control solutions

TERACOM



Environmental Monitoring Module
TCW320E-EMM / TCW320W-EMM

USER MANUAL

1. Product Overview

The TCW320x-EMM is an Environmental Monitoring Module designed for acquisition, processing, and integration of environmental data in networked systems.
The module is built around an API-first architecture, making it suitable for seamless integration with third-party software and automation systems. At the same time, the embedded web interface provides convenient access to all configuration and monitoring functions. The module is intended for unattended operation and programmatic control during normal use, while the web interface is primarily used for configuration, commissioning, diagnostics, and maintenance.
The TCW320x-EMM is a compact and self-contained Environmental Monitoring Module, combining sensor inputs, control outputs, and network connectivity in a single device. It is designed for reliable long-term operation in distributed monitoring and automation applications.
This user manual applies to the following device variants:

- TCW320E-EMM – Ethernet variant
- TCW320W-EMM – Wi-Fi variant

Both variants provide identical monitoring and control functionality and differ only by the network interface.

2. Features

- Integrated web-based configuration interface with HTTPS support for secure access
- API-first module design with Teracom HTTP API for system integration and machine-to-machine communication
- Support for 1-Wire (up to 4 sensors) and Modbus RTU (up to 10 devices)
- 2 analog inputs with 16-bit resolution, 2 digital inputs, and 2 relay outputs
- Email alert notifications
- Certificate-based security (optional)
- Multiple user accounts with different access levels
- 2.4 GHz Wi-Fi or 10/100 Mbps Ethernet connectivity, depending on the device variant
- Remote firmware update support
- Wall mounting, as well as DIN rail and 19-inch rack mounting using dedicated mounting accessories

3. Applications

The TCW320x-EMM is intended for use in applications requiring reliable monitoring of environmental conditions and integration with centralized monitoring and management systems.
Typical applications include:

- Environmental monitoring in data centers and server rooms, including temperature, humidity, and infrastructure-related parameters
- Monitoring of technical rooms and equipment areas in broadcasting facilities
- Environmental and infrastructure monitoring in telecommunications facilities and network nodes
- Integration into centralized monitoring systems for building management and industrial environments
- Remote monitoring of distributed sites where programmatic access to measurement data is required
- Automation and control applications requiring integration with external systems via network-based APIs

The module is suitable for installations where unattended operation, remote access, and system-to-system integration are required.

4. Specifications

- Physical characteristics
  Dimensions: 109 x 32 x 93 mm
  Weight: 110 g
- Environmental limits
  Operating temperature range: -20 to 60 °C
  Operating relative humidity range: 10 to 90 %RH (non-condensing)
  Storage temperature range: -20 to 60 °C
  Storage relative humidity range: 10 to 90 %RH (non-condensing)
  Ingress protection: IP20
- Power requirements
  Operating voltage range (including -15/+20 % according to IEC 62368-1): 10 to 28 VDC
  Current consumption (Ethernet variant): 110 mA @ 12 VDC (with both relays on)
  Current consumption (Wi-Fi variant): 110 mA @ 12 VDC (with both relays on)
- Digital inputs
  Isolation: Non-isolated
  To be used with: Dry contacts, Open drain, and Open collector outputs
  Pull up: Internal
  Sampling rate: 10 ms
  Digital filtering: 30 ms minimum (user-expandable via the web interface)
- Analog inputs
  Isolation: Non-isolated
  Input range: 0 to 10 VDC
  Resolution: 16-bit ADC
  Digital processing: Averaging 5 interleaved samples per channel every 1 second
  Input impedance: ≥ 1 MΩ
- RS-485 interface for Modbus RTU
  Isolation: Non-isolated
  Maximum number of supported Modbus RTU devices: 10
  Supported baud rates: 2400, 4800, 9600, 19200, 38400, 57600
- 1-Wire interface
  Maximum number of supported 1-Wire sensors: 4
  Supply output voltage: 5.0 ± 0.3 VDC
  Maximum output current: 0.2 A
- Relays
  Type: SPST (1 Form A)
  Rated loads: 3 A @ 24 VDC/30 VAC (resistive load)
  Mechanical endurance: 5 million operations (min.)

  Caution: The device does not contain internal overcurrent protection on the relay contact lines. External fuses or short-circuit current-limiting circuit breakers rated to 3 A must be used for overcurrent protection of the connected lines.
- Internal FLASH memory
  Endurance: 100 000 cycles (each configuration change is counted as one memory cycle)
- Wi-Fi interface
  Wi-Fi standard: IEEE 802.11 b/g/n
  Frequency band: 2.4 GHz
  Operating modes: Station (STA) and Access Point (AP)
  Security: WPA/WPA2-PSK

- Ethernet interface
  Ethernet standard: IEEE 802.3
  Supported data rates: 10/100 Mbps
  Auto-negotiation support
  Full-duplex and half-duplex operation

5. Pinout

This section describes the connectors and pin assignments of the TCW320x-EMM module. The pin numbering shown in the figures corresponds to the numbering printed on the device enclosure. Ensure correct wiring according to the pinout before applying power to the device.

Connector 1

Power supply – 2.1 x 5.5 mm DC jack, center positive

Connector 2

Pin 1	Digital Input 1
Pin 2	Ground
Pin 3	Digital Input 2
Pin 4	Analog Input 1
Pin 5	Ground
Pin 6	Analog Input 2

Connector 3

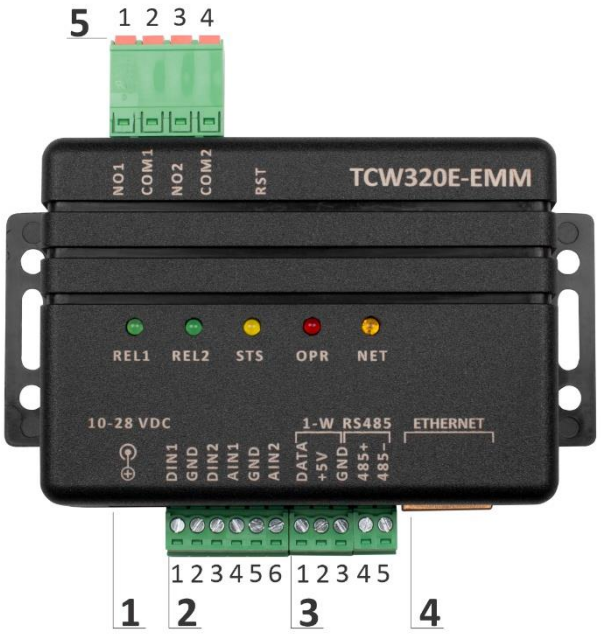
Pin 1	1-Wire Data
Pin 2	+5 V
Pin 3	Ground
Pin 4	RS485+
Pin 5	RS485-

Connector 4

Ethernet connector
(not available on the Wi-Fi variant)

Connector 5

Pin 1	Normally Open Contact Relay 1
Pin 2	Common Contact Relay 1
Pin 3	Normally Open Contact Relay 2
Pin 4	Common Contact Relay 2



The connectors are removable terminal blocks. Ensure that the wiring is performed with the power supply disconnected.

6. LED Indicators

The following indicators show the current status of the module:

- STS (yellow) - Flashes to indicate that the module's main program is running.
- OPR (red) - Steady ON during normal operation. Blinks in sync with the STS LED in the event of a hardware error.
- REL1/REL2 (green) - Illuminates when the corresponding relay is activated (NO and COM contacts are closed).
- NET (orange) - Provides network status information depending on the model

For TCW320E-EMM (Ethernet model):

Steady ON: Network link is established.

Blinking: Indicates network activity (data transfer).

For TCW320W-EMM (Wi-Fi model):

Pattern (ON / OFF)	Wi-Fi Station (STA) Mode	Wi-Fi Access Point (AP) Mode
100ms / 100ms	Wi-Fi is not connected.	AP is not initialized.
100ms / 400ms	Wi-Fi is connected; waiting for DHCP IP address.	—
500ms / 2500ms	IP address assigned (Static or DHCP).	AP is active; no clients connected.
500ms / 500ms	Successful ping to the Gateway.	AP is active; at least one client is connected.

7. Installation and Initial Configuration

The device must be installed by qualified personnel.

It is not designed for direct outdoor installation and must be installed in a suitable enclosure when used in outdoor environments.

The installation process includes mounting the device, connecting it to an IP network, connecting the required peripherals, supplying power, and performing the initial configuration through a web browser.

7.1. Mounting

The TCW320x-EMM must be installed in a clean, dry, and non-flammable environment. Adequate ventilation is recommended, especially in installations with high ambient temperatures.

For wall mounting, use two plastic dowels (e.g. Würth GmbH 0912 802 002) and two dowel screws (e.g. Würth GmbH 0157 06 70) to secure the device. Mechanical dimensions and mounting hole positions are shown in Figure 1 in Appendix D.

A minimum clearance of 50 mm on all sides must be maintained to ensure proper ventilation and electrical isolation, as shown in Figure 2 in Appendix D.

The device can also be mounted on a standard DIN rail (35 mm × 7.55 mm) using a dedicated DIN rail mounting accessory.

For installations in 19-inch racks, the device can be mounted using a dedicated 19-inch rack mounting accessory.

7.2. Connection

Warning! Disconnect the power supply before wiring the device.

Follow these steps for correct wiring:

- Disconnect the power supply
- Connect the wires to the appropriate terminals
- Reconnect the power supply.

Ensure that all wires are securely attached to the terminals and properly tightened. Improper wiring or incorrect configuration may cause permanent damage to the TCW320x-EMM or to connected equipment.

7.2.1 Power supply connection

The TCW320x-EMM must be powered using the SYS1308(N)-2412-W2E power adapter or an equivalent power supply unit. The power supply must be suitable for overvoltage category II and certified for safety compliance.

The power supply device should be capable of withstanding short circuits and secondary circuit overloads. The equipment must be installed in a way that allows easy access to the power supply disconnect during operation.

7.2.2 RS-485 connection

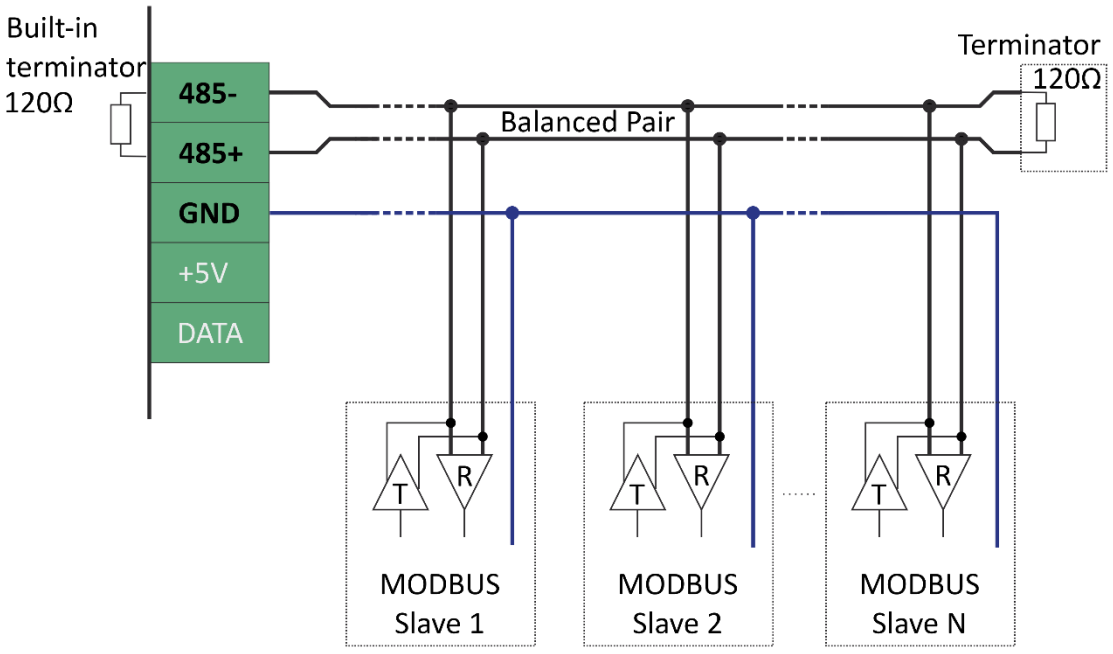
The RS-485 interface of the device is not galvanically isolated from the device power supply ground. When connecting RS-485 devices, special attention must be paid to grounding and potential differences between devices.

RS-485 is a differential serial communication standard suitable for multi-drop networks and operation in electrically noisy industrial environments. The device supports communication with Modbus RTU slave devices over RS-485 and allows connection of up to 247 devices, depending on bus topology and the characteristics of the connected equipment.

Since the RS-485 interface is non-isolated, all devices connected to the same RS-485 bus should share a common reference ground. Connecting devices with different ground potentials may lead to communication errors or permanent damage to the equipment. In installations where ground potential differences are expected, the use of external RS-485 isolation or signal converters is strongly recommended.

To ensure reliable communication, the RS-485 bus must be terminated with 120 Ω termination resistors at both ends of the line. The device includes a built-in 120 Ω termination resistor and should be installed at one end of the bus. A second 120 Ω termination resistor must be installed at the opposite end of the bus.

It is strongly recommended to use twisted-pair cables such as UTP or FTP for RS-485 wiring. When shielded cable is used, the shield should be connected to ground at one point only, typically at the module side, to reduce electromagnetic interference and avoid ground loops.



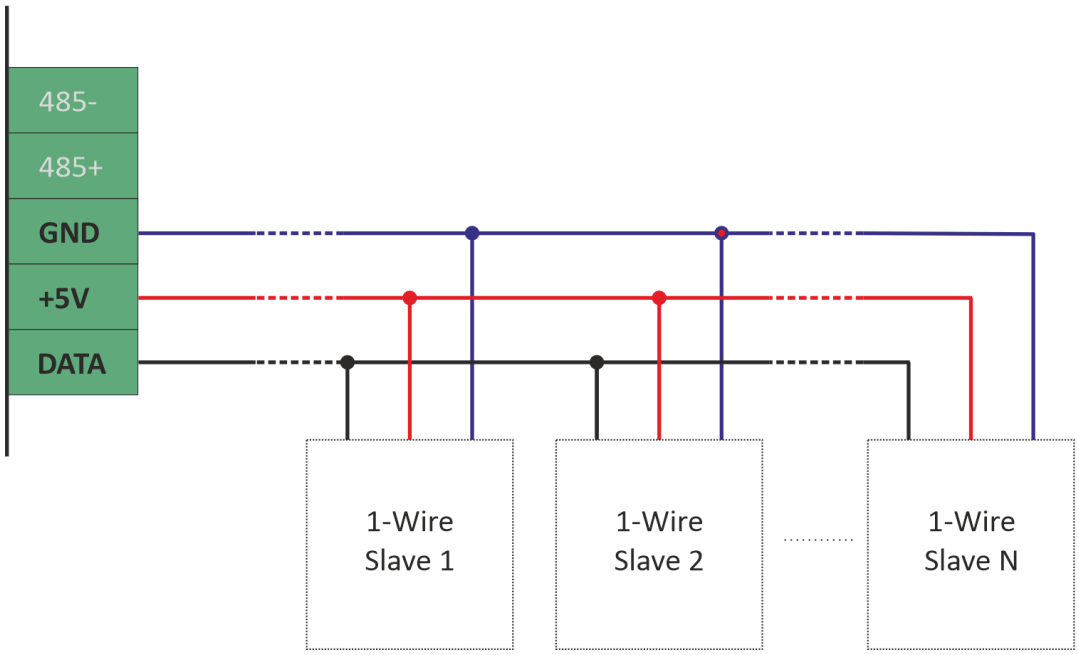
For best reliability, the total RS-485 cable length should be kept within 30 meters. Although operation over longer distances may be possible depending on baud rate and installation conditions, shorter cable lengths significantly improve signal integrity and communication stability.

7.2.3 1-Wire connection

1-Wire, a registered trademark of Analog Devices, Inc., is designed for connecting multiple sensors over short wiring distances. It is not suitable for long distances or environments with significant

electromagnetic interference (EMC). For best practices, refer to the Guidelines for Reliable Long-Line 1-Wire Networks.

The 1-Wire interface is implemented as a three-wire connection consisting of power, ground, and data lines.

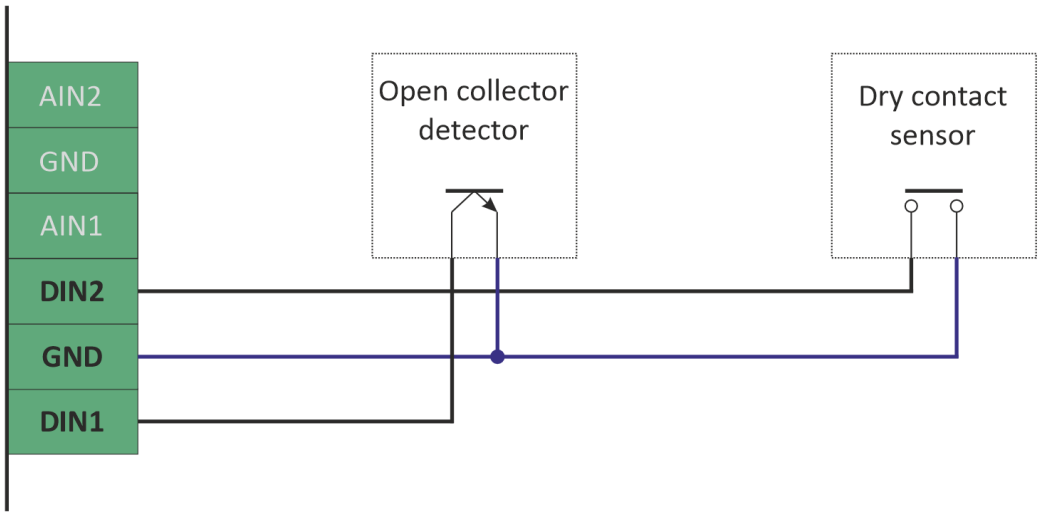


A linear (bus) topology is recommended for reliable operation, with a total cable length of up to 30 meters. This topology provides predictable signal characteristics and stable communication with connected 1-Wire devices.

Star topologies are not recommended, as they may cause signal reflections and communication instability, especially with longer cable lengths or multiple connected devices.

7.2.4 Digital inputs connection

The digital inputs are not isolated from the power supply ground and include internal pull-up resistors.



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 digital input and the other terminal to the corresponding GND terminal.

The digital inputs can also be connected to open-collector or open-drain outputs. In this case, the output transistor pulls the corresponding digital input to GND when activated.

