control solutions

TERACOM

4G LTE wireless I/O module

TCG120-4E / TCG120-4G

USER MANUAL

1. Short description

The TCG120-4 is a 4G LTE Cat.1 wireless I/O module that also supports 2G communication, enabling data connectivity over LTE-FDD, EDGE, and GPRS networks. It features two digital inputs, two analog inputs, a 1-Wire interface for connecting external sensors, and two relays. The relays can be activated remotely via SMS, HTTP API commands, or MQTT messages, or locally based on the status of monitored parameters.

This module is fully compatible with all Teracom 1-Wire sensors, providing a wide range of integration possibilities. It supports an HTTP API, making it suitable for modern IoT applications. The device periodically sends an HTTP/HTTPS POST request to a remote server containing a status file in XML or JSON format.

Additionally, the device supports MQTT for publishing sensor data and receiving commands as a subscriber, enabling real-time monitoring and control. It is compatible with MQTT 3.1.1 and allows uploading client certificates for secure communication with MQTT brokers, making it suitable for integration with cloud platforms such as AWS IoT Core.

The TCG120-4 is a versatile and reliable solution for IoT projects requiring robust wireless communication, flexible integration options, and efficient remote monitoring and control.

2. Features

- Multi-band connectivity
- Flexible setup (USB, SMS, HTTP API)
- 2 digital inputs (dry contact)
- 2 analog inputs (0–60 VDC)
- 2 relays (NO/NC)
- 1-Wire interface (Teracom sensors support)
- HTTP API support
- Periodic HTTP/HTTPS POST (XML/JSON status)
- Remote control via HTTP API
- MQTT v3.1.1 support (publisher/subscriber)
- MQTT client certificate support
- Cloud integration (e.g. AWS IoT Core)
- SMS notifications (up to 5 numbers)
- Email notifications (up to 5 recipients)
- GSM-based geolocation
- Firmware update (USB or mobile network)

3. Applications

The TCG120-4 is a versatile device designed for applications requiring reliable cellular connectivity, remote monitoring, and control. It is suitable for industrial, infrastructure, and IoT environments where Ethernet connectivity is not available or not practical.

- SCADA and industrial automation

  The TCG120-4 integrates into SCADA systems for monitoring and control of industrial processes. It supports data exchange via HTTP API and MQTT, enabling reliable operation in distributed systems and remote locations.
- IoT systems with HTTP API integration

  The TCG120-4 can be integrated with platforms such as ThingsBoard using HTTP API and periodic HTTP/HTTPS POST updates. This enables centralized data collection, visualization, and device management

- Remote infrastructure monitoring

  The TCG120-4 is ideal for monitoring remote assets such as pipelines, tanks, and communication equipment. It enables real-time data reporting, alarm notifications (SMS/email), and remote control over cellular networks.
- Smart agriculture in remote locations

  The device can be used in agricultural environments for irrigation control, soil monitoring, and weather tracking. It supports integration with cloud platforms such as AWS IoT Core using MQTT v3.1.1, including client certificate-based authentication.

4. Specifications

- Physical characteristics

  Dimensions: 109 x 65 x 32 mm

  Weight: 152 g
- Environmental conditions

  Operating temperature range: -20 to 55°C

  Operating temperature range for USB setup: 0 to 40°C

  Storage temperature range: -25 to 60°C

  Ambient relative humidity range: 5 to 85% (non-condensing)
- Standards and Certifications

  Safety: EN 62368-1:2014 + EN 62368-1:2014/AC1-3:2015 + EN 62368-1:2014/A11:2017 + EN 62368-1:2014/AC:2017-03:2017, EN 62311:2008

  EMC: EN 55032:2015 + EN 55032:2015/AC:2016-07 + EN 55032:2015/A11:2020
  EN 55024:2010 + EN 55024:2010/A1:2015
  EN 61000-3-2:2014
  EN 61000-3-3:2013 + EN 61000-3-3:2013/A1:2019

  Radio: EN 301 489-19 V2.1.1, EN 301511 V12.5.1, EN 301 908-1 V11.1.1, EN 301 908-13 V11.1.2, EN 301 908-2 V11.1.2, EN 303 413 V1.1.1

  Green: RoHS compliant
- Warranty

  Warranty period: 3 years
- Power supply

  Operating voltage range (including -15/+20% according to IEC 62368-1): 10 to 28 VDC

  Current consumption: 0.24 A @ 12 VDC
- Cellular interface

  SIM card size: Mini SIM (2FF)

  Antenna connector: SMA-F

  Standards: LTE-FDD, EDGE and GPRS

  TCG120-4E h2.xx bands:

  4G LTE: B1/B3/B5/B7/B8/B20

  2G: B3/B8

  TCG120-4G h2.xx bands:

  4G LTE: B1/B2/B3/B4/B5/B7/B8/B12/B13/B18/B19/B20/B25/B26/B28/ B66/B34/B38/B39/B40/B41

  3G: B1/B2/B4/B5/B6/B8/B19

  2G: B2/B3/B5/B8

⚠ Attention!

The device can be used only in networks that support IPv4 addresses (IPv6 is not supported).
SMS functionality relies on IMS (IP Multimedia Subsystem).
We recommend contacting your mobile operator to verify IPv4 and IMS support.

- Analog inputs
  Isolation: Non-isolated
  Type: Single-ended
  Resolution: 10 bits
  Input range: 0 to 60 VDC
  Accuracy: ±1%
  Sampling period: 500 ms per channel (averaged value of 250 samples)
  Input impedance: 1 MΩ (min.)
- Digital inputs
  Isolation: Non-isolated
  Type: Dry contact
  Sampling period: 10ms
  Digital filtering: 30ms (minimum, expandable)
- Relay outputs
  Type: Form C (N.O. and N.C. contacts)
  Contact current rating: 3 A @ 24 VDC, 30 VAC (resistive load)
  Initial insulation resistance: 100 MΩ (min.) @ 500 VDC
  Mechanical endurance: 10 000 000 operations
  Electrical endurance: 100 000 operations @ 3 A resistive load
  Contact resistance (initial): 50 mΩ max.
  Minimum switching frequency: 0.1 Hz (at rated load)

  ⚠ Attention!

  The device does not contain any internal overcurrent protection facilities on the relays' contact lines. External fuses or short circuit current limiting circuit breakers, rated to 3 A, are to be used for overcurrent protection of the connecting lines.
- 1-Wire interface
  Output voltage: 5.0 ± 0.3 VDC
  Maximum output current: 200 mA
- Internal flash memory
  Endurance: 100,000 cycles (each relay status change or settings save counts as one memory cycle).

5. Installation

The TCG120-4 must be installed by qualified personnel to ensure proper operation and safety. The installation process includes mounting the device, connecting the antenna, wiring the inputs and outputs, supplying power, and completing the configuration via a web browser.

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

Attention! Before inserting the SIM card, ensure that the PIN code is disabled.

5.1. Mounting

The TCG120-4 must be installed in a clean, dry location on a non-flammable surface. If the installation environment has high ambient temperatures, ensure adequate ventilation around the device to maintain optimal operation.

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

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

5.2. Connectors

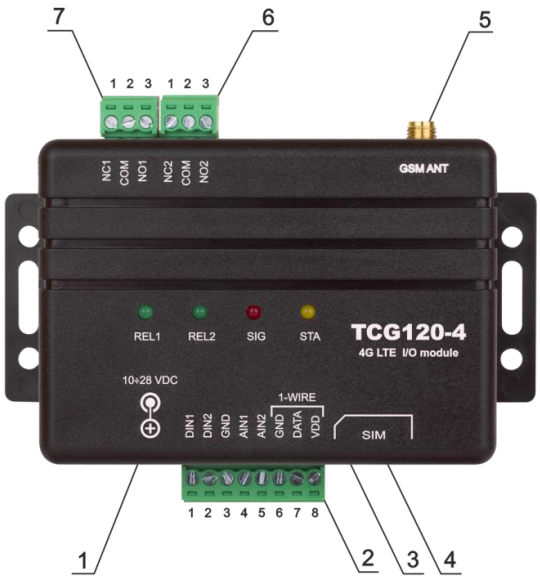
Attention! Always disconnect the power supply from the device before starting any wiring to avoid electrical hazards or equipment damage.

5.2.1 Follow these steps for correct wiring:

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

It is strongly recommended to test and configure the TCG120-4 without connecting it to any controlled devices. Ensure that all wires are securely attached to the terminals and that the terminal screws are properly tightened. Improper wiring or configuration may cause permanent damage to the module, the connected equipment, or both.

Refer to the diagram below for the exact locations of inputs and outputs:



Connector 1 – Power; 2.1 x 5.5 mm DC jack, center positive
Connector 2, Pin 1 - Digital input 1 (DIN1)
Connector 2, Pin 2 - Digital input 2 (DIN2)
Connector 2, Pin 3 - GND
Connector 2, Pin 4 - Analog input 1 (AIN1)
Connector 2, Pin 5 - Analog input 2 (AIN2)
Connector 2, Pin 6 - GND
Connector 2, Pin 7 – 1-Wire data
Connector 2, Pin 8 – 1-Wire power supply
Connector 3 – Mini USB
Connector 4 – SIM card holder
Connector 5 – GSM antenna
Connector 6, Pin 1 – Relay2 NC
Connector 6, Pin 2 – Relay2 COM
Connector 6, Pin 3 – Relay2 NO
Connector 7, Pin 1 – Relay1 NC
Connector 7, Pin 2 – Relay1 COM
Connector 7, Pin 3 – Relay1 NO

5.2.2 Power supply connection

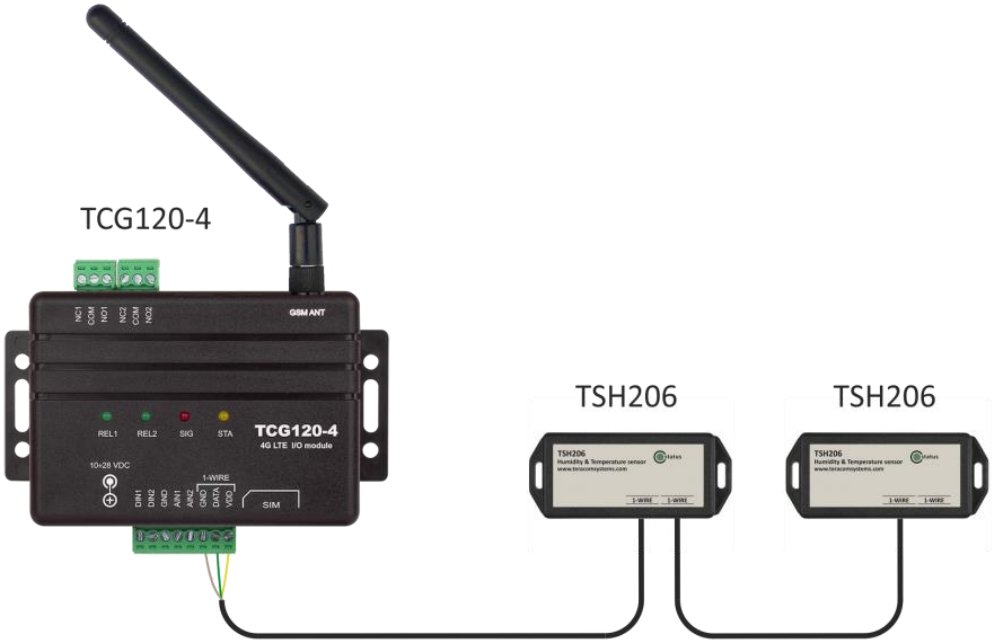
The TCG120-4 must be powered by the adapter SYS1308(N)-2412-W2E or equivalent, suitable for overvoltage category II and certified for safety compliance. The power supply should be able to withstand short circuits and overloads on the secondary circuit. Ensure that the device can be easily disconnected from the power supply during operation.

5.2.3 1-Wire interface

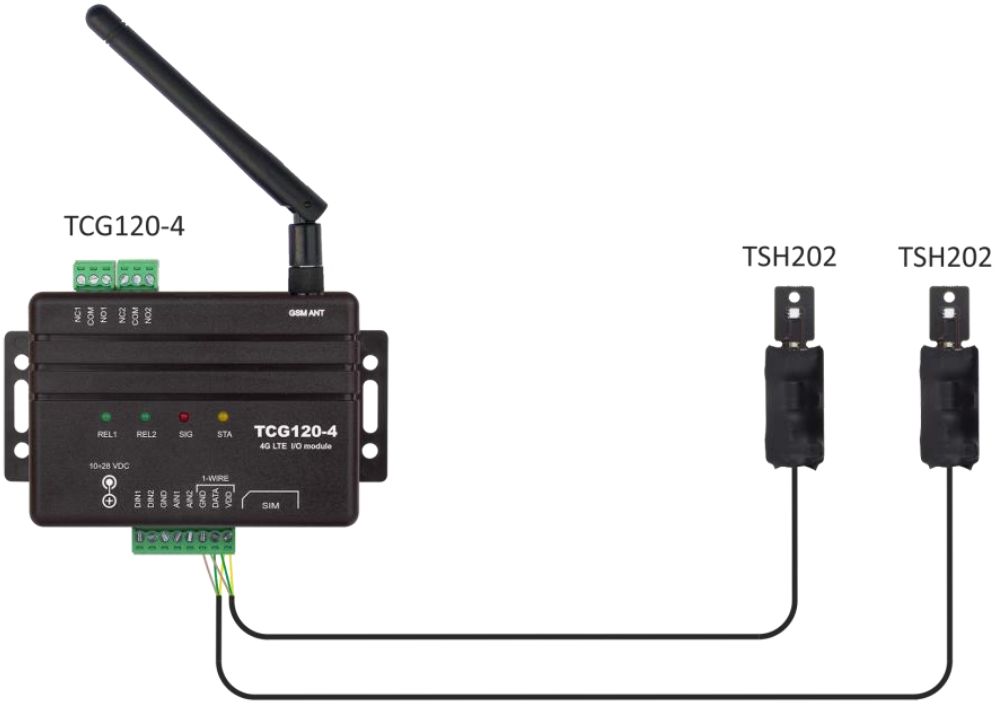
The 1-Wire interface, a registered trademark of Analog Devices Inc., is designed for connecting multiple sensors over short wiring distances. However, it is not suitable for long cable runs or environments with high electrical noise.

The TCG120-4 supports up to four 1-Wire sensors simultaneously. All Teracom 1-Wire sensors are compatible with this device. Once connected, sensors are automatically detected, and the corresponding measurement dimension is assigned.

For multi-sensor setups, it is strongly recommended to use a daisy-chain (linear) topology to ensure proper operation and reliable communication between the device and sensors:



Star topology may be used as a last resort, but only for connecting up to four sensors with a total cable length not exceeding 10 meters:



It is strongly recommended to use only UTP/FTP cables for the 1-Wire interface and to keep the total cable length within 30 meters. While functionality has been observed at longer distances, error-free operation cannot be guaranteed beyond the specified limit.

For best practices and detailed guidance on reliable long-line 1-Wire networks, refer to the guidelines available here.

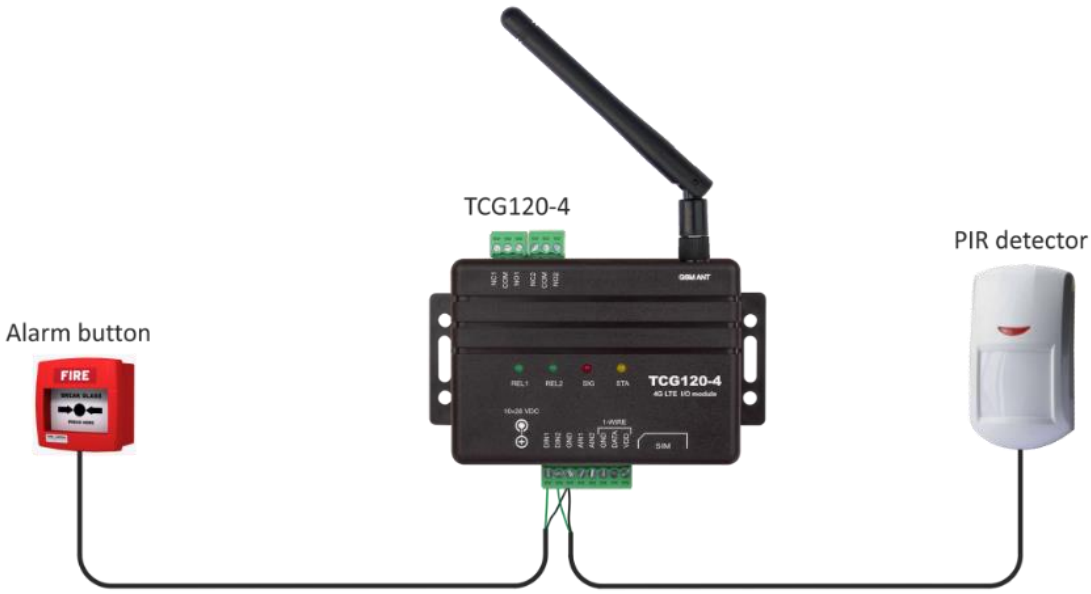
Proper operation is guaranteed only when using Teracom 1-Wire sensors.

5.2.4 Digital inputs connection

Note that all inputs are non-isolated from the power supply ground.

The TCG120-4 digital inputs are intended solely for use with dry contact devices. A dry contact refers to devices with open-drain outputs or relay outputs, such as door contact switches, push buttons, and PIR (passive infrared) detectors.

To connect an alarm button to a TCG120-4 digital input, one side of the button should be connected to the "DIN" terminal, while the other side is connected to the "GND" terminal. The diagram below shows this connection.

