avoid communication collisions, it is advisable to configure both the TCW242 and the sensor with the factory default Modbus RTU communication parameters. This practice ensures smooth operation. The default Modbus RTU communication parameters for the TCW242 are detailed in Section 9.4.1.

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

9.4.2.2 Sensor FW update



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

9.5. MQTT

MQTT (Message Queuing Telemetry Transport) is a lightweight, client-server publish/subscribe messaging transport protocol. It is open, simple, and designed for easy implementation. MQTT is widely used across various industries, including automotive, manufacturing, telecommunications, oil, and gas, among others.

The TCW242 supports both MQTT Publisher and MQTT Subscriber modes. The device can publish monitored values and states to an MQTT broker and receive commands for relay control and device configuration.

The official site for the MQTT protocol is mqtt.org.

For comprehensive explanations suitable for both beginners and professionals, refer to these resources:

- The Easiest Guide to Getting Started with MQTT at emqx.com
- MQTT Essentials at hivemq.com

9.6. Data Logger

The logger utilizes a circular buffer in FLASH memory. When the buffer is full, new data overwrites the oldest entries, ensuring that the FLASH memory continuously stores a complete log. There is no command to clear the log, but a copy of the full log is always available for download.

The number of records that can be stored depends on the length of the descriptions and the types of characters used. In the worst-case scenario—using 15-byte descriptions with characters from the upper range of UTF-8—the total number of records is approximately 52,371. This capacity is sufficient for 36 days of logging at one-minute intervals.

In most cases, the data logger can retain around 71,400 records, which allows for 49 days of logging at one-minute intervals.

New records can be periodically uploaded as a CSV file to a designated HTTP server at intervals of 1, 2, 3, 4, 6, 8, 12, and 24 hours. A semicolon is used as the delimiter in the CSV format.

The first row of the log file always contains the header. All rows, including the header, begin with the record ID and timestamp.

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

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

ID	32-bit unique number for every row (record).	
Time	a time stamp of record, in format dd.mm.yyyy, hh:mm:ss.	
Type of record	following types of records are available:	
	"Time"	for periodical record;
	"Event"	for record initiated by alarm condition;
	"Type"	for header record;
	"Start"	after power-up condition;
	"Restart"	after reset condition;
	"Power Down"	after power-down condition;
	"Bad"	for a problematic record
Channels - values/units	Channels 1 to 24 values/units	
Relays - conditions	Relays 1 to 4 conditions	
Channels - states/units	Channels 1 to 24 states/units	
	For chanel types General, Cumulative and Discrete following states are available:	

0 – “Undefined”
1 – “Normal”
2 – “Indeterminate”
3 – “Warning”
4 – “Minor”
5 – “Major”
6 – “Critical”

For channel type Counter following states are available:

8 – “Undefined”
9 – “Normal”
10 – “Indeterminate”
11 – “Warning”
12 – “Minor”
13 – “Major”
14 – “Critical”

Alarms - values/descriptions Alarms 1 to 24 values/descriptions

The wollowing alarm values are available:
0 – “Undefined”
1 – “Normal”
2 – “Indeterminate”
3 – “Warning”
4 – “Minor”
5 – “Major”
6 – “Critical”

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

ID;Time;Type;Ch1/°C;Ch2/%RH;Ch3;Ch4;Ch5;Ch6;Ch7;Ch8;Ch9;Ch10;Ch11;Ch12;Ch13;Ch14;Ch15;Ch16;Ch17;Ch18;Ch19;Ch20;Ch21;Ch22;Ch23;Ch24....

25114;19.07.2022,16:49:49; Time;25.319;54.512;118.833;229.877;0.000;6587.396;;;;;;;;;;;;;;;;;;;;....

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

ID;Time;Type;...R1,R2,R3,R4....

25114;19.07.2022,16:49:49; Time;1,1,0,0;;;;;;;;;;;;;;;;;;;;....

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

ID;Time;Type;....ChSt1/°C;ChSt2/%RH;ChSt3;ChSt4;ChSt5;ChSt6;ChSt7;ChSt8;ChSt9;ChSt10;ChSt11;ChSt12;ChSt13;ChSt14;ChSt15;ChSt16;ChSt17;ChSt18;ChSt19;ChSt20;ChSt21;ChSt22;ChSt23;ChSt24....

25114;19.07.2022,16:49:49;Time;....3;5;0;2;2;2;;;;;;;;;;;;;;;;;;;;....

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

ID;Time;Type;...Al1/AL1-T;Al2/AL2-H;Al3;Al4;Al5;Al6;Al7;Al8;Al9;Al10;Al11;Al12;Al13;Al14;Al15;Al16;Al17;Al18;Al19;Al20;Al21;Al22;Al23;Al24....

25114;19.07.2022,16:49:49;Time;....3;5;0;0;0;0;;;;;;;;;;;;;;;;;;

10. Factory default settings

TCW242 can be restored to its original factory default settings in three different ways.

10.1. Factory reset without network settings

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

10.2. Factory reset for network settings only

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

10.3. Full factory reset

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

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



The factory default settings are:

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

11. Operating environment

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

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

12. Safety precautions

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

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

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

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

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

To ensure that the device works correctly follow these steps:

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

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

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

13. Maintenance

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

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

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

XML file structure

```
<Monitor>
<script id="eppiocemhmnlbhjplcgkofciiegomcon"/>
<script/>
<script/>
<DeviceInfo>
<DeviceName>TCW242</DeviceName>
<HostName>TCW242-LD </HostName>
<ID>5c:32:c5:00:ac:52</ID>
<FwVer>TCW242-v3.014LD</FwVer>
<MnfInfo>www.teracomsystems.com</MnfInfo>
<SysContact>info@teracomsystems.com</SysContact>
<SysName>tcw242</SysName>
<SysLocation>Location</SysLocation>
</DeviceInfo>
<CH>
<CH1>
<type>3</type>
<description>Production [LM]</description>
<value>8</value>
<valuebin/>
<unit/>
<alarmbin>0</alarmbin>
<alarm/>
<selch>34</selch>
</CH1>
<CH2>
<type>2</type>
<description>V_DI2</description>
<value>closed</value>
<valuebin>1</valuebin>
<unit/>
<alarmbin>3</alarmbin>
<alarm>Warning</alarm>
<selch>32</selch>
</CH2>
<CH3>
<type>2</type>
<description>V_DI3</description>
<value>open</value>
<valuebin>1</valuebin>
<unit/>
<alarmbin>5</alarmbin>
<alarm>Major</alarm>
<selch>33</selch>
</CH3>
<CH4>
<type>2</type>
<description>V_DI4</description>
<value>XXXXXXXX</value>
<valuebin>0</valuebin>
<unit/>
<alarmbin>1</alarmbin>
<alarm>Normal</alarm>
<selch>31</selch>
</CH4>
<CH5>
<type>0</type>
<description>V_AI1</description>
<value>1.000</value>
<valuebin/>
<unit/>
<alarmbin>4</alarmbin>
<alarm>Minor</alarm>
<selch>27</selch>
</CH5>
<CH6>
<type>0</type>
<description>V06</description>
<value>---</value>
<valuebin/>
<unit/>
<alarmbin>2</alarmbin>
<alarm>---</alarm>
```

<selch>0</selch>
</CH6>
<CH7>
<type>0</type>
<description>V07</description>
<value>---</value>
<valuebin/>
<unit/>
<alarmbin>2</alarmbin>
<alarm>---</alarm>
<selch>0</selch>
</CH7>
<CH8>
<type>0</type>
<description>V08</description>
<value>---</value>
<valuebin/>
<unit/>
<alarmbin>2</alarmbin>
<alarm>---</alarm>
<selch>0</selch>
</CH8>
<CH9>
<type>0</type>
<description>V09</description>
<value>---</value>
<valuebin/>
<unit/>
<alarmbin>2</alarmbin>
<alarm>---</alarm>
<selch>0</selch>
</CH9>
<CH10>
<type>0</type>
<description>V10</description>
<value>---</value>
<valuebin/>
<unit/>
<alarmbin>0</alarmbin>
<alarm/>
<selch>0</selch>
</CH10>
<CH11>
<type>0</type>
<description>V11 w23</description>
<value>---</value>
<valuebin/>
<unit/>
<alarmbin>0</alarmbin>
<alarm/>
<selch>0</selch>
</CH11>
<CH12>
<type>0</type>
<description>V12</description>
<value>---</value>
<valuebin/>
<unit/>
<alarmbin>2</alarmbin>
<alarm>---</alarm>
<selch>0</selch>
</CH12>
<CH13>
<type>0</type>
<description>V13</description>
<value>---</value>
<valuebin/>
<unit/>
<alarmbin>0</alarmbin>
<alarm/>
<selch>0</selch>
</CH13>
<CH14>
<type>0</type>
<description>V14</description>
<value>---</value>

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

<CH22>
<type>0</type>
<description>V22</description>
<value>---</value>
<valuebin/>
<unit/>
<alarmbin>0</alarmbin>
<alarm/>
<selch>0</selch>
</CH22>
<CH23>
<type>0</type>
<description>V23</description>
<value>---</value>
<valuebin/>
<unit/>
<alarmbin>0</alarmbin>
<alarm/>
<selch>0</selch>
</CH23>
<CH24>
<type>0</type>
<description>V24</description>
<value>---</value>
<valuebin/>
<unit/>
<alarmbin>2</alarmbin>
<alarm>---</alarm>
<selch>0</selch>
</CH24>
</CH>
<AL>
<AL1>
<description>P1On</description>
<alarmbin>1</alarmbin>
<alarm>Normal</alarm>
<assign>1</assign>
</AL1>
<AL2>
<description>AL_DI2</description>
<alarmbin>3</alarmbin>
<alarm>Warning</alarm>
<assign>2</assign>
</AL2>
<AL3>
<description>AL_DI3</description>
<alarmbin>5</alarmbin>
<alarm>Major</alarm>
<assign>3</assign>
</AL3>
<AL4>
<description>AL_DI4</description>
<alarmbin>1</alarmbin>
<alarm>Normal</alarm>
<assign>4</assign>
</AL4>
<AL5>
<description>AL_AI1_Low</description>
<alarmbin>4</alarmbin>
<alarm>Minor</alarm>
<assign>5</assign>
</AL5>
<AL6>
<description>AL_AI1_High</description>
<alarmbin>2</alarmbin>
<alarm>---</alarm>
<assign>5</assign>
</AL6>
<AL7>
<description>AL_AI2_Low</description>
<alarmbin>2</alarmbin>
<alarm>---</alarm>
<assign>6</assign>
</AL7>
<AL8>
<description>AL_AI2_High</description>

<alarmbin>2</alarmbin>
<alarm>---</alarm>
<assign>6</assign>
</AL8>
<AL9>
<description>AL09</description>
<alarmbin>2</alarmbin>
<alarm>---</alarm>
<assign>1</assign>
</AL9>
<AL10>
<description>AL10_L</description>
<alarmbin>1</alarmbin>
<alarm>Normal</alarm>
<assign>4</assign>
</AL10>
<AL11>
<description>AL10_H</description>
<alarmbin>2</alarmbin>
<alarm>---</alarm>
<assign>4</assign>
</AL11>
<AL12>
<description>AL12_safety</description>
<alarmbin>2</alarmbin>
<alarm>---</alarm>
<assign>8</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>DO1</description>
<value>OFF</value>
<valuebin>0</valuebin>
<pulseWidth>10.0</pulseWidth>
<control>Function 1</control>
</R1>
<R2>
<description>DO2</description>
<value>ON</value>
<valuebin>1</valuebin>
<pulseWidth>10.0</pulseWidth>
<control>AL_DI2</control>
</R2>
<R3>
<description>DO3</description>
<value>OFF</value>
<valuebin>0</valuebin>
<pulseWidth>10.0</pulseWidth>
<control>Function 3</control>
</R3>
<R4>
<description>DO4</description>
<value>OFF</value>
<valuebin>0</valuebin>
<pulseWidth>1.0</pulseWidth>
<control>Watchdog3</control>
</R4>
</R>
<MBR>
<MBR1>
<description>Modbus Relay 1</description>
<value>No connection</value>
<valuebin>0</valuebin>
<control>0</control>
<conn>0</conn>
<valid>0</valid>
</MBR1>
<MBR2>
<description>Modbus Relay 2</description>
<value>No connection</value>
<valuebin>0</valuebin>
<control>0</control>
<conn>0</conn>
<valid>0</valid>
</MBR2>
<MBR3>
<description>Modbus Relay 3</description>
<value>No connection</value>
<valuebin>0</valuebin>

<control>0</control>
<conn>0</conn>
<valid>0</valid>
</MBR3>
<MBR4>
<description>Modbus Relay 4</description>
<value>No connection</value>
<valuebin>0</valuebin>
<control>0</control>
<conn>0</conn>
<valid>0</valid>
</MBR4>
<MBR5>
<description>Modbus Relay 5</description>
<value>No connection</value>
<valuebin>0</valuebin>
<control>0</control>
<conn>0</conn>
<valid>0</valid>
</MBR5>
<MBR6>
<description>Modbus Relay 6</description>
<value>No connection</value>
<valuebin>0</valuebin>
<control>0</control>
<conn>0</conn>
<valid>0</valid>
</MBR6>
<MBR7>
<description>Modbus Relay 7</description>
<value>No connection</value>
<valuebin>0</valuebin>
<control>0</control>
<conn>0</conn>
<valid>0</valid>
</MBR7>
<MBR8>
<description>Modbus Relay 8</description>
<value>No connection</value>
<valuebin>0</valuebin>
<control>0</control>
<conn>0</conn>
<valid>0</valid>
</MBR8>
<MBR9>
<description>Modbus Relay 9</description>
<value>No connection</value>
<valuebin>0</valuebin>
<control>0</control>
<conn>0</conn>
<valid>0</valid>
</MBR9>
<MBR10>
<description>Modbus Relay 10</description>
<value>No connection</value>
<valuebin>0</valuebin>
<control>0</control>
<conn>0</conn>
<valid>0</valid>
</MBR10>
<MBR11>
<description>Modbus Relay 11</description>
<value>No connection</value>
<valuebin>0</valuebin>
<control>0</control>
<conn>0</conn>
<valid>0</valid>
</MBR11>
<MBR12>
<description>Modbus Relay 12</description>
<value>No connection</value>
<valuebin>0</valuebin>
<control>0</control>
<conn>0</conn>
<valid>0</valid>
</MBR12>

<MBR13>
<description>Modbus Relay 13</description>
<value>No connection</value>
<valuebin>0</valuebin>
<control>0</control>
<conn>0</conn>
<valid>0</valid>
</MBR13>
<MBR14>
<description>Modbus Relay 14</description>
<value>No connection</value>
<valuebin>0</valuebin>
<control>0</control>
<conn>0</conn>
<valid>0</valid>
</MBR14>
<MBR15>
<description>Modbus Relay 15</description>
<value>No connection</value>
<valuebin>0</valuebin>
<control>0</control>
<conn>0</conn>
<valid>0</valid>
</MBR15>
<MBR16>
<description>Modbus Relay 16</description>
<value>No connection</value>
<valuebin>0</valuebin>
<control>0</control>
<conn>0</conn>
<valid>0</valid>
</MBR16>
<MBR17>
<description>Modbus Relay 17</description>
<value>No connection</value>
<valuebin>0</valuebin>
<control>0</control>
<conn>0</conn>
<valid>0</valid>
</MBR17>
<MBR18>
<description>Modbus Relay 18</description>
<value>No connection</value>
<valuebin>0</valuebin>
<control>0</control>
<conn>0</conn>
<valid>0</valid>
</MBR18>
<MBR19>
<description>Modbus Relay 19</description>
<value>No connection</value>
<valuebin>0</valuebin>
<control>0</control>
<conn>0</conn>
<valid>0</valid>
</MBR19>
<MBR20>
<description>Modbus Relay 20</description>
<value>No connection</value>
<valuebin>0</valuebin>
<control>0</control>
<conn>0</conn>
<valid>0</valid>
</MBR20>
<MBR21>
<description>Modbus Relay 21</description>
<value>No connection</value>
<valuebin>0</valuebin>
<control>0</control>
<conn>0</conn>
<valid>0</valid>
</MBR21>
<MBR22>
<description>Modbus Relay 22</description>
<value>No connection</value>
<valuebin>0</valuebin>