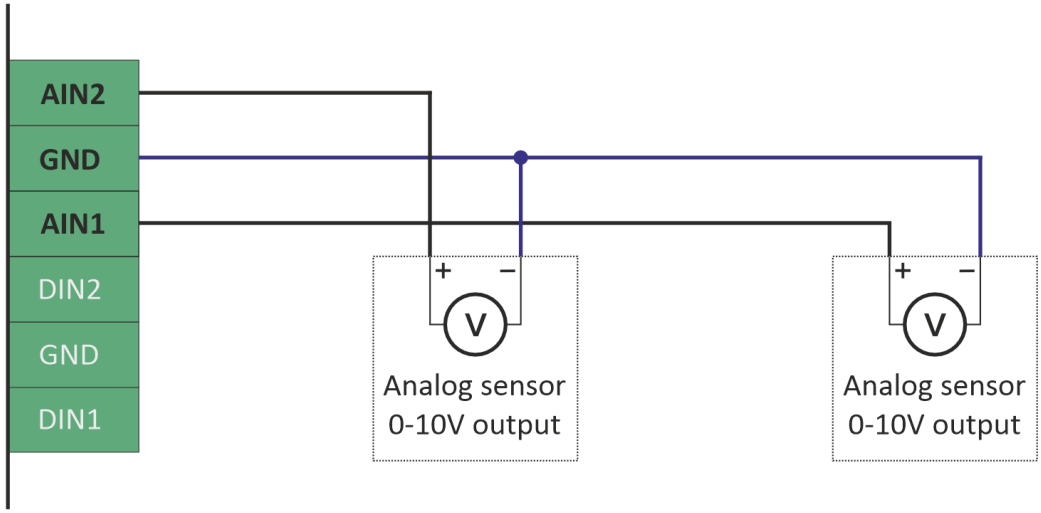
The input state changes when the corresponding digital input is connected to GND through an external contact or transistor output.

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

7.2.5 Analog inputs connection

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

The analog inputs operate in voltage mode (0 to 10 VDC). They can be connected directly to sensors, transmitters, and other devices providing compatible analog voltage outputs.



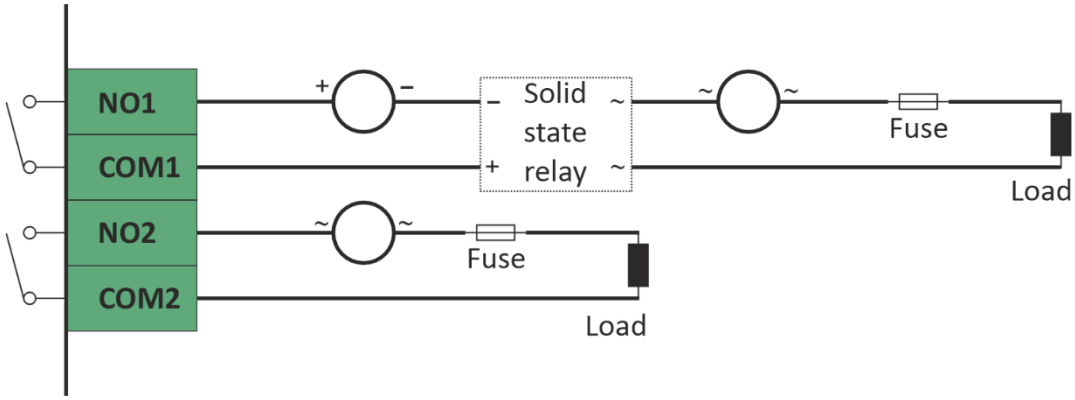
Connect the positive output of the sensor or transmitter to the selected analog input and the negative output to the corresponding GND terminal.

To ensure measurement accuracy, it is recommended to use shielded cables and route the signal wiring away from power cables, contactors, motors, and other sources of electrical interference.

The maximum cable length between the module and the connected device should not exceed 30 m.

7.2.6 Relay connection

The TCW320x-EMM provides two relay outputs with SPST (Single Pole Single Throw) contacts. Each relay is equipped with Normally Open (NO) and Common (COM) terminals.



When the relay is energized, the COM terminal is connected to the NO terminal. When the relay is de-energized, the circuit remains open.

The relay outputs can be used for direct switching of AC and DC loads within the specified ratings, or for controlling external devices such as contactors and solid-state relays (SSRs).

The wiring examples shown below illustrate typical relay connections for DC loads, AC loads, and external solid-state relays.

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.

7.2.7 Ethernet connection

The Ethernet interface is used for network communication.

The TCW320E-EMM is connected to an Ethernet network using a standard Ethernet cable. The Ethernet interface supports 10/100 Mbps operation with auto-negotiation of speed and duplex mode.

The device should be connected to a network switch, router, or directly to a host system, depending on the network topology and application requirements.

The use of standard CAT5e or higher Ethernet cables is recommended.

Ensure that the Ethernet cable is securely connected and routed away from sources of strong electromagnetic interference, such as power cables, contactors, and motors. Improper cable routing may result in communication instability or packet loss.

Network parameters can be configured through the web interface. Depending on the network environment, the device can obtain an IP address automatically via DHCP or be configured with a static IP address.

7.2.8 Wi-Fi connection

The Wi-Fi interface is used for network communication and access to the device web interface on the TCW320W-EMM variant.

The device supports two Wi-Fi operating modes: Access Point (AP) mode and Station (client) mode. By default, the device operates in Access Point (AP) mode.

In Access Point mode, the device creates its own Wi-Fi network, allowing direct connection from a computer or mobile device for initial configuration. This mode is intended for commissioning and setup when no existing Wi-Fi infrastructure is available.

In Station mode, the device connects to an existing Wi-Fi network as a client. In this mode, network communication is performed through the connected wireless network, and the device becomes accessible to other systems on the same IP network.

Wi-Fi configuration parameters, including operating mode and network credentials, are configured through the web interface. After configuration, the device may require a restart for the new settings to take effect.

For reliable operation, ensure adequate Wi-Fi signal strength at the installation location and avoid placing the device near sources of strong electromagnetic interference or metal enclosures that may attenuate the wireless signal.

7.3. Initial network configuration

This section describes how to access the device web interface using the factory default network settings during initial installation.

7.3.1 Default network settings

By default, the devices are shipped with the following network settings, depending on the device variant.

TCW320E-EMM (Ethernet variant):

- IP address: 192.168.1.2
- Subnet mask: 255.255.255.0
- Default gateway: 192.168.1.1
- Web interface user name: admin
- Web interface password: admin

TCW320W-EMM (Wi-Fi variant, Access Point mode):

- Operating mode: Access Point (AP)
- SSID: tcw320ap; Password: tcw320ap
- Security: WPA2-PSK
- IP address: 192.168.4.1
- Subnet mask: 255.255.255.0
- Web interface user name: admin
- Web interface password: admin

These default settings allow direct access to the device web interface during initial installation, without requiring prior network configuration.

7.3.2 Initial access via Ethernet

To access the TCW320E-EMM via Ethernet, connect the device to a computer or to an Ethernet network using a standard Ethernet cable.

If the connected computer is configured to obtain an IP address automatically and no DHCP server is available on the network, assign a temporary static IP address to the computer in the same subnet as the device.

Once the computer and the device are in the same IP network, open a web browser and enter the device IP address. When prompted, log in using the default user name and password specified above.

7.3.3 Initial access via Wi-Fi Access Point

On the TCW320W-EMM, the device operates in Wi-Fi Access Point mode by default.

Connect a computer or mobile device to the Wi-Fi network created by the device using the default SSID specified above. After establishing the wireless connection, open a web browser and access the device web interface using the default IP address. When prompted, log in using the default user name and password specified above.

8. Set up concept

The primary configuration interface of the module is the embedded web interface. It provides access to all configuration parameters, including network settings, connected sensors, alarm conditions, communication services, and system administration.

For reliable operation, it is recommended to configure the module in the following sequence:

- Network configuration
  Configure the Ethernet or Wi-Fi network settings and verify that the module is accessible from the intended network.
- Time configuration
  Configure the time zone and synchronize the module clock using an NTP server or set the time

manually. Accurate system time is required for event timestamps, email notifications, and HTTP POST messages.

- Sensor configuration
  Configure the connected 1-Wire sensors, Modbus RTU devices, analog inputs, and digital inputs. Verify that all measured values are displayed correctly on the Monitoring page.
- Input and output configuration
  Assign descriptions, engineering units, scaling parameters, relay operating modes, and other application-specific settings.
- Alarm conditions
  Configure the trigger conditions, notification settings, communication loss monitoring, and relay actions according to the application requirements.
- Communication services
  Configure the required services, such as SMTP for email notifications and HTTP POST for communication with remote monitoring systems.
- Security and administration
  Configure user accounts, authentication, certificates (if required), and perform a backup of the completed configuration.

After completing the configuration, verify correct operation by checking sensor readings, alarm notifications, relay control, and communication with external systems.

In summary, the recommended setup sequence is:

Network configuration → Time configuration → Sensor configuration → Input & Output configuration → Alarm conditions → Communication services → Security and administration → System verification

9. Web Interface

Access to the device web interface is protected by HTTP(S) authentication.

When a user accesses the device IP address using a web browser over HTTP or HTTPS, the browser displays a standard authentication dialog requesting a user name and password. No separate login page is used.

After entering valid credentials, the web interface becomes accessible and the device pages are loaded in the browser.

The device supports multiple user accounts with different access levels. The access level associated with the authenticated user determines which pages and functions are available in the web interface.

Default login credentials are provided for initial access and should be changed after commissioning to prevent unauthorized access.

9.1. Monitoring

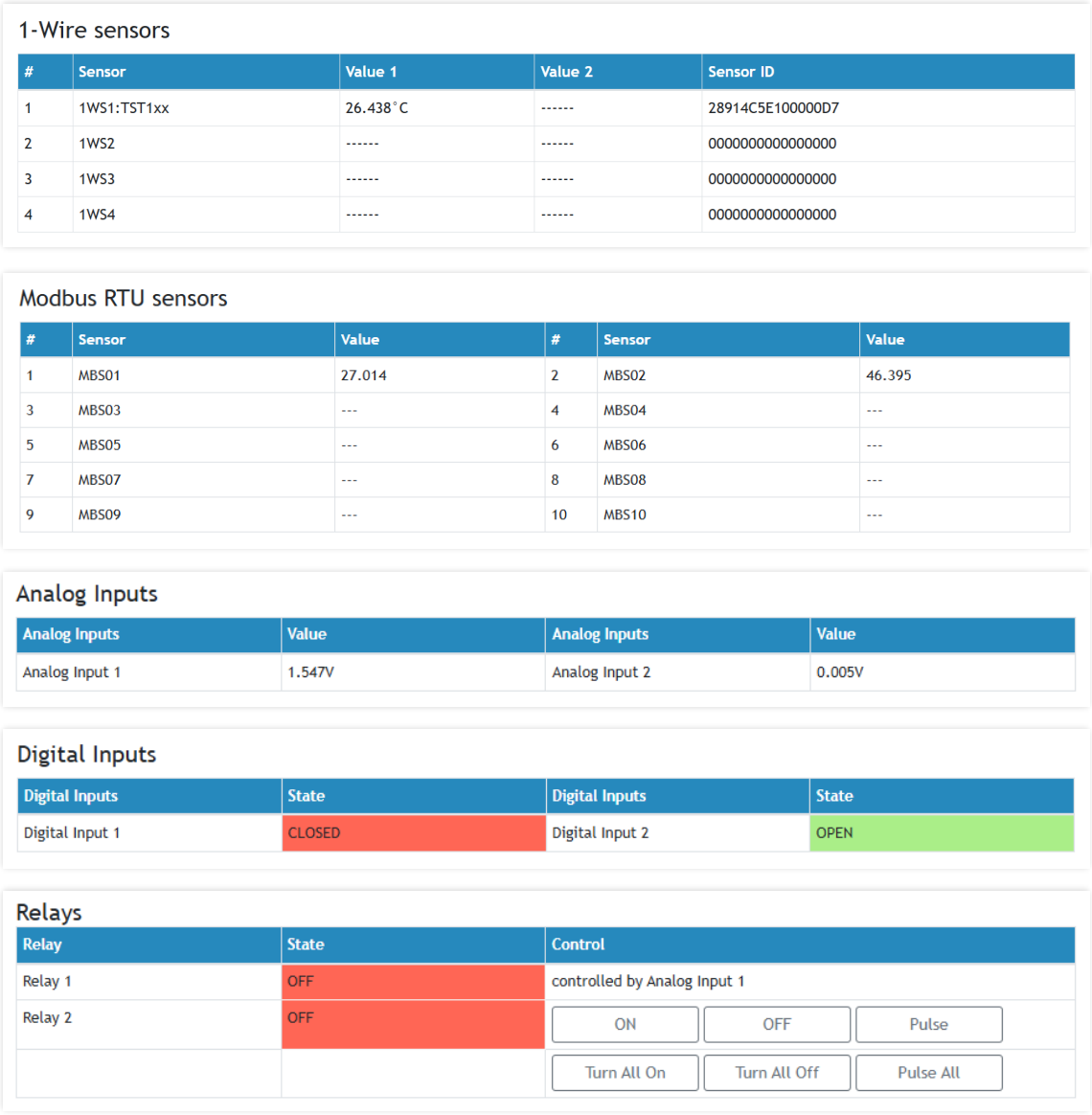
The Monitoring page provides real-time information about the current state of the module inputs, connected sensors, and relay outputs.

This page is intended for observation and status verification of:

- Analog inputs
- Digital inputs
- Connected 1-Wire sensors
- Configured Modbus RTU registers
- Relay outputs

The Monitoring page displays the current values of all enabled measurement sources, including sensor readings, analog input values, digital input states, and relay status. For 1-Wire sensors, the sensor identifier is also displayed to assist with sensor verification and diagnostics.

The displayed content on the Monitoring page is configurable through the System page. Each section can be individually enabled or disabled using dedicated checkboxes. This allows the user to display only the required information and minimize unnecessary information on the page.



1-Wire sensors

#	Sensor	Value 1	Value 2	Sensor ID
1	1WS1:TST1xx	26.438 °C	------	28914C5E100000D7
2	1WS2	------	------	0000000000000000
3	1WS3	------	------	0000000000000000
4	1WS4	------	------	0000000000000000

Modbus RTU sensors

#	Sensor	Value	#	Sensor	Value
1	MBS01	27.014	2	MBS02	46.395
3	MBS03	---	4	MBS04	---
5	MBS05	---	6	MBS06	---
7	MBS07	---	8	MBS08	---
9	MBS09	---	10	MBS10	---

Analog Inputs

Analog Inputs	Value	Analog Inputs	Value
Analog Input 1	1.547V	Analog Input 2	0.005V

Digital Inputs

Digital Inputs	State	Digital Inputs	State
Digital Input 1	CLOSED	Digital Input 2	OPEN

Relays

Relay	State	Control
Relay 1	OFF	controlled by Analog Input 1
Relay 2	OFF	ON / OFF / Pulse
		Turn All On / Turn All Off / Pulse All

The displayed values are refreshed automatically and provide a convenient overview of the current operating condition of the module and all connected devices.

For users logged in with the User account, the Monitoring page is read-only and relay control buttons are not available. Relay states are displayed for monitoring purposes only.

For accounts with higher access levels, relay control functions are available directly from the Monitoring page, allowing manual control of relay outputs when required.

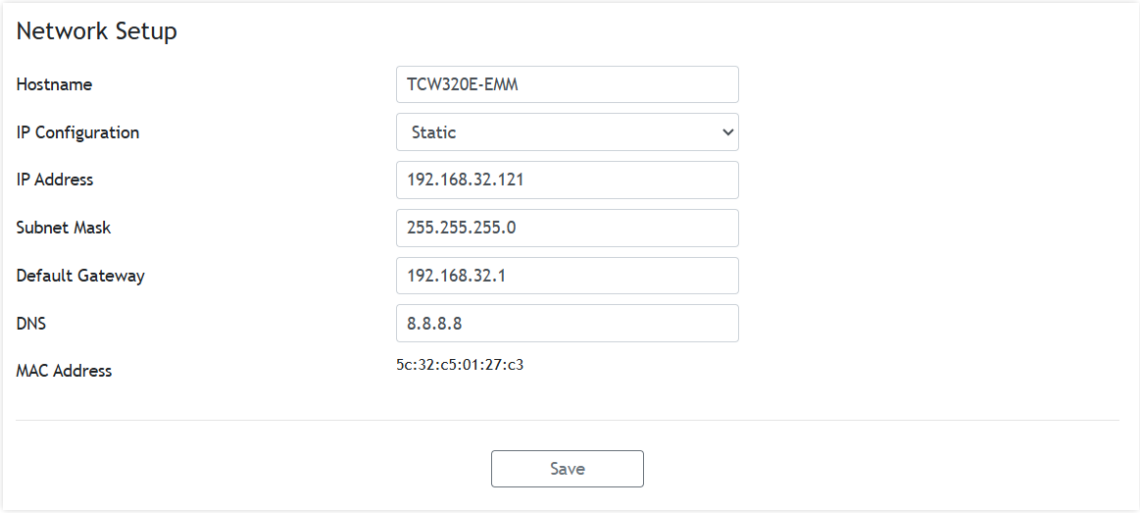
The Monitoring page can be used during normal operation to observe input states, measurement values, alarm-related parameters, and relay status. When relay control functions are available, actions performed on this page directly affect device operation.

9.2. Setup

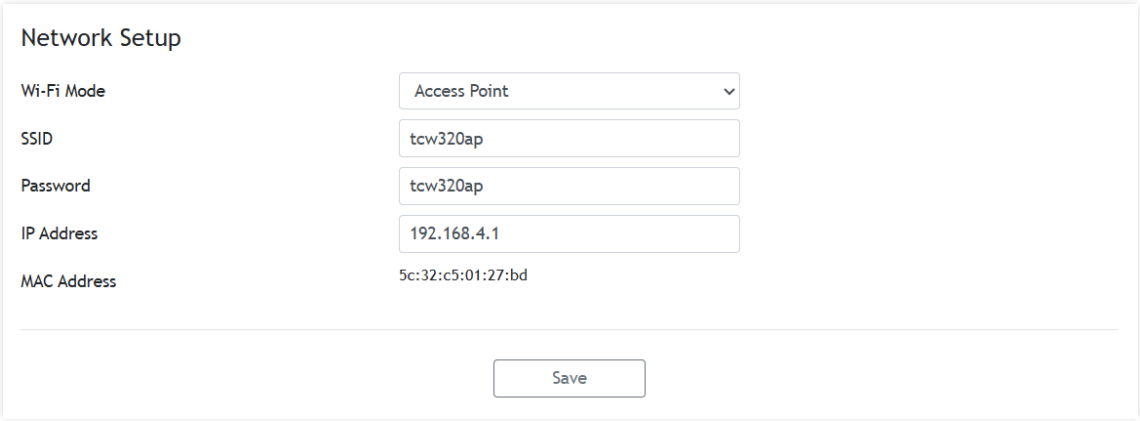
9.2.1 Network settings

The Network settings section is used to configure the network parameters of the device, depending on the selected network interface and operating mode.