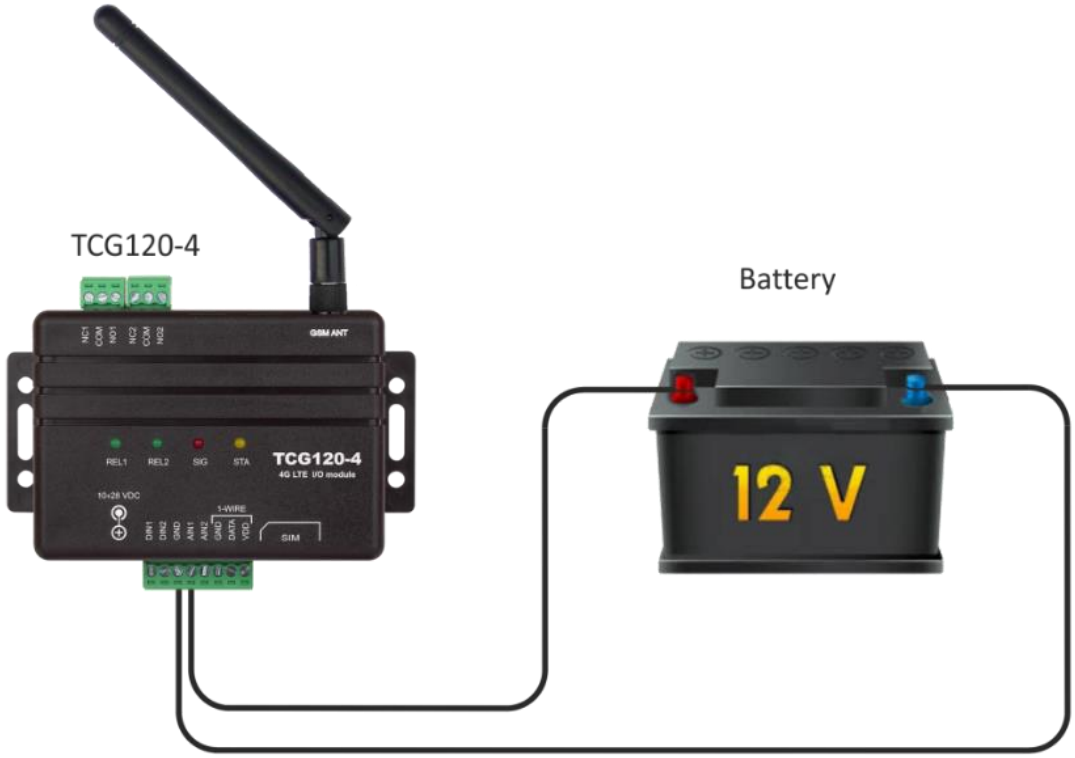


The maximum cable length should not exceed 30 meters.

5.2.5 Analog inputs connection

Note that all inputs are non-isolated from the power supply ground.

The analog inputs of the TCG120-4 can be used to monitor DC voltage up to 60 VDC. The diagram below shows how to connect a battery to the analog input of TCG120-4, with the positive terminal connected to "AIN" and the negative terminal to "GND"



The maximum cable length should not exceed 30 meters.

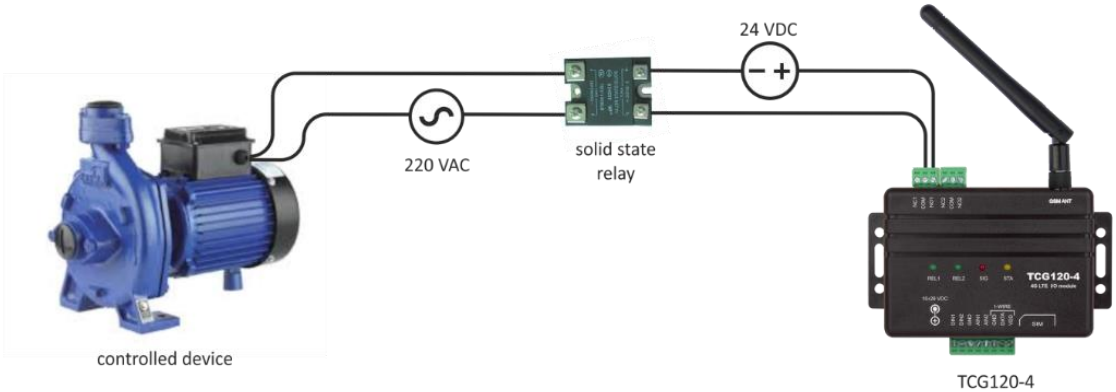
5.2.6 Relays connection

Note that the relay contacts are directly connected to the terminals and no internal overcurrent protection is provided.

For all relays, NO, NC, and COM contacts are available. If the load requires a switching current or voltage higher than the specified limits, an external relay must be used.

When mechanical relays switch inductive loads, such as motors, transformers, or other relays, an arc may form across the relay contacts each time they open. This can lead to accelerated wear, reducing their lifespan. To protect the contacts and extend their service life, it is strongly recommended to use relay contact protection devices when switching inductive loads.

The use of an external solid-state relay is recommended.



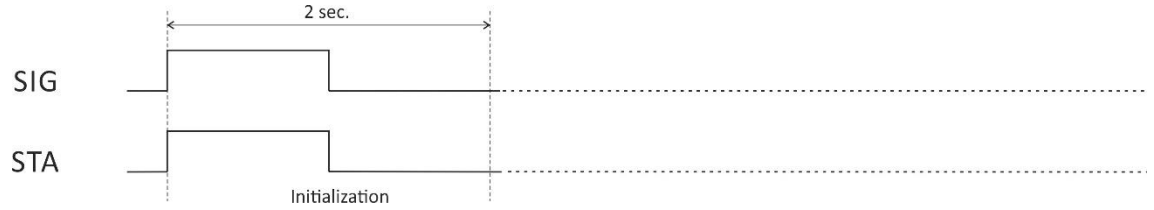
6. LED indicators

The LED indicators provide a visual representation of the module's status:

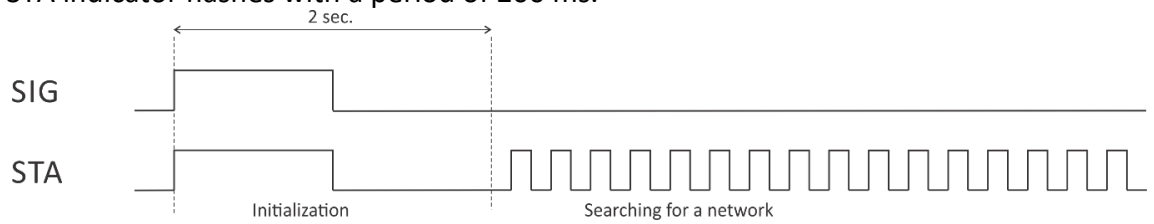
- **REL1/REL2 (green)**: This LED turns on when the corresponding relay is activated, indicating that the NO contact is connected to the COM terminal.
- **SIG (red)**: Indicates the device's status, operates in conjunction with the STA indicator.
- **STA (yellow)**: Indicates the device's status, working in conjunction with the SIG indicator.

The following states are displayed:

- **Module initialization** – during the module initialization process, the SIG (red) and STA (yellow) indicators turn on for one second immediately after power-on. Following this, they turn off for another second.

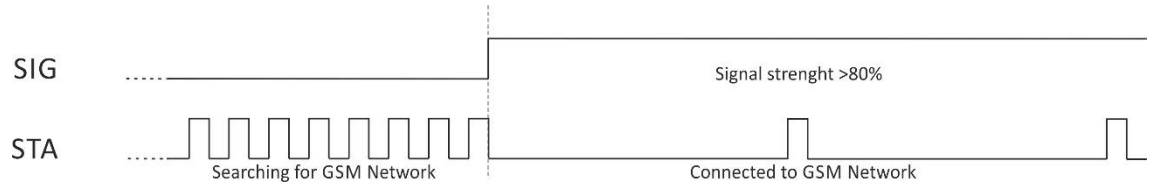


- **Searching for a network** – after the initialization process, the SIG indicator remains off, while the STA indicator flashes with a period of 200 ms.

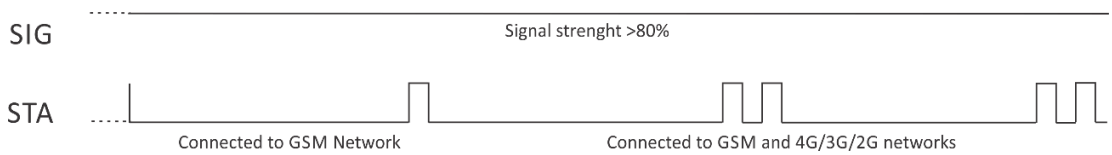


- **Connected to a network** – once the module successfully connects to the network, the STA indicator shows the type of connection, and the SIG indicator shows the signal strength.

  If there is only a GSM connection, the STA indicator flashes once for 200 ms within a 2-second cycle.



  If 4G/3G/2G connections are available, the STA indicator flashes twice for 200 ms within a 2-second cycle.



  At the same time, the SIG indicator provides five states to represent the signal strength:

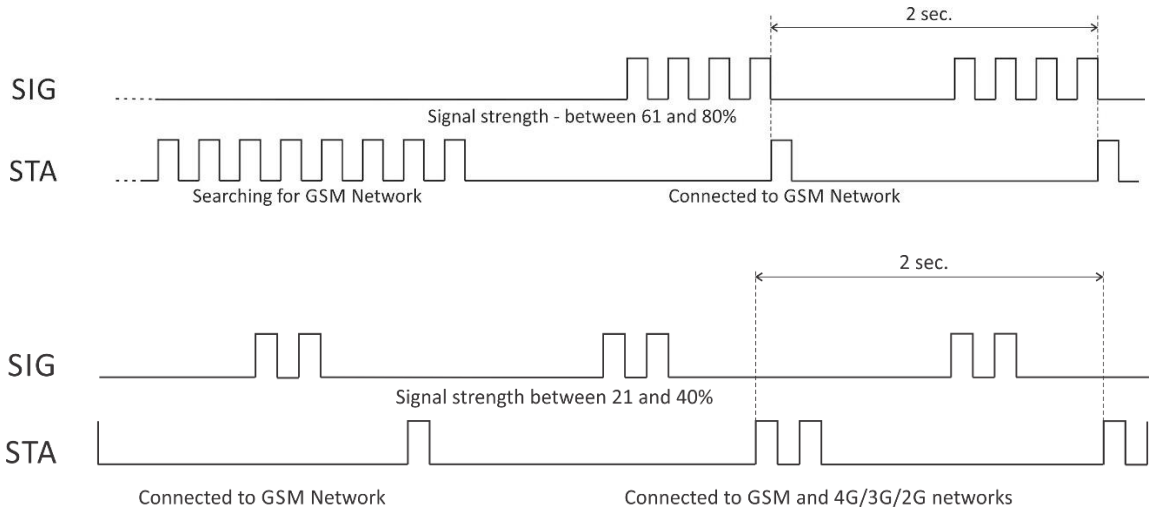
  0–20%: SIG flashes once in a 2-second period.

  21–40%: SIG flashes twice in a 2-second period.

  41–60%: SIG flashes three times in a 2-second period.

  61–80%: SIG flashes four times in a 2-second period.

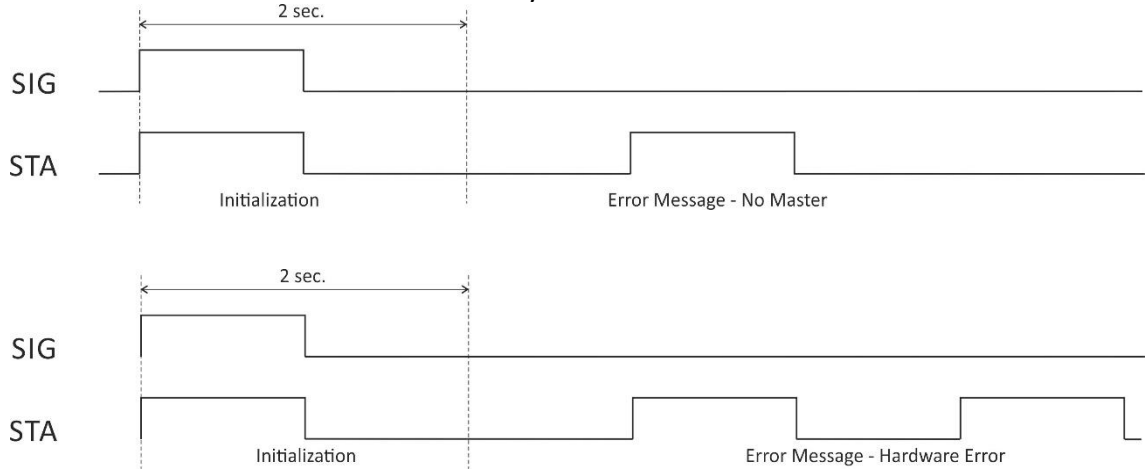
  81–100%: SIG remains on.



- **Error message** - if an error occurs after initialization, the SIG indicator will remain off, and the STA indicator will flash to indicate the type of error.

  STA flashes once for 1 second: Master phone number is not set.

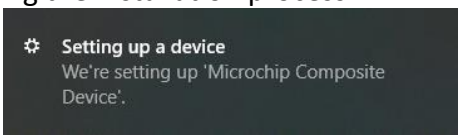
  STA flashes for 1 second within a 2-second cycle: Permanent hardware error.



7. Initial setup via USB

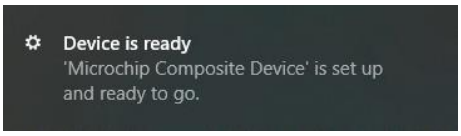
The initial setup of the TCG120-4 is performed using a computer with Windows 10 or newer version of Microsoft Windows. After the module is powered on, connect it to the computer using a USB cable (Mini USB). Once connected, the operating system will automatically begin installing the required drivers for communication with the device.

A notification will appear indicating the installation process:



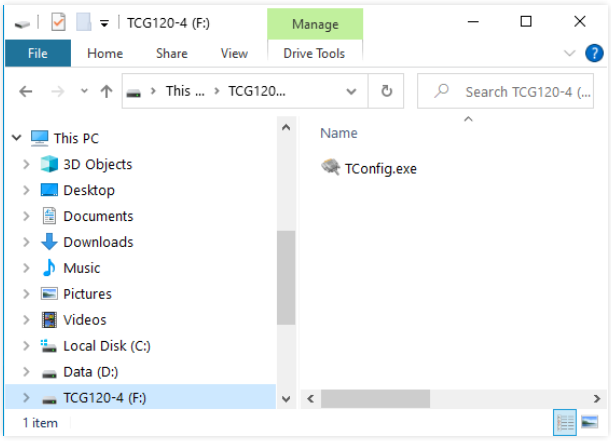
The following drivers will be installed:

- Microchip composite device
- USB Serial Port driver

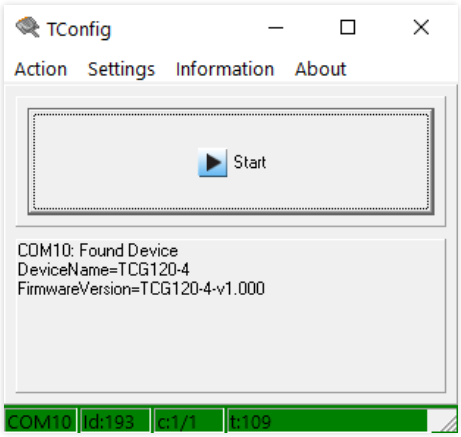


If the USB serial port driver cannot be installed automatically, it must be installed manually. The driver can be downloaded from the TCG120-4 product page at www.teracomsystems.com.

After the driver is successfully installed, the device will be recognized as a mass storage device. The following window will appear on the screen:



The only file stored on the mass storage is a tool called TConfig, which facilitates communication between the TCG120-4 and the PC:



Clicking the “Start” button will open the browser and display the Monitoring page of the TCG120-4.

7.1. Monitoring page

The monitoring page provides real-time information about the current status of the TCG120-4 inputs and outputs. It is divided into four sections: "1-Wire sensors," "Analog inputs," "Digital inputs," and "Relays." Each parameter (sensor, input, or relay) is displayed with a description of up to 15 characters. These descriptions can be customized on the Setup → Input/Output page.

7.1.1 Sensors section

1-Wire sensors

Pos	Description	Value 1	Value 2	ID
1	S1:TST1xx	25.247°C	------	28B3B464080000AB
2	S2:TSH2xx	25.875°C	45.188%RH	015225B71700FF45
3	S3	------	------	0000000000000000
4	S4	------	------	0000000000000000

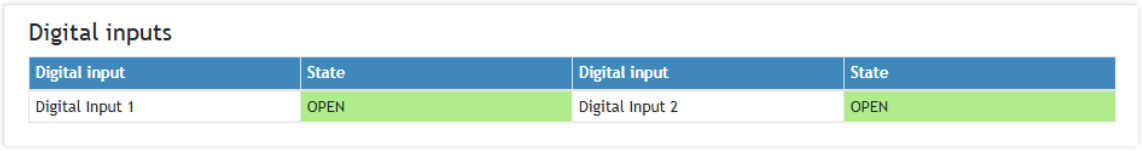
The TCG120-4 supports up to four 1-Wire sensors. All detected sensors are displayed in the "1-Wire sensors" section. These sensors must be configured in the "1-Wire sensors setup" section, accessible via Setup → Sensors.

For each sensor, the page displays a description, values, and ID information. Descriptions can be up to 15 characters long and can be modified on the Setup → Sensors page.

Sensors with a single parameter, such as Teracom temperature sensors, display their readings only in the Value 1 column. Some sensors provide two parameters. For example, TSH2xx temperature and humidity sensors display the temperature reading in the Value 1 column and the humidity reading in the Value 2 column..

7.1.2 Digital inputs section

The digital inputs are used to monitor the state of discrete devices such as motion sensors, door contacts, relay contacts, alarm outputs, and similar devices.



The state of each input is displayed both as text and with color coding (e.g. OPEN / CLOSED). The input descriptions and state texts can be customized in the Setup → Input/Output section.