<control>0</control>
<conn>0</conn>
<valid>0</valid>
</MBR22>
<MBR23>
<description>Modbus Relay 23</description>
<value>No connection</value>
<valuebin>0</valuebin>
<control>0</control>
<conn>0</conn>
<valid>0</valid>
</MBR23>
<MBR24>
<description>Modbus Relay 24</description>
<value>No connection</value>
<valuebin>0</valuebin>
<control>0</control>
<conn>0</conn>
<valid>0</valid>
</MBR24>
</MBR>
<FUNC>
<FUNC1>
<alarmbin>0</alarmbin>
<alarm>Normal</alarm>
</FUNC1>
<FUNC2>
<alarmbin>0</alarmbin>
<alarm>Normal</alarm>
</FUNC2>
<FUNC3>
<alarmbin>0</alarmbin>
<alarm>Normal</alarm>
</FUNC3>
<FUNC4>
<alarmbin>0</alarmbin>
<alarm>Normal</alarm>
</FUNC4>
</FUNC>
<WDG>
<WDG1>
<description>Watchdog1</description>
<modebin>0</modebin>
<mode/>
<host>0.0.0.0</host>
<state>0</state>
</WDG1>
<WDG2>
<description>Watchdog2</description>
<modebin>0</modebin>
<mode/>
<host>0.0.0.0</host>
<state>0</state>
</WDG2>
<WDG3>
<description>Watchdog3</description>
<modebin>0</modebin>
<mode/>
<host>0.0.0.0</host>
<state>0</state>
</WDG3>
<WDG4>
<description>Watchdog4</description>
<modebin>0</modebin>
<mode/>
<host>0.0.0.0</host>
<state>0</state>
</WDG4>
</WDG>
<HTTPPost>
<Key/>
<PostPeriod>60</PostPeriod>
</HTTPPost>
<MQTT>
<Period>65</Period>
</MQTT>

<Sys>
<hwerr>Changes will not take effect until the controller is restarted.</hwerr>
<HighAlarmbin>5</HighAlarmbin>
<HighAlarm>Major</HighAlarm>
</Sys>
<Time>
<Date>04.06.2026</Date>
<Time>13:50:13</Time>
</Time>
</Monitor>

Where:

<CH1>… <CH24> - channels;
<AL1> … <AL24> - alarms;
<R1> … <R4> - relays
<MBR1> … <MBR24> - MODBUS relays
<FUNC1> … <FUNC4> - functions
<WDG1> … <WDG4> - watchdogs
<valuebin> - number values 0 or 1;
<alarmbin> - number values from 0 to 6;
<alarm> - Undefined, Normal, Indeterminate, Warning, Minor, Major, Critical;
<assign>0</assign> - alarm not assigned to any channel;
<selch>0</selch> - channel is not displayed on Monitoring -> Channels section
<state>1</state> - watchdog is activated

JSON file structure


```
{
 "Monitor": {
  "DeviceInfo": {
   "DeviceName": "TCW242",
   "HostName": "TCW242-LD     ",
   "ID": "5c:32:c5:00:ac:52",
   "FwVer": "TCW242-v3.014LD",
   "MnfInfo": "www.teracomsystems.com",
   "SysContact": "info@teracomsystems.com",
   "SysName": "tcw242",
   "SysLocation": "Location"
  },
  "CH": {
   "CH1": {
    "type": "3",
    "description": "Production [LM]",
    "value": "8",
    "valuebin": "",
    "unit": "",
    "alarmbin": "0",
    "alarm": "",
    "selch": "34"
   },
   "CH2": {
    "type": "2",
    "description": "V_DI2",
    "value": "closed",
    "valuebin": "1",
    "unit": "",
    "alarmbin": "3",
    "alarm": "Warning",
    "selch": "32"
   },
   "CH3": {
    "type": "2",
    "description": "V_DI3",
    "value": "open",
    "valuebin": "1",
    "unit": "",
    "alarmbin": "5",
    "alarm": "Major",
    "selch": "33"
   },
   "CH4": {
    "type": "2",
    "description": "V_DI4",
    "value": "XXXXXXXX",
    "valuebin": "0",
    "unit": "",
    "alarmbin": "1",
    "alarm": "Normal",
    "selch": "31"
   },
   "CH5": {
    "type": "0",
    "description": "V_AI1",
    "value": "1.000",
    "valuebin": "",
    "unit": "",
    "alarmbin": "4",
    "alarm": "Minor",
    "selch": "27"
   },
   "CH6": {
    "type": "0",
    "description": "V06",
    "value": "---",
    "valuebin": "",
    "unit": "",
    "alarmbin": "2",
    "alarm": "---",
    "selch": "0"
   },
```

```
"CH7": {
 "type": "0",
 "description": "V07",
 "value": "---",
 "valuebin": "",
 "unit": "",
 "alarmbin": "2",
 "alarm": "---",
 "selch": "0"
},
"CH8": {
 "type": "0",
 "description": "V08",
 "value": "---",
 "valuebin": "",
 "unit": "",
 "alarmbin": "2",
 "alarm": "---",
 "selch": "0"
},
"CH9": {
 "type": "0",
 "description": "V09",
 "value": "---",
 "valuebin": "",
 "unit": "",
 "alarmbin": "2",
 "alarm": "---",
 "selch": "0"
},
"CH10": {
 "type": "0",
 "description": "V10",
 "value": "---",
 "valuebin": "",
 "unit": "",
 "alarmbin": "0",
 "alarm": "",
 "selch": "0"
},
"CH11": {
 "type": "0",
 "description": "V11 w23",
 "value": "---",
 "valuebin": "",
 "unit": "",
 "alarmbin": "0",
 "alarm": "",
 "selch": "0"
},
"CH12": {
 "type": "0",
 "description": "V12",
 "value": "---",
 "valuebin": "",
 "unit": "",
 "alarmbin": "2",
 "alarm": "---",
 "selch": "0"
},
"CH13": {
 "type": "0",
 "description": "V13",
 "value": "---",
 "valuebin": "",
 "unit": "",
 "alarmbin": "0",
 "alarm": "",
 "selch": "0"
},
"CH14": {
 "type": "0",
 "description": "V14",
 "value": "---",
 "valuebin": "",
 "unit": "",
```

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

```
   "description": "V22",
   "value": "---",
   "valuebin": "",
   "unit": "",
   "alarmbin": "0",
   "alarm": "",
   "selch": "0"
  },
  "CH23": {
   "type": "0",
   "description": "V23",
   "value": "---",
   "valuebin": "",
   "unit": "",
   "alarmbin": "0",
   "alarm": "",
   "selch": "0"
  },
  "CH24": {
   "type": "0",
   "description": "V24",
   "value": "---",
   "valuebin": "",
   "unit": "",
   "alarmbin": "2",
   "alarm": "---",
   "selch": "0"
  }
 },
 "AL": {
  "AL1": {
   "description": "P1On",
   "alarmbin": "1",
   "alarm": "Normal",
   "assign": "1"
  },
  "AL2": {
   "description": "AL_DI2",
   "alarmbin": "3",
   "alarm": "Warning",
   "assign": "2"
  },
  "AL3": {
   "description": "AL_DI3",
   "alarmbin": "5",
   "alarm": "Major",
   "assign": "3"
  },
  "AL4": {
   "description": "AL_DI4",
   "alarmbin": "1",
   "alarm": "Normal",
   "assign": "4"
  },
  "AL5": {
   "description": "AL_AI1_Low",
   "alarmbin": "4",
   "alarm": "Minor",
   "assign": "5"
  },
  "AL6": {
   "description": "AL_AI1_High",
   "alarmbin": "2",
   "alarm": "---",
   "assign": "5"
  },
  "AL7": {
   "description": "AL_AI2_Low",
   "alarmbin": "2",
   "alarm": "---",
   "assign": "6"
  },
  "AL8": {
   "description": "AL_AI2_High",
   "alarmbin": "2",
   "alarm": "---",
```

```
  "assign": "6"
 },
 "AL9": {
  "description": "AL09",
  "alarmbin": "2",
  "alarm": "---",
  "assign": "1"
 },
 "AL10": {
  "description": "AL10_L",
  "alarmbin": "1",
  "alarm": "Normal",
  "assign": "4"
 },
 "AL11": {
  "description": "AL10_H",
  "alarmbin": "2",
  "alarm": "---",
  "assign": "4"
 },
 "AL12": {
  "description": "AL12_safety",
  "alarmbin": "2",
  "alarm": "---",
  "assign": "8"
 },
 "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": "DO1",
   "value": "OFF",
   "valuebin": "0",
   "pulseWidth": "10.0",
   "control": "Function 1"
  },
  "R2": {
   "description": "DO2",
   "value": "ON",
   "valuebin": "1",
   "pulseWidth": "10.0",
   "control": "AL_DI2"
  },
  "R3": {
   "description": "DO3",
   "value": "OFF",
   "valuebin": "0",
   "pulseWidth": "10.0",
   "control": "Function 3"
  },
  "R4": {
   "description": "DO4",
   "value": "OFF",
   "valuebin": "0",
   "pulseWidth": "1.0",
   "control": "Watchdog3"
  }
 },
 "MBR": {
  "MBR1": {
   "description": "Modbus Relay 1",
   "value": "No connection",
   "valuebin": "0",
   "control": "0",
   "conn": "0",
   "valid": "0"
  },
  "MBR2": {
   "description": "Modbus Relay  2",
   "value": "No connection",
   "valuebin": "0",
   "control": "0",
   "conn": "0",
   "valid": "0"
  },
  "MBR3": {
   "description": "Modbus Relay  3",
   "value": "No connection",
   "valuebin": "0",
   "control": "0",
   "conn": "0",
```

```
 "valid": "0"
},
"MBR4": {
 "description": "Modbus Relay  4",
 "value": "No connection",
 "valuebin": "0",
 "control": "0",
 "conn": "0",
 "valid": "0"
},
"MBR5": {
 "description": "Modbus Relay  5",
 "value": "No connection",
 "valuebin": "0",
 "control": "0",
 "conn": "0",
 "valid": "0"
},
"MBR6": {
 "description": "Modbus Relay  6",
 "value": "No connection",
 "valuebin": "0",
 "control": "0",
 "conn": "0",
 "valid": "0"
},
"MBR7": {
 "description": "Modbus Relay  7",
 "value": "No connection",
 "valuebin": "0",
 "control": "0",
 "conn": "0",
 "valid": "0"
},
"MBR8": {
 "description": "Modbus Relay  8",
 "value": "No connection",
 "valuebin": "0",
 "control": "0",
 "conn": "0",
 "valid": "0"
},
"MBR9": {
 "description": "Modbus Relay  9",
 "value": "No connection",
 "valuebin": "0",
 "control": "0",
 "conn": "0",
 "valid": "0"
},
"MBR10": {
 "description": "Modbus Relay  10",
 "value": "No connection",
 "valuebin": "0",
 "control": "0",
 "conn": "0",
 "valid": "0"
},
"MBR11": {
 "description": "Modbus Relay  11",
 "value": "No connection",
 "valuebin": "0",
 "control": "0",
 "conn": "0",
 "valid": "0"
},
"MBR12": {
 "description": "Modbus Relay  12",
 "value": "No connection",
 "valuebin": "0",
 "control": "0",
 "conn": "0",
 "valid": "0"
},
"MBR13": {
 "description": "Modbus Relay  13",
```

```
 "value": "No connection",
 "valuebin": "0",
 "control": "0",
 "conn": "0",
 "valid": "0"
},
"MBR14": {
 "description": "Modbus Relay  14",
 "value": "No connection",
 "valuebin": "0",
 "control": "0",
 "conn": "0",
 "valid": "0"
},
"MBR15": {
 "description": "Modbus Relay  15",
 "value": "No connection",
 "valuebin": "0",
 "control": "0",
 "conn": "0",
 "valid": "0"
},
"MBR16": {
 "description": "Modbus Relay  16",
 "value": "No connection",
 "valuebin": "0",
 "control": "0",
 "conn": "0",
 "valid": "0"
},
"MBR17": {
 "description": "Modbus Relay  17",
 "value": "No connection",
 "valuebin": "0",
 "control": "0",
 "conn": "0",
 "valid": "0"
},
"MBR18": {
 "description": "Modbus Relay  18",
 "value": "No connection",
 "valuebin": "0",
 "control": "0",
 "conn": "0",
 "valid": "0"
},
"MBR19": {
 "description": "Modbus Relay  19",
 "value": "No connection",
 "valuebin": "0",
 "control": "0",
 "conn": "0",
 "valid": "0"
},
"MBR20": {
 "description": "Modbus Relay  20",
 "value": "No connection",
 "valuebin": "0",
 "control": "0",
 "conn": "0",
 "valid": "0"
},
"MBR21": {
 "description": "Modbus Relay  21",
 "value": "No connection",
 "valuebin": "0",
 "control": "0",
 "conn": "0",
 "valid": "0"
},
"MBR22": {
 "description": "Modbus Relay  22",
 "value": "No connection",
 "valuebin": "0",
 "control": "0",
 "conn": "0",
```

```
   "valid": "0"
  },
  "MBR23": {
   "description": "Modbus Relay  23",
   "value": "No connection",
   "valuebin": "0",
   "control": "0",
   "conn": "0",
   "valid": "0"
  },
  "MBR24": {
   "description": "Modbus Relay  24",
   "value": "No connection",
   "valuebin": "0",
   "control": "0",
   "conn": "0",
   "valid": "0"
  }
 },
 "FUNC": {
  "FUNC1": {
   "alarmbin": "0",
   "alarm": "Normal"
  },
  "FUNC2": {
   "alarmbin": "0",
   "alarm": "Normal"
  },
  "FUNC3": {
   "alarmbin": "0",
   "alarm": "Normal"
  },
  "FUNC4": {
   "alarmbin": "0",
   "alarm": "Normal"
  }
 },
 "WDG": {
  "WDG1": {
   "description": "Watchdog1",
   "modebin": "0",
   "mode": "",
   "host": "0.0.0.0",
   "state": "0"
  },
  "WDG2": {
   "description": "Watchdog2",
   "modebin": "0",
   "mode": "",
   "host": "0.0.0.0",
   "state": "0"
  },
  "WDG3": {
   "description": "Watchdog3",
   "modebin": "0",
   "mode": "",
   "host": "0.0.0.0",
   "state": "0"
  },
  "WDG4": {
   "description": "Watchdog4",
   "modebin": "0",
   "mode": "",
   "host": "0.0.0.0",
   "state": "0"
  }
 },
 "HTTPPost": {
  "Key": "",
  "PostPeriod": "60"
 },
 "MQTT": {
  "Period": "65"
 },
 "Sys": {
  "hwerr": "Changes will not take effect until the controller is restarted.",
```

```
    "HighAlarmbin": "5",
    "HighAlarm": "Major"
   },
   "Time": {
    "Date": "04.06.2026",
    "Time": "13:51:17"
   }
  }
}
```