For the Ethernet variant, this section allows configuration of IP addressing parameters, including automatic configuration via DHCP or manual assignment of a static IP address.



For the Wi-Fi variant, network settings include selection of the Wi-Fi operating mode, configuration of wireless network parameters, and IP addressing settings. In Station mode, the device connects to an existing wireless network, while in Access Point mode it provides its own wireless network for direct access.



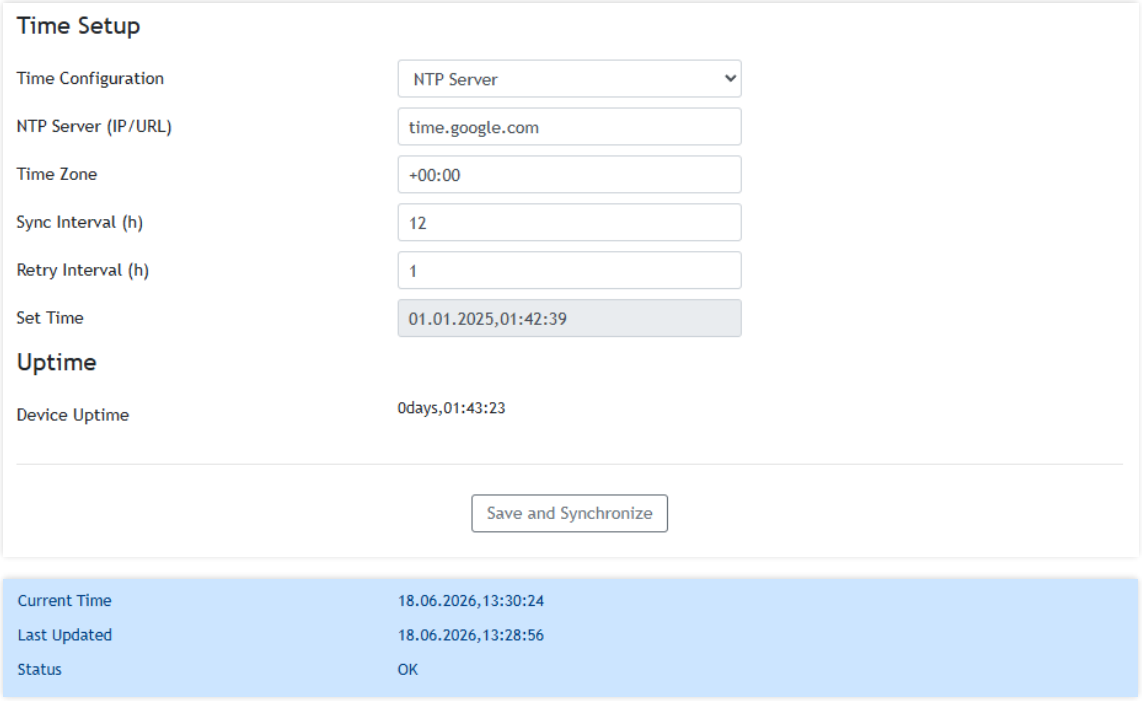
Changes to network settings affect how the device is accessed and how it communicates with other systems on the network. Incorrect configuration may result in loss of connectivity to the web interface.

9.2.2 Time

The Time section is used to configure the system time of the module and to define how time synchronization is performed.

Accurate system time is essential for correct device operation, as it is used for timestamping events, measurements, and notifications. The configured time directly affects the timestamps included in email alert messages and HTTP POST transmissions.

The module supports both manual time configuration and automatic time synchronization using an NTP server. In order to synchronize the time automatically, the device must have access to the Internet. Without Internet connectivity, time synchronization using an external NTP server is not possible.



When NTP synchronization is enabled and the Save and Synchronize button is used, the device updates its internal clock. After a successful synchronization, the current system time is updated and displayed accordingly in the web interface.

It is recommended to use well-known and reliable NTP servers with good response times to ensure accurate and timely synchronization of the system clock.

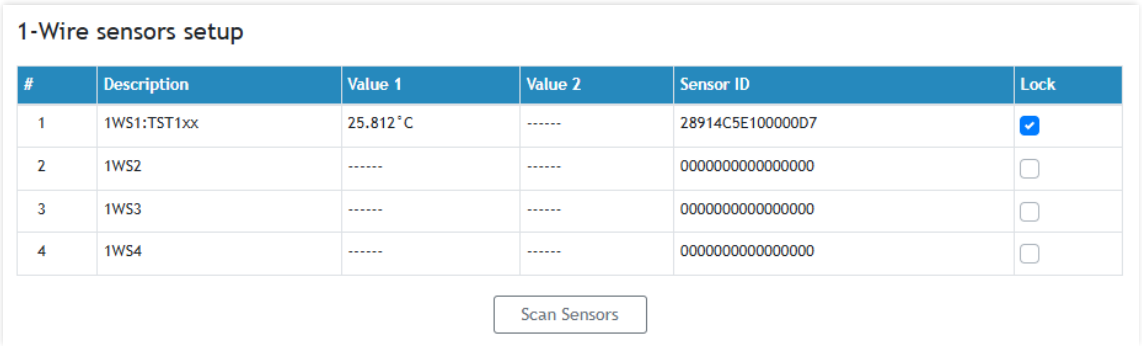
When the device is installed in a closed or internal network without direct Internet access, it is recommended to use the local network gateway as the NTP server, provided that the gateway offers an NTP service. If no NTP service is available within the network, the system time should be set manually.

The time zone setting defines the local time offset applied to the system time and is used for displaying timestamps in the web interface and in generated notifications.

The Time section also displays the current system time, synchronization status, and device uptime information for reference

9.2.3 1-Wire sensors

The 1-Wire sensors page is used to detect, configure, and manage sensors connected to the module via the 1-Wire interface.



This page provides tools for scanning the 1-Wire bus, identifying connected sensors, and assigning them to specific positions. Each detected sensor is identified by its unique ID, and a user-defined description can be assigned to simplify identification. The Scan function initiates a search for connected 1-Wire devices. Detected sensors are listed together with their current measurement values, which are displayed for verification purposes during setup.

It is recommended to add 1-Wire sensors one by one during initial configuration. After each sensor is detected, it is recommended to lock it to its assigned position before adding the next sensor. This helps ensure that sensors remain in the desired order and prevents unintended reordering during subsequent scans.

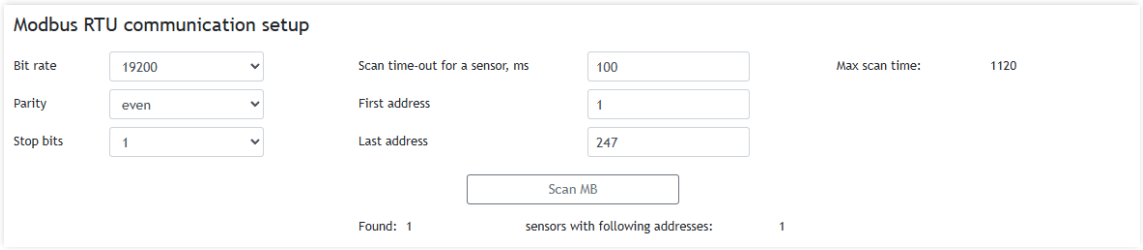
The Lock option prevents automatic reassignment of a detected sensor to a different position, ensuring that sensor mapping remains consistent after rescanning the 1-Wire bus or restarting the device.

Changes made on the 1-Wire sensors page are applied after saving the configuration.

9.2.4 Modbus sensors

This page is used to configure Modbus RTU communication and define the registers read from connected Modbus RTU devices.

Modbus RTU settings



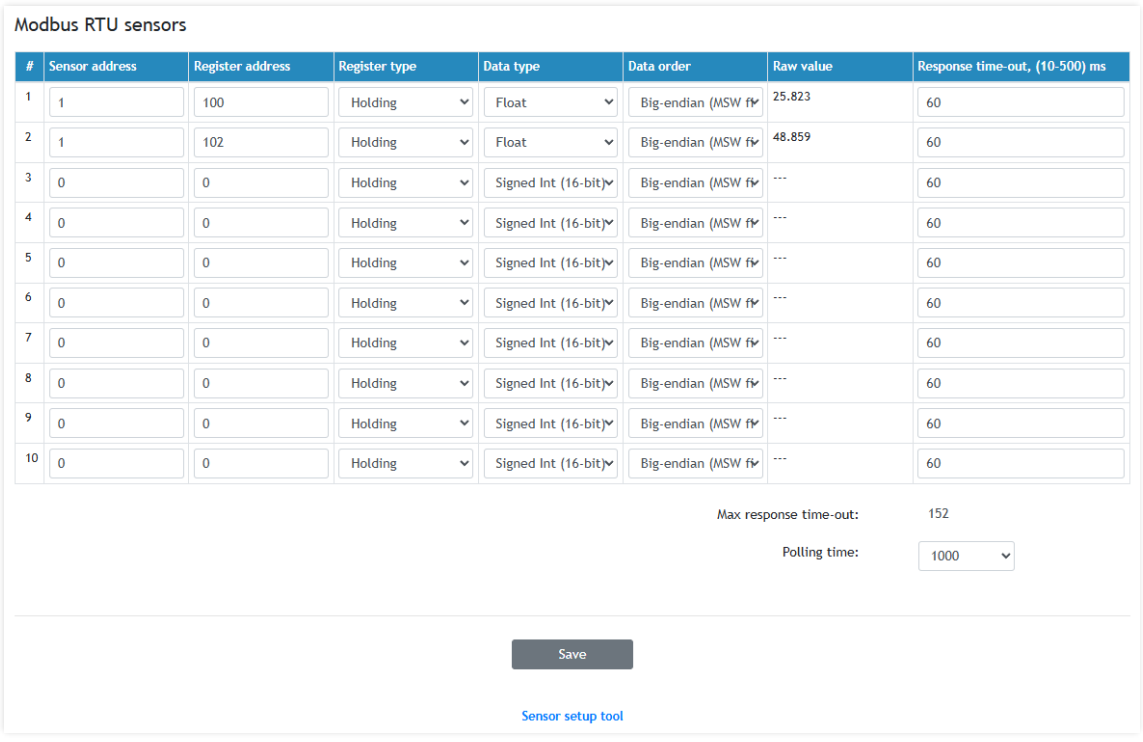
The Modbus RTU settings define the serial communication parameters of the RS-485 interface.

These parameters include baud rate, parity, stop bits, sensor scan time-out, and scan address range. The configured communication parameters must match the settings of the connected Modbus RTU devices. Incorrect settings may result in communication errors or loss of data.

The scan parameters define the address range and response time used when searching for Modbus RTU devices on the RS-485 bus.

The Modbus RTU settings apply globally to the RS-485 interface and affect both device scanning and periodic polling of configured Modbus RTU registers.

Modbus RTU registers



The Modbus RTU registers table is used to define the list of registers that are periodically read from connected Modbus RTU devices.

Each entry in the table represents a single Modbus RTU register and includes parameters such as the device address, register address, register type, data type, data order, and response time-out. These parameters determine how the register value is read and interpreted by the module.

The Raw value column displays the most recently acquired value from the corresponding Modbus register and can be used for communication verification and troubleshooting.

The response time-out parameter defines the maximum time the module waits for a response from the corresponding Modbus RTU device.

The Polling time parameter defines how often all configured Modbus RTU registers are read during normal operation. The Max response time-out value is calculated automatically based on the configured registers and their individual response time-out settings.

The acquired register values are stored internally and are available for monitoring, alarm processing, relay control, API access, and Modbus TCP register mapping.

Changes made in the Modbus RTU registers table are applied after saving the configuration.

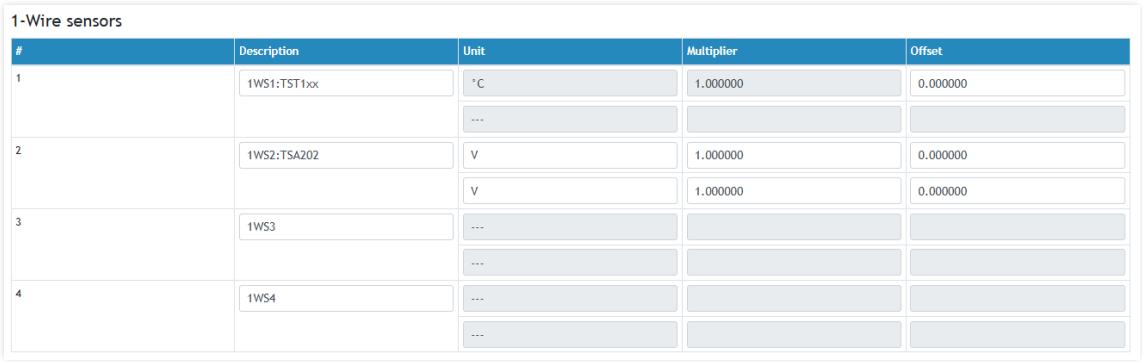
9.2.5 Input & Output

9.2.5.1 1-Wire Sensors

This section is used to configure the description for each sensor, as well as the parameters "Unit", "Multiplier", and "Offset".

The "Offset" parameter is available for all sensors and can be used to compensate measurement errors or adjust the displayed value.

The "Unit" and "Multiplier" parameters are available only for selected sensor types (such as TSA200, TSA202, etc.).


1-Wire sensors

#	Description	Unit	Multiplier	Offset
1	1WS1:TST1xx	°C	1.000000	0.000000

2	1WS2:TSA202	V	1.000000	0.000000
		V	1.000000	0.000000
3	1WS3	---		

4	1WS4	---		

The maximum length of the description field is 15 characters.

The relationship between "Unit", "Multiplier", and "Offset" is defined as follows:

$$SV\ [Un] = (RV \times MU) + OF$$

Where:

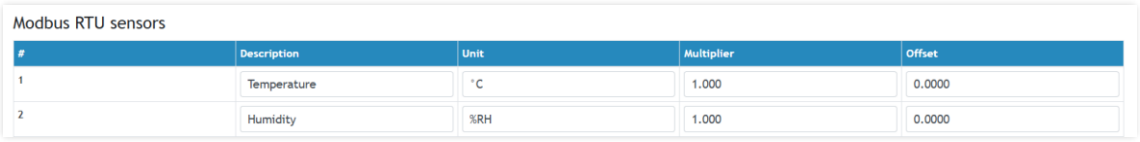
SV – scaled (displayed) value
Un – unit
RV – raw value from the sensor
MU – multiplier
OF – offset

By default, the "Multiplier" is set to 1 and the "Offset" is set to 0, meaning that the raw value is displayed without modification.

A practical example of applying this formula is provided in the Analog Inputs section.

9.2.5.2 Modbus RTU registers

This section is used to configure the description, unit, multiplier, and offset for each configured Modbus RTU register.


Modbus RTU sensors

#	Description	Unit	Multiplier	Offset
1	Temperature	°C	1.000	0.0000
2	Humidity	%RH	1.000	0.0000

The "Offset" parameter is available for all registers and can be used to compensate measurement errors or adjust the displayed value.

The maximum length of the description field is 15 characters.

The relationship between "Unit", "Multiplier", and "Offset" is defined as follows:

SV [Un] = (RV × MU) + OF

Where:

SV – scaled (displayed) value
Un – unit
RV – raw value from the sensor
MU – multiplier
OF – offset

The multiplier is applied first, followed by the offset.

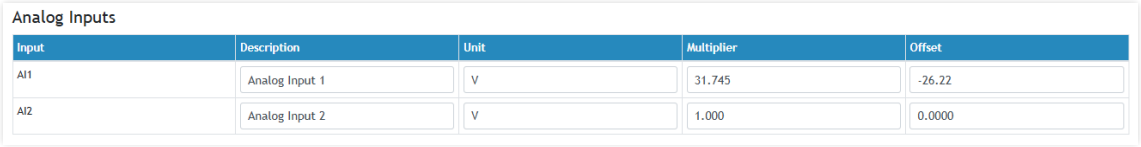
By default, the "Multiplier" is set to 1 and the "Offset" is set to 0, meaning that the raw value is displayed without modification.

A practical example of applying this formula is provided in the Analog Inputs section.

9.2.5.3 Analog Inputs

This section is used to configure the description for the analog inputs, as well as the parameters "Unit", "Multiplier", and "Offset".

The "Offset" parameter is available for all analog inputs and can be used to calibrate the measured values.


Analog Inputs

Input	Description	Unit	Multiplier	Offset
AI1	Analog Input 1	V	31.745	-26.22
AI2	Analog Input 2	V	1.000	0.0000

The maximum length of the description field is 15 characters.

The relationship between "Unit", "Multiplier", and "Offset" is defined as follows:

SV [Un] = (RV × MU) + OF

Where:

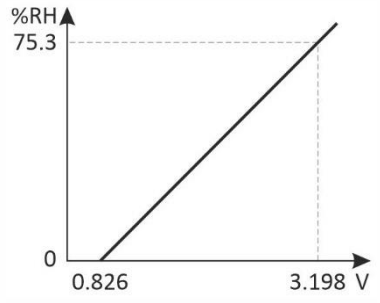
SV – scaled (displayed) value
Un – unit
RV – raw voltage from the analog input
MU – multiplier
OF – offset

By default, the "Multiplier" is set to 1 and the "Offset" is set to 0, meaning that the raw value is displayed without modification.

The following example illustrates how to calculate the multiplier and offset for a typical analog sensor:

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 a raw voltage signal, which must be converted into relative humidity (RH) values.

To achieve this, we use a multiplier and an offset. These parameters enable us to calculate the relative humidity for any voltage within the sensor's operational 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). This corresponds to the slope of the linear relationship between voltage and relative humidity. For this sensor, the line is represented by the equation ΔRH%/ΔV. We can calculate the multiplier as follows:

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

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 [Un] = (RV × MU) + OF, we can solve for the offset:

OF = SV - (RV * MU)

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

OF = 0 - (0.826 * 31.745) = 0 - 26.22 = -26.22

We can also calculate the offset using the other 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

Thus, the formula for this sensor becomes:

SV = RV * 31.745 - 26.22

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

SV = 0.826 * 31.745 - 26.22 = 26.22 - 26.22 = 0 %RH

This confirms that the formula accurately converts voltage readings to their corresponding relative humidity values.