7.1.3 Analog inputs section

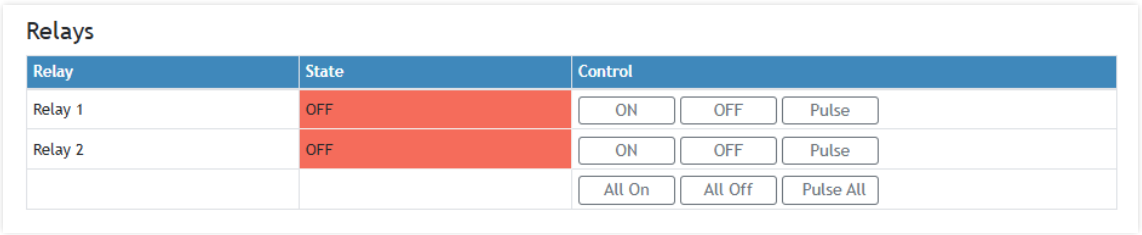
The analog inputs are used to monitor DC voltage sources, such as analog sensors, batteries, power supplies, or solar panels.

The current value of each analog input is displayed in the "Analog inputs" section.

For each analog input, the Unit, Multiplier, and Offset can be configured in the Setup → Input/Output section. The Multiplier and Offset parameters allow scaling and calibration of the measured value, for example to convert voltage readings into engineering units such as temperature, pressure, or level.

7.1.4 Relay section

This section displays the current state of the relays. Each relay can be activated remotely via SMS, HTTP API, or MQTT, or locally based on the status of a monitored parameter. For relays activated locally, a text description of the controlling parameter is shown instead of control buttons.

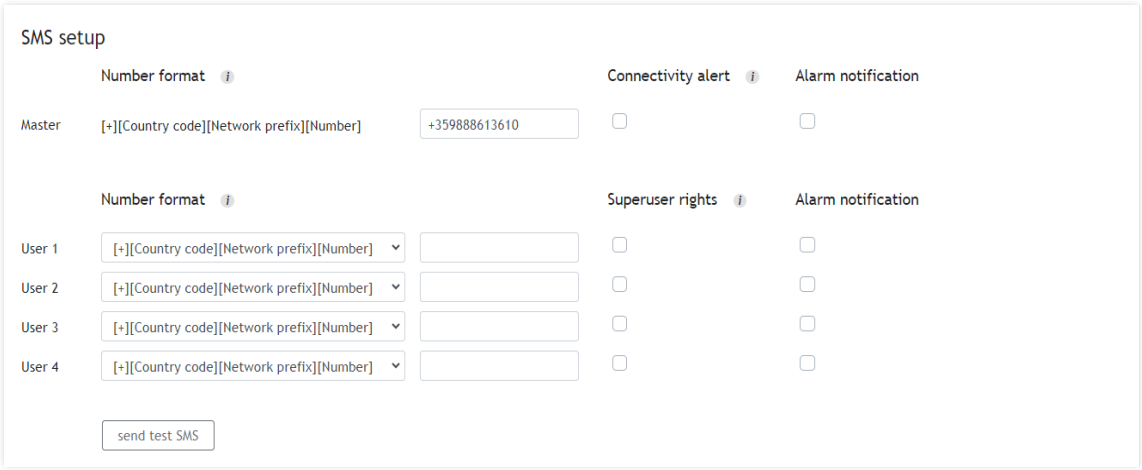


Pulse duration and parameters for local relay activation can be configured individually for each relay in the Relay outputs section on the Setup → Input/Output page.

7.2. Setup

7.2.1 SMS

In this section, you can configure SMS alarm recipients.



The Master recipient has special privileges, including the ability to modify device settings using SMS commands. By default, the Master receives SMS alarm messages, though this can be disabled by unchecking the alarm notification checkbox. Additionally, the Master will receive SMS alerts for internet connectivity loss if the "Connectivity alert" checkbox is enabled, which it is by default.

The Master's phone number must be entered in the E.164 international format (ITU-T Recommendation), consisting of a "+" sign followed by the country code and subscriber number, with a maximum length of 15 digits, to ensure global uniqueness and proper routing of calls and messages.

Up to four additional recipients can be set to receive SMS alarms when monitored parameters enter an alarm state. To enable this, the "Alarm Notification" checkbox must be activated for each number. These recipients can also query parameter states/values by sending an SMS request.

Granting "Superuser rights" to any of these four recipients allows them to send SMS commands with similar permissions to the Master, except for creating new users or changing the Master. Numbers for these users can be entered in either the E.164 international format (consisting of a "+" sign followed by the country code and subscriber number) or as short codes.

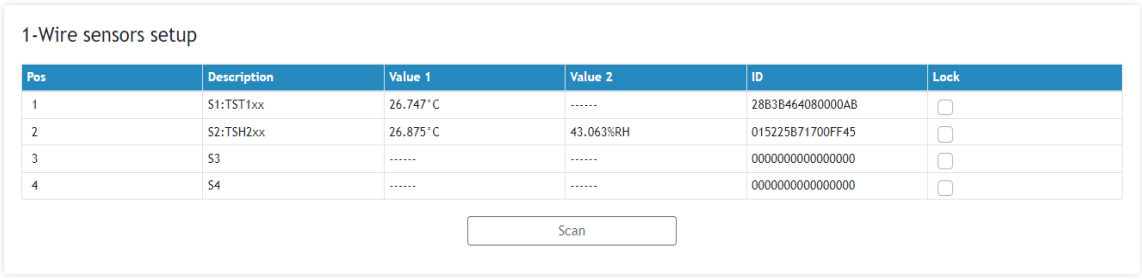
Short codes are typically 3–12 digits long and are used for high-volume application-to-person (A2P) messaging within the same country or region. These codes cannot be used for international messaging and are valid only for communication within the country where they are based.

A "Send test SMS" button allows you to send a test SMS to all recipients to verify the configuration.

All SMS commands, their syntax, and the corresponding responses are detailed in the "Setup via SMS" section.

Note: SMS is not a reliable communication method and is not recommended as a primary monitoring technology.

7.2.2 Sensors



Detection of connected 1-Wire sensors occurs automatically at startup or can be triggered manually by pressing the "Scan" button. All detected sensors are displayed in ascending order according to their unique IDs.

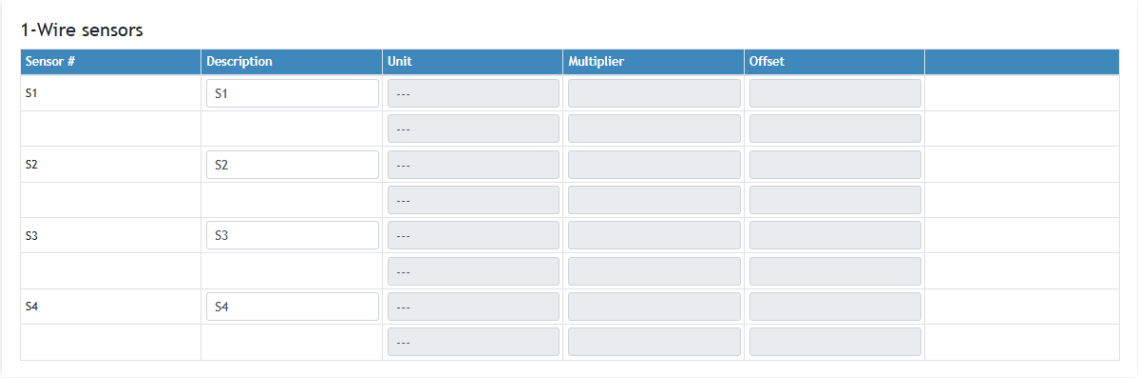
It is possible to lock a 1-Wire sensor into a specific position. To do this, sensors must be added one by one. After each addition, perform a new scan, and then lock the newly detected sensor to its position. Once all sensors are locked, removing one sensor "in the middle" will not affect the positions of the others after a device restart.

This feature is especially valuable when the TCG120-4 is integrated into a monitoring and control system managed via HTTP API commands, ensuring consistent sensor identification and positioning.

7.2.3 Input/Output

7.2.3.1 Sensors

This section allows configuration of the parameters for 1-Wire sensors.


1-Wire sensors

Sensor #	Description	Unit	Multiplier	Offset	
S1	S1	---			

S2	S2	---			

S3	S3	---			

S4	S4	---			

For each sensor, a description of up to 15 characters can be set. These descriptions will appear on the Monitoring page, Setup → Conditions page, in XML/JSON data, as well as in SMS and email alerts.

For certain sensors, additional fields - Unit, Multiplier, and Offset are available to convert raw sensor values into meaningful units. The scaled value is calculated using the following formula:

SV = RV × MU + OF

Where:

SV – scaled (displayed) value;

Unit – measurement unit;

RV – raw value from the sensor;

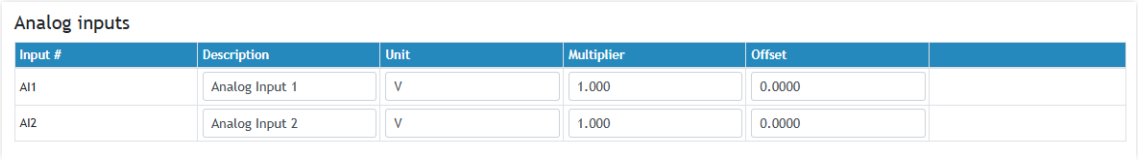
MU – multiplier;

OF – offset.

A practical example of using the Multiplier and Offset parameters is provided in the Analog inputs section.

7.2.3.2 Analog inputs

This section allows configuration of parameters for analog inputs, including descriptions, units, multiplier, and offset.


Analog inputs

Input #	Description	Unit	Multiplier	Offset	
AI1	Analog Input 1	V	1.000	0.0000	
AI2	Analog Input 2	V	1.000	0.0000	

For each analog input, a description of up to 15 characters can be set. These descriptions will appear on the Monitoring page, Setup → Conditions page, in XML/JSON data, as well as in SMS and email alerts.

The fields Unit, Multiplier, and Offset can be configured to convert raw input values into meaningful engineering units or to correct the displayed analog input readings.

The scaled value is calculated using the following formula:

SV = RV × MU + OF

Where:
SV – scaled (displayed) value;
Unit – measurement unit;
RV – raw input value;
MU – multiplier;
OF – offset.

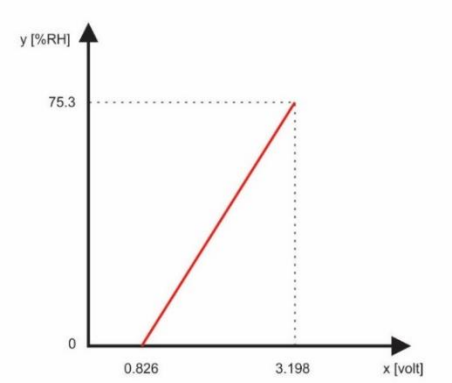
Example:

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

VOUT = 0.826 V at 0 %RH

VOUT = 3.198 V at 75.3 %RH

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



Calculation of the multiplier (MU)

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

The multiplier is calculated as follows:

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

Calculation of the offset (OF)

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

SV = RV × MU + OF

the offset can be calculated as:

OF = SV − (RV × MU)

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

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

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

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

Final formula

The final conversion formula for this sensor is:

SV = RV × 31.745 − 26.22

Verification

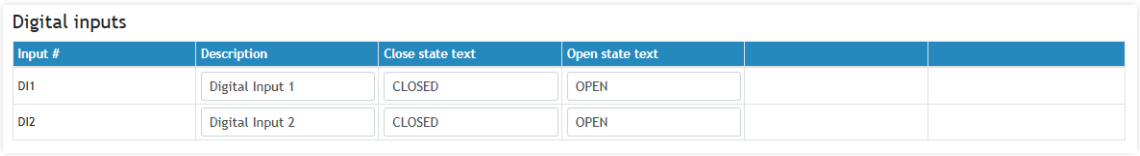
To verify the correctness of the formula, we can check the case where VOUT = 0.826 V (0 %RH):

$$SV = 0.826 \times 31.745 - 26.22 = 26.22 - 26.22 = 0\ \%RH$$

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

7.2.3.3 Digital inputs

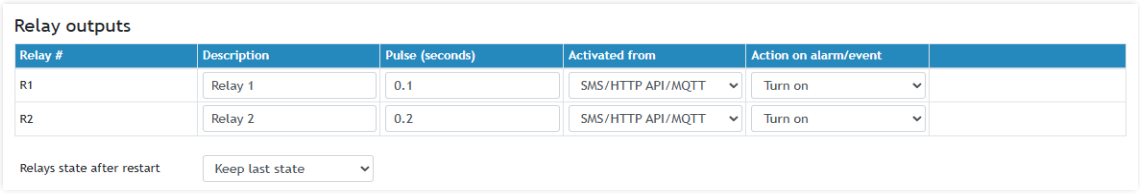
This section allows configuration of parameters for digital inputs, including descriptions and state texts.



For each digital input, a description of up to 15 characters and state texts of up to 15 characters can be set. These descriptions and state texts will appear on the Monitoring page, Setup → Conditions page, in XML/JSON data, as well as in SMS and email alerts.

7.2.3.4 Relay outputs

For each relay, a description of up to 15 characters can be set. These descriptions will appear on the Monitoring page, in XML/JSON data, as well as in SMS and email alerts.



The pulse duration for each relay can be set individually, with a resolution of 0.1 seconds and a maximum value of 3600 seconds.

Relays can be activated either remotely via SMS, HTTP API, or MQTT messages, or locally based on the status of a monitored parameter. This configuration is done using the "Activated from" drop-down menu, which offers the following options:

- SMS/HTTP API/MQTT – allows relay outputs to be activated remotely via SMS from an authorized number, through HTTP API commands, or by MQTT messages.
- Sensor parameter – the relay is activated based on the value measured by a specific 1-Wire sensor parameter, following the range rules configured in Setup → Conditions. For sensors with multiple parameters, each parameter can be selected separately (e.g. S11, S12, S21, S22).
- Analog input – the relay is activated based on the value measured by a specific analog input, following the range rules configured in Setup → Conditions.
- Digital input – the relay mirrors the state of the selected digital input.
- Any alarm – the relay is activated when any configured alarm condition is met.

The action performed when a relay is activated by a local condition can be selected from the "Action on alarm/event" drop-down menu.

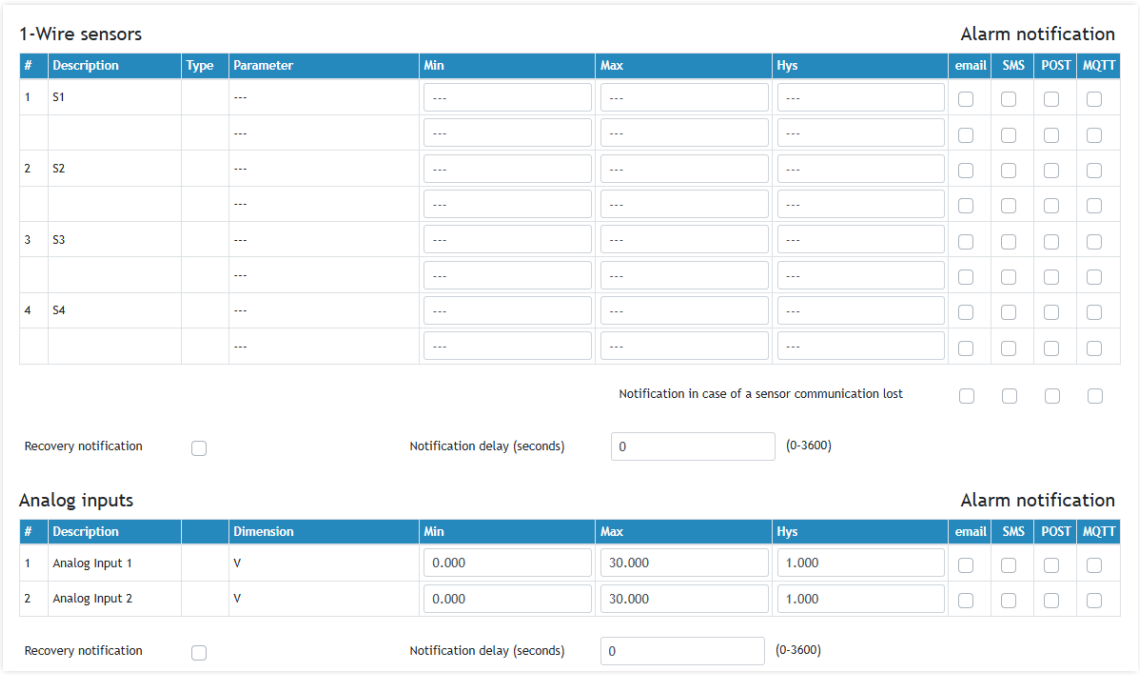
All changes in these settings must be saved by pressing the "Save" button.

7.2.4 Conditions

This section is used to configure conditions for sensors, analog inputs, and digital inputs. It allows the setup of rules that define how and when alarms or actions are triggered based on monitored parameters.

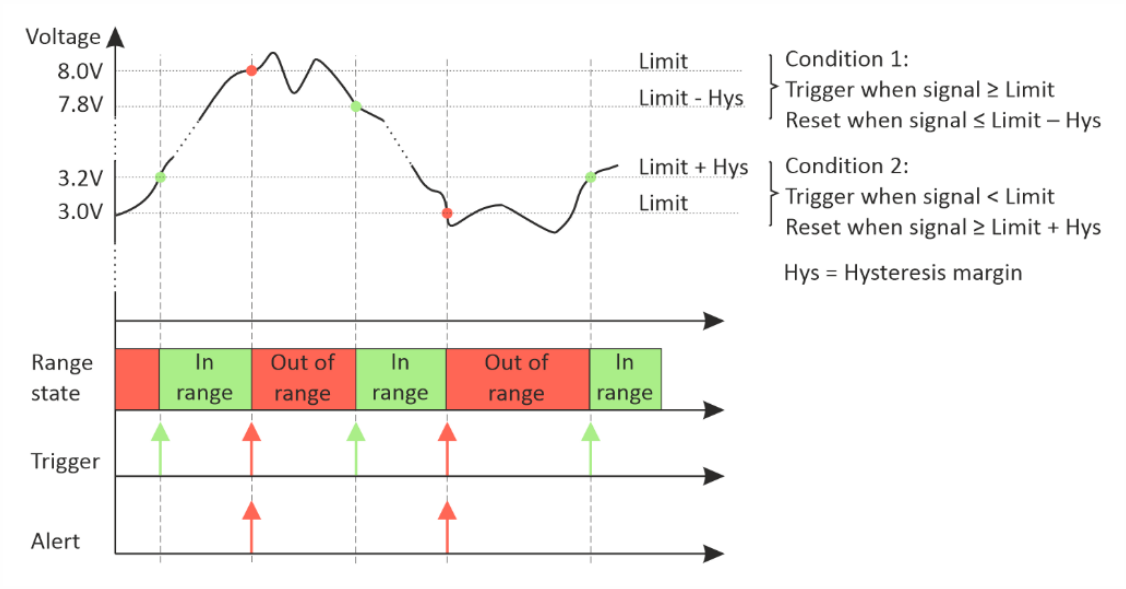
7.2.4.1 Sensors and analog inputs

This section is used to configure trigger conditions, recovery conditions, and alarm notification settings for sensors and analog inputs



Each sensor or analog input has fields for configuring trigger conditions (Min, Max, and Hys) and alarm notification methods.