Modbus TCP/IP full address table

Parameter	FC	PDU decimal address	Data size	Data
Relay 1	01,05,15	100	Discrete	
Relay 2	01,05,15	101	Discrete	
Relay 3	01,05,15	102	Discrete	
Relay 4	01,05,15	103	Discrete	
MB relay 1	01,05	200	Discrete	See Note 1 after the table.
MB relay 2	01,05	201	Discrete	—"—
MB relay 3	01,05	202	Discrete	—"—
MB relay 4	01,05	203	Discrete	—"—
MB relay 5	01,05	204	Discrete	—"—
MB relay 6	01,05	205	Discrete	—"—
MB relay 7	01,05	206	Discrete	—"—
MB relay 8	01,05	207	Discrete	—"—
MB relay 9	01,05	208	Discrete	—"—
MB relay 10	01,05	209	Discrete	—"—
MB relay 11	01,05	210	Discrete	—"—
MB relay 12	01,05	211	Discrete	—"—
MB relay 13	01,05	212	Discrete	—"—
MB relay 14	01,05	213	Discrete	—"—
MB relay 15	01,05	214	Discrete	—"—
MB relay 16	01,05	215	Discrete	—"—
MB relay 17	01,05	216	Discrete	—"—
MB relay 18	01,05	217	Discrete	—"—
MB relay 19	01,05	218	Discrete	—"—
MB relay 20	01,05	219	Discrete	—"—
MB relay 21	01,05	220	Discrete	—"—
MB relay 22	01,05	221	Discrete	—"—
MB relay 23	01,05	222	Discrete	—"—
MB relay 24	01,05	223	Discrete	—"—
Digital input 1	02	100	Discrete	
Digital input 2	02	101	Discrete	
Digital input 3	02	102	Discrete	
Digital input 4	02	103	Discrete	
Relay 1 description	03,16	10000	16 bytes UTF-8	
Relay 2 description	03,16	10008	16 bytes UTF-8	
Relay 3 description	03,16	10016	16 bytes UTF-8	
Relay 4 description	03,16	10024	16 bytes UTF-8	
Relay 1 pulse width	03,16	10100	32-bit Float	

Relay 2 pulse width	03,16	10102	32-bit Float	
Relay 3 pulse width	03,16	10104	32-bit Float	
Relay 4 pulse width	03,16	10106	32-bit Float	
Relay 1 activated from	03,06,16	10200	16-bit unsign int	webHttpApi(0), al01(1), al02(2), al03(3), al04(4), al05(5), al06(6), al07(7), al08(8), al09(9), al10(10), al11(11), al12(12), al13(13), al14(14), al15(15), al16(16), al17(17), al18(18), al19(19), al20(20), al21(21), al22(22), al23(23), al24(24), anyAlarm(25), schedule1(26), schedule2(27), schedule3(28), schedule4(29)
Relay 2 activated from	03,06,16	10201	16-bit unsign int	—"—
Relay 3 activated from	03,06,16	10202	16-bit unsign int	—"—
Relay 4 activated from	03,06,16	10203	16-bit unsign int	—"—
Relay 1 action on alarm	03,06,16	10300	16-bit unsign int	on(0), pulse(2)
Relay 2 action on alarm	03,06,16	10301	16-bit unsign int	on(0), pulse(2)
Relay 3 action on alarm	03,06,16	10302	16-bit unsign int	on(0), pulse(2)
Relay 4 action on alarm	03,06,16	10303	16-bit unsign int	on(0), pulse(2)
Relays state after restart	03,06	10400	16-bit unsign int	off(0), on(1), laststate(2)
Digital input 1 Description	03,16	12000	16 bytes UTF-8	
Digital input 2 Description	03,16	12008	16 bytes UTF-8	
Digital input 3 Description	03,16	12016	16 bytes UTF-8	
Digital input 4 Description	03,16	12024	16 bytes UTF-8	
Digital input 1 closed state description	03,16	12100	16 bytes UTF-8	
Digital input 2 closed state description	03,16	12108	16 bytes UTF-8	
Digital input 3 closed state description	03,16	12116	16 bytes UTF-8	
Digital input 4 closed state description	03,16	12124	16 bytes UTF-8	
Digital input 1 open state description	03,16	12200	16 bytes UTF-8	
Digital input 2 open state description	03,16	12208	16 bytes UTF-8	
Digital input 3 open state description	03,16	12216	16 bytes UTF-8	

Digital input 4 open state description	03,16	12224	16 bytes UTF-8	
Digital input 1 mode	03,06,16	12300	16-bit unsign int	openclosed(0), risingEdge(1), fallingEdge(2), bothEdges(3)
Digital input 2 mode	03,06,16	12301	16-bit unsign int	—"—
Digital input 3 mode	03,06,16	12302	16-bit unsign int	—"—
Digital input 4 mode	03,06,16	12303	16-bit unsign int	—"—
Digital input 1 close to open delay	03,16	12400	32-bit unsign int	
Digital input 2 close to open delay	03,16	12402	32-bit unsign int	
Digital input 3 close to open delay	03,16	12404	32-bit unsign int	
Digital input 4 close to open delay	03,16	12406	32-bit unsign int	
Digital input 1 open to close delay	03,16	12500	32-bit unsign int	
Digital input 2 open to close delay	03,16	12502	32-bit unsign int	
Digital input 3 open to close delay	03,16	12504	32-bit unsign int	
Digital input 4 open to close delay	03,16	12506	32-bit unsign int	
Digital input 1 counter init value	03,16	12600	32-bit unsign int	
Digital input 2 counter init value	03,16	12602	32-bit unsign int	
Digital input 3 counter init value	03,16	12604	32-bit unsign int	
Digital input 4 counter init value	03,16	12606	32-bit unsign int	
Digital input 1 counter value	03	12700	32-bit unsign int	
Digital input 2 counter value	03	12702	32-bit unsign int	
Digital input 3 counter value	03	12704	32-bit unsign int	
Digital input 4 counter value	03	12706	32-bit unsign int	
Analog input 1 description	03,16	14000	16 bytes UTF-8	
Analog input 2 description	03,16	14008	16 bytes UTF-8	
Analog input 3 description	03,16	14016	16 bytes UTF-8	
Analog input 4 description	03,16	14024	16 bytes UTF-8	
Analog input 1 multiplier	03,16	14100	32-bit Float	
Analog input 2 multiplier	03,16	14102	32-bit Float	
Analog input 3 multiplier	03,16	14104	32-bit Float	
Analog input 4 multiplier	03,16	14106	32-bit Float	
Analog input 1 offset	03,16	14200	32-bit Float	
Analog input 2 offset	03,16	14202	32-bit Float	
Analog input 3 offset	03,16	14204	32-bit Float	
Analog input 4 offset	03,16	14206	32-bit Float	
Analog input 1 mode(V/mA)	03,06,16	14300	16-bit unsign int	0-10V(0), 4-20mA(1)
Analog input 2 mode(V/mA)	03,06,16	14301	16-bit unsign int	0-10V(0), 4-20mA(1)
Analog input 3 mode(V/mA)	03,06,16	14302	16-bit unsign int	0-10V(0), 4-20mA(1)
Analog input 4 mode(V/mA)	03,06,16	14303	16-bit unsign int	0-10V(0), 4-20mA(1)

Analog input 1 value	03	14400	32-bit Float	
Analog input 2 value	03	14402	32-bit Float	
Analog input 3 value	03	14404	32-bit Float	
Analog input 4 value	03	14406	32-bit Float	
MB sensor 1 description	03,16	16000	16 bytes UTF-8	
MB sensor 2 description	03,16	16008	16 bytes UTF-8	
MB sensor 3 description	03,16	16016	16 bytes UTF-8	
MB sensor 4 description	03,16	16024	16 bytes UTF-8	
MB sensor 5 description	03,16	16032	16 bytes UTF-8	
MB sensor 6 description	03,16	16040	16 bytes UTF-8	
MB sensor 7 description	03,16	16048	16 bytes UTF-8	
MB sensor 8 description	03,16	16056	16 bytes UTF-8	
MB sensor 9 description	03,16	16064	16 bytes UTF-8	
MB sensor 10 description	03,16	16072	16 bytes UTF-8	
MB sensor 11 description	03,16	16080	16 bytes UTF-8	
MB sensor 12 description	03,16	16088	16 bytes UTF-8	
MB sensor 13 description	03,16	16096	16 bytes UTF-8	
MB sensor 14 description	03,16	16104	16 bytes UTF-8	
MB sensor 15 description	03,16	16112	16 bytes UTF-8	
MB sensor 16 description	03,16	16120	16 bytes UTF-8	
MB sensor 17 description	03,16	16128	16 bytes UTF-8	
MB sensor 18 description	03,16	16136	16 bytes UTF-8	
MB sensor 19 description	03,16	16144	16 bytes UTF-8	
MB sensor 20 description	03,16	16152	16 bytes UTF-8	
MB sensor 21 description	03,16	16160	16 bytes UTF-8	
MB sensor 22 description	03,16	16168	16 bytes UTF-8	
MB sensor 23 description	03,16	16176	16 bytes UTF-8	
MB sensor 24 description	03,16	16184	16 bytes UTF-8	
MB sensor 1 multiplier	03,16	16200	32-bit Float	
MB sensor 2 multiplier	03,16	16202	32-bit Float	
MB sensor 3 multiplier	03,16	16204	32-bit Float	
MB sensor 4 multiplier	03,16	16206	32-bit Float	
MB sensor 5 multiplier	03,16	16208	32-bit Float	
MB sensor 6 multiplier	03,16	16210	32-bit Float	
MB sensor 7 multiplier	03,16	16212	32-bit Float	
MB sensor 8 multiplier	03,16	16214	32-bit Float	
MB sensor 9 multiplier	03,16	16216	32-bit Float	
MB sensor 10 multiplier	03,16	16218	32-bit Float	
MB sensor 11 multiplier	03,16	16220	32-bit Float	
MB sensor 12 multiplier	03,16	16222	32-bit Float	
MB sensor 13 multiplier	03,16	16224	32-bit Float	
MB sensor 14 multiplier	03,16	16226	32-bit Float	
MB sensor 15 multiplier	03,16	16228	32-bit Float	
MB sensor 16 multiplier	03,16	16230	32-bit Float	
MB sensor 17 multiplier	03,16	16232	32-bit Float	

MB sensor 18 multiplier	03,16	16234	32-bit Float	
MB sensor 19 multiplier	03,16	16236	32-bit Float	
MB sensor 20 multiplier	03,16	16238	32-bit Float	
MB sensor 21 multiplier	03,16	16240	32-bit Float	
MB sensor 22 multiplier	03,16	16242	32-bit Float	
MB sensor 23 multiplier	03,16	16244	32-bit Float	
MB sensor 24 multiplier	03,16	16246	32-bit Float	
MB sensor 1 offset	03,16	16300	32-bit Float	
MB sensor 2 offset	03,16	16302	32-bit Float	
MB sensor 3 offset	03,16	16304	32-bit Float	
MB sensor 4 offset	03,16	16306	32-bit Float	
MB sensor 5 offset	03,16	16308	32-bit Float	
MB sensor 6 offset	03,16	16310	32-bit Float	
MB sensor 7 offset	03,16	16312	32-bit Float	
MB sensor 8 offset	03,16	16314	32-bit Float	
MB sensor 9 offset	03,16	16316	32-bit Float	
MB sensor 10 offset	03,16	16318	32-bit Float	
MB sensor 11 offset	03,16	16320	32-bit Float	
MB sensor 12 offset	03,16	16322	32-bit Float	
MB sensor 13 offset	03,16	16324	32-bit Float	
MB sensor 14 offset	03,16	16326	32-bit Float	
MB sensor 15 offset	03,16	16328	32-bit Float	
MB sensor 16 offset	03,16	16330	32-bit Float	
MB sensor 17 offset	03,16	16332	32-bit Float	
MB sensor 18 offset	03,16	16334	32-bit Float	
MB sensor 19 offset	03,16	16336	32-bit Float	
MB sensor 20 offset	03,16	16338	32-bit Float	
MB sensor 21 offset	03,16	16340	32-bit Float	
MB sensor 22 offset	03,16	16342	32-bit Float	
MB sensor 23 offset	03,16	16344	32-bit Float	
MB sensor 24 offset	03,16	16346	32-bit Float	
MB sensor 1 value	03	16400	32-bit Float	
MB sensor 2 value	03	16402	32-bit Float	
MB sensor 3 value	03	16404	32-bit Float	
MB sensor 4 value	03	16406	32-bit Float	
MB sensor 5 value	03	16408	32-bit Float	
MB sensor 6 value	03	16410	32-bit Float	
MB sensor 7 value	03	16412	32-bit Float	
MB sensor 8 value	03	16414	32-bit Float	
MB sensor 9 value	03	16416	32-bit Float	
MB sensor 10 value	03	16418	32-bit Float	
MB sensor 11 value	03	16420	32-bit Float	
MB sensor 12 value	03	16422	32-bit Float	
MB sensor 13 value	03	16424	32-bit Float	
MB sensor 14 value	03	16426	32-bit Float	
MB sensor 15 value	03	16428	32-bit Float	

MB sensor 16 value	03	16430	32-bit Float	
MB sensor 17 value	03	16432	32-bit Float	
MB sensor 18 value	03	16434	32-bit Float	
MB sensor 19 value	03	16436	32-bit Float	
MB sensor 20 value	03	16438	32-bit Float	
MB sensor 21 value	03	16440	32-bit Float	
MB sensor 22 value	03	16442	32-bit Float	
MB sensor 23 value	03	16444	32-bit Float	
MB sensor 24 value	03	16446	32-bit Float	
MB sensor 1 counter	03	16500	32-bit unsign int	
MB sensor 2 counter	03	16502	32-bit unsign int	
MB sensor 3 counter	03	16504	32-bit unsign int	
MB sensor 4 counter	03	16506	32-bit unsign int	
MB sensor 5 counter	03	16508	32-bit unsign int	
MB sensor 6 counter	03	16510	32-bit unsign int	
MB sensor 7 counter	03	16512	32-bit unsign int	
MB sensor 8 counter	03	16514	32-bit unsign int	
MB sensor 9 counter	03	16516	32-bit unsign int	
MB sensor 10 counter	03	16518	32-bit unsign int	
MB sensor 11 counter	03	16520	32-bit unsign int	
MB sensor 12 counter	03	16522	32-bit unsign int	
MB sensor 13 counter	03	16524	32-bit unsign int	
MB sensor 14 counter	03	16526	32-bit unsign int	
MB sensor 15 counter	03	16528	32-bit unsign int	
MB sensor 16 counter	03	16530	32-bit unsign int	
MB sensor 17 counter	03	16532	32-bit unsign int	
MB sensor 18 counter	03	16534	32-bit unsign int	
MB sensor 19 counter	03	16536	32-bit unsign int	
MB sensor 20 counter	03	16538	32-bit unsign int	
MB sensor 21 counter	03	16540	32-bit unsign int	
MB sensor 22 counter	03	16542	32-bit unsign int	
MB sensor 23 counter	03	16544	32-bit unsign int	
MB sensor 24 counter	03	16546	32-bit unsign int	
Channel 1 type	03,06,16	18000	16-bit unsign int	general(0), discrete(2) , counter(3)
Channel 2 type	03,06,16	18001	16-bit unsign int	—"—
Channel 3 type	03,06,16	18002	16-bit unsign int	—"—
Channel 4 type	03,06,16	18003	16-bit unsign int	—"—
Channel 5 type	03,06,16	18004	16-bit unsign int	—"—
Channel 6 type	03,06,16	18005	16-bit unsign int	—"—
Channel 7 type	03,06,16	18006	16-bit unsign int	—"—
Channel 8 type	03,06,16	18007	16-bit unsign int	—"—
Channel 9 type	03,06,16	18008	16-bit unsign int	—"—
Channel 10 type	03,06,16	18009	16-bit unsign int	—"—
Channel 11 type	03,06,16	18010	16-bit unsign int	—"—
Channel 12 type	03,06,16	18011	16-bit unsign int	—"—