9.2.5.4 Digital Inputs

This section is used to configure the description and state labels for each digital input.

Each digital input can be assigned a description of up to 15 characters to simplify identification.

Digital Inputs

Input	Description	Closed state description	Open state description	
DI1	Digital Input 1	CLOSED	OPEN	
DI2	Digital Input 2	CLOSED	OPEN	

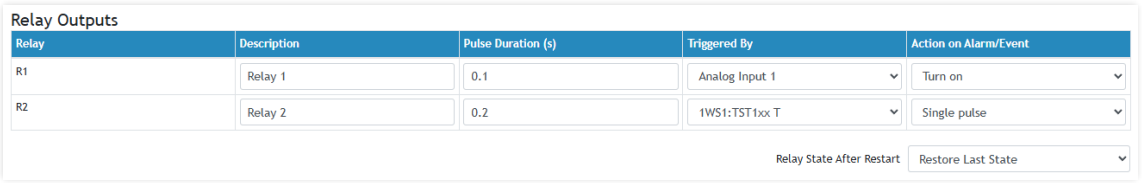
The "Closed state description" and "Open state description" fields define the text displayed on the Monitoring page for the corresponding input state. Each field allows up to 15 characters.

These parameters allow the displayed input states to be adapted to the specific application and improve the readability of the Monitoring page.

9.2.5.5 Relay Outputs

This section is used to configure the parameters of the relay outputs.

A description of up to 15 characters can be assigned to each relay to simplify identification.



The pulse duration can be configured individually for each relay, with a resolution of 0.1 seconds.

Relays can be activated either manually (remote control) or automatically based on monitored parameters. By default, all relays are set to manual control, as indicated by the "Triggered By" field.

For automatic operation, relay activation can be linked to conditions defined in the “Setup → Conditions” section. Depending on the selected source, the relay can be triggered by:

- **Sxy** – where “S” represents a 1-Wire sensor, “x” is the 1-Wire sensor number (1 to 4), and “y” is the channel number (1 or 2). The relay is activated based on the measured value and the rules defined in “Setup → Conditions”
- **MBSz** – the relay is activated based on the value of the specified Modbus RTU register (z = 1 to 10), according to the configured conditions
- **Analog input z** – the relay is activated based on the value measured by the specified analog input (z = 1 or 2), according to the configured conditions
- **Digital input z** – the relay follows the state of the specified digital input (z = 1 or 2)
- **Any alarm** – the relay is activated when any defined alarm condition is triggered

The action performed by the relay upon activation is defined in the "Action on Alarm/Event" field. Available actions include turning the relay on, turning it off, or generating a pulse with the configured pulse duration.

The Relay State After Restart parameter defines the startup state of all relay outputs after the module is restarted.

The following options are available:

- All Relays OFF – all relay outputs are switched off after startup
- All Relays ON – all relay outputs are switched on after startup
- Restore Last State – each relay restores the state it had immediately before the restart

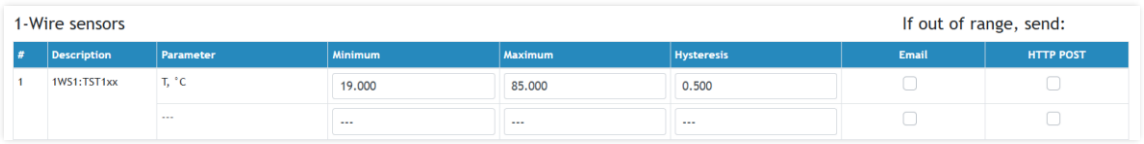
Changes made on this page are applied after saving the configuration.

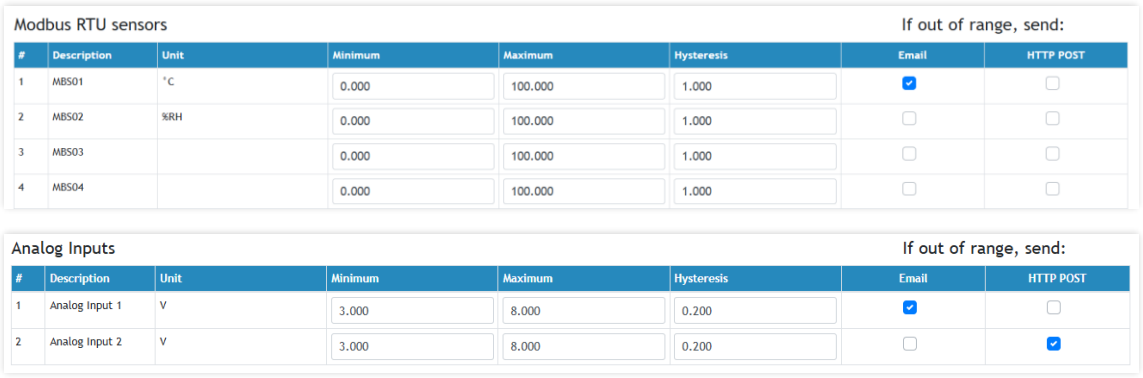
9.2.6 Conditions

This section is dedicated to configuring the trigger and alert conditions for 1-Wire sensors, Modbus sensors, analog inputs, and digital inputs.

9.2.6.1 Sensors and analog inputs

This section is used to define operating limits, alarm conditions, and notification settings for 1-Wire sensors, Modbus RTU registers, and analog inputs.





For each monitored parameter, the module provides two groups of settings:

- Trigger conditions – defined by the Minimum, Maximum, and Hysteresis parameters
- Notification settings – defined by the Email and HTTP POST checkboxes in the If out of range, send section

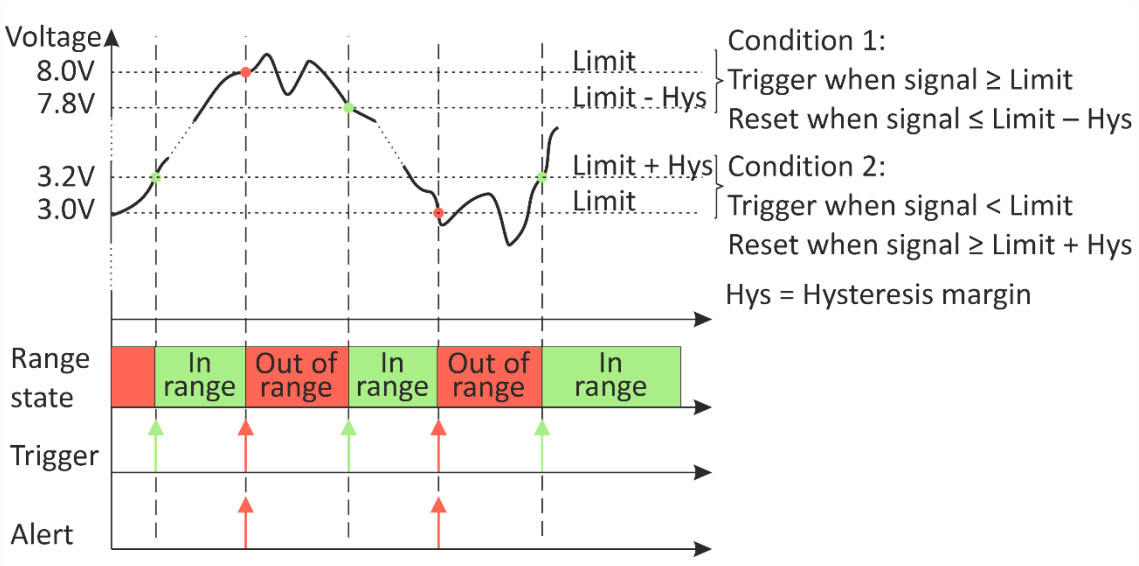
The Minimum and Maximum values define the acceptable operating range of the monitored parameter.

A Maximum condition is triggered when the measured value exceeds the configured upper limit.

A Minimum condition is triggered when the measured value falls below the configured lower limit.

In both cases, the parameter is considered out of range.

The Hysteresis parameter defines the margin required for returning to the normal operating range. This prevents repeated triggering when the measured value fluctuates around the configured limit.



For a minimum limit, the alarm condition is cleared when:

Measured Value ≥ Minimum + Hysteresis

For a maximum limit, the alarm condition is cleared when:

Measured Value ≤ Maximum − Hysteresis

Example – Temperature Control Using a Relay

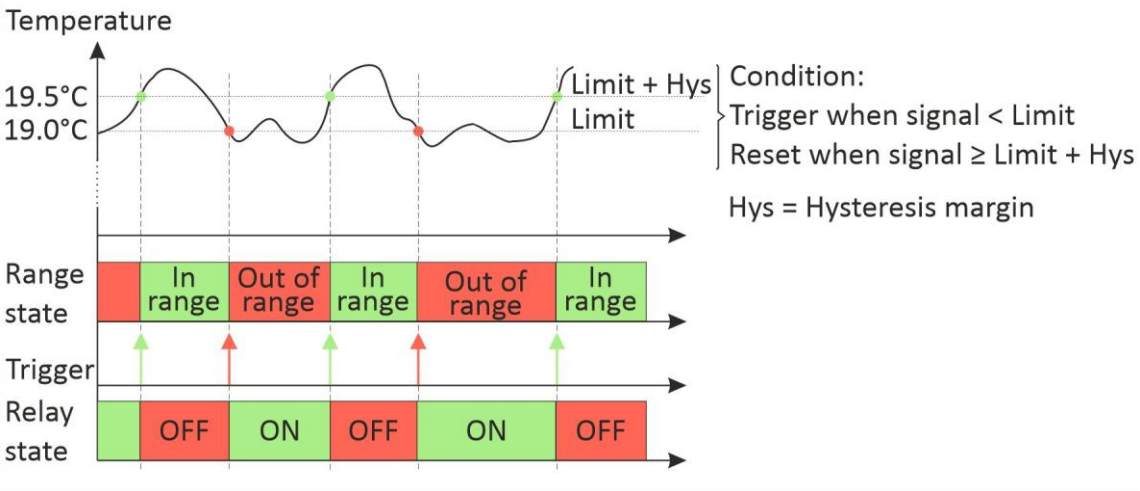
The following example illustrates the interaction between trigger conditions and relay control.

A TCW320W-EMM module is connected to a TST100 temperature sensor and a heater. The desired minimum room temperature is 19.0 °C.

The first 1-Wire sensor is configured with:

- Minimum = 19.0 °C
- Maximum = 85.0 °C
- Hysteresis = 0.5 °C

Relay 1 is configured to be triggered by Sensor 1.



When the module starts and the measured temperature is below 19.0 °C, the temperature is considered out of range and Relay 1 is activated. The heater turns on and the room temperature begins to rise.

The relay remains active until the temperature reaches 19.5 °C (19.0 + 0.5). At this point the parameter returns to the valid operating range and the relay is switched off.

As the room cools down and the temperature falls below 19.0 °C again, the parameter becomes out of range, Relay 1 is activated, and the heating cycle repeats.

To avoid unwanted high-temperature alarms in this application, the Maximum value should be configured sufficiently above the expected operating temperature.

Alarm Notifications

When a parameter becomes out of range, the module can generate notifications independently through the available communication services.

The following notification methods are available:

- Email notification
- HTTP/HTTPS POST notification

Each notification method can be enabled or disabled individually using the corresponding checkbox.

Recovery Notification

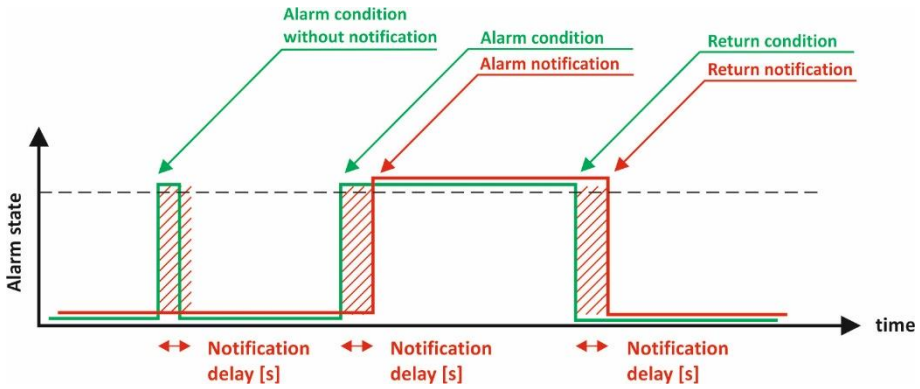
The Recovery Notification option enables notifications when a parameter returns to the valid operating range after an alarm condition.

When this option is disabled, notifications are generated only when an alarm condition occurs.

When enabled, the module sends an additional notification when the alarm condition is cleared.

Notification Delay

The Notification Delay parameter defines the amount of time that an alarm condition must remain active before a notification is generated.



This function acts as a filter for short-duration alarm conditions and helps prevent false or unnecessary notifications caused by temporary signal fluctuations.

If the alarm condition disappears before the configured delay expires, no notification is generated.

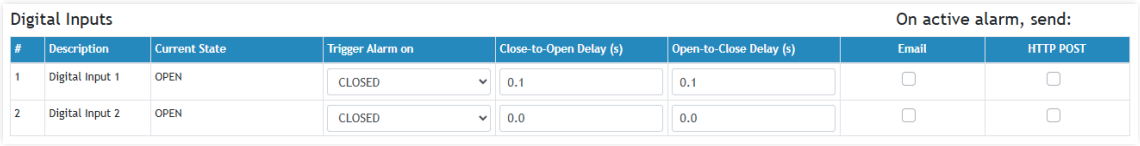
Communication Loss Monitoring

For 1-Wire sensors and Modbus RTU registers, additional alarm options are available to detect communication failures.

When enabled, the module generates notifications if communication with the corresponding sensor or Modbus RTU device is lost. Email and HTTP POST notifications can be configured independently for these events.

9.2.6.2 Digital inputs

This section is used to configure alarm conditions and notification settings for the digital inputs.


Digital Inputs — On active alarm, send:

#	Description	Current State	Trigger Alarm on	Close-to-Open Delay (s)	Open-to-Close Delay (s)	Email	HTTP POST
1	Digital Input 1	OPEN	CLOSED	0.1	0.1	☐	☐
2	Digital Input 2	OPEN	CLOSED	0.0	0.0	☐	☐

For each digital input, the alarm state is selected using the Trigger Alarm on parameter. The alarm condition can be triggered when the input is either OPEN or CLOSED.

When the input enters the configured alarm state, the module can generate notifications through the available communication services. The following notification methods are available:

- Email notification
- HTTP/HTTPS POST notification

Each notification method can be enabled or disabled independently using the corresponding checkbox.

When a digital input is in the alarm state, the corresponding input on the Monitoring page is highlighted to indicate the active alarm condition.

Recovery Notification

The Recovery Notification option enables notifications when a digital input returns from the alarm state to its normal state.

When this option is disabled, notifications are generated only when the alarm condition occurs.

When enabled, the module sends an additional notification when the input returns to the normal state.

Notification Delay

The Notification Delay parameter defines the amount of time that an alarm condition must remain active before a notification is generated.

This function acts as a filter for short-duration alarm conditions and helps prevent false or unnecessary notifications caused by temporary input transitions.

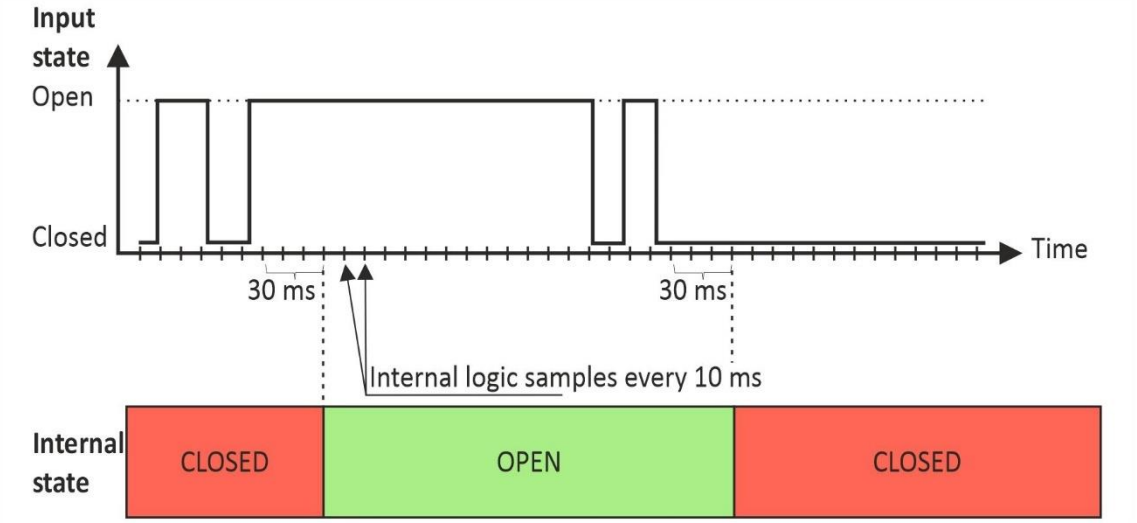
If the alarm condition disappears before the configured delay expires, no notification is generated.

Input Filtering and Debouncing

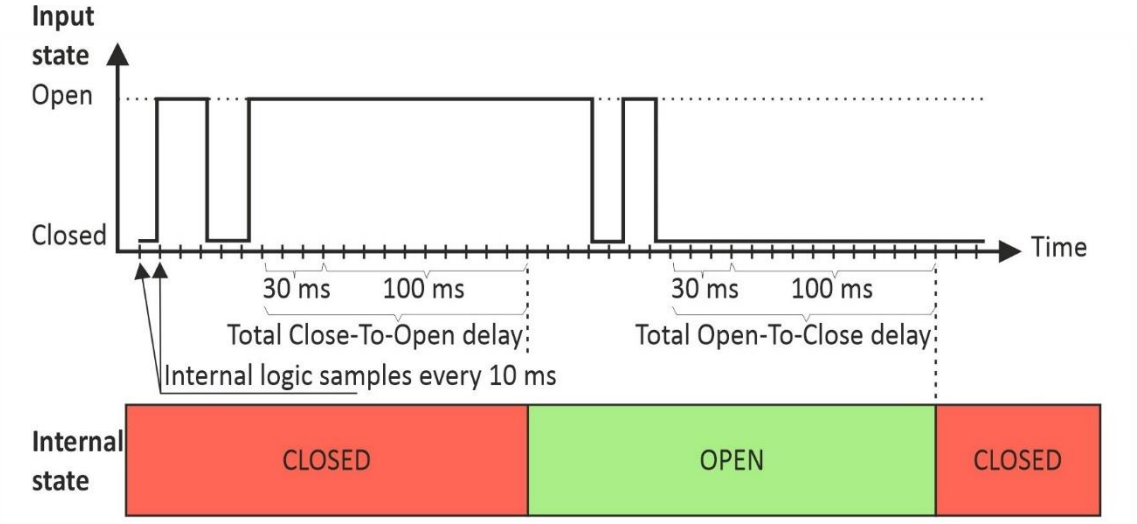
The module continuously samples the digital inputs every 10 ms.