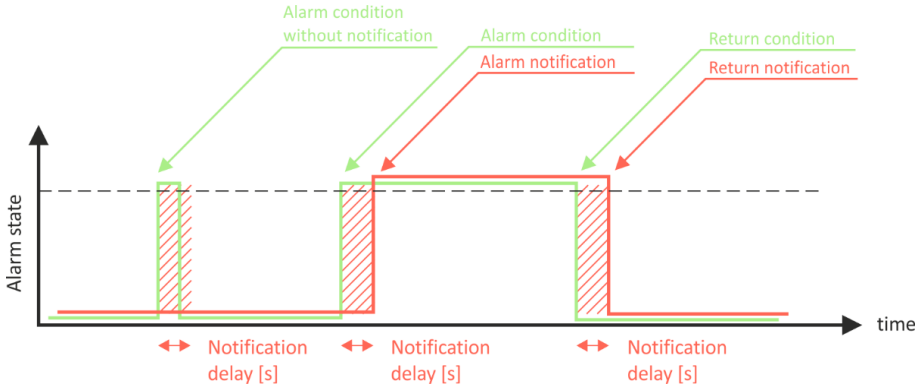
- Min and Max: These define the boundaries of the acceptable range for the monitored parameter.
  A Max trigger condition occurs when the value exceeds the upper limit.
  A Min trigger condition occurs when the value falls below the lower limit.
  In either case, the monitored parameter is considered out of range.
- Return condition: The parameter is considered back in range when:
  The value rises above (Min + Hys) after falling below the Min threshold.
  The value falls below (Max – Hys) after exceeding the Max threshold.
- Hysteresis (Hys): This setting prevents repeated triggering caused by small fluctuations around the threshold values, ensuring stable operation.



For each sensor or analog input, four independent alarm notification methods are available during an alarm condition: email, SMS, POST (HTTP POST with XML/JSON data), and MQTT publish. Each notification method can be enabled individually using a checkbox.

A global Recovery notification option is available for all sensors and analog inputs. If enabled, notifications will also be sent when the monitored parameter returns to the defined range.

A global Notification delay parameter can be configured for all sensors and analog inputs. This parameter delays the alarm notification to prevent alerts caused by short-lived alarm conditions.



The Notification delay starts when a trigger condition occurs. An alarm notification is sent only if the monitored parameter remains out of range for the entire configured delay period. If the parameter returns to the defined range before the delay expires, no notification is sent.

<u>Example – room temperature control:</u>

In this example, the TCG120-4, a TST100 sensor, and a heater are used to maintain a room temperature of at least 19°C. The initial room temperature is 17°C.

Configuration:

The TST100 sensor is assigned to the first position in the 1-Wire sensors section.

Relay1 is configured for local activation based on Sensor1 parameter 1 (S11).

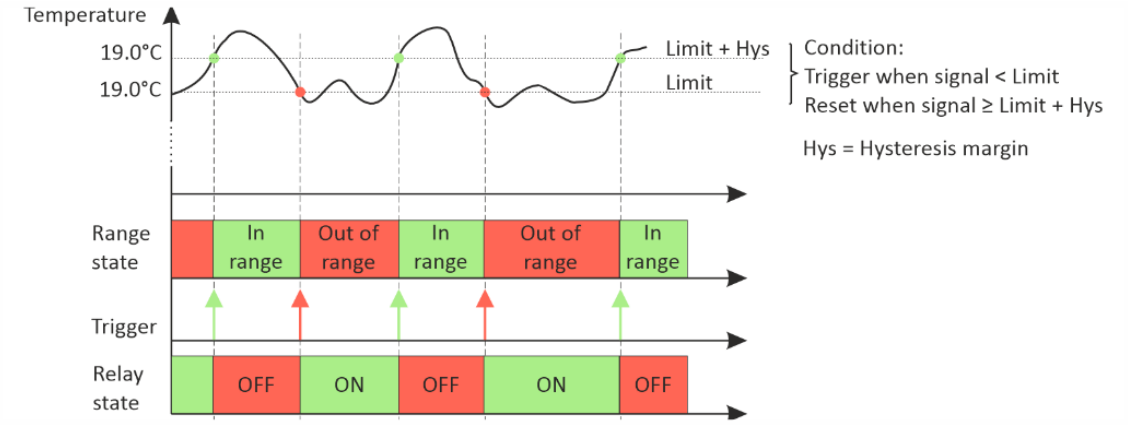
The following condition parameters are configured for Sensor1 parameter 1:

Min = 19, Max = 100, and Hys = 0.5.

Email notification is enabled for this condition.

Operation:

After power-up, the measured temperature (17°C) is below the Min threshold (19°C) and is therefore considered out of range. This creates a trigger condition and activates Relay1, switching the heater on.



As the temperature rises and reaches 19.5°C (19.0 + 0.5), the return condition is met (Min + Hys), and the temperature is considered back in range. The trigger condition is cleared and Relay1 is deactivated, switching the heater off.

As the room cools down and the temperature falls below 19°C again, the parameter becomes out of range again, creating a new trigger condition. Relay1 is activated again, switching the heater on. Since Email notification is enabled, an alarm notification is also sent.

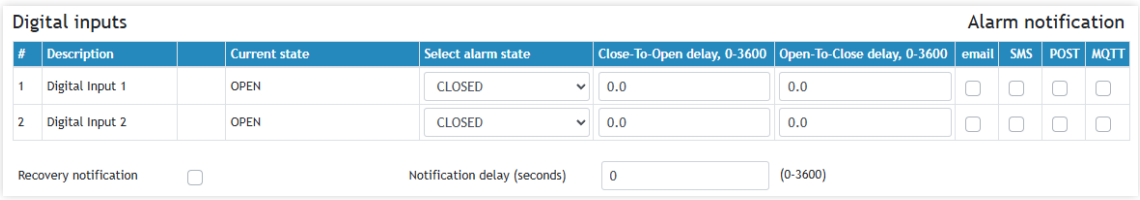
The Max value is intentionally set much higher than the target temperature to avoid unwanted trigger conditions near the operating threshold.

7.2.4.2 Digital inputs

This section is used to configure alarm notification parameters for digital inputs.

For each digital input, an alarm state must be selected. The available state texts are configured in the Setup → Input/Output section.

When a digital input enters the selected alarm state, four independent alarm notification methods are available: email, SMS, POST (HTTP POST with XML/JSON data), and MQTT publish. Each notification method can be enabled individually using a checkbox.



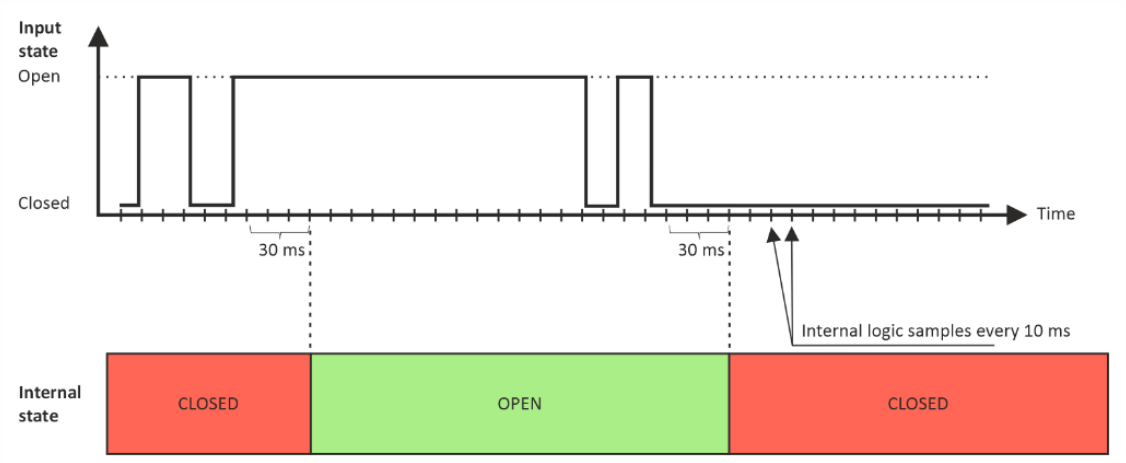
A global Recovery notification option is available for all digital inputs. If enabled, a notification will also be sent when the input returns from the alarm state to its normal state.

A global Notification delay parameter can be configured for all digital inputs. The notification delay starts when the selected alarm state is detected. A notification is sent only if the input remains in the alarm state for the entire configured delay period. If the input returns to its normal state before the delay expires, no notification is sent.

When a digital input is in an alarm state, its state is highlighted in red on the Monitoring page.

To prevent false state changes caused by short disturbances or noise, digital inputs use filtering based on internal sampling. The inputs are sampled every 10 ms. A fixed digital filter is always

applied, and a change in input state is accepted only if the same value is read in three consecutive samples (30 ms).

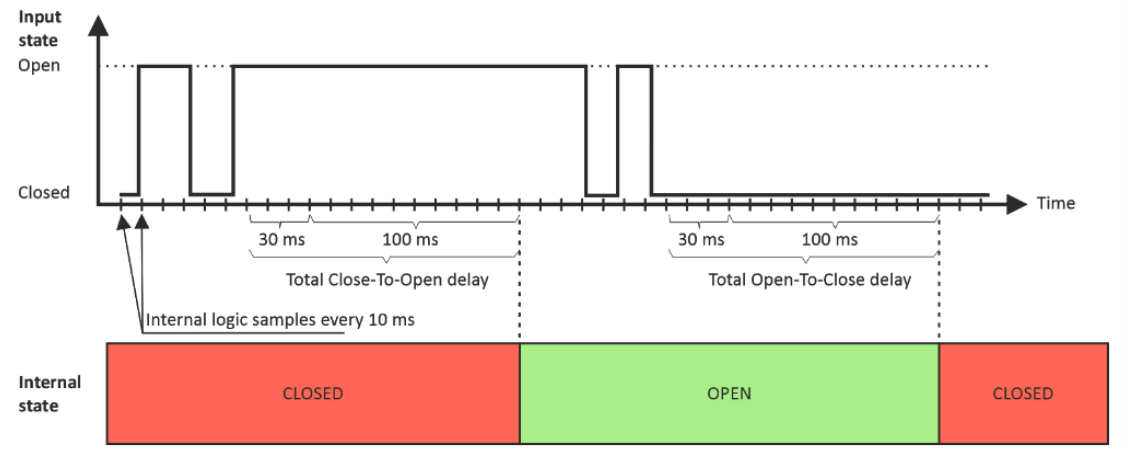


Additional filtering can be configured separately for both state transitions:

- Low-to-high delay – additional delay applied when the input changes from low level to high level.
- High-to-low delay – additional delay applied when the input changes from high level to low level.

These delays are added to the fixed 30 ms digital filter and can be configured with a resolution of 0.1 seconds. By default, both are set to 0 seconds.

In the example below, both Low-to-high and High-to-low delays are configured to 0.1 seconds (100 ms). The total Close-to-Open and Open-to-Close delay becomes 130 ms (30 ms fixed filtering + 100 ms additional delay). This means that a change in input state is accepted only if the new state is read consistently in 13 consecutive samples (13 × 10 ms = 130 ms). This mechanism prevents false triggering caused by short disturbances or noise on the input signal.

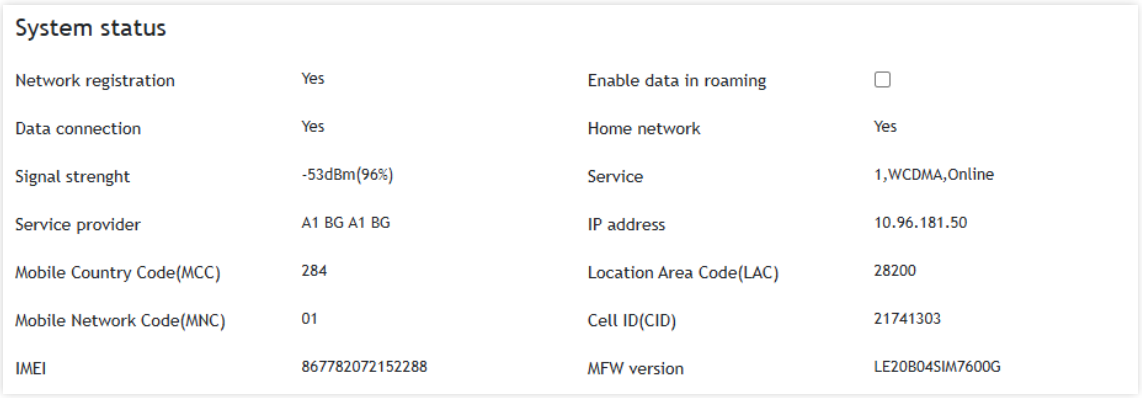


7.2.5 System

This page provides status information about the device's operation, including signal strength, IP address, service provider, and other relevant system information. It also provides options for configuring global settings related to network and communication services.

7.2.5.1 System status

This section provides status information about the module's connection to the mobile network, including information such as the IMEI and the modem firmware version.



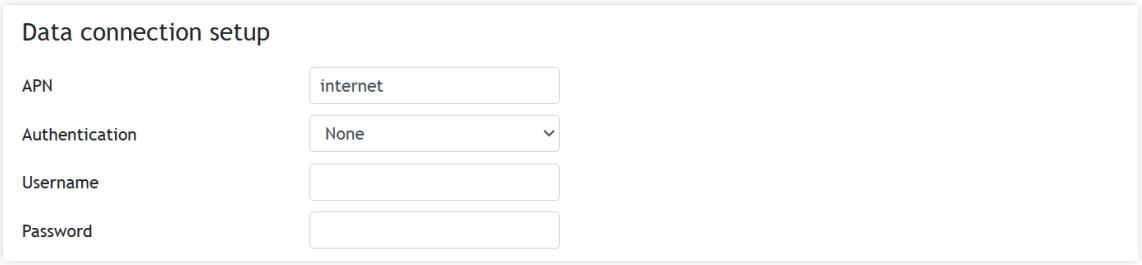
The only configurable setting in this section is the “Enable data in roaming” checkbox. By default, this option is disabled to prevent additional charges for data usage while roaming.

If the SIM card is used outside its home network, “Enable data in roaming” must be enabled to allow access to data services such as email, HTTP POST, MQTT, NTP, and others.

7.2.5.2 Data connection setup

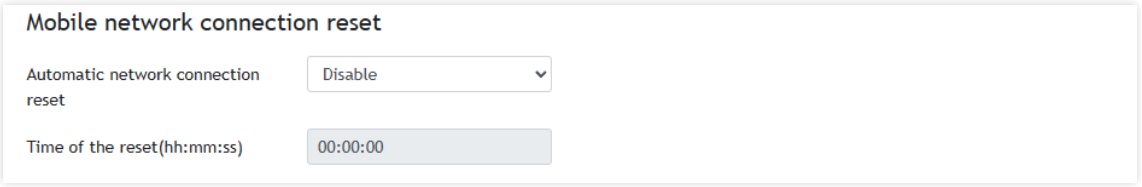
The settings in this section depend on the requirements of the mobile operator. We recommend consulting your service provider for the correct configuration parameters.

By default, the APN (Access Point Name) is set to "internet". Some mobile operators may also require an authentication method, username, and password to establish the data connection.



7.2.5.3 Automatic mobile network reconnection

In areas with weak signal strength or unstable mobile network coverage, the data connection may become unresponsive or disconnected for extended periods.



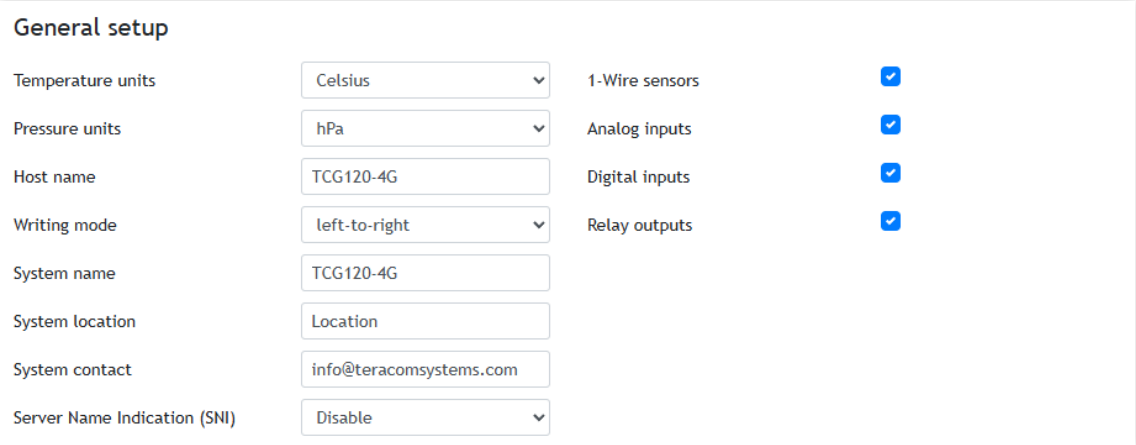
Enabling automatic network connection reset helps restore communication by forcing the mobile modem to reconnect to the network. This improves long-term reliability in installations where network quality is poor or intermittent.