Channel 13 type	03,06,16	18012	16-bit unsign int	—"—
Channel 14 type	03,06,16	18013	16-bit unsign int	—"—
Channel 15 type	03,06,16	18014	16-bit unsign int	—"—
Channel 16 type	03,06,16	18015	16-bit unsign int	—"—
Channel 17 type	03,06,16	18016	16-bit unsign int	—"—
Channel 18 type	03,06,16	18017	16-bit unsign int	—"—
Channel 19 type	03,06,16	18018	16-bit unsign int	—"—
Channel 20 type	03,06,16	18019	16-bit unsign int	—"—
Channel 21 type	03,06,16	18020	16-bit unsign int	—"—
Channel 22 type	03,06,16	18021	16-bit unsign int	—"—
Channel 23 type	03,06,16	18022	16-bit unsign int	—"—
Channel 24 type	03,06,16	18023	16-bit unsign int	—"—
Channel 1 description	03,16	18100	16 bytes UTF-8	
Channel 2 description	03,16	18108	16 bytes UTF-8	
Channel 3 description	03,16	18116	16 bytes UTF-8	
Channel 4 description	03,16	18124	16 bytes UTF-8	
Channel 5 description	03,16	18132	16 bytes UTF-8	
Channel 6 description	03,16	18140	16 bytes UTF-8	
Channel 7 description	03,16	18148	16 bytes UTF-8	
Channel 8 description	03,16	18156	16 bytes UTF-8	
Channel 9 description	03,16	18164	16 bytes UTF-8	
Channel 10 description	03,16	18172	16 bytes UTF-8	
Channel 11 description	03,16	18180	16 bytes UTF-8	
Channel 12 description	03,16	18188	16 bytes UTF-8	
Channel 13 description	03,16	18196	16 bytes UTF-8	
Channel 14 description	03,16	18204	16 bytes UTF-8	
Channel 15 description	03,16	18212	16 bytes UTF-8	
Channel 16 description	03,16	18220	16 bytes UTF-8	
Channel 17 description	03,16	18228	16 bytes UTF-8	
Channel 18 description	03,16	18236	16 bytes UTF-8	
Channel 19 description	03,16	18244	16 bytes UTF-8	
Channel 20 description	03,16	18252	16 bytes UTF-8	
Channel 21 description	03,16	18260	16 bytes UTF-8	
Channel 22 description	03,16	18268	16 bytes UTF-8	
Channel 23 description	03,16	18276	16 bytes UTF-8	
Channel 24 description	03,16	18284	16 bytes UTF-8	
Channel 1 parameter 1	03,06,16	18300	16-bit unsign int	none(0), s01(3), s02(4), s03(5), s04(6), s05(7), s06(8), s07(9), s08(10), s09(11), s10(12), s11(13), s12(14), s13(15), s14(16), s15(17), s16(18), s17(19), s18(20), s19(21), s20(22), s21(23), s22(24), s23(25), s24(26), a01(27), a02(28), a03(29), a04(30), d01(31), d02(32), d03(33), d04(34)

Channel 2 parameter 1	03,06,16	18301	16-bit unsign int	—"—
Channel 3 parameter 1	03,06,16	18302	16-bit unsign int	—"—
Channel 4 parameter 1	03,06,16	18303	16-bit unsign int	—"—
Channel 5 parameter 1	03,06,16	18304	16-bit unsign int	—"—
Channel 6 parameter 1	03,06,16	18305	16-bit unsign int	—"—
Channel 7 parameter 1	03,06,16	18306	16-bit unsign int	—"—
Channel 8 parameter 1	03,06,16	18307	16-bit unsign int	—"—
Channel 9 parameter 1	03,06,16	18308	16-bit unsign int	—"—
Channel 10 parameter 1	03,06,16	18309	16-bit unsign int	—"—
Channel 11 parameter 1	03,06,16	18310	16-bit unsign int	—"—
Channel 12 parameter 1	03,06,16	18311	16-bit unsign int	—"—
Channel 13 parameter 1	03,06,16	18312	16-bit unsign int	—"—
Channel 14 parameter 1	03,06,16	18313	16-bit unsign int	—"—
Channel 15 parameter 1	03,06,16	18314	16-bit unsign int	—"—
Channel 16 parameter 1	03,06,16	18315	16-bit unsign int	—"—
Channel 17 parameter 1	03,06,16	18316	16-bit unsign int	—"—
Channel 18 parameter 1	03,06,16	18317	16-bit unsign int	—"—
Channel 19 parameter 1	03,06,16	18318	16-bit unsign int	—"—
Channel 20 parameter 1	03,06,16	18319	16-bit unsign int	—"—
Channel 21 parameter 1	03,06,16	18320	16-bit unsign int	—"—
Channel 22 parameter 1	03,06,16	18321	16-bit unsign int	—"—
Channel 23 parameter 1	03,06,16	18322	16-bit unsign int	—"—
Channel 24 parameter 1	03,06,16	18323	16-bit unsign int	—"—
Channel 1 op 1	03,06,16	18400	16-bit unsign int	none(0), multiplication(1), division(2), sum(3), subtract(4)
Channel 2 op 1	03,06,16	18401	16-bit unsign int	—"—
Channel 3 op 1	03,06,16	18402	16-bit unsign int	—"—
Channel 4 op 1	03,06,16	18403	16-bit unsign int	—"—
Channel 5 op 1	03,06,16	18404	16-bit unsign int	—"—
Channel 6 op 1	03,06,16	18405	16-bit unsign int	—"—
Channel 7 op 1	03,06,16	18406	16-bit unsign int	—"—
Channel 8 op 1	03,06,16	18407	16-bit unsign int	—"—
Channel 9 op 1	03,06,16	18408	16-bit unsign int	—"—
Channel 10 op 1	03,06,16	18409	16-bit unsign int	—"—
Channel 11 op 1	03,06,16	18410	16-bit unsign int	—"—
Channel 12 op 1	03,06,16	18411	16-bit unsign int	—"—
Channel 13 op 1	03,06,16	18412	16-bit unsign int	—"—
Channel 14 op 1	03,06,16	18413	16-bit unsign int	—"—
Channel 15 op 1	03,06,16	18414	16-bit unsign int	—"—
Channel 16 op 1	03,06,16	18415	16-bit unsign int	—"—
Channel 17 op 1	03,06,16	18416	16-bit unsign int	—"—
Channel 18 op 1	03,06,16	18417	16-bit unsign int	—"—
Channel 19 op 1	03,06,16	18418	16-bit unsign int	—"—
Channel 20 op 1	03,06,16	18419	16-bit unsign int	—"—
Channel 21 op 1	03,06,16	18420	16-bit unsign int	—"—
Channel 22 op 1	03,06,16	18421	16-bit unsign int	—"—

Channel 23 op 1	03,06,16	18422	16-bit unsign int	—"—
Channel 24 op 1	03,06,16	18423	16-bit unsign int	—"—
Channel 1 parameter 2	03,06,16	18500	16-bit unsign int	none(0), s01(3), s02(4), s03(5), s04(6), s05(7), s06(8), s07(9), s08(10), s09(11), s10(12), s11(13), s12(14), s13(15), s14(16), s15(17), s16(18), s17(19), s18(20), s19(21), s20(22), s21(23), s22(24), s23(25), s24(26), a01(27), a02(28), a03(29), a04(30), d01(31), d02(32), d03(33), d04(34)
Channel 2 parameter 2	03,06,16	18501	16-bit unsign int	—"—
Channel 3 parameter 2	03,06,16	18502	16-bit unsign int	—"—
Channel 4 parameter 2	03,06,16	18503	16-bit unsign int	—"—
Channel 5 parameter 2	03,06,16	18504	16-bit unsign int	—"—
Channel 6 parameter 2	03,06,16	18505	16-bit unsign int	—"—
Channel 7 parameter 2	03,06,16	18506	16-bit unsign int	—"—
Channel 8 parameter 2	03,06,16	18507	16-bit unsign int	—"—
Channel 9 parameter 2	03,06,16	18508	16-bit unsign int	—"—
Channel 10 parameter 2	03,06,16	18509	16-bit unsign int	—"—
Channel 11 parameter 2	03,06,16	18510	16-bit unsign int	—"—
Channel 12 parameter 2	03,06,16	18511	16-bit unsign int	—"—
Channel 13 parameter 2	03,06,16	18512	16-bit unsign int	—"—
Channel 14 parameter 2	03,06,16	18513	16-bit unsign int	—"—
Channel 15 parameter 2	03,06,16	18514	16-bit unsign int	—"—
Channel 16 parameter 2	03,06,16	18515	16-bit unsign int	—"—
Channel 17 parameter 2	03,06,16	18516	16-bit unsign int	—"—
Channel 18 parameter 2	03,06,16	18517	16-bit unsign int	—"—
Channel 19 parameter 2	03,06,16	18518	16-bit unsign int	—"—
Channel 20 parameter 2	03,06,16	18519	16-bit unsign int	—"—
Channel 21 parameter 2	03,06,16	18520	16-bit unsign int	—"—
Channel 22 parameter 2	03,06,16	18521	16-bit unsign int	—"—
Channel 23 parameter 2	03,06,16	18522	16-bit unsign int	—"—
Channel 24 parameter 2	03,06,16	18523	16-bit unsign int	—"—
Channel 1 op 2	03,06,16	18600	16-bit unsign int	none(0), multiplication(1), division(2), sum(3), subtract(4)
Channel 2 op 2	03,06,16	18601	16-bit unsign int	—"—
Channel 3 op 2	03,06,16	18602	16-bit unsign int	—"—
Channel 4 op 2	03,06,16	18603	16-bit unsign int	—"—
Channel 5 op 2	03,06,16	18604	16-bit unsign int	—"—
Channel 6 op 2	03,06,16	18605	16-bit unsign int	—"—
Channel 7 op 2	03,06,16	18606	16-bit unsign int	—"—
Channel 8 op 2	03,06,16	18607	16-bit unsign int	—"—
Channel 9 op 2	03,06,16	18608	16-bit unsign int	—"—
Channel 10 op 2	03,06,16	18609	16-bit unsign int	—"—

Channel 11 op 2	03,06,16	18610	16-bit unsign int	—"—
Channel 12 op 2	03,06,16	18611	16-bit unsign int	—"—
Channel 13 op 2	03,06,16	18612	16-bit unsign int	—"—
Channel 14 op 2	03,06,16	18613	16-bit unsign int	—"—
Channel 15 op 2	03,06,16	18614	16-bit unsign int	—"—
Channel 16 op 2	03,06,16	18615	16-bit unsign int	—"—
Channel 17 op 2	03,06,16	18616	16-bit unsign int	—"—
Channel 18 op 2	03,06,16	18617	16-bit unsign int	—"—
Channel 19 op 2	03,06,16	18618	16-bit unsign int	—"—
Channel 20 op 2	03,06,16	18619	16-bit unsign int	—"—
Channel 21 op 2	03,06,16	18620	16-bit unsign int	—"—
Channel 22 op 2	03,06,16	18621	16-bit unsign int	—"—
Channel 23 op 2	03,06,16	18622	16-bit unsign int	—"—
Channel 24 op 2	03,06,16	18623	16-bit unsign int	—"—
Channel 1 coeff 1	03,16	18700	32-bit Float	
Channel 2 coeff 1	03,16	18702	32-bit Float	
Channel 3 coeff 1	03,16	18704	32-bit Float	
Channel 4 coeff 1	03,16	18706	32-bit Float	
Channel 5 coeff 1	03,16	18708	32-bit Float	
Channel 6 coeff 1	03,16	18710	32-bit Float	
Channel 7 coeff 1	03,16	18712	32-bit Float	
Channel 8 coeff 1	03,16	18714	32-bit Float	
Channel 9 coeff 1	03,16	18716	32-bit Float	
Channel 10 coeff 1	03,16	18718	32-bit Float	
Channel 11 coeff 1	03,16	18720	32-bit Float	
Channel 12 coeff 1	03,16	18722	32-bit Float	
Channel 13 coeff 1	03,16	18724	32-bit Float	
Channel 14 coeff 1	03,16	18726	32-bit Float	
Channel 15 coeff 1	03,16	18728	32-bit Float	
Channel 16 coeff 1	03,16	18730	32-bit Float	
Channel 17 coeff 1	03,16	18732	32-bit Float	
Channel 18 coeff 1	03,16	18734	32-bit Float	
Channel 19 coeff 1	03,16	18736	32-bit Float	
Channel 20 coeff 1	03,16	18738	32-bit Float	
Channel 21 coeff 1	03,16	18740	32-bit Float	
Channel 22 coeff 1	03,16	18742	32-bit Float	
Channel 23 coeff 1	03,16	18744	32-bit Float	
Channel 24 coeff 1	03,16	18746	32-bit Float	
Channel 1 op 3	03,06,16	18800	16-bit unsign int	none(0), multiplication(1), division(2), sum(3), subtract(4)
Channel 2 op 3	03,06,16	18801	16-bit unsign int	—"—
Channel 3 op 3	03,06,16	18802	16-bit unsign int	—"—
Channel 4 op 3	03,06,16	18803	16-bit unsign int	—"—
Channel 5 op 3	03,06,16	18804	16-bit unsign int	—"—
Channel 6 op 3	03,06,16	18805	16-bit unsign int	—"—