A built-in digital filter is applied to all inputs. A change of state is accepted only when the same value is detected in three consecutive samples. As a result, the fixed filtering time is 30 ms.

The Close-to-Open Delay and Open-to-Close Delay parameters allow additional filtering to be applied independently for each transition direction. These delays are added to the fixed 30 ms filtering time.



When both delay parameters are set to 0.0 s, only the fixed 30 ms filter is applied. In this case, an input state change is accepted after three consecutive identical samples.



In the example above, an additional delay of 0.1 s (100 ms) is configured. The total filtering time becomes:

Total delay = 30 ms + 100 ms = 130 ms

The new input state is accepted only if it remains stable for the entire 130 ms period.

This mechanism provides effective debouncing and helps prevent false triggering caused by contact bounce, electrical noise, or short disturbances on the input signal.

9.3. Services

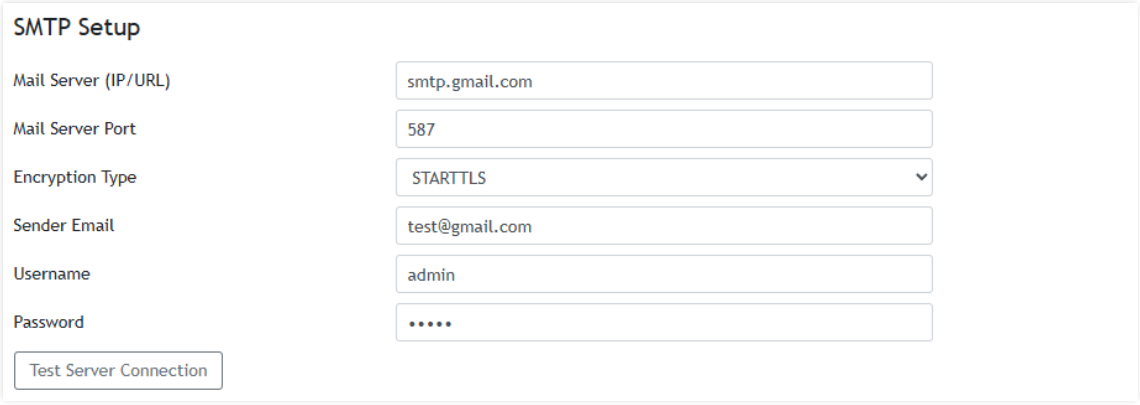
The Services section is used to configure the external communication services of the module, including email notifications and HTTP POST data transmission to remote systems.

9.3.1 SMTP

The SMTP section is used to configure the email notification service of the module. It includes SMTP server settings, recipient configuration, and customization of the email subject and message content using predefined variables. Email notifications are generated when alarm conditions defined in the module are triggered.

9.3.1.1 SMTP Setup

The SMTP Setup section is used to configure the parameters required for sending email notifications.



The mail server can be specified either by hostname (for example, smtp.gmail.com) or by IP address. The SMTP port must correspond to the requirements of the selected mail service provider. If communication cannot be established using the configured port, consult the mail service provider or network administrator for the correct settings.

The Encryption Type parameter defines whether the connection to the mail server is encrypted. The module supports TLS 1.0, TLS 1.1, TLS 1.2, and STARTTLS using RSA-based key exchange, agreement, and authentication, ensuring compatibility with most public SMTP servers.

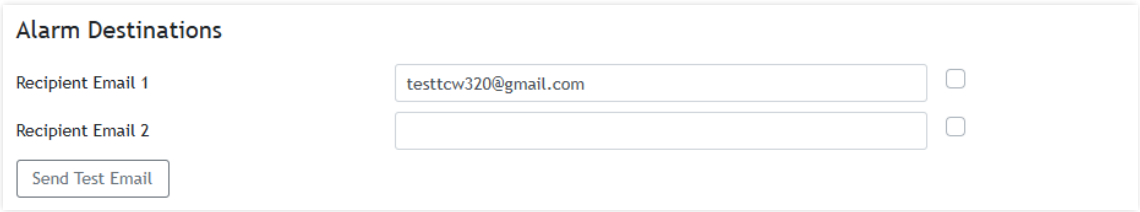
The Sender Email, Username, and Password fields contain the authentication credentials used to access the SMTP server. Each field supports up to 128 characters.

The Test Server Connection button can be used to verify the SMTP configuration. The module attempts to establish a connection to the configured mail server and reports the result of the test. During this test, the sender and recipient email addresses are identical.

Changes made in this section are applied after saving the configuration.

9.3.1.2 Alarm destination

The Alarm Destinations section is used to configure the email recipients for alarm notifications generated by the module.



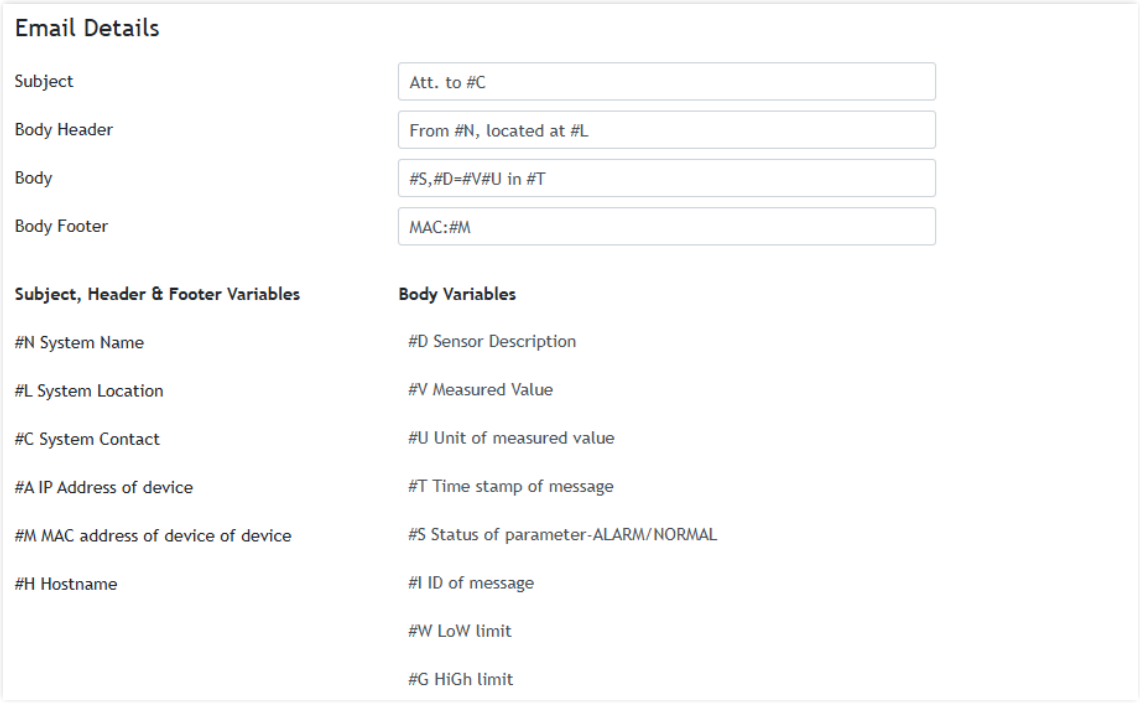
Up to two recipient email addresses can be configured. Each recipient can be enabled or disabled independently using the corresponding checkbox. Disabled recipients do not receive alarm notifications.

The Send Test Email button can be used to verify the SMTP configuration and recipient settings. When pressed, the module sends a test message to all enabled recipients and reports the result of the operation.

It is recommended to use the test function after any change to the SMTP server settings or recipient addresses to confirm proper email delivery.

9.3.1.3 E-mail details

The Email Details section is used to customize the content of alarm notification emails generated by the module.



The email message consists of four parts:

- Subject – email subject line
- Body Header – text inserted at the beginning of the message body
- Body – main alarm notification text
- Body Footer – text appended at the end of the message body

Each field may contain both fixed text and predefined variables. During message generation, the module replaces the variables with the corresponding current values and inserts the resulting text into the email.

The available variables are grouped according to their intended use:

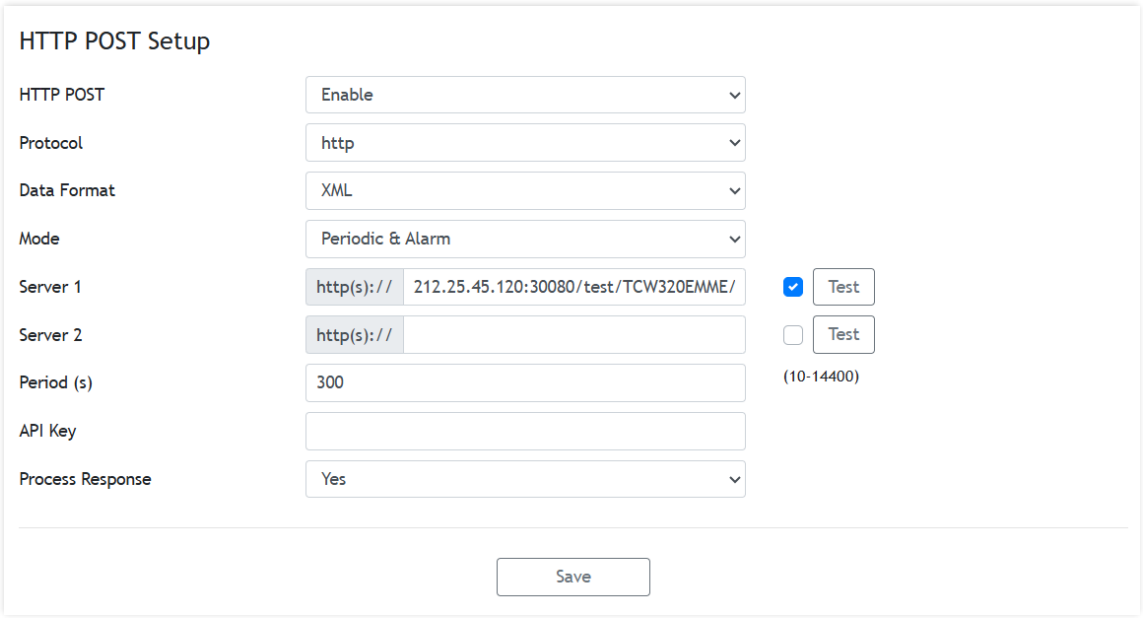
- Subject, Header & Footer Variables – provide general information about the module, such as system name, location, contact information, IP address, MAC address, and hostname
- Body Variables – provide information related to the alarm event, including the parameter description, measured value, engineering unit, timestamp, alarm status, message identifier, and configured alarm limits

Variables can be combined with ordinary text to create customized notification messages suitable for the specific application.

The default email template provides a concise alarm notification containing the affected parameter, measured value, engineering unit, timestamp, and device identification information.

9.3.2 HTTP POST

The HTTP POST section is used to configure automatic transmission of measurement data and alarm notifications to external servers using the HTTP or HTTPS protocol.



The module can periodically upload XML or JSON data to one or two remote servers. When HTTPS is selected, communication is secured using TLS 1.0, TLS 1.1, or TLS 1.2 with RSA-based authentication and key exchange.

The HTTP POST parameter enables or disables the service. The Protocol parameter selects HTTP or HTTPS communication, while the Data Format parameter defines whether the transmitted payload is generated in XML or JSON format.

The Mode parameter determines when data is transmitted:

- Periodic & Alarm – data is transmitted periodically and additionally whenever an alarm condition occurs
- Periodic Only – only periodic transmissions are performed
- Alarm Only – data is transmitted only when an alarm condition occurs

Up to two destination servers can be configured. Each server can be enabled or disabled independently using the corresponding checkbox. The Test button can be used to verify communication with the configured server.

The Period parameter defines the interval between periodic transmissions and can be configured between 10 and 14,400 seconds.

The API Key field allows a user-defined identification string to be included in the transmitted XML or JSON payload. This value can be used by the receiving system for device identification, authentication, or data processing purposes.

When Process Response is enabled, the module processes the response returned by the remote server. This allows the server to issue commands to the module through the HTTP POST communication channel. The supported commands are described in the HTTP API Commands section.

Changes made in this section are applied after saving the configuration.

9.4. Administration

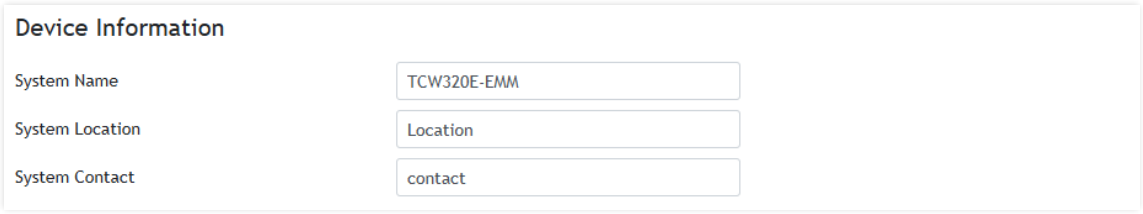
The Administration section is used to configure system-level parameters of the module and perform maintenance operations.

This section is available only to users logged in with the Administrator account. It provides access to device identification settings, user account management, configuration backup and restore functions, firmware updates, and device restart operations.

9.4.1 System

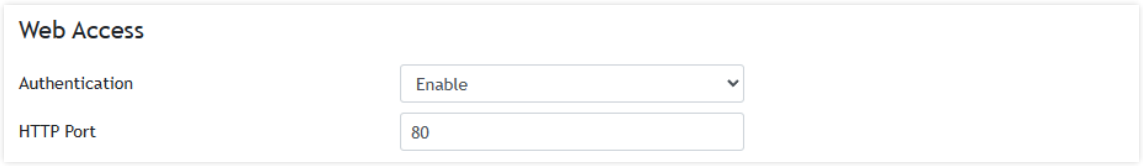
The System page is used to configure general device parameters, web access settings, display preferences, and maintenance functions.

Device Information



The System Name, System Location, and System Contact fields are used to identify the module and its installation. These parameters may be included in email notifications, HTTP POST messages, and other system-generated information.

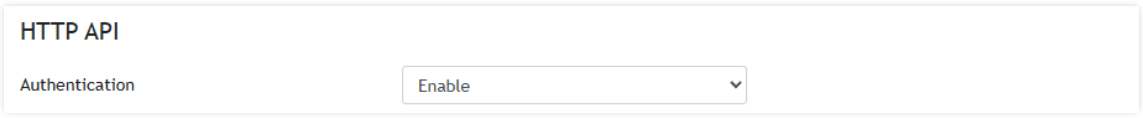
Web Access



The Authentication parameter enables or disables user authentication for access to the web interface.

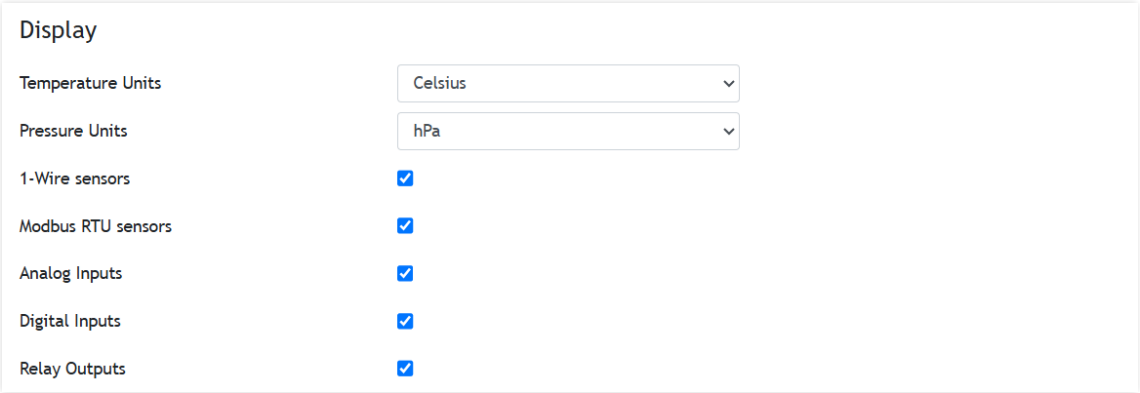
The HTTP Port parameter defines the TCP port used by the embedded web server. By default, HTTP communication uses port 80. If the port number is changed, the new port must be included in the URL used to access the device.

HTTP API



The Authentication parameter determines whether authentication is required when accessing the HTTP API. When enabled, valid user credentials must be provided when executing HTTP API commands.

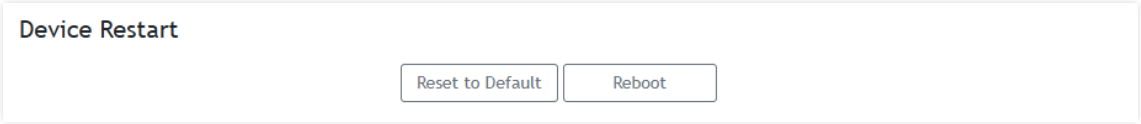
Display



The Temperature Units and Pressure Units parameters define the engineering units used for displaying measured values throughout the web interface.

The checkboxes in this section determine which information blocks are displayed on the Monitoring page. Individual sections for 1-Wire sensors, Modbus RTU registers, analog inputs, digital inputs, and relay outputs can be enabled or disabled according to the application requirements.