When enabled, the device performs one automatic mobile network connection reset per day at the configured time.

By default, this feature is disabled.

7.2.5.4 General setup

This section is used to configure general system parameters and the visibility of monitored objects.



The Temperature units and Pressure units settings define the measurement units used for supported sensors and displayed values.

The Host name parameter defines the device name used for identification in the local network.

The Writing mode parameter defines the text direction used by the user interface.

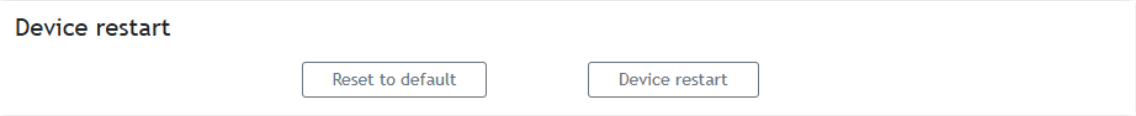
The System name, System location, and System contact parameters provide identification and contact information for the device. These values are included in XML/JSON status data and can be used by external monitoring systems.

The Server Name Indication (SNI) setting controls whether the device sends the server name during the TLS handshake when establishing secure HTTPS or MQTT connections. This option should be enabled only if required by the remote server.

The checkboxes for 1-Wire sensors, Analog inputs, Digital inputs, and Relay outputs control whether the corresponding sections are visible on the Monitoring page.

7.2.5.5 Device restart

This section provides two maintenance functions for the device.

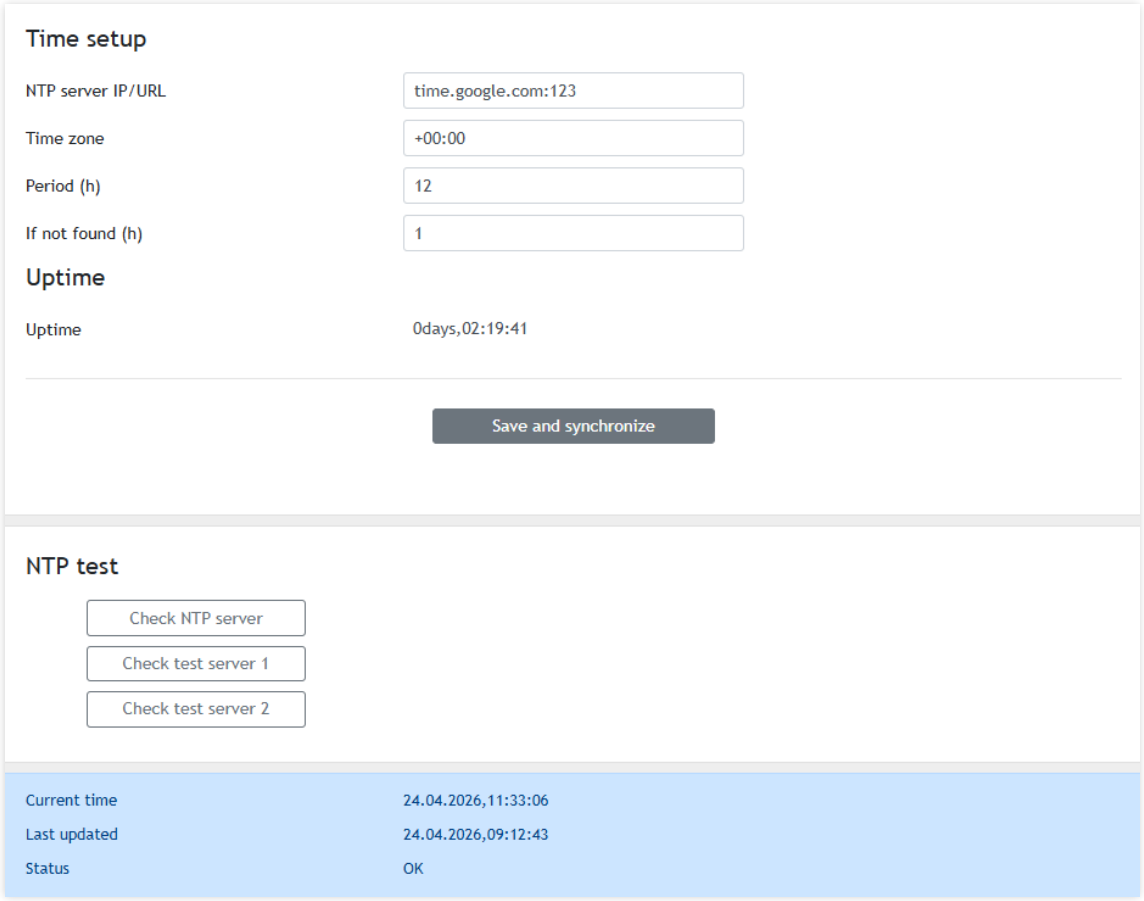


The Device restart button performs a software restart of the module without changing the current configuration. This function can be used after configuration changes or for troubleshooting purposes.

The Reset to default button restores all device settings to their factory default values. This action deletes all user-defined settings, including communication parameters, alarm conditions, and relay configuration. Use this function with caution.

7.2.6 NTP

This section is used to configure and monitor time synchronization using the Network Time Protocol (NTP).



The NTP server IP/URL field defines the address of the NTP server used for time synchronization. The default server is time.google.com:123.

The Time zone parameter defines the local time offset from UTC.

The Period (h) parameter defines the interval, in hours, between periodic synchronization attempts with the NTP server.

The If not found (h) parameter defines the retry interval, in hours, if the NTP server cannot be reached during a synchronization attempt.

The Uptime field shows the time elapsed since the last device restart.

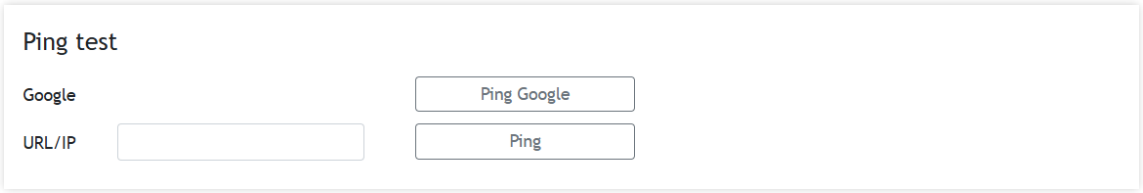
Pressing the Save and synchronize button saves the current NTP settings and immediately starts a synchronization attempt.

The NTP test section provides diagnostic functions for testing connectivity to the configured NTP server or to predefined test servers.

The status panel displays the current synchronized time, the time of the last successful synchronization, and the current synchronization status.

7.2.7 Ping

This page provides a diagnostic tool for testing network connectivity from the device to remote hosts. It can be used to verify internet access, DNS resolution, and reachability of external servers.



The Ping Google button performs a connectivity test to a predefined external server (Google) and can be used as a quick check for internet access.

The URL/IP field allows entering a custom hostname or IP address for testing. Pressing the Ping button sends an ICMP echo request to the specified destination and displays the result.

This function is useful for troubleshooting network configuration problems, verifying APN settings, and checking connectivity to remote services such as HTTP servers, MQTT brokers, or NTP servers.

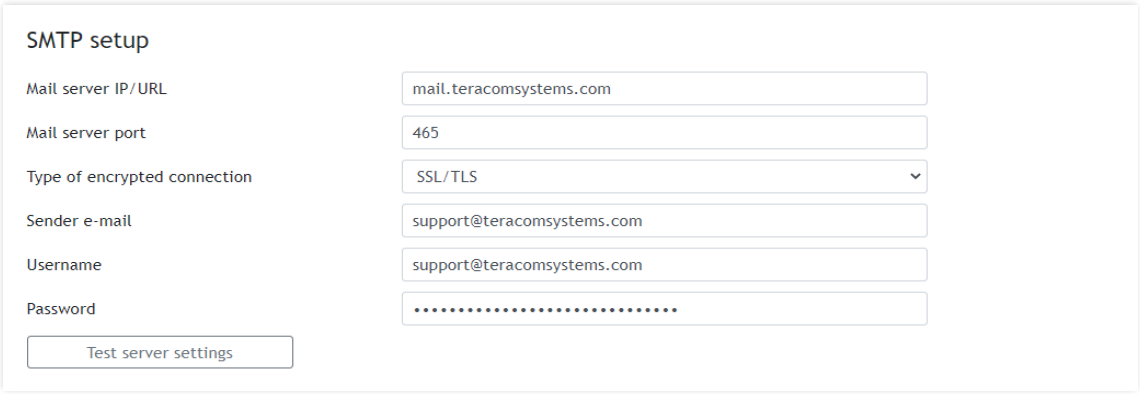
7.3. Services

7.3.1 SMTP

This page is used to configure SMTP server settings, alarm email recipients, and the content of email alarm notifications.

7.3.1.1 SMTP setup

This section is used to configure the SMTP server settings required for sending email alarm notifications.



The Mail server IP/URL field defines the address of the SMTP server. The server can be specified either by IP address or hostname.

The Mail server port field defines the TCP port used for the SMTP connection. The default value is 25, but other ports may be required depending on the mail service provider.

The Type of encrypted connection setting defines the security mode used for the SMTP connection. Depending on the server configuration, the connection can be unencrypted or protected by SSL/TLS.

The Sender email field defines the email address that will appear as the sender of all alarm notifications.

If the SMTP server requires authentication, the Username and Password fields must be configured with valid credentials.

The Test server settings button performs a connection test using the configured SMTP parameters and can be used to verify that the server settings are correct before enabling email notifications.

7.3.1.2 Alarm destination

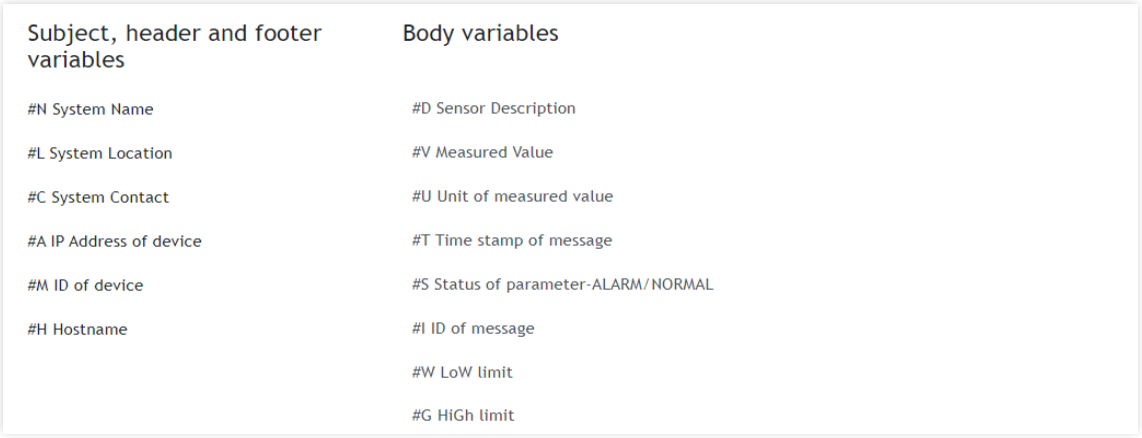
This section is used to configure the recipient email addresses for alarm notifications.

Up to five recipient email addresses can be configured. Each recipient can be enabled or disabled individually using the corresponding checkbox. Only enabled recipients will receive alarm notifications.

The Test email button sends a test message to all enabled recipients using the currently configured SMTP settings. This function can be used to verify both the SMTP configuration and the recipient email addresses before enabling alarm notifications.

7.3.1.3 E-mail details

This section is used to configure the content and format of email alarm notifications.



The email message consists of four parts: Subject, Body header, Body, and Body footer.

Each field can contain both fixed text and dynamic variables (keys). During message generation, these variables are automatically replaced with the current device, parameter, or alarm information. This allows creating customized alarm messages with relevant real-time data.

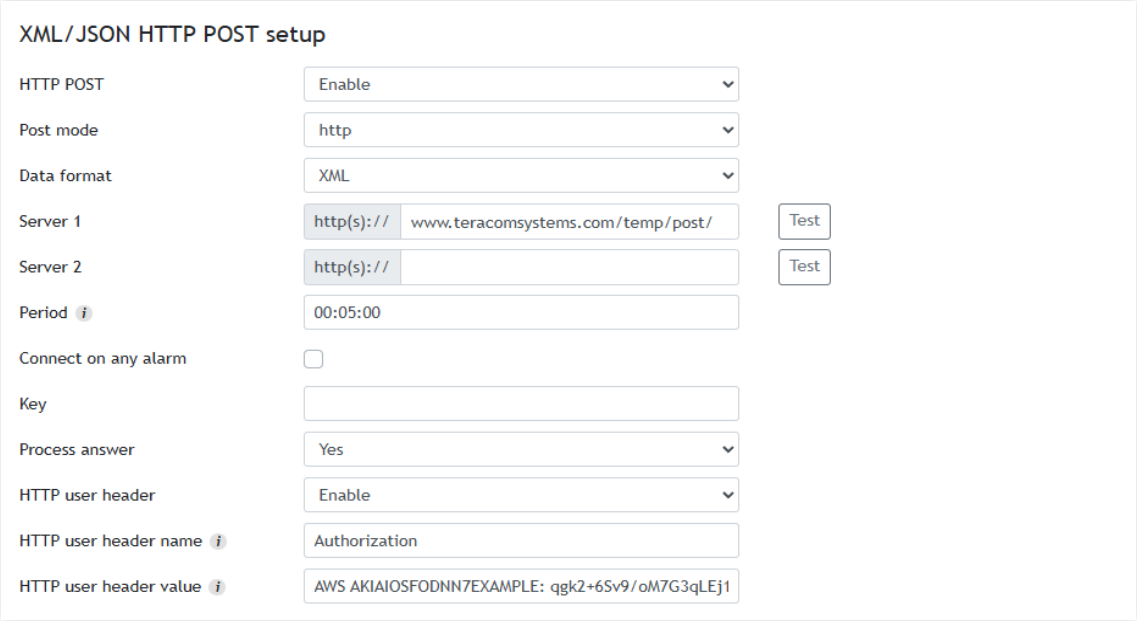
The available variables are grouped according to where they can be used:

- Subject, header, and footer variables – used for general device information such as system name, location, contact, IP address, device ID, and hostname.
- Body variables – used for alarm-specific information such as parameter description, measured value, unit, timestamp, status, message ID, and configured limits.

Using these variables allows the generated email notifications to include all important information required for identification of the alarm source and analysis of the alarm condition.

7.3.2 HTTP POST

This page is used to configure periodic or alarm-triggered HTTP POST transmission of device status data in XML or JSON format to remote servers



The transmitted data contains the current status of all monitored parameters together with system information. The preferred data format can be selected from the Data format drop-down menu.

The remote servers can be addressed using either a domain name or an IP address. Two server addresses can be configured for redundancy or parallel integration with multiple systems. The Test button can be used to verify the connectivity to each configured server.

The upload interval, defined by the Period parameter, can be configured from 1 minute to 48 hours. This parameter defines how often the remote server receives status updates from the TCG120-4. Shorter intervals provide more up-to-date information but increase mobile data traffic. The Period parameter can also be changed remotely by command.

If the Connect on any alarm option is enabled, an HTTP POST request is sent immediately when any alarm condition occurs, regardless of the configured period.

The Key field is user-defined. Its value is included in the XML/JSON data and can be used for additional device identification by the remote server.

If the Process Answer option is enabled, the TCG120-4 processes and executes commands received in the server response to an HTTP/HTTPS POST request. This allows remote control and configuration directly from the server side.

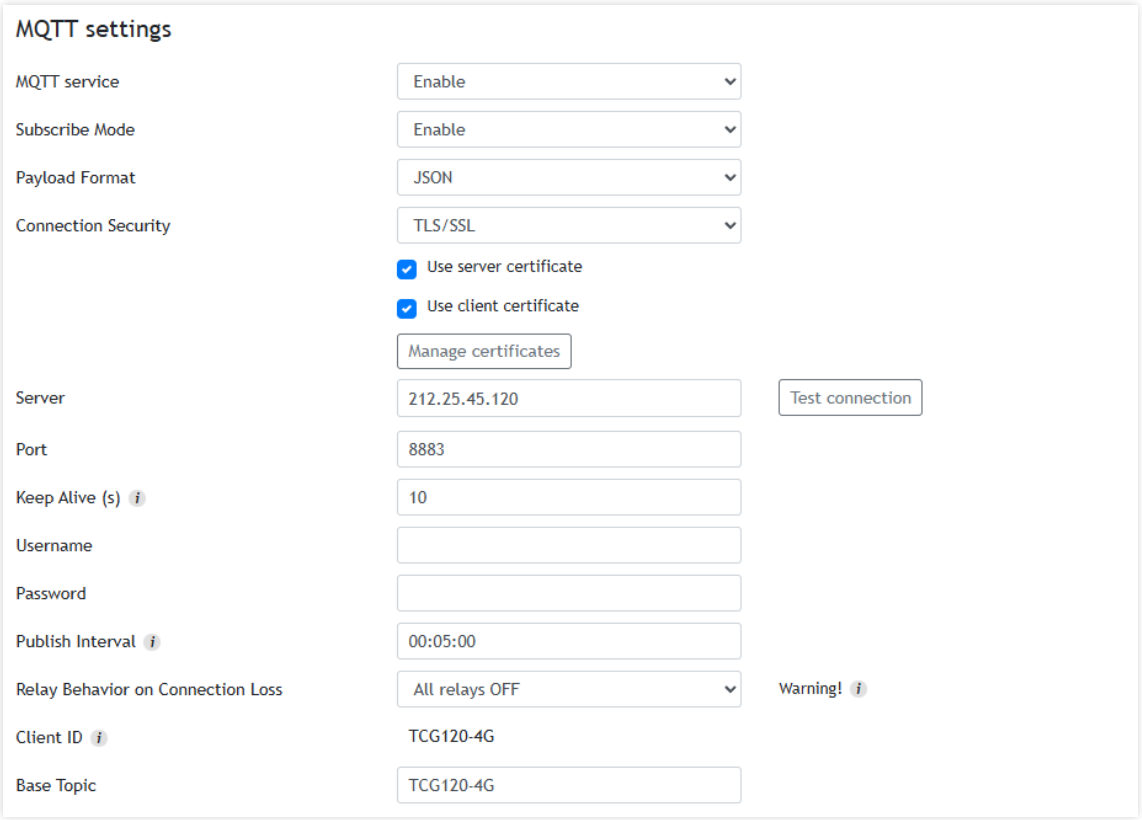
The device also supports a custom static HTTP header. A user-defined header name and value can be configured and included in every HTTP POST request. This can be used for server identification, API keys, access tokens, or other integration requirements. By default, this option is disabled.