Channel 7 op 3	03,06,16	18806	16-bit unsign int	—"—
Channel 8 op 3	03,06,16	18807	16-bit unsign int	—"—
Channel 9 op 3	03,06,16	18808	16-bit unsign int	—"—
Channel 10 op 3	03,06,16	18809	16-bit unsign int	—"—
Channel 11 op 3	03,06,16	18810	16-bit unsign int	—"—
Channel 12 op 3	03,06,16	18811	16-bit unsign int	—"—
Channel 13 op 3	03,06,16	18812	16-bit unsign int	—"—
Channel 14 op 3	03,06,16	18813	16-bit unsign int	—"—
Channel 15 op 3	03,06,16	18814	16-bit unsign int	—"—
Channel 16 op 3	03,06,16	18815	16-bit unsign int	—"—
Channel 17 op 3	03,06,16	18816	16-bit unsign int	—"—
Channel 18 op 3	03,06,16	18817	16-bit unsign int	—"—
Channel 19 op 3	03,06,16	18818	16-bit unsign int	—"—
Channel 20 op 3	03,06,16	18819	16-bit unsign int	—"—
Channel 21 op 3	03,06,16	18820	16-bit unsign int	—"—
Channel 22 op 3	03,06,16	18821	16-bit unsign int	—"—
Channel 23 op 3	03,06,16	18822	16-bit unsign int	—"—
Channel 24 op 3	03,06,16	18823	16-bit unsign int	—"—
Channel 1 coeff 2	03,16	18900	32-bit Float	
Channel 2 coeff 2	03,16	18902	32-bit Float	
Channel 3 coeff 2	03,16	18904	32-bit Float	
Channel 4 coeff 2	03,16	18906	32-bit Float	
Channel 5 coeff 2	03,16	18908	32-bit Float	
Channel 6 coeff 2	03,16	18910	32-bit Float	
Channel 7 coeff 2	03,16	18912	32-bit Float	
Channel 8 coeff 2	03,16	18914	32-bit Float	
Channel 9 coeff 2	03,16	18916	32-bit Float	
Channel 10 coeff 2	03,16	18918	32-bit Float	
Channel 11 coeff 2	03,16	18920	32-bit Float	
Channel 12 coeff 2	03,16	18922	32-bit Float	
Channel 13 coeff 2	03,16	18924	32-bit Float	
Channel 14 coeff 2	03,16	18926	32-bit Float	
Channel 15 coeff 2	03,16	18928	32-bit Float	
Channel 16 coeff 2	03,16	18930	32-bit Float	
Channel 17 coeff 2	03,16	18932	32-bit Float	
Channel 18 coeff 2	03,16	18934	32-bit Float	
Channel 29 coeff 2	03,16	18936	32-bit Float	
Channel 20 coeff 2	03,16	18938	32-bit Float	
Channel 21 coeff 2	03,16	18940	32-bit Float	
Channel 22 coeff 2	03,16	18942	32-bit Float	
Channel 23 coeff 2	03,16	18944	32-bit Float	
Channel 24 coeff 2	03,16	18946	32-bit Float	
Channel 1 unit	03,16	19000	16 bytes UTF-8	
Channel 2 unit	03,16	19008	16 bytes UTF-8	
Channel 3 unit	03,16	19016	16 bytes UTF-8	
Channel 4 unit	03,16	19024	16 bytes UTF-8	

Channel 5 unit	03,16	19032	16 bytes UTF-8	
Channel 6 unit	03,16	19040	16 bytes UTF-8	
Channel 7 unit	03,16	19048	16 bytes UTF-8	
Channel 8 unit	03,16	19056	16 bytes UTF-8	
Channel 9 unit	03,16	19064	16 bytes UTF-8	
Channel 10 unit	03,16	19072	16 bytes UTF-8	
Channel 11 unit	03,16	19080	16 bytes UTF-8	
Channel 12 unit	03,16	19088	16 bytes UTF-8	
Channel 13 unit	03,16	19096	16 bytes UTF-8	
Channel 14 unit	03,16	19104	16 bytes UTF-8	
Channel 15 unit	03,16	19112	16 bytes UTF-8	
Channel 16 unit	03,16	19120	16 bytes UTF-8	
Channel 17 unit	03,16	19128	16 bytes UTF-8	
Channel 18 unit	03,16	19136	16 bytes UTF-8	
Channel 19 unit	03,16	19144	16 bytes UTF-8	
Channel 20 unit	03,16	19152	16 bytes UTF-8	
Channel 21 unit	03,16	19160	16 bytes UTF-8	
Channel 22 unit	03,16	19168	16 bytes UTF-8	
Channel 23 unit	03,16	19176	16 bytes UTF-8	
Channel 24 unit	03,16	19184	16 bytes UTF-8	
Channel 1 value	03	19300	32-bit Float	
Channel 2 value	03	19302	32-bit Float	
Channel 3 value	03	19304	32-bit Float	
Channel 4 value	03	19306	32-bit Float	
Channel 5 value	03	19308	32-bit Float	
Channel 6 value	03	19310	32-bit Float	
Channel 7 value	03	19312	32-bit Float	
Channel 8 value	03	19314	32-bit Float	
Channel 9 value	03	19316	32-bit Float	
Channel 10 value	03	19318	32-bit Float	
Channel 11 value	03	19320	32-bit Float	
Channel 12 value	03	19322	32-bit Float	
Channel 13 value	03	19324	32-bit Float	
Channel 14 value	03	19326	32-bit Float	
Channel 15 value	03	19328	32-bit Float	
Channel 16 value	03	19330	32-bit Float	
Channel 17 value	03	19332	32-bit Float	
Channel 18 value	03	19334	32-bit Float	
Channel 19 value	03	19336	32-bit Float	
Channel 20 value	03	19338	32-bit Float	
Channel 21 value	03	19340	32-bit Float	
Channel 22 value	03	19342	32-bit Float	
Channel 23 value	03	19344	32-bit Float	
Channel 24 value	03	19346	32-bit Float	
Channel 1 counter	03	19400	32-bit unsign int	
Channel 2 counter	03	19402	32-bit unsign int	

Channel 3 counter	03	19404	32-bit unsign int	
Channel 4 counter	03	19406	32-bit unsign int	
Channel 5 counter	03	19408	32-bit unsign int	
Channel 6 counter	03	19410	32-bit unsign int	
Channel 7 counter	03	19412	32-bit unsign int	
Channel 8 counter	03	19414	32-bit unsign int	
Channel 9 counter	03	19416	32-bit unsign int	
Channel 10 counter	03	19418	32-bit unsign int	
Channel 11 counter	03	19420	32-bit unsign int	
Channel 12 counter	03	19422	32-bit unsign int	
Channel 13 counter	03	19424	32-bit unsign int	
Channel 14 counter	03	19426	32-bit unsign int	
Channel 15 counter	03	19428	32-bit unsign int	
Channel 16 counter	03	19430	32-bit unsign int	
Channel 17 counter	03	19432	32-bit unsign int	
Channel 18 counter	03	19434	32-bit unsign int	
Channel 19 counter	03	19436	32-bit unsign int	
Channel 20 counter	03	19438	32-bit unsign int	
Channel 21 counter	03	19440	32-bit unsign int	
Channel 22 counter	03	19442	32-bit unsign int	
Channel 23 counter	03	19444	32-bit unsign int	
Channel 24 counter	03	19446	32-bit unsign int	
Channel 1 alarm status	03	19500	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Channel 2 alarm status	03	19501	16-bit unsign int	—"—
Channel 3 alarm status	03	19502	16-bit unsign int	—"—
Channel 4 alarm status	03	19503	16-bit unsign int	—"—
Channel 5 alarm status	03	19504	16-bit unsign int	—"—
Channel 6 alarm status	03	19505	16-bit unsign int	—"—
Channel 7 alarm status	03	19506	16-bit unsign int	—"—
Channel 8 alarm status	03	19507	16-bit unsign int	—"—
Channel 9 alarm status	03	19508	16-bit unsign int	—"—
Channel 10 alarm status	03	19509	16-bit unsign int	—"—
Channel 11 alarm status	03	19510	16-bit unsign int	—"—
Channel 12 alarm status	03	19511	16-bit unsign int	—"—
Channel 13 alarm status	03	19512	16-bit unsign int	—"—
Channel 14 alarm status	03	19513	16-bit unsign int	—"—
Channel 15 alarm status	03	19514	16-bit unsign int	—"—
Channel 16 alarm status	03	19515	16-bit unsign int	—"—
Channel 17 alarm status	03	19516	16-bit unsign int	—"—
Channel 18 alarm status	03	19517	16-bit unsign int	—"—
Channel 19 alarm status	03	19518	16-bit unsign int	—"—
Channel 20 alarm status	03	19519	16-bit unsign int	—"—
Channel 21 alarm status	03	19520	16-bit unsign int	—"—
Channel 22 alarm status	03	19521	16-bit unsign int	—"—

Channel 23 alarm status	03	19522	16-bit unsign int	—"—
Channel 24 alarm status	03	19523	16-bit unsign int	—"—
Alarm 1 description	03,16	20000	16 bytes UTF-8	
Alarm 2 description	03,16	20008	16 bytes UTF-8	
Alarm 3 description	03,16	20016	16 bytes UTF-8	
Alarm 4 description	03,16	20024	16 bytes UTF-8	
Alarm 5 description	03,16	20032	16 bytes UTF-8	
Alarm 6 description	03,16	20040	16 bytes UTF-8	
Alarm 7 description	03,16	20048	16 bytes UTF-8	
Alarm 8 description	03,16	20056	16 bytes UTF-8	
Alarm 9 description	03,16	20064	16 bytes UTF-8	
Alarm 10 description	03,16	20072	16 bytes UTF-8	
Alarm 11 description	03,16	20080	16 bytes UTF-8	
Alarm 12 description	03,16	20088	16 bytes UTF-8	
Alarm 13 description	03,16	20096	16 bytes UTF-8	
Alarm 14 description	03,16	20104	16 bytes UTF-8	
Alarm 15 description	03,16	20112	16 bytes UTF-8	
Alarm 16 description	03,16	20120	16 bytes UTF-8	
Alarm 17 description	03,16	20128	16 bytes UTF-8	
Alarm 18 description	03,16	20136	16 bytes UTF-8	
Alarm 19 description	03,16	20144	16 bytes UTF-8	
Alarm 20 description	03,16	20152	16 bytes UTF-8	
Alarm 21 description	03,16	20160	16 bytes UTF-8	
Alarm 22 description	03,16	20168	16 bytes UTF-8	
Alarm 23 description	03,16	20176	16 bytes UTF-8	
Alarm 24 description	03,16	20184	16 bytes UTF-8	
Alarm 1 condition 1 channel	03,06,16	20200	16-bit unsign int	none(0), v01(1), v02(2), v03(3), v04(4), v05(5), v06(6), v07(7), v08(8), v09(9), v10(10), v11(11), v12(12), v13(13), v14(14), v15(15), v16(16), v17(17), v18(18), v19(19), v20(20), v21(21), v22(22), v23(23), v24(24)
Alarm 2 condition 1 channel	03,06,16	20201	16-bit unsign int	—"—
Alarm 3 condition 1 channel	03,06,16	20202	16-bit unsign int	—"—
Alarm 4 condition 1 channel	03,06,16	20203	16-bit unsign int	—"—
Alarm 5 condition 1 channel	03,06,16	20204	16-bit unsign int	—"—
Alarm 6 condition 1 channel	03,06,16	20205	16-bit unsign int	—"—
Alarm 7 condition 1 channel	03,06,16	20206	16-bit unsign int	—"—
Alarm 8 condition 1 channel	03,06,16	20207	16-bit unsign int	—"—
Alarm 9 condition 1 channel	03,06,16	20208	16-bit unsign int	—"—
Alarm 10 condition 1 channel	03,06,16	20209	16-bit unsign int	—"—
Alarm 11 condition 1 channel	03,06,16	20210	16-bit unsign int	—"—
Alarm 12 condition 1 channel	03,06,16	20211	16-bit unsign int	—"—
Alarm 13 condition 1 channel	03,06,16	20212	16-bit unsign int	—"—
Alarm 14 condition 1 channel	03,06,16	20213	16-bit unsign int	—"—

Alarm 15 condition 1 channel	03,06,16	20214	16-bit unsign int	—"—
Alarm 16 condition 1 channel	03,06,16	20215	16-bit unsign int	—"—
Alarm 17 condition 1 channel	03,06,16	20216	16-bit unsign int	—"—
Alarm 18 condition 1 channel	03,06,16	20217	16-bit unsign int	—"—
Alarm 19 condition 1 channel	03,06,16	20218	16-bit unsign int	—"—
Alarm 20 condition 1 channel	03,06,16	20219	16-bit unsign int	—"—
Alarm 21 condition 1 channel	03,06,16	20220	16-bit unsign int	—"—
Alarm 22 condition 1 channel	03,06,16	20221	16-bit unsign int	—"—
Alarm 23 condition 1 channel	03,06,16	20222	16-bit unsign int	—"—
Alarm 24 condition 1 channel	03,06,16	20223	16-bit unsign int	—"—
Alarm 1 condition 1 operand	03,06,16	20300	16-bit unsign int	larger(0), less(1)
Alarm 2 condition 1 operand	03,06,16	20301	16-bit unsign int	larger(0), less(1)
Alarm 3 condition 1 operand	03,06,16	20302	16-bit unsign int	larger(0), less(1)
Alarm 4 condition 1 operand	03,06,16	20303	16-bit unsign int	larger(0), less(1)
Alarm 5 condition 1 operand	03,06,16	20304	16-bit unsign int	larger(0), less(1)
Alarm 6 condition 1 operand	03,06,16	20305	16-bit unsign int	larger(0), less(1)
Alarm 7 condition 1 operand	03,06,16	20306	16-bit unsign int	larger(0), less(1)
Alarm 8 condition 1 operand	03,06,16	20307	16-bit unsign int	larger(0), less(1)
Alarm 9 condition 1 operand	03,06,16	20308	16-bit unsign int	larger(0), less(1)
Alarm 10 condition 1 operand	03,06,16	20309	16-bit unsign int	larger(0), less(1)
Alarm 11 condition 1 operand	03,06,16	20310	16-bit unsign int	larger(0), less(1)
Alarm 12 condition 1 operand	03,06,16	20311	16-bit unsign int	larger(0), less(1)
Alarm 13 condition 1 operand	03,06,16	20312	16-bit unsign int	larger(0), less(1)
Alarm 14 condition 1 operand	03,06,16	20313	16-bit unsign int	larger(0), less(1)
Alarm 15 condition 1 operand	03,06,16	20314	16-bit unsign int	larger(0), less(1)
Alarm 16 condition 1 operand	03,06,16	20315	16-bit unsign int	larger(0), less(1)
Alarm 17 condition 1 operand	03,06,16	20316	16-bit unsign int	larger(0), less(1)
Alarm 18 condition 1 operand	03,06,16	20317	16-bit unsign int	larger(0), less(1)
Alarm 19 condition 1 operand	03,06,16	20318	16-bit unsign int	larger(0), less(1)
Alarm 20 condition 1 operand	03,06,16	20319	16-bit unsign int	larger(0), less(1)
Alarm 21 condition 1 operand	03,06,16	20320	16-bit unsign int	larger(0), less(1)
Alarm 22 condition 1 operand	03,06,16	20321	16-bit unsign int	larger(0), less(1)
Alarm 23 condition 1 operand	03,06,16	20322	16-bit unsign int	larger(0), less(1)
Alarm 24 condition 1 operand	03,06,16	20323	16-bit unsign int	larger(0), less(1)
Alarm 1 condition 1 limit	03,16	20400	32-bit Float	
Alarm 2 condition 1 limit	03,16	20402	32-bit Float	
Alarm 3 condition 1 limit	03,16	20404	32-bit Float	
Alarm 4 condition 1 limit	03,16	20406	32-bit Float	
Alarm 5 condition 1 limit	03,16	20408	32-bit Float	
Alarm 6 condition 1 limit	03,16	20410	32-bit Float	
Alarm 7 condition 1 limit	03,16	20412	32-bit Float	
Alarm 8 condition 1 limit	03,16	20414	32-bit Float	
Alarm 9 condition 1 limit	03,16	20416	32-bit Float	
Alarm 10 condition 1 limit	03,16	20418	32-bit Float	
Alarm 11 condition 1 limit	03,16	20420	32-bit Float	
Alarm 12 condition 1 limit	03,16	20422	32-bit Float	