Device Restart



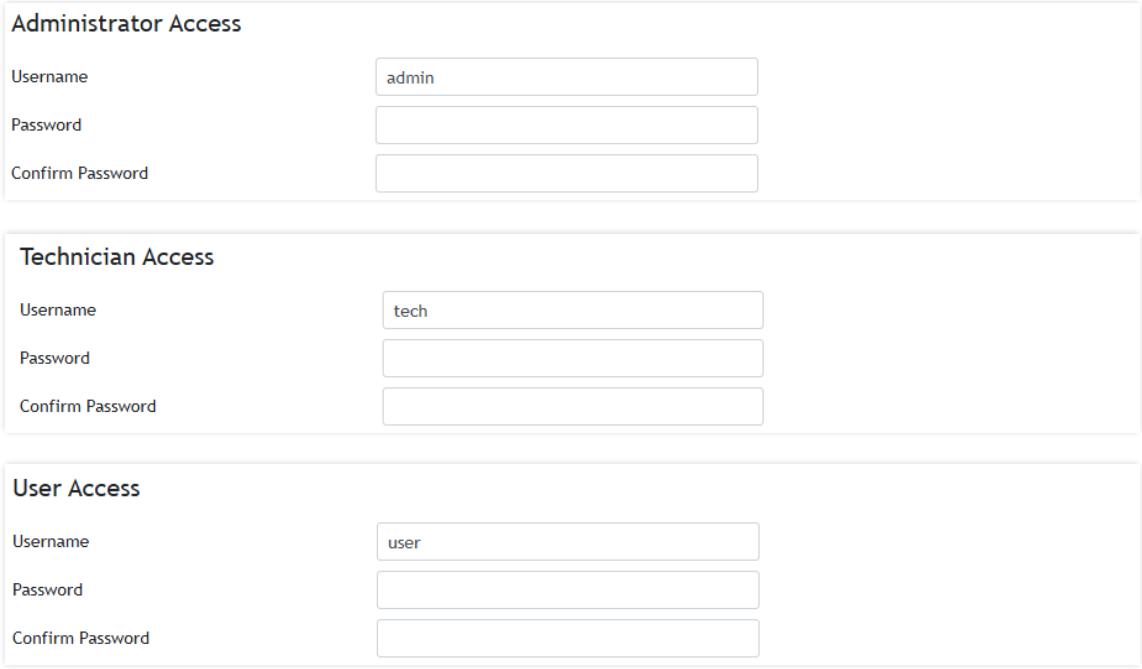
The Reboot button performs a software restart of the module while preserving the current configuration.

The Reset to Default button restores the factory default configuration while preserving the current network settings.

Detailed information about the available reset methods and the factory default settings is provided in Section 11, Factory Default Settings.

9.4.2 User management

The User management section is used to configure access credentials for the device web interface.



The device supports three predefined user accounts with different access levels: user, tech, and admin.

By default, the password for each account is identical to the corresponding user name. It is strongly recommended to change the default passwords during the initial device configuration.

A user account is considered active only if a user name is defined. If no user name is configured for a given account, that account cannot be used to access the web interface.

Passwords are configured by entering a new password and confirming it. Changes are applied after saving the configuration.

Access rights are defined as follows:

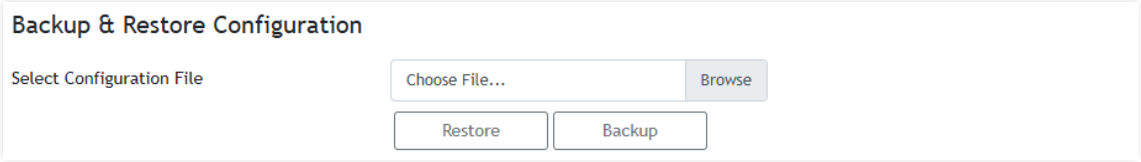
- The user account has access only to the Monitoring section
- The tech account has access to the Monitoring, Setup, and Services sections
- The admin account has unrestricted access to all sections of the web interface

Operations related to firmware updates, backup and restore functions, user management, and system administration are available only to the admin account.

Restricting administrative functions to the administrator account improves the overall security of the device.

9.4.3 Backup and restore

The Backup and Restore section is used to export and import the device configuration.



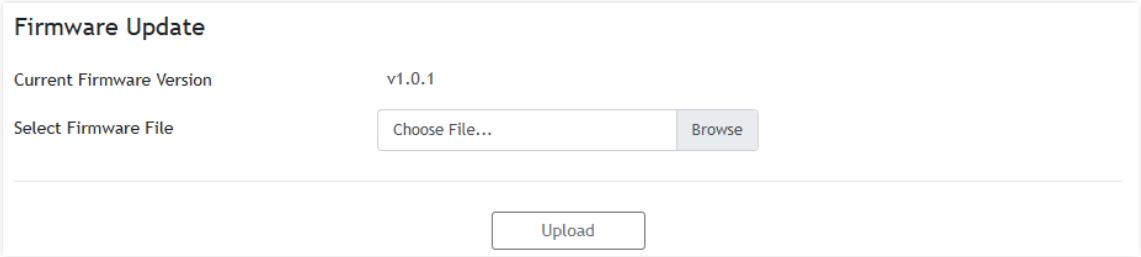
The configuration is stored in an XML file containing the device parameters. When performing a backup operation, the current configuration is saved to an XML file that can be downloaded and stored externally.

The XML configuration file can also be edited manually if required, allowing selective modification of specific parameters before restoring it to the device. This makes it possible to modify only selected configuration parameters while preserving all other settings unchanged.

The restore function applies the parameters from the uploaded XML file to the device configuration. This feature is especially useful for replicating identical or similar configurations across multiple devices, reducing setup time and minimizing configuration errors.

9.4.4 Firmware update

The Firmware Update section is used to update the device firmware.



The currently installed firmware version is displayed for reference. To perform a firmware update, select the appropriate firmware file and press the Upload button. During the update process, the device must not be powered off or restarted, as this may result in incomplete firmware installation.

After a successful firmware update, the device automatically restarts and activates the new firmware version.

Firmware update operations are available only to users with administrative privileges.

10. HTTP API

The HTTP API allows external software to monitor the module, retrieve measurement data, control relay outputs, and configure device parameters using standard HTTP or HTTPS communication.

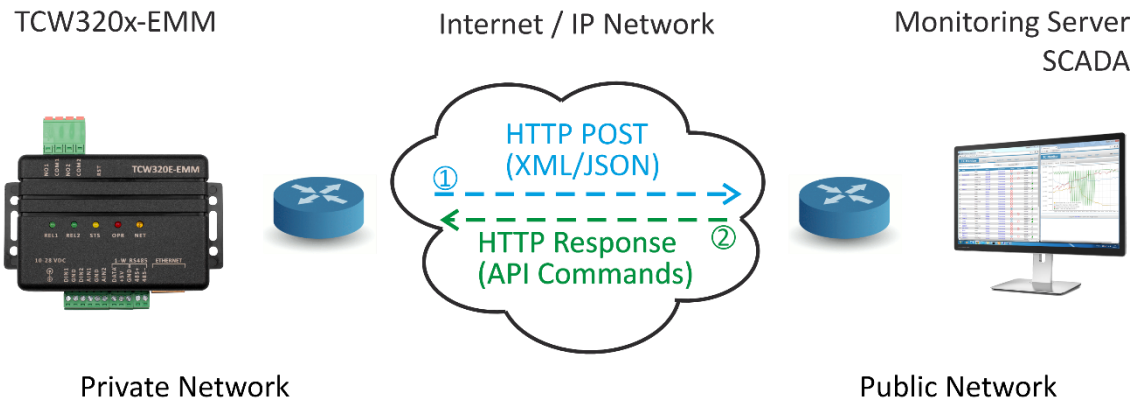
10.1. HTTP POST

The module can periodically transmit XML or JSON status data to one or two remote monitoring servers using HTTP or HTTPS POST requests.

Unlike HTTP GET communication, where the monitoring software initiates the connection to the module, HTTP POST works in the opposite direction - the module always initiates the communication. This makes HTTP POST particularly suitable for installations where the module is located behind a router, firewall, or cellular network and cannot be accessed directly from the Internet.

During each communication session, the module sends its current XML or JSON status data to the monitoring server. The server may return HTTP API commands, allowing remote control of relay outputs, modification of configuration parameters, or continue the current communication session.

The typical HTTP POST communication architecture is shown below.



10.2. Communication session

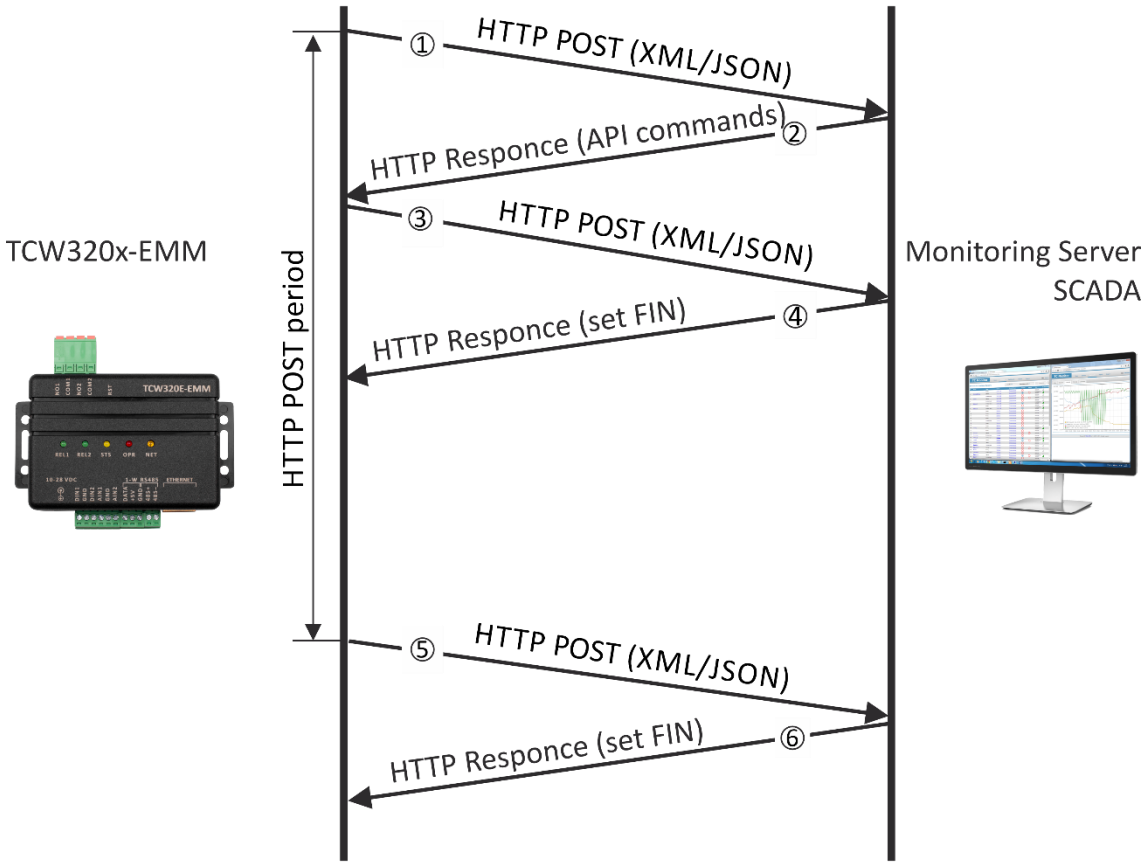
The following example illustrates a typical HTTP POST communication session when the Process Response option is enabled.

Each communication session begins when the module sends an HTTP POST request containing its current XML or JSON status data. The monitoring server processes the received data and returns an HTTP response.

The monitoring server controls the duration of the communication session. The HTTP response may contain one or more HTTP API commands or the set FIN command. The set FIN command explicitly terminates the current communication session and may be returned in response to the very first HTTP POST request. In this case, the communication session ends immediately after a single request/response exchange.

If the server returns one or more HTTP API commands instead of set FIN, the module executes the received commands and immediately sends another HTTP POST request containing the updated device status. This exchange continues until the server finally returns the set FIN command.

The communication sequence shown below illustrates a session in which the server returns an HTTP API command before terminating the communication with the set FIN command.



- ① The module sends an HTTP POST request containing its current XML or JSON status data. The request is initiated either periodically or by an alarm condition.
- ② The monitoring server processes the received data and returns an HTTP response containing one or more HTTP API commands.
- ③ After executing the received commands, the module immediately sends another HTTP POST request containing the updated device status.
- ④ The monitoring server returns an HTTP response containing the **set FIN** command, indicating that no further commands are pending. The current communication session is terminated.
- ⑤ At the next configured HTTP POST interval, or when triggered by an alarm condition, the module starts a new communication session by sending another HTTP POST request.
- ⑥ Since no further actions are required, the monitoring server immediately returns an HTTP response containing the **set FIN** command, terminating the new communication session after a single request/response exchange.

10.3. “Real-time” mode

The "Real-time" mode provides fast bidirectional communication between the module and the monitoring server by replacing the normal periodic HTTP POST operation with short communication cycles.

The mode is initiated by the monitoring server. Instead of terminating the communication session with the set FIN command, the server returns the following HTTP API commands:

set ddly=xx

get <filename>

where:

xx is the delay before the next HTTP POST request (3 to 59 seconds);

<filename> can be one of the following files:

dummy.xml

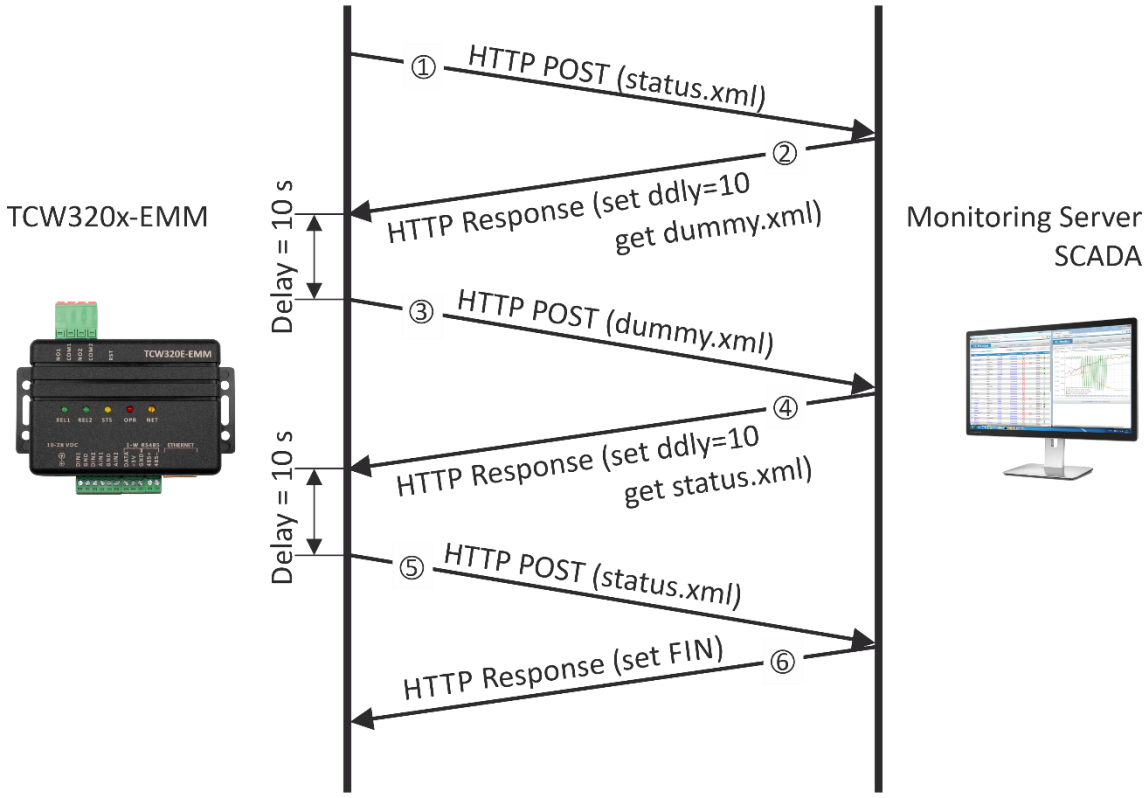
dummy.json

status.xml

status.json

After waiting for the specified delay, the module sends a new HTTP POST request containing the requested file. The server may continue the "Real-time" mode by returning another set ddly=xx command together with a file request, or terminate the communication by returning the set FIN command.

The communication sequence is illustrated below.



- ① The module sends an HTTP POST request containing the current device status.
- ② The monitoring server returns set ddly=10 and requests dummy.xml.
- ③ After the specified delay, the module sends dummy.xml.
- ④ The monitoring server returns set ddly=10 and requests status.xml.
- ⑤ After the specified delay, the module sends status.xml.
- ⑥ The monitoring server returns the set FIN command, terminating the "Real-time" mode and the current communication session.

The dummy.xml and dummy.json files contain no measurement data. They are intended to maintain frequent communication with minimal network traffic when updated device status information is not required. The monitoring server can request the complete device status at any time by returning get status.xml or get status.json.

While the module operates in "Real-time" mode, normal periodic and alarm-triggered HTTP POST requests to the same monitoring server are suspended. If a second monitoring server is configured, it continues to receive periodic and alarm-triggered HTTP POST requests according to its own configuration.

Important

Every HTTP response that keeps the module in "Real-time" mode must include a valid set ddly=xx command. Otherwise, after processing the response, the module immediately sends the next HTTP POST request without delay, which may generate unnecessary network traffic and increase the load on the monitoring server.

10.4. Testing HTTP POST

The following PHP script can be used to verify that the module is able to send HTTP POST requests successfully. The script accepts the XML or JSON data transmitted by the module, saves it as a timestamped file on the web server, and returns the set FIN command to terminate the communication session.

PHP script:

```
<?php
/**
 * HTTP POST receiver – accepts XML data from the module
 * and saves it as a timestamped file on the server.
 * Example output filename: 20260625143022_status.xml
 */
// ── Settings ──────────────────────────────────────────────
// Name of the file sent by the module
define("FILENAME", 'status.xml');
// Folder where received files will be stored ('' = current directory)
define("FOLDER", '');
// Separator between folder path and filename ('' = none)
define("SEPARATOR", '');
// Response message on successful file write
define("STR_SUCCESS", 'set FIN');
// Response message on file write failure
define("STR_ERROR", 'error');
// ── Request handling ──────────────────────────────────────
if($_SERVER['REQUEST_METHOD'] == 'POST'){
    // Generate a unique filename using a timestamp prefix (format:
YYYYMMDDHHmmss_status.xml)
    $datePrefix = date('YmdHis', strtotime('now'));
    $pathname = FOLDER.SEPARATOR.$datePrefix.'_'.FILENAME;
    // Read the raw POST body (XML data from the module)
    $postdata = file_get_contents("php://input");
    // Open file for writing (creates new or overwrites existing)
    $handle = fopen($pathname, 'w+');
    // Strip the surrounding quotes added by var_export()
    $content = var_export($postdata, true);
    fwrite($handle, substr($content, 1, strlen($content)-2));
    fclose($handle);
    // Return result to the module
    echo (($handle === false) ? STR_ERROR : STR_SUCCESS)."\r\n";
} else {
    // GET request – used to verify the script is reachable
    echo "The PHP script is working!";
}
?>
```

To test the HTTP POST functionality, follow these steps:

- Save the PHP script as post.php.
- Upload the file to a web server with PHP support.