7.3.3 MQTT

This page is used to configure MQTT communication for publishing device data and receiving relay control commands. The TCG120-4 can publish monitored parameter values and states to an MQTT broker and can subscribe to topics for remote relay control.

7.3.3.1 MQTT connection settings

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



The TCG120-4 supports both MQTT publishing (sending device data to the broker) and MQTT subscribing (receiving relay control commands from the broker). By default, both services are disabled.

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

The Port field defines the MQTT broker communication port. The default port is 1883 for unencrypted communication.

The Payload Format parameter defines the format of the published MQTT messages. The supported formats are JSON and plain text.

The Connection Security parameter defines whether the connection to the broker is unencrypted or protected by TLS. For secure MQTT connections, server certificate validation and client certificate authentication can be enabled separately. The required certificates are managed through the Manage certificates page.

The Keep Alive (s) parameter defines the interval for maintaining the MQTT connection and detecting connection loss.

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

The Relay Behavior on Connection Loss parameter defines the relay safety behavior if the connection to the MQTT broker is lost. Three options are available: All relays OFF, All relays ON, and Keep last state.

The Base Topic field defines the root MQTT topic used for all published and subscribed messages. All parameter-specific topics are created under this root topic.

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

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

7.3.3.2 MQTT topics and data publishing

This section is used to configure the MQTT topics and publishing behavior for 1-Wire sensors, analog inputs, digital inputs, and relay outputs.

Each group has its own topic name (topic prefix), which is appended to the configured Base Topic. Each individual parameter or relay also has its own Topic field, defining its unique topic identifier within the group.

The complete MQTT topic is formed as follows:

Base Topic / Group Topic / Parameter Topic / Type

where:

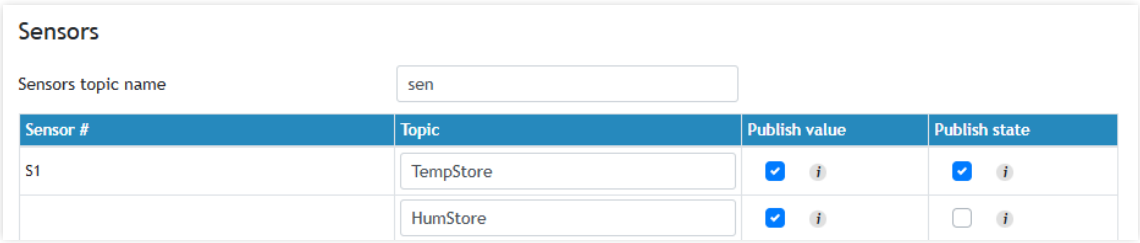
- Base Topic – the root MQTT topic configured in the general MQTT settings;
- Group Topic – the topic prefix for sensors, analog inputs, digital inputs, or relays;
- Parameter Topic – the unique topic assigned to the specific parameter;
- Type – defines whether the published data contains a value or a state.

The published payload can be sent either in plain text or JSON format, depending on the selected Payload Format setting. In JSON mode, each MQTT message contains only a single item (value or state).

Example – 1-Wire sensor publishing

For a sensor with description TempStore, using:

- Base Topic = TRC
- Group Topic = sen
- Parameter Topic = TempStore



Value topic: TRC/sen/TempStore/value

Payload (plain text): 28.5

Payload (JSON): {"value":"28.5"}

Alarm state topic: TRC/sen/TempStore/state

Payload (plain text): 1

Payload (JSON): {"state":"1"}

where:

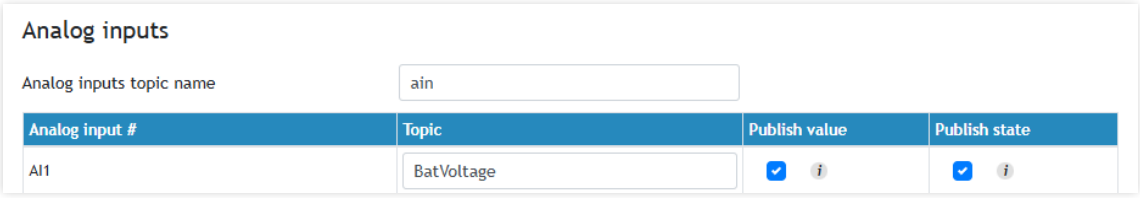
0 = normal state

1 = alarm state

Example – Analog inputs publishing

For an analog input with description BatVoltage, using:

- Base Topic = TRC
- Group Topic = ain
- Parameter Topic = BatVoltage



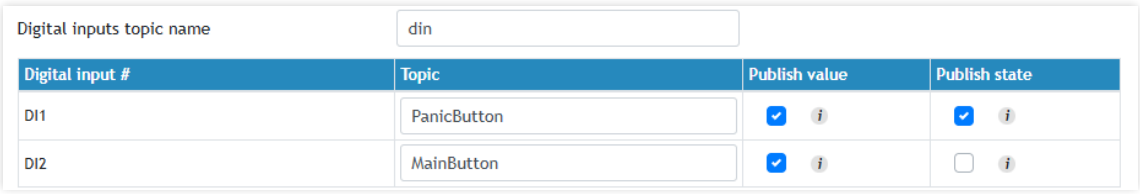
Value topic: TRC/ain/BatVoltage/value
Payload (plain text): 12.5
Payload (JSON): {"value":"12.5"}

Alarm state topic: TRC/ain/BatVoltage/state
Payload (plain text): 1
Payload (JSON): {"state":"1"}
where:
0 = normal state
1 = alarm state

Example – Digital inputs publishing

For a digital input with description PanicButton, using:

- Base Topic = TRC
- Group Topic = din
- Parameter Topic = PanicButton



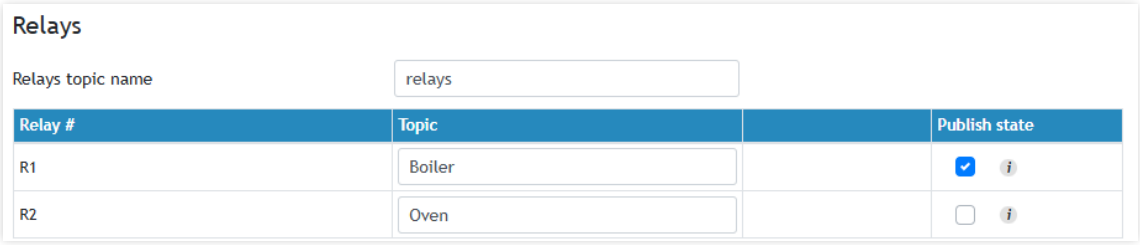
Value topic (current input state): TRC/din/PanicButton/value
Payload (plain text): 1
Payload (JSON): {"value":"1"}
where:
0 = closed state
1 = open state

Alarm state topic: TRC/din/PanicButton/state
Payload (plain text): 1
Payload (JSON): {"state":"1"}
where:
0 = normal state
1 = alarm state

Example – Relays publishing

For a relay with description Boiler, using:

- Base Topic = TRC
- Group Topic = relays
- Parameter Topic = Boiler



State topic: TRC/relays/Boiler/state

Payload (plain text): 1

Payload (JSON): {"state":"1"}

where:

0 = relay OFF

1 = relay ON

7.3.3.3 MQTT subscribe mode

This section describes the MQTT subscribe functionality used for remote relay control and device command execution. When the MQTT subscribe service is enabled, the TCG120-4 subscribes to predefined MQTT topics and executes the received commands.

Two types of subscription topics are supported:

- Relay control topics – used for direct control of the built-in relays;
- API command topic – used for remote device configuration and control through text commands.

Relay control using MQTT subscribe

The built-in relays of the TCG120-4 can be controlled through dedicated MQTT topics. Each relay uses its own topic, formed as follows:

Base Topic / relays / Relay Topic / command

where:

- Base Topic – the root MQTT topic configured in the general MQTT settings;
- Relay Topic – the topic assigned to the specific relay.

Example:

For Relay 1 with topic Boiler and Base Topic TRC,

the subscribe topic will be: TRC/relays/Boiler/command

Publishing the following values to this topic controls the relay:

- 0 – turns the relay OFF
- 1 – turns the relay ON

Device configuration and control using MQTT subscribe

The TCG120-4 also supports remote configuration and control through a common API topic. The API subscribe topic is formed as:

Base Topic / api

Example:

TRC/api

The commands accepted through the MQTT API topic are the same as those described in the HTTP API section. Publishing a text command to this topic causes the controller to read and execute it in the same way as an HTTP API command.

Examples:

- r1=pl – activates Relay 1 in pulse mode
- r2=on – turns Relay 2 ON
- pper=15 – changes the HTTP POST period to 15 seconds

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

7.3.4 GSM geolocation

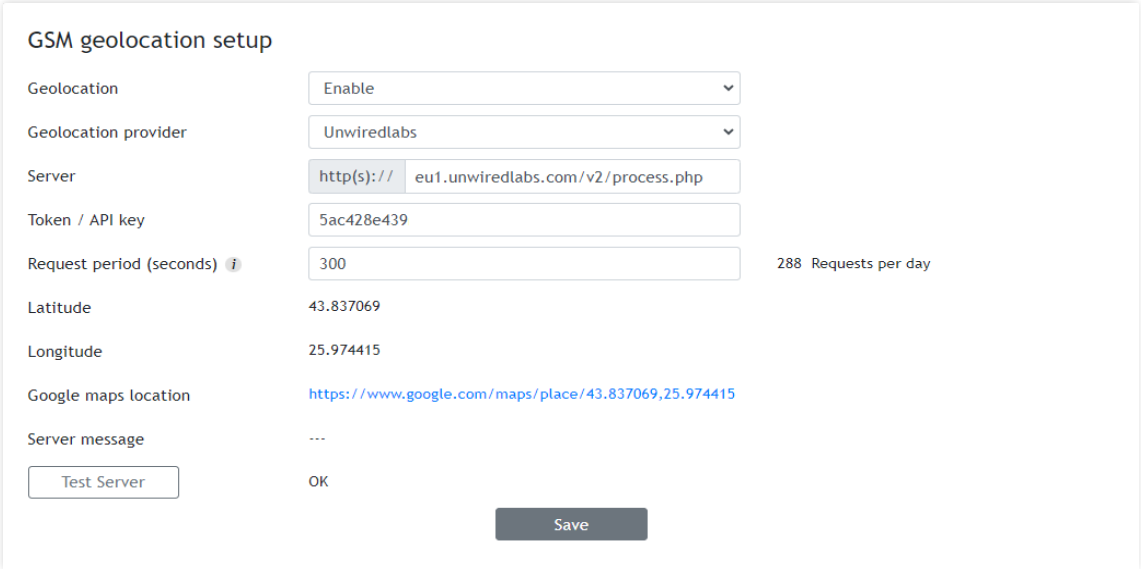
GSM geolocation provides a way to determine the approximate position of the device without relying on GPS. Instead, it uses information from the mobile network, such as the serving cell and neighboring cell towers. This data is sent to a third-party geolocation service provider, which returns the estimated device coordinates with a certain level of accuracy.

Although GSM geolocation is less precise than GPS, it has the advantage of working indoors and in locations where GPS reception is limited or unavailable. This makes it suitable for applications such as tracking parcels, refrigerated display cases, refrigerators, machinery, and equipment operating in enclosed or covered environments.

The TCG120-4 supports the following GSM geolocation service providers:

- Google
- Unwired Labs

To use this service, registration with the selected provider is required, and the corresponding API key must be configured.

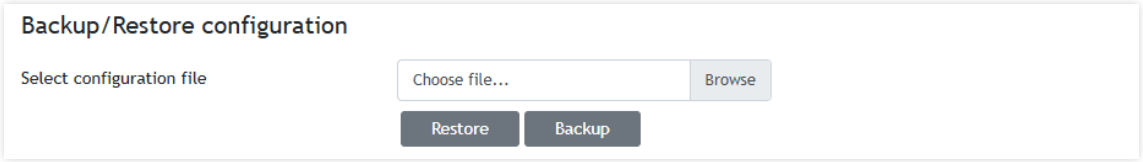


7.4. Administration

This page provides administrative functions for device maintenance, including configuration backup and restore, as well as firmware updates

7.4.1 Backup/Restore configuration

The TCG120-4 allows backup and restoration of all user-defined settings. The configuration is stored in an XML file, which can be downloaded for backup purposes or uploaded later to restore the device configuration.

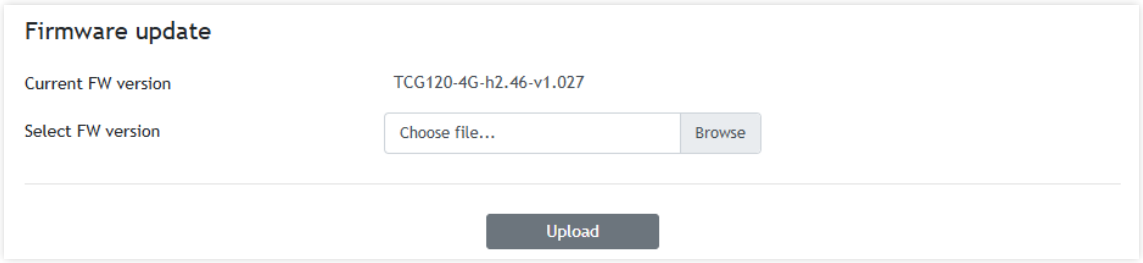


The backup file can also be used to copy the same configuration to multiple TCG120-4 devices, making it especially useful for batch deployment or replacing a device with minimal setup time.

It is also possible to restore only selected settings by manually editing the XML backup file and removing the parameters that should not be restored. This option is intended for advanced users familiar with XML structure and editing. In this case, only the parameters present in the XML file will be applied to the device.

7.4.2 FW update

The TCG120-4 supports firmware updates through the web interface. The current firmware version is displayed on this page.



To update the firmware:

- Download the latest firmware file from www.teracomsystems.com.
- Select the firmware file using the Browse button.
- Press the Upload button to start the update process.

Important:

Do not turn off the power supply or interrupt the update process while the firmware is being uploaded and installed. Interrupting the power supply during the update may cause permanent damage to the device.

8. SMS configuration and control

The TCG120-4 supports configuration, control, and status reporting via SMS commands. SMS commands can be used to change settings, control relay outputs, request parameter status, configure communication services, and perform firmware updates over the air.

For security reasons, SMS commands are accepted only from phone numbers configured in the SMS Setup section. Depending on the command, different access levels may be required: Master, Superuser, or User.

SMS commands must be entered exactly as shown in the syntax descriptions. In the command examples below, the symbol ␣ represents a single space character and must be replaced with an actual space when sending the SMS.

The following sections describe all supported SMS commands, their syntax, required access rights, and response examples

8.1. User management commands

These commands are used to configure and manage the authorized users for SMS, email, and call notifications, as well as their access rights for SMS control and configuration.

- **Set new master number**
  Rights: Master
  Syntax: set␣master␣<number>
  Where: <number> is a mobile number in the international format
  Example
  Command: *set master +359885885885*
  Answer: *You are the new master!*
- **Set SMS users numbers** - used to add or delete SMS users
  Rights: Master
  Syntax: set␣sms␣user␣<user>:<number>
  Where: <user> can be u1, u2, u3 or u4
  <number> is a mobile number in the international format
  Example
  Command: *set sms user u1:+359885887766*
  Answer: *u1:+359885887766,u2,u3,u4*
  Command: *set sms user u2:+359885999888*
  Answer: *u1:+359885887766, u2:+359885999888,u3,u4*
  Command: *set sms user u1:*
  Answer: *u1, u2:+359885999888,u3,u4*
- **Set superuser rights for users** - used to activate or deactivate superuser rights for configured SMS users
  Rights: Master
  Syntax: set␣suser␣<user>=<cmd>
  Where: <user> can be u1, u2, u3 or u4
  <cmd> is “on” for activation or “off” for deactivation
  Example
  Command: *set suser u1=on*
  Answer: *m:+ 359885885885,su1:+359885887767,u2,u3,u4*
  Command: *set suser u1=off*
  Answer: *m:+ 359885885885,u1:+359885887767,u2,u3,u4*
  Command: *set suser u1=off,u2=on,u3=off,u4=off*
  Answer: *m:+ 359885885885,u1:+359885887761,su2:+359885887762, u3:+359885887763,u4:+359885887764*
- **Display SMS users**
  Rights: Master, Superusers, Users
  Syntax: display␣sms␣users
  Example
  Command: *display sms users*
  Answer: *m:+359885885885,u1:+359885887766,u2:+359885999888,u3,u4*

- **Set email users** - used to add or delete email users
  Rights: Master, Superusers
  Syntax: set␣email␣user␣<user>:<email>
  Where: <user> can be e1, e2, e3, e4 or e5
  <email> - a valid email address
  Example
  Command: *set email user e1:mail1@teracomsystems.com*
  Answer: *e1:mail1@teracomsystems.com*
  Command: *set email user e2:mail2@teracomsystems.com*
  Answer: *e2:mail2@teracomsystems.com*
  Command: *set email user e2:*
  Answer: *e2:*
- **Display email users** - used to display the email address of a selected user
  Rights: Master, Superusers, Users
  Syntax: display␣email␣<user>
  Where: <user> can be e1, e2, e3, e4 or e5
  Example
  Command: *display email e1*
  Answer: *e1:mail1@teracomsystems.com*
  Command: *display email e2*
  Answer: *e2:mail2@teracomsystems.com*

8.2. Status commands

These commands are used to request status information about the device, monitored parameters, and communication settings without changing the current configuration.