Alarm 13 condition 1 limit	03,16	20424	32-bit Float	
Alarm 14 condition 1 limit	03,16	20426	32-bit Float	
Alarm 15 condition 1 limit	03,16	20428	32-bit Float	
Alarm 16 condition 1 limit	03,16	20430	32-bit Float	
Alarm 17 condition 1 limit	03,16	20432	32-bit Float	
Alarm 18 condition 1 limit	03,16	20434	32-bit Float	
Alarm 19 condition 1 limit	03,16	20436	32-bit Float	
Alarm 20 condition 1 limit	03,16	20438	32-bit Float	
Alarm 21 condition 1 limit	03,16	20440	32-bit Float	
Alarm 22 condition 1 limit	03,16	20442	32-bit Float	
Alarm 23 condition 1 limit	03,16	20444	32-bit Float	
Alarm 23 condition 1 limit	03,16	20446	32-bit Float	
Alarm 1 condition 1 hysteresis	03,16	20500	32-bit Float	
Alarm 2 condition 1 hysteresis	03,16	20502	32-bit Float	
Alarm 3 condition 1 hysteresis	03,16	20504	32-bit Float	
Alarm 4 condition 1 hysteresis	03,16	20506	32-bit Float	
Alarm 5 condition 1 hysteresis	03,16	20508	32-bit Float	
Alarm 6 condition 1 hysteresis	03,16	20510	32-bit Float	
Alarm 7 condition 1 hysteresis	03,16	20512	32-bit Float	
Alarm 8 condition 1 hysteresis	03,16	20514	32-bit Float	
Alarm 9 condition 1 hysteresis	03,16	20516	32-bit Float	
Alarm 10 condition 1 hysteresis	03,16	20518	32-bit Float	
Alarm 11 condition 1 hysteresis	03,16	20520	32-bit Float	
Alarm 12 condition 1 hysteresis	03,16	20522	32-bit Float	
Alarm 13 condition 1 hysteresis	03,16	20524	32-bit Float	
Alarm 14 condition 1 hysteresis	03,16	20526	32-bit Float	
Alarm 15 condition 1 hysteresis	03,16	20528	32-bit Float	
Alarm 16 condition 1 hysteresis	03,16	20530	32-bit Float	
Alarm 17 condition 1 hysteresis	03,16	20532	32-bit Float	
Alarm 18 condition 1 hysteresis	03,16	20534	32-bit Float	
Alarm 19 condition 1 hysteresis	03,16	20536	32-bit Float	
Alarm 20 condition 1 hysteresis	03,16	20538	32-bit Float	
Alarm 21 condition 1 hysteresis	03,16	20540	32-bit Float	
Alarm 22 condition 1 hysteresis	03,16	20542	32-bit Float	
Alarm 23 condition 1 hysteresis	03,16	20544	32-bit Float	
Alarm 24 condition 1 hysteresis	03,16	20546	32-bit Float	
Alarm 1 condition 1 discrete al. state	03,06,16	20600	16-bit unsign int	open(1), closed(0)
Alarm 2 condition 1 discrete al. state	03,06,16	20601	16-bit unsign int	open(1), closed(0)
Alarm 3 condition 1 discrete al. state	03,06,16	20602	16-bit unsign int	open(1), closed(0)
Alarm 4 condition 1 discrete al. state	03,06,16	20603	16-bit unsign int	open(1), closed(0)
Alarm 5 condition 1 discrete al. state	03,06,16	20604	16-bit unsign int	open(1), closed(0)

Alarm 6 condition 1 discrete al. state	03,06,16	20605	16-bit unsign int	open(1), closed(0)
Alarm 7 condition 1 discrete al. state	03,06,16	20606	16-bit unsign int	open(1), closed(0)
Alarm 8 condition 1 discrete al. state	03,06,16	20607	16-bit unsign int	open(1), closed(0)
Alarm 9 condition 1 discrete al. state	03,06,16	20608	16-bit unsign int	open(1), closed(0)
Alarm 10 condition 1 discrete al. state	03,06,16	20609	16-bit unsign int	open(1), closed(0)
Alarm 11 condition 1 discrete al. state	03,06,16	20610	16-bit unsign int	open(1), closed(0)
Alarm 12 condition 1 discrete al. state	03,06,16	20611	16-bit unsign int	open(1), closed(0)
Alarm 13 condition 1 discrete al. state	03,06,16	20612	16-bit unsign int	open(1), closed(0)
Alarm 14 condition 1 discrete al. state	03,06,16	20613	16-bit unsign int	open(1), closed(0)
Alarm 15 condition 1 discrete al. state	03,06,16	20614	16-bit unsign int	open(1), closed(0)
Alarm 16 condition 1 discrete al. state	03,06,16	20615	16-bit unsign int	open(1), closed(0)
Alarm 17 condition 1 discrete al. state	03,06,16	20616	16-bit unsign int	open(1), closed(0)
Alarm 18 condition 1 discrete al. state	03,06,16	20617	16-bit unsign int	open(1), closed(0)
Alarm 19 condition 1 discrete al. state	03,06,16	20618	16-bit unsign int	open(1), closed(0)
Alarm 20 condition 1 discrete al. state	03,06,16	20619	16-bit unsign int	open(1), closed(0)
Alarm 21 condition 1 discrete al. state	03,06,16	20620	16-bit unsign int	open(1), closed(0)
Alarm 22 condition 1 discrete al. state	03,06,16	20621	16-bit unsign int	open(1), closed(0)
Alarm 23 condition 1 discrete al. state	03,06,16	20622	16-bit unsign int	open(1), closed(0)
Alarm 24 condition 1 discrete al. state	03,06,16	20623	16-bit unsign int	open(1), closed(0)
Alarm 1 function	03,06,16	20700	16-bit unsign int	none(0}, and(1), or(2)
Alarm 2 function	03,06,16	20701	16-bit unsign int	none(0}, and(1), or(2)
Alarm 3 function	03,06,16	20702	16-bit unsign int	none(0}, and(1), or(2)
Alarm 4 function	03,06,16	20703	16-bit unsign int	none(0}, and(1), or(2)
Alarm 5 function	03,06,16	20704	16-bit unsign int	none(0}, and(1), or(2)
Alarm 6 function	03,06,16	20705	16-bit unsign int	none(0}, and(1), or(2)
Alarm 7 function	03,06,16	20706	16-bit unsign int	none(0}, and(1), or(2)
Alarm 8 function	03,06,16	20707	16-bit unsign int	none(0}, and(1), or(2)
Alarm 9 function	03,06,16	20708	16-bit unsign int	none(0}, and(1), or(2)
Alarm 10 function	03,06,16	20709	16-bit unsign int	none(0}, and(1), or(2)
Alarm 11 function	03,06,16	20710	16-bit unsign int	none(0}, and(1), or(2)
Alarm 12 function	03,06,16	20711	16-bit unsign int	none(0}, and(1), or(2)
Alarm 13 function	03,06,16	20712	16-bit unsign int	none(0}, and(1), or(2)

Alarm 14 function	03,06,16	20713	16-bit unsign int	none(0}, and(1), or(2)
Alarm 15 function	03,06,16	20714	16-bit unsign int	none(0}, and(1), or(2)
Alarm 16 function	03,06,16	20715	16-bit unsign int	none(0}, and(1), or(2)
Alarm 17 function	03,06,16	20716	16-bit unsign int	none(0}, and(1), or(2)
Alarm 18 function	03,06,16	20717	16-bit unsign int	none(0}, and(1), or(2)
Alarm 19 function	03,06,16	20718	16-bit unsign int	none(0}, and(1), or(2)
Alarm 20 function	03,06,16	20719	16-bit unsign int	none(0}, and(1), or(2)
Alarm 21 function	03,06,16	20720	16-bit unsign int	none(0}, and(1), or(2)
Alarm 22 function	03,06,16	20721	16-bit unsign int	none(0}, and(1), or(2)
Alarm 23 function	03,06,16	20722	16-bit unsign int	none(0}, and(1), or(2)
Alarm 24 function	03,06,16	20723	16-bit unsign int	none(0}, and(1), or(2)
Alarm 1 condition 2 channel	03,06,16	20800	16-bit unsign int	none(0), v01(1), v02(2), v03(3), v04(4), v05(5), v06(6), v07(7), v08(8), v09(9), v10(10), v11(11), v12(12), v13(13), v14(14), v15(15), v16(16), v17(17), v18(18), v19(19), v20(20), v21(21), v22(22), v23(23), v24(24)
Alarm 2 condition 2 channel	03,06,16	20801	16-bit unsign int	—"—
Alarm 3 condition 2 channel	03,06,16	20802	16-bit unsign int	—"—
Alarm 4 condition 2 channel	03,06,16	20803	16-bit unsign int	—"—
Alarm 5 condition 2 channel	03,06,16	20804	16-bit unsign int	—"—
Alarm 6 condition 2 channel	03,06,16	20805	16-bit unsign int	—"—
Alarm 7 condition 2 channel	03,06,16	20806	16-bit unsign int	—"—
Alarm 8 condition 2 channel	03,06,16	20807	16-bit unsign int	—"—
Alarm 9 condition 2 channel	03,06,16	20808	16-bit unsign int	—"—
Alarm 10 condition 2 channel	03,06,16	20809	16-bit unsign int	—"—
Alarm 11 condition 2 channel	03,06,16	20810	16-bit unsign int	—"—
Alarm 12 condition 2 channel	03,06,16	20811	16-bit unsign int	—"—
Alarm 13 condition 2 channel	03,06,16	20812	16-bit unsign int	—"—
Alarm 14 condition 2 channel	03,06,16	20813	16-bit unsign int	—"—
Alarm 15 condition 2 channel	03,06,16	20814	16-bit unsign int	—"—
Alarm 16 condition 2 channel	03,06,16	20815	16-bit unsign int	—"—
Alarm 17 condition 2 channel	03,06,16	20816	16-bit unsign int	—"—
Alarm 18 condition 2 channel	03,06,16	20817	16-bit unsign int	—"—
Alarm 19 condition 2 channel	03,06,16	20818	16-bit unsign int	—"—
Alarm 20 condition 2 channel	03,06,16	20819	16-bit unsign int	—"—
Alarm 21 condition 2 channel	03,06,16	20820	16-bit unsign int	—"—
Alarm 22 condition 2 channel	03,06,16	20821	16-bit unsign int	—"—
Alarm 23 condition 2 channel	03,06,16	20822	16-bit unsign int	—"—
Alarm 24 condition 2 channel	03,06,16	20823	16-bit unsign int	—"—
Alarm 1 condition 2 operand	03,06,16	20900	16-bit unsign int	larger(0), less(1)
Alarm 2 condition 2 operand	03,06,16	20901	16-bit unsign int	larger(0), less(1)
Alarm 3 condition 2 operand	03,06,16	20902	16-bit unsign int	larger(0), less(1)
Alarm 4 condition 2 operand	03,06,16	20903	16-bit unsign int	larger(0), less(1)

Alarm 5 condition 2 operand	03,06,16	20904	16-bit unsign int	larger(0), less(1)
Alarm 6 condition 2 operand	03,06,16	20905	16-bit unsign int	larger(0), less(1)
Alarm 7 condition 2 operand	03,06,16	20906	16-bit unsign int	larger(0), less(1)
Alarm 8 condition 2 operand	03,06,16	20907	16-bit unsign int	larger(0), less(1)
Alarm 9 condition 2 operand	03,06,16	20908	16-bit unsign int	larger(0), less(1)
Alarm 10 condition 2 operand	03,06,16	20909	16-bit unsign int	larger(0), less(1)
Alarm 11 condition 2 operand	03,06,16	20910	16-bit unsign int	larger(0), less(1)
Alarm 12 condition 2 operand	03,06,16	20911	16-bit unsign int	larger(0), less(1)
Alarm 13 condition 2 operand	03,06,16	20912	16-bit unsign int	larger(0), less(1)
Alarm 14 condition 2 operand	03,06,16	20913	16-bit unsign int	larger(0), less(1)
Alarm 15 condition 2 operand	03,06,16	20914	16-bit unsign int	larger(0), less(1)
Alarm 16 condition 2 operand	03,06,16	20915	16-bit unsign int	larger(0), less(1)
Alarm 17 condition 2 operand	03,06,16	20916	16-bit unsign int	larger(0), less(1)
Alarm 18 condition 2 operand	03,06,16	20917	16-bit unsign int	larger(0), less(1)
Alarm 19 condition 2 operand	03,06,16	20918	16-bit unsign int	larger(0), less(1)
Alarm 20 condition 2 operand	03,06,16	20919	16-bit unsign int	larger(0), less(1)
Alarm 21 condition 2 operand	03,06,16	20920	16-bit unsign int	larger(0), less(1)
Alarm 22 condition 2 operand	03,06,16	20921	16-bit unsign int	larger(0), less(1)
Alarm 23 condition 2 operand	03,06,16	20922	16-bit unsign int	larger(0), less(1)
Alarm 24 condition 2 operand	03,06,16	20923	16-bit unsign int	larger(0), less(1)
Alarm 1 condition 2 limit	03,16	21000	32-bit Float	
Alarm 2 condition 2 limit	03,16	21002	32-bit Float	
Alarm 3 condition 2 limit	03,16	21004	32-bit Float	
Alarm 4 condition 2 limit	03,16	21006	32-bit Float	
Alarm 5 condition 2 limit	03,16	21008	32-bit Float	
Alarm 6 condition 2 limit	03,16	21010	32-bit Float	
Alarm 7 condition 2 limit	03,16	21012	32-bit Float	
Alarm 8 condition 2 limit	03,16	21014	32-bit Float	
Alarm 9 condition 2 limit	03,16	21016	32-bit Float	
Alarm 10 condition 2 limit	03,16	21018	32-bit Float	
Alarm 11 condition 2 limit	03,16	21020	32-bit Float	
Alarm 12 condition 2 limit	03,16	21022	32-bit Float	
Alarm 13 condition 2 limit	03,16	21024	32-bit Float	
Alarm 14 condition 2 limit	03,16	21026	32-bit Float	
Alarm 15 condition 2 limit	03,16	21028	32-bit Float	
Alarm 16 condition 2 limit	03,16	21030	32-bit Float	
Alarm 17 condition 2 limit	03,16	21032	32-bit Float	
Alarm 18 condition 2 limit	03,16	21034	32-bit Float	
Alarm 19 condition 2 limit	03,16	21036	32-bit Float	
Alarm 20 condition 2 limit	03,16	21038	32-bit Float	
Alarm 21 condition 2 limit	03,16	21040	32-bit Float	
Alarm 22 condition 2 limit	03,16	21042	32-bit Float	
Alarm 23 condition 2 limit	03,16	21044	32-bit Float	
Alarm 24 condition 2 limit	03,16	21046	32-bit Float	
Alarm 1 condition 2 hysteresis	03,16	21100	32-bit Float	
Alarm 2 condition 2 hysteresis	03,16	21102	32-bit Float	