- Verify that the script is accessible by opening its URL in a web browser (for example, https://yourserver.com/post.php).
  If the script is installed correctly, the following message is displayed:
  The PHP script is working!
- Configure the module to use the script URL as the HTTP POST destination and press Test HTTP POST on the HTTP POST configuration page.
- If the test is successful, the module displays OK next to the Test HTTP POST button. The script also creates a timestamped XML or JSON file in the same directory, for example:
  20260625143022_status.xml
  The generated file contains the HTTP POST payload received from the module and can be used to verify the transmitted data or assist in application development.

10.5. HTTP GET

The module provides its current status as XML or JSON data through the HTTP API. External software can retrieve this information at any time using standard HTTP or HTTPS GET requests.

The following URLs are available:

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

The XML and JSON file formats are described in Appendix A and Appendix B, respectively.

In addition to retrieving the current device status, HTTP GET requests can also be used to execute the same HTTP API commands that are supported during HTTP POST communication. Commands are appended to the request URL as query parameters.

The general format is:

http(s)://device.ip.address/status.xml?command=value

Example:

http://192.168.1.2/status.xml?pper=300

The example above sets the HTTP POST period to 300 seconds.

The complete list of supported HTTP API commands is provided in Appendix C.

10.6. HTTP API Authentication

HTTP API authentication can be enabled or disabled from the Setup → System page. When authentication is enabled, valid user credentials must be included with every HTTP API request. If authentication is disabled, HTTP API requests can be executed without credentials.

The authentication information is passed using the “a” parameter in the request URL, where the user name and password are separated by a colon.

HTTP API authentication	Request format
Enabled	http(s)://device.ip.address/status.xml?a=username:password
Disabled	http(s)://device.ip.address/status.xml

Example:

http://192.168.1.2/status.xml?a=admin:admin&pper=120

The example above authenticates using the admin account and sets the HTTP POST period to 120 seconds.

10.7. HTTP API Commands and File Structures

The complete list of HTTP API commands is provided in Appendix C.

The XML file structure is described in Appendix A.

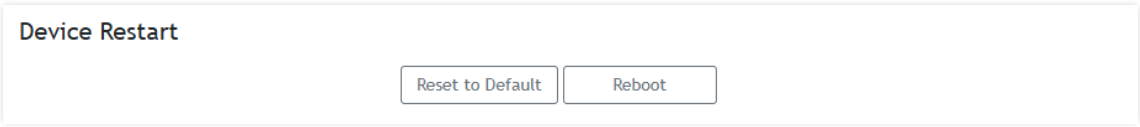
The JSON file structure is described in Appendix B.

11. Device reset

The module provides several reset methods intended for different recovery and maintenance scenarios. Depending on the selected method, different groups of parameters are restored to their factory default values.

Factory reset operations can be initiated either from the web interface or by using the RESET button on the device. This allows recovery from configuration errors while minimizing the risk of losing access to the device.

The available reset methods are described in the following sections.



11.1. Factory reset without network settings

Selecting the “Reset to default” option in the Setup/System section restores all configuration parameters to their factory default values, except for the network settings.

This reset method is intended for situations where the device configuration must be cleared while preserving the existing network access.

11.2. Factory reset for network settings only

If the RESET button is pressed and held for more than 5 seconds while the device is powered on, all network-related parameters are restored to their factory default values.

This reset method affects only the network configuration and does not modify other device settings such as Modbus configuration parameters or user accounts.

The factory reset for network settings is intended for recovery scenarios in which network access to the device has been lost due to incorrect configuration.

11.3. Full factory reset

A full factory reset restores all device parameters, including network settings and user credentials, to their factory default values.

To perform a full factory reset, follow these steps:

- Press and hold the RESET button.
- Apply power to the device while holding the button.
- The OPR LED flashes rapidly for 5 seconds.
- Keep the RESET button pressed until the OPR LED turns off and the STS LED lights up steadily, indicating the reset process is completed.
- Release the button and allow the device to restart.
- After the restart, the device operates with factory default settings.

11.4. Factory default settings

After a full factory reset, the device is configured with the following default settings.

Web interface credentials

- Username: admin
- Password: admin

Network settings Ethernet variant (TCW320E-EMM):

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

Network settings Wi-Fi variant (TCW320W-EMM, Access Point mode):

- SSID: tcw320ap
- Password: tcw320ap
- Security: WPA2-PSK
- IP address: 192.168.4.1
- Subnet mask: 255.255.255.0

Other settings

After a full factory reset, the Modbus RTU communication parameters are restored to the following default values, in accordance with the Modbus over Serial Line Specification:

- Baud rate: 19200
- Data bits: 8
- Parity: Even
- Stop bits: 1

12. Operating environment

The device is intended for operation in a Pollution Degree 2 environment, at altitudes up to 2000 meters above sea level.

When the device is integrated into a larger system, all other system components must comply with applicable EMC (Electromagnetic Compatibility) requirements and be suitable for operation under the same environmental conditions.

13. Safety precautions

The device must not be used for medical, life-support, or life-saving applications, or in any system where a failure of the device could result in serious injury or loss of life.

To reduce the risk of fire, only flexible stranded conductors with a minimum cross-sectional area of 0.5 mm² must be used for wiring.

To prevent electric shock and fire hazards, the device must not be exposed to liquids, rain, or excessive moisture. Objects containing liquids must not be placed on or near the device.

Ensure adequate ventilation around the device. Failure to maintain the recommended free space around the device may result in overheating and potential damage. Sufficient space must also be provided to allow safe connection and disconnection of cables.

The device must be operated strictly within the limits specified in this documentation. Operation outside the specified conditions may impair the protection provided by the device and lead to unreliable operation.

Teracom OOD does not guarantee correct operation of the product if it is used under conditions that differ from those specified by the manufacturer. In no event shall Teracom OOD be held responsible or liable for indirect or consequential damages resulting from the use or application of this product.

14. Maintenance

The device does not require routine maintenance during normal operation.

Periodically inspect the device to ensure that it is in good operating condition, all electrical connections are secure, and there are no signs of mechanical damage.

Clean the device using a dry cloth only. Do not use liquid or aerosol cleaners. Magnetic or electrostatic cleaning devices, as well as abrasive materials, must not be used.

Following these maintenance guidelines helps ensure reliable operation and extends the service life of the device.

15. Disposal and recycling

Please recycle all applicable materials in accordance with local regulations.



Do not dispose of the device as regular household waste.
Electronic components should be properly recycled to minimize environmental impact.

status.xml:

```
<Monitor>
<script id="eppiocemhmnlbhjplcgkofciiegomcon"/>
<script/>
<script/>
<DeviceInfo>
<DeviceName>TCW320E-EMM</DeviceName>
<HostName>TCW320E-EMM</HostName>
<ID>5c:32:c5:01:1c:c4</ID>
<FwVer>v1.0.1</FwVer>
<MnfInfo>www.teracomsystems.com</MnfInfo>
<SysContact>contact</SysContact>
<SysName>TCW320E-EMM</SysName>
<SysLocation>Location</SysLocation>
</DeviceInfo>
<OWS>
<OWS1>
<description>1WS1</description>
<id>0000000000000000</id>
<item1>
<value>---</value>
<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item1>
<item2>
<value>---</value>
<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item2>
</OWS1>
<OWS2>
<description>1WS2</description>
<id>0000000000000000</id>
<item1>
<value>---</value>
<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item1>
<item2>
<value>---</value>
<unit>---</unit>
<alarm>0</alarm>
```

<min>---</min>
<max>---</max>
<hys>---</hys>
</item2>
</OWS2>
<OWS3>
<description>1WS3</description>
<id>0000000000000000</id>
<item1>
<value>---</value>
<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item1>
<item2>
<value>---</value>
<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item2>
</OWS3>
<OWS4>
<description>1WS4</description>
<id>0000000000000000</id>
<item1>
<value>---</value>
<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item1>
<item2>
<value>---</value>
<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item2>
</OWS4>
</OWS>
<MBS>
<MBS1>
<description>MBS01</description>
<addr>0</addr>
<reg>0</reg>
<value>---</value>

<unit/>
<alarm>0</alarm>
<min>0.000</min>
<max>100.000</max>
<hys>1.000</hys>
</MBS1>
<MBS2>
<description>MBS02</description>
<addr>0</addr>
<reg>0</reg>
<value>---</value>
<unit/>
<alarm>0</alarm>
<min>0.000</min>
<max>100.000</max>
<hys>1.000</hys>
</MBS2>
<MBS3>
<description>MBS03</description>
<addr>0</addr>
<reg>0</reg>
<value>---</value>
<unit/>
<alarm>0</alarm>
<min>0.000</min>
<max>100.000</max>
<hys>1.000</hys>
</MBS3>
<MBS4>
<description>MBS04</description>
<addr>0</addr>
<reg>0</reg>
<value>---</value>
<unit/>
<alarm>0</alarm>
<min>0.000</min>
<max>100.000</max>
<hys>1.000</hys>
</MBS4>
<MBS5>
<description>MBS05</description>
<addr>0</addr>
<reg>0</reg>
<value>---</value>
<unit/>
<alarm>0</alarm>
<min>0.000</min>
<max>100.000</max>
<hys>1.000</hys>
</MBS5>
<MBS6>
<description>MBS06</description>

<addr>0</addr>
<reg>0</reg>
<value>---</value>
<unit/>
<alarm>0</alarm>
<min>0.000</min>
<max>100.000</max>
<hys>1.000</hys>
</MBS6>
<MBS7>
<description>MBS07</description>
<addr>0</addr>
<reg>0</reg>
<value>---</value>
<unit/>
<alarm>0</alarm>
<min>0.000</min>
<max>100.000</max>
<hys>1.000</hys>
</MBS7>
<MBS8>
<description>MBS08</description>
<addr>0</addr>
<reg>0</reg>
<value>---</value>
<unit/>
<alarm>0</alarm>
<min>0.000</min>
<max>100.000</max>
<hys>1.000</hys>
</MBS8>
<MBS9>
<description>MBS09</description>
<addr>0</addr>
<reg>0</reg>
<value>---</value>
<unit/>
<alarm>0</alarm>
<min>0.000</min>
<max>100.000</max>
<hys>1.000</hys>
</MBS9>
<MBS10>
<description>MBS10</description>
<addr>0</addr>
<reg>0</reg>
<value>---</value>
<unit/>
<alarm>0</alarm>
<min>0.000</min>
<max>100.000</max>
<hys>1.000</hys>

```
</MBS10>
</MBS>
<AI>
<AI1>
<description>Analog Input 1</description>
<value>0.003</value>
<unit>V</unit>
<multiplier>1.000</multiplier>
<offset>0.0000</offset>
<alarm>0</alarm>
<min>0.000</min>
<max>10.000</max>
<hys>1.000</hys>
</AI1>
<AI2>
<description>Analog Input 2</description>
<value>0.005</value>
<unit>V</unit>
<multiplier>1.000</multiplier>
<offset>0.0000</offset>
<alarm>0</alarm>
<min>0.000</min>
<max>10.000</max>
<hys>1.000</hys>
</AI2>
</AI>
<DI>
<DI1>
<description>Digital Input 1</description>
<value>OPEN</value>
<valuebin>1</valuebin>
<alarmState>CLOSED</alarmState>
<alarm>0</alarm>
</DI1>
<DI2>
<description>Digital Input 2</description>
<value>OPEN</value>
<valuebin>1</valuebin>
<alarmState>CLOSED</alarmState>
<alarm>0</alarm>
</DI2>
</DI>
<R>
<R1>
<description>Relay 1</description>
<value>OFF</value>
<valuebin>0</valuebin>
<pulseWidth>0.1</pulseWidth>
<control>0</control>
</R1>
<R2>
<description>Relay 2</description>
```

<value>OFF</value>
<valuebin>0</valuebin>
<pulseWidth>0.2</pulseWidth>
<control>0</control>
</R2>
</R>
<HTTPPost>
<Key/>
<PostPeriod>300</PostPeriod>
</HTTPPost>
<hwerr/>
<Alarmed>0</Alarmed>
<Scanning>Scan Sensors</Scanning>
<Time>
<Date>01.01.2025</Date>
<Time>04:28:19</Time>
</Time>
</Monitor>

status.json:


```
{
 "Monitor": {
  "DeviceInfo": {
   "DeviceName": "TCW320E-EMM",
   "HostName": "TCW320E-EMM",
   "ID": "5c:32:c5:01:1c:c4",
   "FwVer": "v1.0.1",
   "MnfInfo": "www.teracomsystems.com",
   "SysContact": "contact",
   "SysName": "TCW320E-EMM",
   "SysLocation": "Location"
  },
  "OWS": {
   "OWS1": {
    "description": "1WS1",
    "id": "0000000000000000",
    "item1": {
     "value": "---",
     "unit": "---",
     "alarm": "0",
     "min": "---",
     "max": "---",
     "hys": "---"
    },
    "item2": {
     "value": "---",
     "unit": "---",
     "alarm": "0",
     "min": "---",
     "max": "---",
     "hys": "---"
    }
   },
   "OWS2": {
    "description": "1WS2",
    "id": "0000000000000000",
    "item1": {
     "value": "---",
     "unit": "---",
     "alarm": "0",
     "min": "---",
     "max": "---",
     "hys": "---"
    },
    "item2": {
     "value": "---",
     "unit": "---",
     "alarm": "0",
     "min": "---",
     "max": "---",
```

```
    "hys": "---"
   }
  },
  "OWS3": {
   "description": "1WS3",
   "id": "0000000000000000",
   "item1": {
    "value": "---",
    "unit": "---",
    "alarm": "0",
    "min": "---",
    "max": "---",
    "hys": "---"
   },
   "item2": {
    "value": "---",
    "unit": "---",
    "alarm": "0",
    "min": "---",
    "max": "---",
    "hys": "---"
   }
  },
  "OWS4": {
   "description": "1WS4",
   "id": "0000000000000000",
   "item1": {
    "value": "---",
    "unit": "---",
    "alarm": "0",
    "min": "---",
    "max": "---",
    "hys": "---"
   },
   "item2": {
    "value": "---",
    "unit": "---",
    "alarm": "0",
    "min": "---",
    "max": "---",
    "hys": "---"
   }
  }
 },
 "MBS": {
  "MBS1": {
   "description": "MBS01",
   "addr": "0",
   "reg": "0",
   "value": "---",
   "unit": "",
   "alarm": "0",
```

```
  "min": "0.000",
  "max": "100.000",
  "hys": "1.000"
 },
 "MBS2": {
  "description": "MBS02",
  "addr": "0",
  "reg": "0",
  "value": "---",
  "unit": "",
  "alarm": "0",
  "min": "0.000",
  "max": "100.000",
  "hys": "1.000"
 },
 "MBS3": {
  "description": "MBS03",
  "addr": "0",
  "reg": "0",
  "value": "---",
  "unit": "",
  "alarm": "0",
  "min": "0.000",
  "max": "100.000",
  "hys": "1.000"
 },
 "MBS4": {
  "description": "MBS04",
  "addr": "0",
  "reg": "0",
  "value": "---",
  "unit": "",
  "alarm": "0",
  "min": "0.000",
  "max": "100.000",
  "hys": "1.000"
 },
 "MBS5": {
  "description": "MBS05",
  "addr": "0",
  "reg": "0",
  "value": "---",
  "unit": "",
  "alarm": "0",
  "min": "0.000",
  "max": "100.000",
  "hys": "1.000"
 },
 "MBS6": {
  "description": "MBS06",
  "addr": "0",
  "reg": "0",
```

```
    "value": "---",
    "unit": "",
    "alarm": "0",
    "min": "0.000",
    "max": "100.000",
    "hys": "1.000"
   },
   "MBS7": {
    "description": "MBS07",
    "addr": "0",
    "reg": "0",
    "value": "---",
    "unit": "",
    "alarm": "0",
    "min": "0.000",
    "max": "100.000",
    "hys": "1.000"
   },
   "MBS8": {
    "description": "MBS08",
    "addr": "0",
    "reg": "0",
    "value": "---",
    "unit": "",
    "alarm": "0",
    "min": "0.000",
    "max": "100.000",
    "hys": "1.000"
   },
   "MBS9": {
    "description": "MBS09",
    "addr": "0",
    "reg": "0",
    "value": "---",
    "unit": "",
    "alarm": "0",
    "min": "0.000",
    "max": "100.000",
    "hys": "1.000"
   },
   "MBS10": {
    "description": "MBS10",
    "addr": "0",
    "reg": "0",
    "value": "---",
    "unit": "",
    "alarm": "0",
    "min": "0.000",
    "max": "100.000",
    "hys": "1.000"
   }
  },
```

```
"AI": {
 "AI1": {
  "description": "Analog Input 1",
  "value": "0.003",
  "unit": "V",
  "multiplier": "1.000",
  "offset": "0.0000",
  "alarm": "0",
  "min": "0.000",
  "max": "10.000",
  "hys": "1.000"
 },
 "AI2": {
  "description": "Analog Input 2",
  "value": "0.005",
  "unit": "V",
  "multiplier": "1.000",
  "offset": "0.0000",
  "alarm": "0",
  "min": "0.000",
  "max": "10.000",
  "hys": "1.000"
 }
},
"DI": {
 "DI1": {
  "description": "Digital Input 1",
  "value": "OPEN",
  "valuebin": "1",
  "alarmState": "CLOSED",
  "alarm": "0"
 },
 "DI2": {
  "description": "Digital Input 2",
  "value": "OPEN",
  "valuebin": "1",
  "alarmState": "CLOSED",
  "alarm": "0"
 }
},
"R": {
 "R1": {
  "description": "Relay 1",
  "value": "OFF",
  "valuebin": "0",
  "pulseWidth": "0.1",
  "control": "0"
 },
 "R2": {
  "description": "Relay 2",
  "value": "OFF",
  "valuebin": "0",
```

```
      "pulseWidth": "0.2",
      "control": "0"
     }
    },
    "HTTPPost": {
     "Key": "",
     "PostPeriod": "300"
    },
    "hwerr": "",
    "Alarmed": "0",
    "Scanning": "Scan Sensors",
    "Time": {
     "Date": "01.01.2025",
     "Time": "04:31:33"
    }
   }
}
```

HTTP API Commands

This appendix lists all HTTP API commands supported by the TCW320x-EMM.