- **Status of system** - used to request the current system status of the device
  Rights: Master, Superusers, Users
  Syntax: status␣system
  Example
  Command: *status system*
  Answer: *06.06.2026,16:09:06,4G=y,ss=80%,fw=1.026*
- **Status of parameter** - used to request the current status or value of a digital input (di), analog input (ai), relay (r), or sensor parameter (s)
  Rights: Master, Superusers, Users
  Syntax: status␣<param>
  Where: <param> is one of: di1, di2, ai1, ai2, r1, r2, s11, s12, s21, s22, s31, s32, s41, s42
  Example
  Command: *status di1*
  Answer: *di1(Garage_door)=CLOSED*
  Command: *status s22*
  Answer: *s22(Office)=34.5%RH*

8.3. Relay control commands

These commands are used to control the built-in relay outputs by switching them on, off, or activating them in pulse mode.

- **Set a relay** - used to switch the selected relay output ON or OFF

  Rights: Master, Superusers, Users
  Syntax: set␣<relay>=<state>␣[option]
  Where: <relay> is r1 or r2
  <state> is on, off
  [option] -w, execute the command without sending an SMS response
  Example
  Command: *set r1=on*
  Answer: *r1=on,r2=off*
  Command: *set r1=off*
  Answer: *r1=off,r2=off*
  Command: *set r1=off,r2=on*
  Answer: *r1=off,r2=on*
  Command: *set r1=on -w*
  Answer: *No response*
  Command: *set r1=off,r2=off -w*
  Answer: *No response*

- **Pulse a relay** - used to activate the selected relay for its configured pulse duration

  Rights: Master, Superusers, Users
  Syntax: set␣<pulse relay>=<state>␣[option]
  Where: <pulse relay> is pl1 or pl2
  <state> is on
  [option] -w, execute the command without sending an SMS response
  Example
  Command: *set pl1=on*
  Answer: *r1=on,r2=off*
  Command: *set pl1=on,pl2=on*
  Answer: *r1=on,r2=on*
  Command: *set pl1=on -w*
  Answer: *No response*
  Command: *set pl1=on,pl2=on -w*
  Answer: *No response*

8.4. HTTP POST commands

These commands are used to configure and control the HTTP POST service, including server URLs, transmission period, and enabling or disabling data transmission

- **Set POST URL** - used to configure the URL for XML/JSON HTTP POST transmission.

  Rights: Master, Superusers
  Syntax: set␣purl=<link> or set␣purl2=<link>
  Where: *<link> is the address of remote server (domain or IP)*
  Example
  Command: *set purl=www.teracomsystems.com:8801/POSTtest.php*
  Answer: *purl=www.teracomsystems.com:8801/POSTtest.php,POST=on, period=00:01:00*
  Command: *set purl2=www.teracomsystems.com:8802/POSTtest.php*
  Answer: *purl2=www.teracomsystems.com:8802/POSTtest.php,POST=on, period=00:01:00*

- **Status POST URL** - used to check the configured HTTP POST destination URL
  Rights: Master, Superusers, Users
  Syntax: status_purl or status_purl2
  Example
  Command: *status purl*
  Answer: *purl=www.teracomsystems.com:8801/POSTtest.php, POST=on,period=00:01:00*
  Command: *status purl2*
  Answer: *purl2=www.teracomsystems.com:8802/POSTtest.php, POST=on,period=00:05:00*
- **Set POST period** (seconds) – used to set the HTTP POST transmission interval in seconds
  Rights: Master, Superusers
  Syntax: set_pper=<value>
  Where: <value> is a number between 60 and 172800 (seconds)
  Example
  Command: *set pper=120*
  Answer: *POST=off,period=00:02:00*
- **Set POST period** (hh:mm:ss) – used to set the POST transmission interval in hh:mm:ss format
  Rights: Master, Superusers
  Syntax: set_pperh=<value>
  Where: <value> is a time between 00:01:00 and 48:00:00
  Example
  Command: *set pperh=00:05:00*
  Answer: *POST=off,period=00:05:00*
- **Enable POST** – used to enable XML/JSON HTTP POST transmission
  Rights: Master, Superusers
  Syntax: set_POST=on
  Example
  Command: *set POST=on*
  Answer: *POST=on,period=00:02:00*
- **Disable POST** – used to disable XML/JSON HTTP POST transmission
  Rights: Master, Superusers
  Syntax: set_POST=off
  Example
  Command: *set POST=off*
  Answer: *POST=off,period=00:02:00*

8.5. MQTT configuration commands

These commands are used to configure and control the MQTT service, including enabling or disabling the service, setting the publish interval, selecting the data format, and configuring the MQTT server connection parameters.

- **Enable MQTT** – used to enable the MQTT service
  Rights: Master, Superusers
  Syntax: set_mqtt=on
  Example
  Command: *set mqtt=on*

Answer: *mqtt=on,mdata=json,period=00:05:00*

- **Disable MQTT** – used to disable the MQTT service
  Rights: Master, Superusers
  Syntax: set_mqtt=off
  Example
  Command: *set mqtt=off*
  Answer: *mqtt=off,mdata=json,period=00:05:00*
- **Set MQTT publish period** – used to set the MQTT data publishing interval
  Rights: Master, Superusers
  Syntax: set_mper=<time>
  Where: <time> is between 00:01:00 and 48:00:00 (hh:mm:ss)
  Example
  Command: *set mper=00:05:00*
  Answer: *mqtt=on,mdata=json,period=00:05:00*
- **Set MQTT data format** – used to select the MQTT payload format
  Rights: Master, Superusers
  Syntax: set_mdata=<format>
  Where: <format> is "json" or "plain"
  Example
  Command: *set mdata=plain*
  Answer: *mqtt=on,mdata=plain,period=00:05:00*
- **Set MQTT server** – used to configure the MQTT broker address
  Rights: Master, Superusers
  Syntax: set_murl=<link>
  Where: <link> is the address of remote server (domain or IP)
  Example
  Command: *set murl=www.teracomsystems.com*
  Answer: *mqtt=on,murl=www.teracomsystems.com,port=1883*
- **Set MQTT port** – used to configure the MQTT broker port
  Rights: Master, Superusers
  Syntax: set_mport=<port>
  Where: <port> is a MQTT port
  Example
  Command: *set mport=8883*
  Answer: *mqtt=on,murl=www.teracomsystems.com,port=8883*

8.6. Time, system and test commands

These commands are used to configure system time settings, control device operation, manage network-related functions, and verify communication services.

- **Set time server** – used to configure the NTP server address and port
  Rights: Master, Superusers
  Syntax: set_ts=<url>:<port>
  Where: <url> is the NTP server address (domain or IP)
  <port> is the NTP port (typically 123)
  Example
  Command: *set ts=time.google.com:123*

Answer: *ts=time.google.com:123,tz=+02:00*

- **Set time zone** – used to configure the device time zone
  Rights: Master, Superusers
  Syntax: set_tz=±hh:mm
  Example
  Command: *set tz=+03:00*
  Answer: *ts=time.google.com:123,tz=+03:00*
- **Restart device** – used to restart the device
  Rights: Master, Superusers
  Syntax: restart
  Example
  Command: *restart*
  Answer: *Device is restarting!*
- **Set mobile network connection reset** – used to enable or disable automatic daily network reconnection
  Rights: Master, Superusers
  Syntax: set_mncr=<status>,hh:mm:ss
  Where: <status> is "on" or "off"
  Example
  Command: *set mncr=on,10:00:00*
  Answer: *mncr=on,10:00:00*
  Command: *set mncr=off,11:00:00*
  Answer: *mncr=off,11:00:00*
- **Send test email** – used to send a test email to configured email users
  Rights: Master, Superusers
  Syntax: test_email
  Example
  Command: *test email*
  Answer: *Emails are sending!*
- **Send test SMS** – used to send a test SMS to authorized users
  Rights: Master, Superusers
  Syntax: test_sms
  Example
  Command: *test sms*
  Answer: *This is a test SMS!*

8.7. Firmware and advanced configuration commands

These commands are used for firmware updates, advanced device configuration, and setting alarm-related parameters.

- **Update firmware** – used to update the device firmware over the air
  Rights: Master, Superusers
  Syntax: update_<URL>
  Where: <URL> is a valid public URL pointing to a firmware (.cod) file
  Example
  Command: *update www.teracomsystems.com/docs/TCG120-4-v1.000-P-S.cod*
  Answer 1: *Downloading firmware...*

Answer 2: *Firmware file downloaded. Updating...*

Possible responses:

File corrupt or wrong version!
Can't connect to server!
Download time out!
4G/3G/2G is not connected!
Connection lost!
Response timeout!
Socket error!

- **Set sensor notification delay** – used to configure the notification delay for sensor alarms

  Rights: Master, Superusers
  Syntax: set_delsen=<value>
  Where: <value> is the notification delay in seconds (0–3600)
  Example
  Command: *set delsen=5*
  Answer: *delsen=5*

- **Set sensor limits** – used to configure Min, Max, and Hysteresis values for sensor parameters

  Rights: Master, Superusers
  Syntax: set_lspt=n<min>,x<max>,y<hys>
  Where: p is the sensor number (1–4)
  t is the parameter number (1 or 2)
  n defines the Min value
  x defines the Max value
  y defines the Hysteresis value
  Example – sensor 1, parameter 1 (temperature)
  Command: *set ls11=n25.0,x35.0,y1.0*
  Answer: *ls11=n25.0,x35.0,y1.0*
  Command: *set ls11=n31.0*
  Answer: *ls11=n31.0,x35.0,y1.0*
  Example – sensor 1, parameter 2 (humidity)
  Command: *set ls12=n45.0,x60.0,y1.0*
  Answer: *ls12=n45.0,x60.0,y1.0*

- **Execute API command** – used to execute HTTP API commands via SMS

  Rights: Master, Superusers
  Syntax: set_api_<cmd>=<value>
  Where: <cmd> is the API command
  <value> is the API value
  Example – set APN
  Command: *set api APN=internet*
  Answer: *OK*
  Example – enable email notification for digital input 1
  Command: *set api d1m=1*
  Answer: *OK*

Example – disable email notification for digital input 1

Command: *set api d1m=0*

Answer: *OK*

Possible responses:

OK

Wrong or missing parameters!

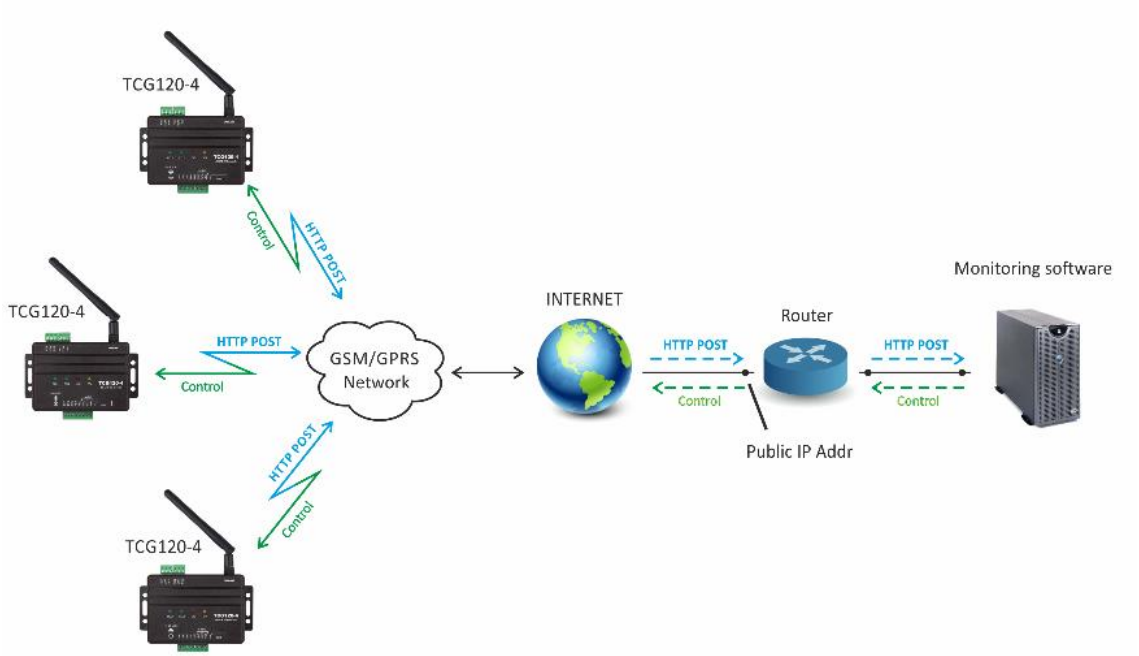
Unknown command!

9. Protocols and API

9.1. HTTP/HTTPS API

HTTP/HTTPS is a widely used protocol for remote monitoring and control in SCADA systems based on client–server architecture. The TCG120-4 operates as an HTTP/HTTPS client, enabling seamless integration with SCADA software using the HTTP/HTTPS protocol.

Below is an example of a typical monitoring application:



9.1.1 Periodic HTTP/HTTPS POST

While this service is active, the module periodically sends HTTP/HTTPS POST requests to a server. These POST requests contain XML or JSON data describing the current status of the monitored parameters. The HTTP/HTTPS POST can also be sent in response to triggered alarm conditions, similar to an SNMP trap.

After each HTTP/HTTPS POST request, the server responds according to the HTTP/HTTPS protocol. The response may include API command(s).

An important parameter in this communication is the HTTP/HTTPS POST interval. A shorter interval increases data traffic but provides more up-to-date information and control of the device. The server can modify the HTTP/HTTPS POST interval using the appropriate command, allowing flexible control of the communication process.

9.1.2 API commands

The commands are transmitted in the body of the HTTP/HTTPS response to a POST request. Most commands follow the structure:

set yyy=xxx
Where:

- yyy represents the command;
- xxx represents the parameter value.

Example:
set r1=on
This command activates relay 1.

Multiple commands can be sent simultaneously by concatenating them with the "&" character.

Example:
set r1=on&r2=on
This command activates both relay 1 and relay 2.

Two exceptions to this structure are the following commands:

- set FIN – used for session termination;
- get <filename> – used to retrieve a specific file from the controller.

These commands cannot be concatenated and must be sent on separate lines.

Example:
set r1=on&r2=on
set FIN

These commands activate relay 1 and relay 2, and then terminate the session.

Note: Commands are executed only if the "Process Answer" option is enabled. Otherwise, the device closes the connection immediately after receiving the response without processing any commands.
Example list of commands:

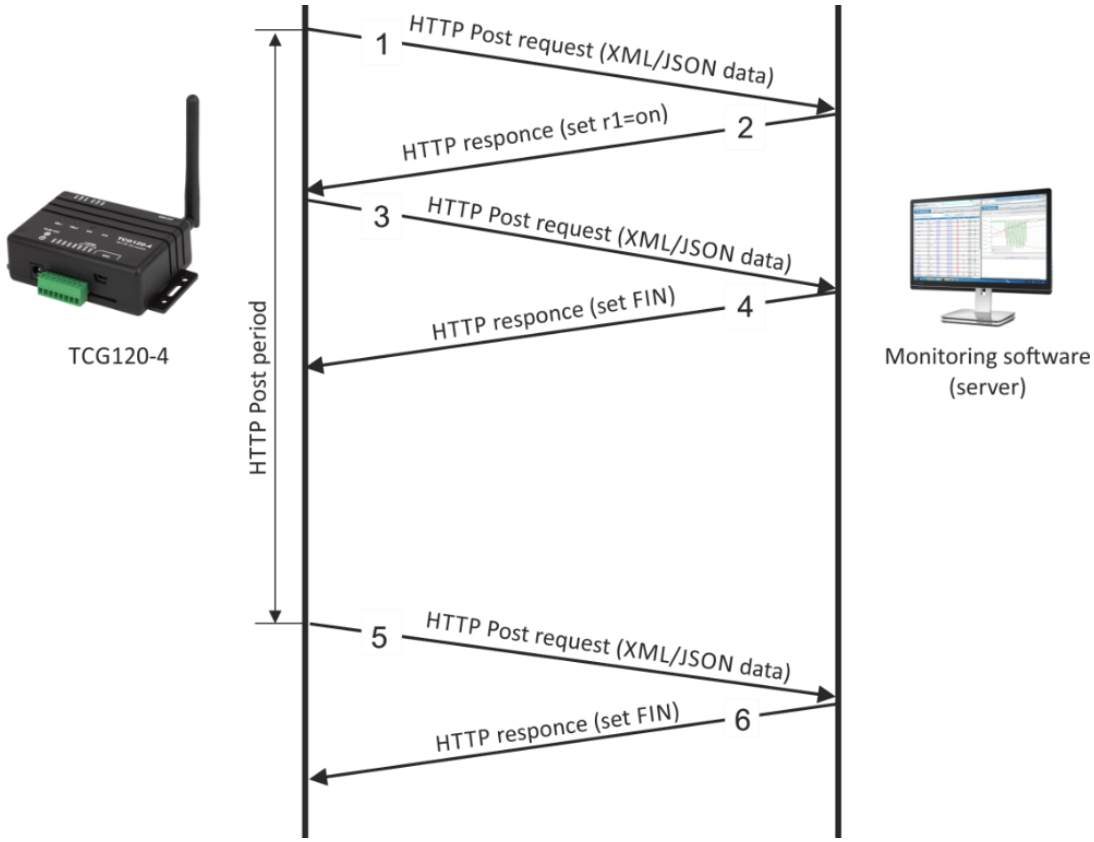
Command	**Description**
r**n**=xxx	Turn relay **n** ON (xxx=on) or OFF (xxx=off) (**n** is 1 or 2 for the respective relay) r2=on – will turn relay 2 ON r1=off – will turn relay 1 OFF r1=on&r2=on – will turn all relays ON r1=off&r2=off – will turn all relays OFF
r**n**=tg	Toggle relay **n** (**n** is 1 or 2 for the respective relay) r2= tg – will toggle relay 2 r1=tg&r2=tg – will toggle relay 1 and 2
r**n**=pl	Pulse relay **n** (**n** is 1 or 2 for the respective relay) r1=pl – will pulse relay 1 r1=pl&r2=pl will pulse relay 1 and 2
pper=x	XML/JSON HTTP POST period in seconds (x is between 60 and 172800)

	pper=600 – will set POST period to 600 seconds
ddly=xx	Set the delay time for sending the next XML/JSON HTTP POST in "real-time" mode after receiving a response from the server. xx is a number between 3 and 59 seconds
save	Save all previous changes (except relays' one) in the FLASH memory. **As every save reflects the FLASH cycles (endurance), this command should be used very carefully.** pper=120&save – will set POST period to 120 seconds and save it
get <filename>	Returns <filename> in the next HTTP/HTTPS POST. The <filename> could be dummy.xml, dummy.json, status.xml or status.json.
FIN	Terminate the session

The complete list of supported commands is provided in Appendix C. The structures of status.xml and status.json are described in Appendix A and Appendix B, respectively.