Alarm 3 condition 2 hysteresis	03,16	21104	32-bit Float	
Alarm 4 condition 2 hysteresis	03,16	21106	32-bit Float	
Alarm 5 condition 2 hysteresis	03,16	21108	32-bit Float	
Alarm 6 condition 2 hysteresis	03,16	21110	32-bit Float	
Alarm 7 condition 2 hysteresis	03,16	21112	32-bit Float	
Alarm 8 condition 2 hysteresis	03,16	21114	32-bit Float	
Alarm 9 condition 2 hysteresis	03,16	21116	32-bit Float	
Alarm 10 condition 2 hysteresis	03,16	21118	32-bit Float	
Alarm 11 condition 2 hysteresis	03,16	21120	32-bit Float	
Alarm 12 condition 2 hysteresis	03,16	21122	32-bit Float	
Alarm 13 condition 2 hysteresis	03,16	21124	32-bit Float	
Alarm 14 condition 2 hysteresis	03,16	21126	32-bit Float	
Alarm 15 condition 2 hysteresis	03,16	21128	32-bit Float	
Alarm 16 condition 2 hysteresis	03,16	21130	32-bit Float	
Alarm 17 condition 2 hysteresis	03,16	21132	32-bit Float	
Alarm 18 condition 2 hysteresis	03,16	21134	32-bit Float	
Alarm 19 condition 2 hysteresis	03,16	21136	32-bit Float	
Alarm 20 condition 2 hysteresis	03,16	21138	32-bit Float	
Alarm 21 condition 2 hysteresis	03,16	21140	32-bit Float	
Alarm 22 condition 2 hysteresis	03,16	21142	32-bit Float	
Alarm 23 condition 2 hysteresis	03,16	21144	32-bit Float	
Alarm 24 condition 2 hysteresis	03,16	21146	32-bit Float	
Alarm 1 condition 2 discrete al. state	03,06,16	21200	16-bit unsign int	open(1), closed(0)
Alarm 2 condition 2 discrete al. state	03,06,16	21201	16-bit unsign int	open(1), closed(0)
Alarm 3 condition 2 discrete al. state	03,06,16	21202	16-bit unsign int	open(1), closed(0)
Alarm 4 condition 2 discrete al. state	03,06,16	21203	16-bit unsign int	open(1), closed(0)
Alarm 5 condition 2 discrete al. state	03,06,16	21204	16-bit unsign int	open(1), closed(0)
Alarm 6 condition 2 discrete al. state	03,06,16	21205	16-bit unsign int	open(1), closed(0)
Alarm 7 condition 2 discrete al. state	03,06,16	21206	16-bit unsign int	open(1), closed(0)
Alarm 8 condition 2 discrete al. state	03,06,16	21207	16-bit unsign int	open(1), closed(0)
Alarm 9 condition 2 discrete al. state	03,06,16	21208	16-bit unsign int	open(1), closed(0)
Alarm 10 condition 2 discrete al. state	03,06,16	21209	16-bit unsign int	open(1), closed(0)
Alarm 11 condition 2 discrete al. state	03,06,16	21210	16-bit unsign int	open(1), closed(0)
Alarm 12 condition 2 discrete al. state	03,06,16	21211	16-bit unsign int	open(1), closed(0)
Alarm 13 condition 2 discrete al. state	03,06,16	21212	16-bit unsign int	open(1), closed(0)
Alarm 14 condition 2 discrete al. state	03,06,16	21213	16-bit unsign int	open(1), closed(0)

Alarm 15 condition 2 discrete al. state	03,06,16	21214	16-bit unsign int	open(1), closed(0)
Alarm 16 condition 2 discrete al. state	03,06,16	21215	16-bit unsign int	open(1), closed(0)
Alarm 17 condition 2 discrete al. state	03,06,16	21216	16-bit unsign int	open(1), closed(0)
Alarm 18 condition 2 discrete al. state	03,06,16	21217	16-bit unsign int	open(1), closed(0)
Alarm 19 condition 2 discrete al. state	03,06,16	21218	16-bit unsign int	open(1), closed(0)
Alarm 20 condition 2 discrete al. state	03,06,16	21219	16-bit unsign int	open(1), closed(0)
Alarm 21 condition 2 discrete al. state	03,06,16	21220	16-bit unsign int	open(1), closed(0)
Alarm 22 condition 2 discrete al. state	03,06,16	21221	16-bit unsign int	open(1), closed(0)
Alarm 23 condition 2 discrete al. state	03,06,16	21222	16-bit unsign int	open(1), closed(0)
Alarm 24 condition 2 discrete al. state	03,06,16	21223	16-bit unsign int	open(1), closed(0)
Alarm 1 type	03,06,16	21300	16-bit unsign int	warning(3), minor(4), major(5), critical(6)
Alarm 2 type	03,06,16	21301	16-bit unsign int	—"—
Alarm 3 type	03,06,16	21302	16-bit unsign int	—"—
Alarm 4 type	03,06,16	21303	16-bit unsign int	—"—
Alarm 5 type	03,06,16	21304	16-bit unsign int	—"—
Alarm 6 type	03,06,16	21305	16-bit unsign int	—"—
Alarm 7 type	03,06,16	21306	16-bit unsign int	—"—
Alarm 8 type	03,06,16	21307	16-bit unsign int	—"—
Alarm 9 type	03,06,16	21308	16-bit unsign int	—"—
Alarm 10 type	03,06,16	21309	16-bit unsign int	—"—
Alarm 11 type	03,06,16	21310	16-bit unsign int	—"—
Alarm 12 type	03,06,16	21311	16-bit unsign int	—"—
Alarm 13 type	03,06,16	21312	16-bit unsign int	—"—
Alarm 14 type	03,06,16	21313	16-bit unsign int	—"—
Alarm 15 type	03,06,16	21314	16-bit unsign int	—"—
Alarm 16 type	03,06,16	21315	16-bit unsign int	—"—
Alarm 17 type	03,06,16	21316	16-bit unsign int	—"—
Alarm 18 type	03,06,16	21317	16-bit unsign int	—"—
Alarm 19 type	03,06,16	21318	16-bit unsign int	—"—
Alarm 20 type	03,06,16	21319	16-bit unsign int	—"—
Alarm 21 type	03,06,16	21320	16-bit unsign int	—"—
Alarm 22 type	03,06,16	21321	16-bit unsign int	—"—
Alarm 23 type	03,06,16	21322	16-bit unsign int	—"—
Alarm 24 type	03,06,16	21323	16-bit unsign int	—"—
Alarm 1 assigned to channel	03,06,16	21400	16-bit unsign int	none(0), v01(1), v02(2), v03(3), v04(4), v05(5), v06(6), v07(7), v08(8),v09(9), v10(10),

				v11(11), v12(12), v13(13), v14(14), v15(15), v16(16),v17(17), v18(18), v19(19), v20(20), v21(21), v22(22), v23(23), v24(24)
Alarm 2 assigned to channel	03,06,16	21401	16-bit unsign int	—"—
Alarm 3 assigned to channel	03,06,16	21402	16-bit unsign int	—"—
Alarm 4 assigned to channel	03,06,16	21403	16-bit unsign int	—"—
Alarm 5 assigned to channel	03,06,16	21404	16-bit unsign int	—"—
Alarm 6 assigned to channel	03,06,16	21405	16-bit unsign int	—"—
Alarm 7 assigned to channel	03,06,16	21406	16-bit unsign int	—"—
Alarm 8 assigned to channel	03,06,16	21407	16-bit unsign int	—"—
Alarm 9 assigned to channel	03,06,16	21408	16-bit unsign int	—"—
Alarm 10 assigned to channel	03,06,16	21409	16-bit unsign int	—"—
Alarm 11 assigned to channel	03,06,16	21410	16-bit unsign int	—"—
Alarm 12 assigned to channel	03,06,16	21411	16-bit unsign int	—"—
Alarm 13 assigned to channel	03,06,16	21412	16-bit unsign int	—"—
Alarm 14 assigned to channel	03,06,16	21413	16-bit unsign int	—"—
Alarm 15 assigned to channel	03,06,16	21414	16-bit unsign int	—"—
Alarm 16 assigned to channel	03,06,16	21415	16-bit unsign int	—"—
Alarm 17 assigned to channel	03,06,16	21416	16-bit unsign int	—"—
Alarm 18 assigned to channel	03,06,16	21417	16-bit unsign int	—"—
Alarm 19 assigned to channel	03,06,16	21418	16-bit unsign int	—"—
Alarm 20 assigned to channel	03,06,16	21419	16-bit unsign int	—"—
Alarm 21 assigned to channel	03,06,16	21420	16-bit unsign int	—"—
Alarm 22 assigned to channel	03,06,16	21421	16-bit unsign int	—"—
Alarm 23 assigned to channel	03,06,16	21422	16-bit unsign int	—"—
Alarm 24 assigned to channel	03,06,16	21423	16-bit unsign int	—"—
Alarm 1 delay	03,16	21500	32-bit Float	
Alarm 2 delay	03,16	21502	32-bit Float	
Alarm 3 delay	03,16	21504	32-bit Float	
Alarm 4 delay	03,16	21506	32-bit Float	
Alarm 5 delay	03,16	21508	32-bit Float	
Alarm 6 delay	03,16	21510	32-bit Float	
Alarm 7 delay	03,16	21512	32-bit Float	
Alarm 8 delay	03,16	21514	32-bit Float	
Alarm 9 delay	03,16	21516	32-bit Float	
Alarm 10 delay	03,16	21518	32-bit Float	
Alarm 11 delay	03,16	21520	32-bit Float	
Alarm 12 delay	03,16	21522	32-bit Float	
Alarm 13 delay	03,16	21524	32-bit Float	
Alarm 14 delay	03,16	21526	32-bit Float	
Alarm 15 delay	03,16	21528	32-bit Float	
Alarm 16 delay	03,16	21530	32-bit Float	
Alarm 17 delay	03,16	21532	32-bit Float	
Alarm 18 delay	03,16	21534	32-bit Float	
Alarm 19 delay	03,16	21536	32-bit Float	

Alarm 20 delay	03,16	21538	32-bit Float	
Alarm 21 delay	03,16	21540	32-bit Float	
Alarm 22 delay	03,16	21542	32-bit Float	
Alarm 23 delay	03,16	21544	32-bit Float	
Alarm 24 delay	03,16	21546	32-bit Float	
Alarm 1 action on return	03,06,16	21600	16-bit unsign int	no(0), yes(1)
Alarm 2 action on return	03,06,16	21601	16-bit unsign int	no(0), yes(1)
Alarm 3 action on return	03,06,16	21602	16-bit unsign int	no(0), yes(1)
Alarm 4 action on return	03,06,16	21603	16-bit unsign int	no(0), yes(1)
Alarm 5 action on return	03,06,16	21604	16-bit unsign int	no(0), yes(1)
Alarm 6 action on return	03,06,16	21605	16-bit unsign int	no(0), yes(1)
Alarm 7 action on return	03,06,16	21606	16-bit unsign int	no(0), yes(1)
Alarm 8 action on return	03,06,16	21607	16-bit unsign int	no(0), yes(1)
Alarm 9 action on return	03,06,16	21608	16-bit unsign int	no(0), yes(1)
Alarm 10 action on return	03,06,16	21609	16-bit unsign int	no(0), yes(1)
Alarm 11 action on return	03,06,16	21610	16-bit unsign int	no(0), yes(1)
Alarm 12 action on return	03,06,16	21611	16-bit unsign int	no(0), yes(1)
Alarm 13 action on return	03,06,16	21612	16-bit unsign int	no(0), yes(1)
Alarm 14 action on return	03,06,16	21613	16-bit unsign int	no(0), yes(1)
Alarm 15 action on return	03,06,16	21614	16-bit unsign int	no(0), yes(1)
Alarm 16 action on return	03,06,16	21615	16-bit unsign int	no(0), yes(1)
Alarm 17 action on return	03,06,16	21616	16-bit unsign int	no(0), yes(1)
Alarm 18 action on return	03,06,16	21617	16-bit unsign int	no(0), yes(1)
Alarm 19 action on return	03,06,16	21618	16-bit unsign int	no(0), yes(1)
Alarm 20 action on return	03,06,16	21619	16-bit unsign int	no(0), yes(1)
Alarm 21 action on return	03,06,16	21620	16-bit unsign int	no(0), yes(1)
Alarm 22 action on return	03,06,16	21621	16-bit unsign int	no(0), yes(1)
Alarm 23 action on return	03,06,16	21622	16-bit unsign int	no(0), yes(1)
Alarm 24 action on return	03,06,16	21623	16-bit unsign int	no(0), yes(1)
Alarm 1 action 1	03,06,16	21700	16-bit unsign int	none(0), trapcond1(1), trapcond2(2), trapcond1and2(3), postiostate(4), mqttpublish(6)
Alarm 2 action 1	03,06,16	21701	16-bit unsign int	—"—
Alarm 3 action 1	03,06,16	21702	16-bit unsign int	—"—
Alarm 4 action 1	03,06,16	21703	16-bit unsign int	—"—
Alarm 5 action 1	03,06,16	21704	16-bit unsign int	—"—
Alarm 6 action 1	03,06,16	21705	16-bit unsign int	—"—
Alarm 7 action 1	03,06,16	21706	16-bit unsign int	—"—
Alarm 8 action 1	03,06,16	21707	16-bit unsign int	—"—
Alarm 9 action 1	03,06,16	21708	16-bit unsign int	—"—
Alarm 10 action 1	03,06,16	21709	16-bit unsign int	—"—
Alarm 11 action 1	03,06,16	21710	16-bit unsign int	—"—
Alarm 12 action 1	03,06,16	21711	16-bit unsign int	—"—
Alarm 13 action 1	03,06,16	21712	16-bit unsign int	—"—

Alarm 14 action 1	03,06,16	21713	16-bit unsign int	—"—
Alarm 15 action 1	03,06,16	21714	16-bit unsign int	—"—
Alarm 16 action 1	03,06,16	21715	16-bit unsign int	—"—
Alarm 17 action 1	03,06,16	21716	16-bit unsign int	—"—
Alarm 18 action 1	03,06,16	21717	16-bit unsign int	—"—
Alarm 19 action 1	03,06,16	21718	16-bit unsign int	—"—
Alarm 20 action 1	03,06,16	21719	16-bit unsign int	—"—
Alarm 21 action 1	03,06,16	21720	16-bit unsign int	—"—
Alarm 22 action 1	03,06,16	21721	16-bit unsign int	—"—
Alarm 23 action 1	03,06,16	21722	16-bit unsign int	—"—
Alarm 24 action 1	03,06,16	21723	16-bit unsign int	—"—
Alarm 1 action 2	03,06,16	21800	16-bit unsign int	none(0), trapcond1(1), trapcond2(2), trapcond1and2(3), postiostate(4), mqttpublish(6)
Alarm 2 action 2	03,06,16	21801	16-bit unsign int	—"—
Alarm 3 action 2	03,06,16	21802	16-bit unsign int	—"—
Alarm 4 action 2	03,06,16	21803	16-bit unsign int	—"—
Alarm 5 action 2	03,06,16	21804	16-bit unsign int	—"—
Alarm 6 action 2	03,06,16	21805	16-bit unsign int	—"—
Alarm 7 action 2	03,06,16	21806	16-bit unsign int	—"—
Alarm 8 action 2	03,06,16	21807	16-bit unsign int	—"—
Alarm 9 action 2	03,06,16	21808	16-bit unsign int	—"—
Alarm 10 action 2	03,06,16	21809	16-bit unsign int	—"—
Alarm 11 action 2	03,06,16	21810	16-bit unsign int	—"—
Alarm 12 action 2	03,06,16	21811	16-bit unsign int	—"—
Alarm 13 action 2	03,06,16	21812	16-bit unsign int	—"—
Alarm 14 action 2	03,06,16	21813	16-bit unsign int	—"—
Alarm 15 action 2	03,06,16	21814	16-bit unsign int	—"—
Alarm 16 action 2	03,06,16	21815	16-bit unsign int	—"—
Alarm 17 action 2	03,06,16	21816	16-bit unsign int	—"—
Alarm 18 action 2	03,06,16	21817	16-bit unsign int	—"—
Alarm 19 action 2	03,06,16	21818	16-bit unsign int	—"—
Alarm 20 action 2	03,06,16	21819	16-bit unsign int	—"—
Alarm 21 action 2	03,06,16	21820	16-bit unsign int	—"—
Alarm 22 action 2	03,06,16	21821	16-bit unsign int	—"—
Alarm 23 action 2	03,06,16	21822	16-bit unsign int	—"—
Alarm 24 action 2	03,06,16	21823	16-bit unsign int	—"—
Alarm 1 action 3	03,06,16	21900	16-bit unsign int	none(0), trapcond1(1), trapcond2(2), trapcond1and2(3), postiostate(4), mqttpublish(6)
Alarm 2 action 3	03,06,16	21901	16-bit unsign int	—"—
Alarm 3 action 3	03,06,16	21902	16-bit unsign int	—"—
Alarm 4 action 3	03,06,16	21903	16-bit unsign int	—"—