HTTP API commands can be used in two ways:

- As parameters in HTTP GET requests sent directly to the controller.
- As commands returned by a monitoring server in response to an HTTP POST request. In this case, the commands are executed only when the Process Response option is enabled.

Multiple commands can be combined in a single request or response by separating them with the & character.

All command names are case-sensitive.

Command	Description
Relay control	
ron=**x**	Turns relay *x* ON. *x* is the relay number (1 or 2). Examples: ron=1 – turns relay 1 ON. ron=1&ron=2 – turns both relays ON.
rof=**x**	Turns relay *x* OFF. *x* is the relay number (1 or 2). Examples: rof=1 – turns relay 1 OFF. rof=1&rof=2 – turns both relays OFF.
rpl=**x**	Pulses relay *x*. *x* is the relay number (1 or 2). Example: rpl=1 – pulses relay 1.
rtg=**x**	Toggles relay *x*. *x* is the relay number (1 or 2). Example: rtg=1 – toggles relay 1.
1-Wire	
sc1=1	Starts a scan for all connected 1-Wire sensors.
Modbus RTU communication	
bitrs=**x**	Sets the Modbus RTU bit rate (*x* = 2400, 4800, 9600, 19200, 38400 or 57600).
pars=**x**	Sets the parity (*x* = 0 – none, 2 – even, 3 – odd).
stbs=**x**	Sets the number of stop bits (*x* = 1 – one stop bit, 3 – two stop bits).
Modbus Sensors setup	
mbsa**n**=**x**	Sets the Modbus address of sensor *n* (*n* = 1 to 10, *x* = 1 to 247).
s**n**reg1a=**x**	Set Modbus sensor *n* Data register address (n = 1 to 10); *x* = (0 to 65 535).
s**n**reg1t=**x**	Set Modbus sensor *n* RegisterType *n* = 1 to 10; *x* = 0 – Holding Register, 1 – Input Register, 2 – Coil.
s**n**pto1=**x**	Sets the response timeout for Modbus sensor *n* (*n* = 1 to 10, *x* = 10 to 500 ms).

s**n**dt1=**x**	Sets the data type for Modbus sensor *n* (*n* = 1 to 10). *x* = 0 – Float, 1 – Signed 16-bit Integer, 2 – Unsigned 16-bit Integer, 3 – Signed 32-bit Integer, 4 – Unsigned 32-bit Integer, 5 – BCD16, 6 – BCD32.
s**n**do1=**x**	Sets the word order for Modbus sensor *n* (*n* = 1 to 10). *x* = 0 – Least Significant Word First (LSWF), 1 – Most Significant Word First (MSWF).
Input/Output setup	
mbsd**n**=**x**	Sets the description for Modbus sensor *n*. (*n* = 1 to 10 for the respective sensor) Example: mbsd1=S1-Temp – sets the description of Modbus sensor 1 to "S1-Temp".
mbsu**n**=**x**	Sets the unit for Modbus sensor *n*. (*n* = 1 to 10 for the respective sensor) Example: mbsu1=°C – sets the unit of Modbus sensor 1 to °C.
mbsm**n**=**x**	Sets the multiplier for Modbus sensor *n*. (*n* = 1 to 10 for the respective sensor) Example: msm1=2.000 – sets the multiplier of Modbus sensor 1 to 2.000.
mbso**n**=**x**	Sets the offset for Modbus sensor n. (*n* = 1 to 10 for the respective sensor) Example: mbso1=0.65 – sets the offset of Modbus sensor 1 to 0.65.
sd**n**=**x**	Sets the description for 1-Wire sensor *n* (*n* = 1 to 4 for the respective sensor) Example: sd1=S1-Temp – sets the description of 1-Wire sensor 1 to "S1-Temp".
su**np**=**x**	Sets the unit for parameter *p* of 1-Wire sensor *n*. (Some 1-Wire sensors report two parameters, such as temperature and humidity. In this case, parameter *p* selects which parameter the unit applies to.) (*n* = 1 to 4 for the respective sensor, *p* = 1 or 2) Example: su11=°C – sets the unit of parameter 1 for 1-Wire sensor 1 to °C.
so**np**=**x**	Sets the offset for parameter *p* of 1-Wire sensor *n*. (Some 1-Wire sensors report two parameters, such as temperature and humidity. In this case, parameter *p* selects which parameter the offset applies to.) (*n* = 1 to 4 for the respective sensor, *p* = 1 or 2) Example: so11=0.85 – sets the offset of parameter 1 for 1-Wire sensor 1 to 0.85.
ad**n**=**x**	Sets the description for analog input *n*. (*n* = 1 or 2 for the respective analog input)

	Example: ad1=AN1 – sets the description of analog input 1 to "AN1".
au**n**=**x**	Sets the unit for analog input *n*. (*n* = 1 or 2 for the respective analog input) Example: au1=VDC – sets the unit of analog input 1 to VDC.
am**n**=**x**	Sets the multiplier for analog input *n*. (*n* = 1 or 2 for the respective analog input) Example: am1=2.45 – sets the multiplier of analog input 1 to 2.45.
ao**n**=**x**	Sets the offset for analog input *n*. (*n* = 1 or 2 for the respective analog input) Example: ao1=0.76 – sets the offset of analog input 1 to 0.76.
dd**n**=**x**	Sets the description for digital input *n*. (*n* = 1 or 2 for the respective digital input) Example: dd1=DIN1 – sets the description of digital input 1 to "DIN1".
dl**n**=**x**	Sets the low-level description for digital input *n*. (*n* = 1 or 2 for the respective digital input) Example: dl1=CLOSED – sets the low-level description of digital input 1 to "CLOSED".
dh**n**=**x**	Sets the high-level description for digital input *n*. (*n* = 1 or 2 for the respective digital input) Example: dh1=OPEN – sets the high-level description of digital input 1 to "OPEN".
rd**n**=**x**	Sets the description for relay *n*. (*n* = 1 or 2 for the respective relay) Example: rd1=REL1 – sets the description of relay 1 to "REL1".
pw**n**=**x**	Sets the pulse time for relay *n* in seconds. (*n* = 1 or 2 for the respective relay) Example: pw1=0.8 – sets the pulse time of relay 1 to 0.8 seconds.
rc**n**=**x**	Sets the alarm source that activates relay *n*. (n = 1 or 2 for the respective relay) Where *x* is: HTTP API (0) S11 (1), S12 (2), S21 (3), S22 (4) S31 (5), S32 (6), S41 (7), S42 (8) MBS01 (9) to MBS10 (18) Analog input 1 (19), Analog input 2 (20) Digital input 1 (21), Digital input 2 (22) Any Alarm (23) Example: rc1=11 – sets relay 1 to be activated by MB sensor 3. rc2=0 – sets relay 2 to be activated by the HTTP API.

aae**n**=**x**	Sets the action performed by relay n when activated by an alarm. (*n* = 1 or 2 for the respective relay, *x* = 0 – Turn ON, 2 – Single pulse) Example: aaen1=0 – sets relay 1 to turn ON when activated by an alarm.
Conditions setup	
sn**np**=**x**	Sets the minimum value for parameter *p* of 1-Wire sensor *n*. (Some 1-Wire sensors report two parameters, such as temperature and humidity. In this case, parameter *p* selects which parameter the limit applies to.) (*n* = 1 to 4 for the respective sensor, *p* = 1 or 2) Example: sn12=30.0 – sets the minimum value of parameter 2 for 1-Wire sensor 1 to 30.0.
sx**np**=**x**	Sets the maximum value for parameter *p* of 1-Wire sensor *n*. (Some 1-Wire sensors report two parameters, such as temperature and humidity. In this case, parameter *p* selects which parameter the limit applies to.) (*n* = 1 to 4 for the respective sensor, *p* = 1 or 2) Example: sx42=40.0 – sets the maximum value of parameter 2 for 1-Wire sensor 4 to 40.0.
sy**np**=**x**	Sets the hysteresis value for parameter *p* of 1-Wire sensor *n*. (Some 1-Wire sensors report two parameters, such as temperature and humidity. In this case, parameter p selects which parameter the hysteresis applies to.) (*n* = 1 to 4 for the respective sensor, *p* = 1 or 2) Example: sy31=2.0 – sets the hysteresis value of parameter 1 for 1-Wire sensor 3 to 2.0.
delsen=**x**	Sets the notification delay for 1-Wire sensors. (*x* = 0 to 3600 seconds) Example: delmbsen=35 – sets the notification delay for Modbus sensors to 35 seconds.
mbsn**n**=**x**	Sets the minimum value for Modbus sensor *n*. (*n* = 1 to 10 for the respective Modbus sensor) Example: mbsn1=10.2 – sets the minimum value of Modbus sensor 1 to 10.2.
mbsx**n**=**x**	Sets the maximum value for Modbus sensor *n*. (*n* = 1 to 10 for the respective Modbus sensor) Example: mbsx2=50.6 – sets the maximum value of Modbus sensor 2 to 50.6.

mbsy**n**=**x**	Sets the hysteresis value for Modbus sensor *n*. (*n* = 1 to 10 for the respective Modbus sensor) Example: mbsy3=2.0 – sets the hysteresis value of Modbus sensor 3 to 2.0.
delmbsen=**x**	Sets the notification delay for Modbus sensors. (*x* = 0 to 3600 seconds) Example: delmbsen=30 – sets the notification delay for Modbus sensors to 30 seconds.
vn**n**=**x**	Sets the minimum value for analog input *n*. (*n* = 1 or 2 for the respective analog input) Example: vn1=10.0 – sets the minimum value of analog input 1 to 10.0.
vx**n**=**x**	Sets the maximum value for analog input *n*. (*n* = 1 or 2 for the respective analog input) Example: vx2=20.0 – sets the maximum value of analog input 2 to 20.0.
vy**n**=**x**	Sets the hysteresis value for analog input *n*. (*n* = 1 or 2 for the respective analog input) Example: vy1=1.0 – sets the hysteresis value of analog input 1 to 1.0.
delanl=**x**	Sets the notification delay for analog inputs. (*x* = 0 to 3600 seconds) Example: delanl=60 – sets the notification delay for analog inputs to 60 seconds.
ds**n**=**x**	Selects the alarm state for digital input *n*. (*n* = 1 or 2 for the respective digital input, *x* = 0 – Low level, 1 – High level) Example: ds1=1 – sets the alarm state of digital input 1 to High level.
dda**n**=**x**	Sets the Close-to-Open delay for digital input *n*. (*n* = 1 or 2 for the respective digital input, *x*= 0 to 3600 seconds) Example: dda1=10 – sets the Close-to-Open delay of digital input 1 to 10 seconds.
ddd**n**=**x**	Sets the Open-to-Close delay for digital input n. (*n* = 1 or 2 for the respective digital input, *x* = 0 to 3600 seconds) Example: ddd2=5 – sets the Open-to-Close delay of digital input 2 to 5 seconds.
deldig=**x**	Sets the notification delay for digital inputs. (*x* = 0 to 3600 seconds)

	Example: deldig=30 – sets the notification delay for digital inputs to 30 seconds.
System setup	
WAU=**x**	Enables or disables Web Access Authentication. (*x* = 0 – Disable, 1 – Enable) Example: WAU=1 – enables Web Access Authentication.
hPrt=**x**	Sets the HTTP port. Example: hPrt=8080 – sets the HTTP port to 8080.
XAU=**x**	Enables or disables HTTP API authentication. (*x* = 0 – Disable, 1 – Enable) Example: XAU=1 – enables HTTP API authentication.
tu=**x**	Sets the temperature unit. (*x* = 0 – Celsius, 1 – Fahrenheit) Example: tu=0 – sets the temperature unit to Celsius.
pu=**x**	Sets the pressure unit. (x = 0 – hPa, 1 – mbar, 2 – mmHg) Example: pu=2 – sets the pressure unit to mmHg.
hn=**x**	Sets the host name. (*x* = up to 31 characters) Example: hn=TCW320 – sets the host name to "TCW320".
sn=**x**	Sets the system name. (*x* = up to 31 characters) Example: sn=BoilerRoom – sets the system name to "BoilerRoom".
sl=**x**	Sets the system location. (*x*= up to 31 characters) Example: sl=Warehouse – sets the system location to "Warehouse".
sc=**x**	Sets the system contact. (*x* = up to 31 characters) Example: sc=John Smith – sets the system contact to "John Smith".
Time setup	
TS=**x**	Sets the NTP server address. (*x* = IP address or hostname with optional port, up to 63 characters) Example: TS=time.google.com:123 – sets the NTP server address to time.google.com:123.
TZ=**hh:mm**	Sets the time zone. (*hh:mm* = UTC offset) Example: TZ=+03:00 – sets the time zone to UTC+03:00.

TD= **hh:mm:ss**	Sets the current time. Example: TD=14:30:00 – sets the current time to 14:30:00.
TC=**x**	Enables or disables NTP time synchronization. (*x* = 0 – Disable, 1 – Enable) Example: TC=1 – enables NTP time synchronization.
interval1=**x**	Sets the NTP time synchronization interval in hours. Example: interval1=12 – sets the NTP time synchronization interval to 12 hours.
interval2=**x**	Sets the retry interval for NTP time synchronization if the previous attempt fails. Example: interval2=1 – retries NTP time synchronization after 1 hour if the previous attempt fails.
HTTP POST	
pmet=**x**	Enables or disables HTTP POST. (*x* = 0 – Disable, 1 – Enable) xample: pmet=1 – enables HTTP POST.
pushtls=**x**	Selects the HTTP POST protocol. (*x* = 0 – HTTP, 1 – HTTPS) Example: pushtls=1 – sets the HTTP POST protocol to HTTPS.
dataf=**x**	Sets the data format for HTTP POST. (*x* = 0 – XML, 1 – JSON) Example: dataf=1 – sets the HTTP POST data format to JSON.
purl1=**x**	Sets the URL for HTTP POST to Server 1. (*x* = full URL, up to 127 characters) Example: purl1=http://212.25.45.120:30181/xampp/1/pushtest.php – sets the URL for HTTP POST to Server 1.
purl2=**x**	Sets the URL for HTTP POST to Server 2. (*x* = full URL, up to 127 characters) Example: purl2=http://212.25.45.120:30181/xampp/2/pushtest.php – sets the URL for HTTP POST to Server 2.
pper=**x**	Sets the HTTP POST interval in seconds. (*x* = 10 to 14400) Example: pper=600 – sets the HTTP POST interval to 600 seconds.
pperh=**hh:mm:ss**	Sets the HTTP POST interval in *hh:mm:ss* format. Example: pperh=00:30:00 – sets the HTTP POST interval to 30 minutes.
dk=**x**	Sets the HTTP POST key. (*x* = up to 127 characters)

	Example: dk=MySecretKey – sets the HTTP POST key to "MySecretKey".
pans=**x**	Enables or disables processing of the HTTP POST server response. (*x* = 0 – Disable, 1 – Enable) Example: pans=1 – enables processing of the HTTP POST server response.
permode=**x**	Sets the HTTP POST transmission mode. (*x* = 0 – Period & Alarm, 1 – Period, 2 – Alarm) Example: permode=2 – sets the HTTP POST transmission mode to Alarm.
save	Saves all configuration changes (except relay states) to the internal flash memory. Because flash memory has a limited write endurance, use this command only when necessary. Example: pper=120&save – sets the HTTP POST interval to 120 seconds and saves the configuration.
set FIN	Terminates the HTTP POST session. Example: set FIN – terminates the current HTTP POST session.

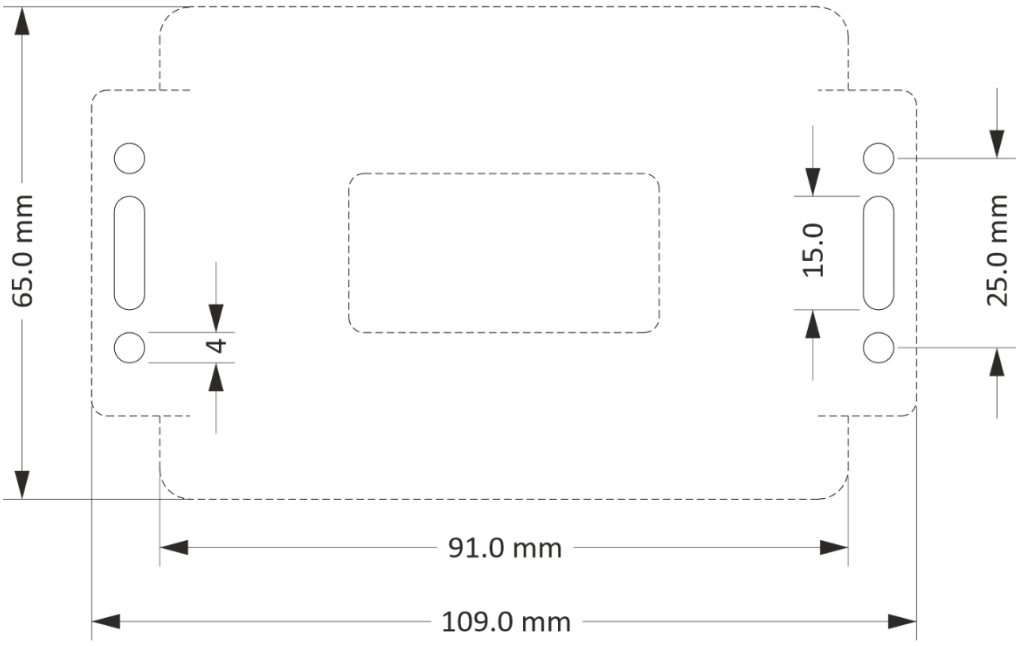

65.0 mm
4
15.0
25.0 mm
91.0 mm
109.0 mm

Fig.1

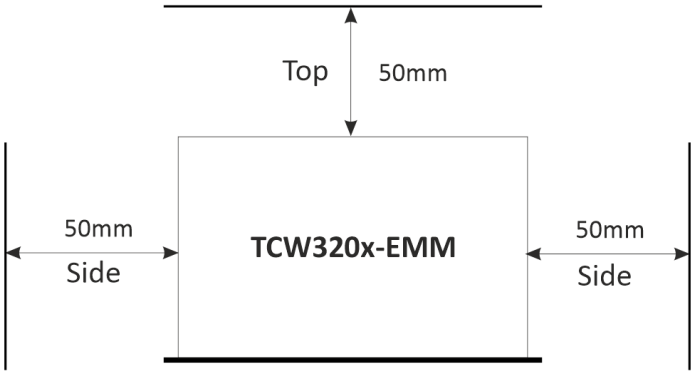

Top
50mm
50mm
Side
TCW320x-EMM
50mm
Side

Fig.2