Alarm 5 action 3	03,06,16	21904	16-bit unsign int	—"—
Alarm 6 action 3	03,06,16	21905	16-bit unsign int	—"—
Alarm 7 action 3	03,06,16	21906	16-bit unsign int	—"—
Alarm 8 action 3	03,06,16	21907	16-bit unsign int	—"—
Alarm 9 action 3	03,06,16	21908	16-bit unsign int	—"—
Alarm 10 action 3	03,06,16	21909	16-bit unsign int	—"—
Alarm 11 action 3	03,06,16	21910	16-bit unsign int	—"—
Alarm 12 action 3	03,06,16	21911	16-bit unsign int	—"—
Alarm 13 action 3	03,06,16	21912	16-bit unsign int	—"—
Alarm 14 action 3	03,06,16	21913	16-bit unsign int	—"—
Alarm 15 action 3	03,06,16	21914	16-bit unsign int	—"—
Alarm 16 action 3	03,06,16	21915	16-bit unsign int	—"—
Alarm 17 action 3	03,06,16	21916	16-bit unsign int	—"—
Alarm 18 action 3	03,06,16	21917	16-bit unsign int	—"—
Alarm 19 action 3	03,06,16	21918	16-bit unsign int	—"—
Alarm 20 action 3	03,06,16	21919	16-bit unsign int	—"—
Alarm 21 action 3	03,06,16	21920	16-bit unsign int	—"—
Alarm 22 action 3	03,06,16	21921	16-bit unsign int	—"—
Alarm 23 action 3	03,06,16	21922	16-bit unsign int	—"—
Alarm 24 action 3	03,06,16	21923	16-bit unsign int	—"—
Alarm 1 status	03	22000	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 2 status	03	22001	16-bit unsign int	—"—
Alarm 3 status	03	22002	16-bit unsign int	—"—
Alarm 4 status	03	22003	16-bit unsign int	—"—
Alarm 5 status	03	22004	16-bit unsign int	—"—
Alarm 6 status	03	22005	16-bit unsign int	—"—
Alarm 7 status	03	22006	16-bit unsign int	—"—
Alarm 8 status	03	22007	16-bit unsign int	—"—
Alarm 9 status	03	22008	16-bit unsign int	—"—
Alarm 10 status	03	22009	16-bit unsign int	—"—
Alarm 11 status	03	22010	16-bit unsign int	—"—
Alarm 12 status	03	22011	16-bit unsign int	—"—
Alarm 13 status	03	22012	16-bit unsign int	—"—
Alarm 14 status	03	22013	16-bit unsign int	—"—
Alarm 15 status	03	22014	16-bit unsign int	—"—
Alarm 16 status	03	22015	16-bit unsign int	—"—
Alarm 17 status	03	22016	16-bit unsign int	—"—
Alarm 18 status	03	22017	16-bit unsign int	—"—
Alarm 19 status	03	22018	16-bit unsign int	—"—
Alarm 20 status	03	22019	16-bit unsign int	—"—
Alarm 21 status	03	22020	16-bit unsign int	—"—
Alarm 22 status	03	22021	16-bit unsign int	—"—
Alarm 23 status	03	22022	16-bit unsign int	—"—
Alarm 24 status	03	22023	16-bit unsign int	—"—

MB relay 1 description	03,16	22200	16 bytes UTF-8	
MB relay 2 description	03,16	22208	16 bytes UTF-8	
MB relay 3 description	03,16	22216	16 bytes UTF-8	
MB relay 4 description	03,16	22224	16 bytes UTF-8	
MB relay 5 description	03,16	22232	16 bytes UTF-8	
MB relay 6 description	03,16	22240	16 bytes UTF-8	
MB relay 7 description	03,16	22248	16 bytes UTF-8	
MB relay 8 description	03,16	22256	16 bytes UTF-8	
MB relay 9 description	03,16	22264	16 bytes UTF-8	
MB relay 10 description	03,16	22272	16 bytes UTF-8	
MB relay 11 description	03,16	22280	16 bytes UTF-8	
MB relay 12 description	03,16	22288	16 bytes UTF-8	
MB relay 13 description	03,16	22296	16 bytes UTF-8	
MB relay 14 description	03,16	22304	16 bytes UTF-8	
MB relay 15 description	03,16	22312	16 bytes UTF-8	
MB relay 16 description	03,16	22320	16 bytes UTF-8	
MB relay 17 description	03,16	22328	16 bytes UTF-8	
MB relay 18 description	03,16	22336	16 bytes UTF-8	
MB relay 19 description	03,16	22344	16 bytes UTF-8	
MB relay 20 description	03,16	22352	16 bytes UTF-8	
MB relay 21 description	03,16	22360	16 bytes UTF-8	
MB relay 22 description	03,16	22368	16 bytes UTF-8	
MB relay 23 description	03,16	22376	16 bytes UTF-8	
MB relay 24 description	03,16	22384	16 bytes UTF-8	
MB relay 1 activated from	03,06,16	22400	16-bit unsign int	webHttpApi(0), al01(1), al02(2), al03(3), al04(4), al05(5), al06(6), al07(7), al08(8), al09(9), al10(10), al11(11), al12(12), al13(13), al14(14), al15(15), al16(16), al17(17), al18(18), al19(19), al20(20), al21(21), al22(22), al23(23), al24(24), anyAlarm(25), schedule1(26), schedule2(27), schedule3(28), schedule4(29)
MB relay 2 activated from	03,06,16	22401	16-bit unsign int	—"—
MB relay 3 activated from	03,06,16	22402	16-bit unsign int	—"—
MB relay 4 activated from	03,06,16	22403	16-bit unsign int	—"—
MB relay 5 activated from	03,06,16	22404	16-bit unsign int	—"—
MB relay 6 activated from	03,06,16	22405	16-bit unsign int	—"—
MB relay 7 activated from	03,06,16	22406	16-bit unsign int	—"—
MB relay 8 activated from	03,06,16	22407	16-bit unsign int	—"—

MB relay 9 activated from	03,06,16	22408	16-bit unsign int	—"—
MB relay 10 activated from	03,06,16	22409	16-bit unsign int	—"—
MB relay 11 activated from	03,06,16	22410	16-bit unsign int	—"—
MB relay 12 activated from	03,06,16	22411	16-bit unsign int	—"—
MB relay 13 activated from	03,06,16	22412	16-bit unsign int	—"—
MB relay 14 activated from	03,06,16	22413	16-bit unsign int	—"—
MB relay 15 activated from	03,06,16	22414	16-bit unsign int	—"—
MB relay 16 activated from	03,06,16	22415	16-bit unsign int	—"—
MB relay 17 activated from	03,06,16	22416	16-bit unsign int	—"—
MB relay 18 activated from	03,06,16	22417	16-bit unsign int	—"—
MB relay 19 activated from	03,06,16	22418	16-bit unsign int	—"—
MB relay 20 activated from	03,06,16	22419	16-bit unsign int	—"—
MB relay 21 activated from	03,06,16	22420	16-bit unsign int	—"—
MB relay 22 activated from	03,06,16	22421	16-bit unsign int	—"—
MB relay 23 activated from	03,06,16	22422	16-bit unsign int	—"—
MB relay 24 activated from	03,06,16	22423	16-bit unsign int	—"—
MB relays state after restart	03,06	22450	16-bit unsign int	off(0), on(1), laststate(2)
Watchdog 1 state	03	22500	16-bit unsign int	false(0), true(1)
Watchdog 2 state	03	22501	16-bit unsign int	false(0), true(1)
Watchdog 3 state	03	22502	16-bit unsign int	false(0), true(1)
Watchdog 4 state	03	22503	16-bit unsign int	false(0), true(1)
Function 1 state	03	23000	16-bit unsign int	false(0), true(1)
Function 2 state	03	23001	16-bit unsign int	false(0), true(1)
Function 3 state	03	23002	16-bit unsign int	false(0), true(1)
Function 4 state	03	23003	16-bit unsign int	false(0), true(1)
Function 1 alarm state	03	23100	16-bit unsign int	normal(0), alarm(1)
Function 2 alarm state	03	23101	16-bit unsign int	normal(0), alarm(1)
Function 3 alarm state	03	23102	16-bit unsign int	normal(0), alarm(1)
Function 4 alarm state	03	23103	16-bit unsign int	normal(0), alarm(1)
Save configuration	03,06	24000	16-bit unsign int	unsaved(0), saved(1)
Restart device	03,06	24001	16-bit unsign int	cancel(0), restart(1)
HW error	03	24002	16-bit unsign int	noErr(0), hwErr(1)
Device ID	03	24100	18 bytes UTF-8	Example: 5c:32:c5:00:ac:52
Hostname	03	24200	16 bytes UTF-8	
Device IP	03	24300	16 bytes UTF-8	Example: 192.168.1.2

Note 1: For the MODBUS relays, command 01 allows only one relay.
If there is no connection to the MODBUS relay, on the commands Read Coils and Write Single Coil, the TCW242 will return error code 04 (SLAVE DEVICE FAILURE).

Appendix D

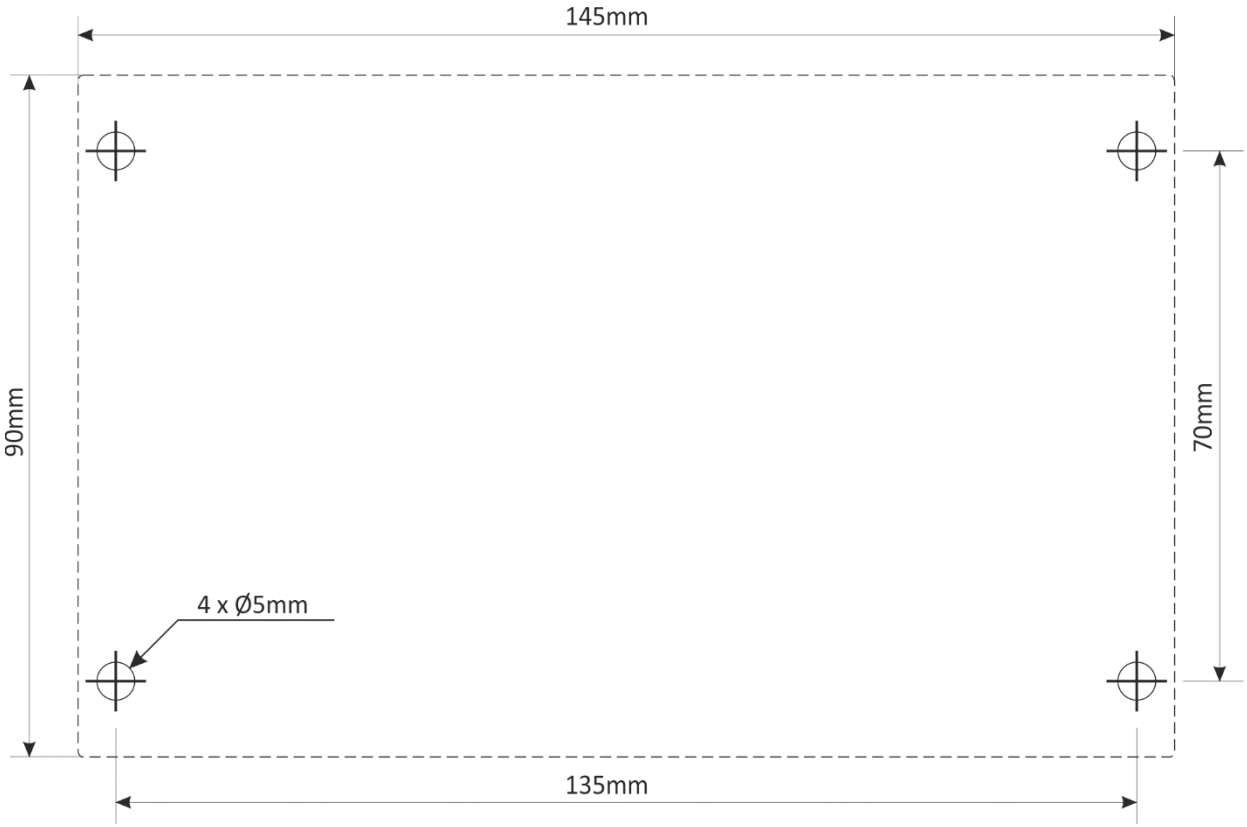


Fig.1

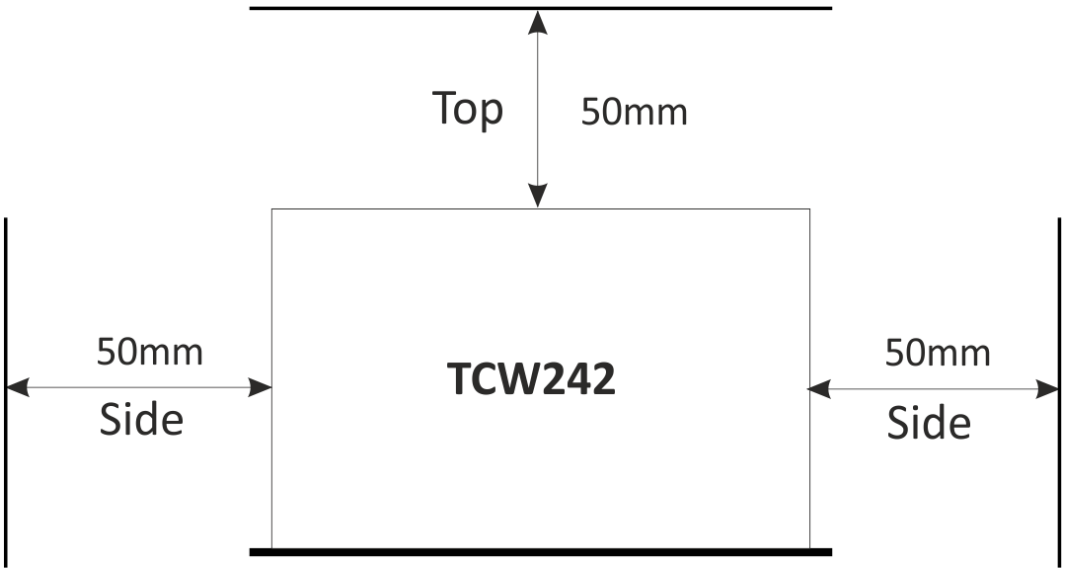


Fig.2