9.1.3 Communication session for a periodic HTTP/HTTPS POST

The following example illustrates a typical communication session between the TCG120-4 and a remote server when the "Process Answer" option is enabled:



- **Step 1:** The device sends an HTTP/HTTPS POST request with XML/JSON data to the remote server. The request is triggered either by the configured POST period or by an alarm condition.
- **Step 2:** The server responds with an HTTP/HTTPS message containing the command **"set r1=on"** in the body, without including the **"set FIN"** command.
- **Step 3:** After executing the received command, the device immediately sends a new HTTP/HTTPS POST request with updated XML/JSON data, reflecting the new relay state.

- **Step 4:** The server responds with another HTTP/HTTPS message containing the **“set FIN”** command, indicating that no further commands are pending and the session can be closed.
- **Step 5:** At the next POST interval, the device initiates a new HTTP/HTTPS POST request to the server.
- **Step 6:** The server responds with **“set FIN”**, indicating that there are no pending commands and the session is closed.

9.1.4 Testing HTTP/HTTPS POST functionality

To test the HTTP/HTTPS POST functionality, follow the steps below:

- Save the following code as POST.php:

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

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

- Upload the POST.php file to a web server that supports PHP and is accessible from the Internet. Popular choices for Windows include UniServer and XAMPP.
- To verify that the script is working, open the URL (e.g., www.yourserverURL.com/POST.php) in a web browser. A page displaying "The PHP script is working!" should appear.
- Enable the XML/JSON HTTP/HTTPS POST service and enter www.yourserverURL.com/POST.php in the Server1 field.
- Click the "Test HTTP POST" button.

If the HTTP/HTTPS POST request is successfully received and processed, the button will display "OK". Additionally, an XML/JSON file will be created in the same directory as POST.php, with a timestamped filename, for example: 20260506103318_status.xml

9.1.5 “Real-time” mode

The module can operate in a so-called “real-time” mode. In this mode, communication with the server becomes more dynamic, allowing near real-time data processing and command execution.

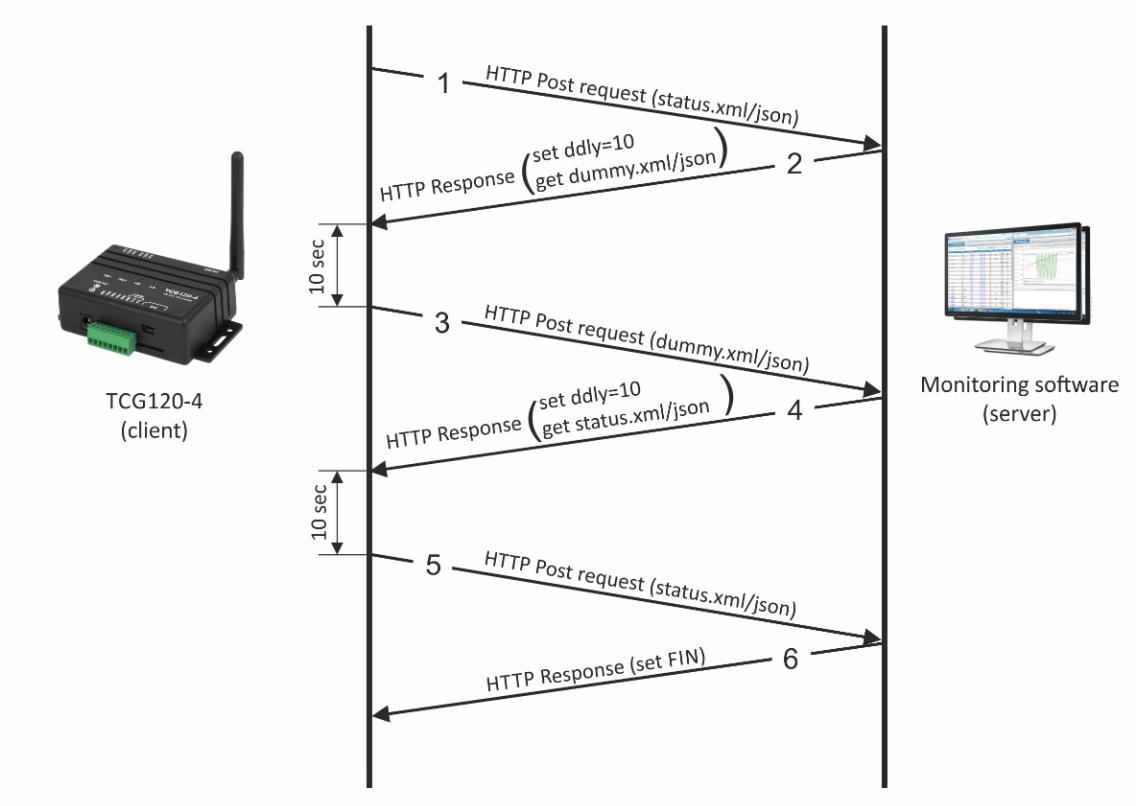
To enable “real-time” mode, the server must respond to a periodic HTTP/HTTPS POST request with the following two commands instead of “set FIN”:

set ddly=xx

get <filename>

Where:

- xx is a number between 3 and 59 seconds;
- <filename> is one of: dummy.xml, dummy.json, status.xml, or status.json.



The files dummy.xml and dummy.json are empty and are used to minimize data traffic in “real-time” mode.

Example:

If the module receives:

set ddly=10

get dummy.xml

it will send an HTTP/HTTPS POST request with dummy.xml after a delay of 10 seconds.

For this and each subsequent POST request, the server may respond with:

- set FIN – terminates the “real-time” mode;
- A new set of commands (set ddly=xx and get <filename>) – schedules the next POST request after xx seconds with the specified file;
- Another valid combination of API commands.

Attention!

If the response does not include ddly=xx (where xx is between 3 and 59 seconds), the controller will start sending HTTP/HTTPS POST requests as quickly as possible using the last requested file. This behavior is not recommended.

While the device is in “real-time” mode, all periodic and alarm-triggered HTTP/HTTPS POST requests to the same server are suspended.

If a second server is configured, it will continue to operate with periodic and alarm-triggered HTTP/HTTPS POST requests as usual.

9.2. Full list of HTTP API commands

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

9.3. File structures

The XML file structure is described in Appendix A.

The JSON file structure is described in Appendix B.

9.4. MQTT

MQTT is a lightweight publish/subscribe messaging protocol based on a client–server architecture. It is an open and easy-to-implement protocol.

MQTT is widely used in industries such as automotive, manufacturing, oil and gas, and telecommunications.

For more information about MQTT, visit: www.mqtt.org

10. Firmware update

The TCG120-4 supports firmware updates via the Web interface and over the air.

For firmware updates via the Web interface, refer to section 7.4.2. FW update.

For firmware updates over the air (requires 4G LTE/3G/2G connection), follow the steps below:

- Upload the update file (with a .cod extension) to a public HTTP server;
- Send the firmware update command (see section 8. SMS Configuration and Control for syntax).
- Attention! Only Master and Superusers are authorized to send this command.

  The firmware file will be downloaded and verified. The download takes approximately 3 minutes. If the file is valid, the Master will receive a confirmation SMS message. The update procedure takes about 2 minutes. After the firmware update is complete (approximately 5 minutes total), the TCG120-4 will restart.

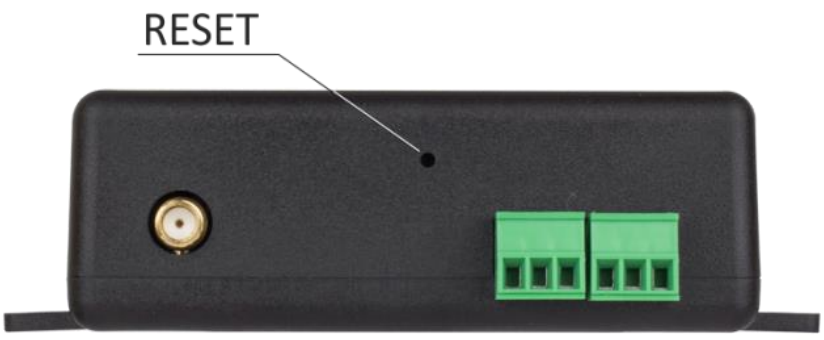
Important: Do not turn off the power supply during the update. Interrupting the power supply may cause permanent damage to the device.

11. Factory default settings

The TCG120-4 can be restored to its factory default settings by following these steps:

- Turn off the power supply;
- Press and hold the RESET button, then turn on the power supply;
- The STA and SIG LEDs will turn ON;
- Release the RESET button.

The module will then restore its factory default settings.



12. Operating environment

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

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

13. Safety

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

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

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

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

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

To ensure proper operation of the device, follow these steps:

- ensure that the device is installed correctly, refer to this user manual;
- log into the device via a web browser;
- perform the necessary setup;
- short the “Din1” and “GND” terminals;
- install sensor on the 1-Wire bus;
- navigate to the “Monitoring page” of the Web interface. The correct parameter values should be displayed, and the “SIG” and “STA” LEDs should indicate the "Connected to Network" status.

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

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

14. Maintenance

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

Clean the device only with a dry cloth. Do not use liquid or aerosol cleaners. Do not use magnetic or static cleaning devices (dust removers) or any abrasive materials.

Following these guidelines will help maintain the device in optimal condition and ensure long-term reliable operation.

The XML file (status.xml) structure:

```
<Monitor>
<DeviceInfo>
<DeviceName>TCG120-4</DeviceName>
<HostName>TCG120-4G</HostName>
<ID>867782072152288</ID>
<FwVer>TCG120-4G-h2.46-v1.027</FwVer>
<MnfInfo>www.teracomsystems.com</MnfInfo>
<SysContact>info@teracomsystems.com</SysContact>
<SysName>TCG120-4G</SysName>
<SysLocation>Location</SysLocation>
</DeviceInfo>
<S>
<S1>
<description>S1</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>
</S1>
<S2>
<description>S2</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>
</S2>
<S3>
<description>S3</description>
<id>0000000000000000</id>
<item1>
<value>---</value>
<unit>---</unit>
```

```
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item1>
<item2>
<value>---</value>
<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item2>
</S3>
<S4>
<description>S4</description>
<id>0000000000000000</id>
<item1>
<value>---</value>
<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item1>
<item2>
<value>---</value>
<unit>---</unit>
<alarm>0</alarm>
<min>---</min>
<max>---</max>
<hys>---</hys>
</item2>
</S4>
</S>
<AI>
<AI1>
<description>Analog Input 1</description>
<value>0.028</value>
<unit>V</unit>
<multiplier>1.000</multiplier>
<offset>0.0000</offset>
<alarm>0</alarm>
<min>0.000</min>
<max>30.000</max>
<hys>1.000</hys>
</AI1>
<AI2>
<description>Analog Input 2</description>
<value>0.017</value>
<unit>V</unit>
<multiplier>1.000</multiplier>
<offset>0.0000</offset>
<alarm>0</alarm>
<min>0.000</min>
<max>30.000</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>
<HTTPPush>
<Key/>
<PushPeriod>300</PushPeriod>
</HTTPPush>
<MQTT>
<Period>300</Period>
</MQTT>
<signalpercent>90</signalpercent>
<hwerr/>
<Alarmed>0</Alarmed>
<Scannig>Scan</Scannig>
<Time>
<Date>18.05.2026</Date>
<Time>06:44:51</Time>
</Time>
<NetworkInfo>
<Name>A1 BG A1 BG</Name>
<SC>
<MCC>284</MCC>
<MNC>01</MNC>
<LAC>28201</LAC>
<CID>13806321</CID>
<SQ>-57</SQ>
</SC>
<Latitude>---</Latitude>
<Longitude>---</Longitude>
</NetworkInfo>
</Monitor>

Where:

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

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

The JSON file (status.json) structure (all values are transmitted as strings):


```
{
  "Monitor": {
    "DeviceInfo": {
      "DeviceName": "TCG120-4",
      "HostName": "TCG120-4G",
      "ID": "867782072152288",
      "FwVer": "TCG120-4G-h2.46-v1.027",
      "MnfInfo": "www.teracomsystems.com",
      "SysContact": "info@teracomsystems.com",
      "SysName": "TCG120-4G",
      "SysLocation": "Location"
    },
    "S": {
      "S1": {
        "description": "S1",
        "id": "0000000000000000",
        "item1": {
          "value": "---",
          "unit": "---",
          "alarm": "0",
          "min": "---",
          "max": "---",
          "hys": "---"
        },
        "item2": {
          "value": "---",
          "unit": "---",
          "alarm": "0",
          "min": "---",
          "max": "---",
          "hys": "---"
        }
      },
      "S2": {
        "description": "S2",
        "id": "0000000000000000",
        "item1": {
          "value": "---",
          "unit": "---",
          "alarm": "0",
          "min": "---",
          "max": "---",
          "hys": "---"
        },
        "item2": {
          "value": "---",
          "unit": "---",
          "alarm": "0",
          "min": "---",
          "max": "---",
          "hys": "---"
        }
      },
      "S3": {
        "description": "S3",
        "id": "0000000000000000",
        "item1": {
```

```
        "value": "---",
        "unit": "---",
        "alarm": "0",
        "min": "---",
        "max": "---",
        "hys": "---"
      },
      "item2": {
        "value": "---",
        "unit": "---",
        "alarm": "0",
        "min": "---",
        "max": "---",
        "hys": "---"
      }
    },
    "S4": {
      "description": "S4",
      "id": "0000000000000000",
      "item1": {
        "value": "---",
        "unit": "---",
        "alarm": "0",
        "min": "---",
        "max": "---",
        "hys": "---"
      },
      "item2": {
        "value": "---",
        "unit": "---",
        "alarm": "0",
        "min": "---",
        "max": "---",
        "hys": "---"
      }
    }
      },
  "AI": {
    "AI1": {
      "description": "Analog Input 1",
      "value": "0.028",
      "unit": "V",
      "multiplier": "1.000",
      "offset": "0.0000",
      "alarm": "0",
      "min": "0.000",
      "max": "30.000",
      "hys": "1.000"
    },
    "AI2": {
      "description": "Analog Input 2",
      "value": "0.017",
      "unit": "V",
      "multiplier": "1.000",
      "offset": "0.0000",
      "alarm": "0",
      "min": "0.000",
      "max": "30.000",
      "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"
      }
    },
    "HTTPPush": {
      "Key": "",
      "PushPeriod": "300"
    },
    "MQTT": {
      "Period": "300"
    },
    "signalpercent": "90",
    "hwerr": "",
    "Alarmed": "0",
    "Scannig": "Scan",
    "Time": {
      "Date": "18.05.2026",
      "Time": "06:39:16"
    },
    "NetworkInfo": {
      "Name": "A1 BG A1 BG",
      "SC": {
        "MCC": "284",
        "MNC": "01",
        "LAC": "28201",
        "CID": "13806321",
        "SQ": "-57"
      },
      "Latitude": "---",
      "Longitude": "---"
    }
  }
}
```

Full list of HTTP API commands

TCG120-4 supports following HTTP commands (case sensitive):

Command	Description
Monitoring	
r**n**=xxx	Turn relay **n** ON (xxx=on) or OFF (xxx=off) (**n** is 1 or 2 for the respective relay) r1=on – will turn relay 1 ON r2=off – will turn relay 2 OFF r1=on&r2=on – will turn all relays ON r1=off&r2=off – will turn all relays OFF
r**n**=tg	Toggle relay **n** (**n** is 1 or 2 for the respective relay) r2= tg – will toggle relay 2 r1=tg&r2=tg – will toggle relay 1 and 2
r**n**=pl	Pulse relay **n** (**n** is 1 or 2 for the respective relay) r1=pl – will pulse relay 1 r1=pl&r2=pl – will pulse relay 1 and 2
SMS/Call	
unumf**n**=x	Set SMS number format for Users - short codes (x=0) or E.164 international format (x=1) (**n** is 1,2,3 or 4 for the respective user) SMS number format for Master is E.164 international standard only and can't be changed. unumf1=0 – set short code number format for User1 unumf2=1 – set E.164 international format for User2
sr**n**=xxx	Set SMS number **n** (**n** is 0 for master and 1,2,3 or 4 for the respective user) sr0=+359885180709 – set Master number. E.164 international format is mandatory. sr1=57575701 – set User1 number in short code format. Before this command must be set SMS number format - short codes or E.164 international format.
conal=x	Set connectivity alert for Master - activation (x=1) or deactivation (x=0)
suser**n**=x	Set superuser rights for users - activation (x=1) or deactivation (x=0) suser1=1 – activation superuser right for User 1 suser2=0 – deactivation superuser right for User 2
ual**n**=x	Activation (x=1) or deactivation (x=0) of alarm notification (**n** is 0 for master and 1,2,3 or 4 for the respective user) ual0=1 – activation alarm notification for Master ual1=1 – deactivation alarm notification for User 1
cu**n**=xxx	Set phone call number **n** (**n** is 1,2,3,4 or 5 for the respective user) Number format is E.164 international standard only and can't be changed. cu1=+359886455736 - set phone call number for user 1
Input/Output	
sd**n**=xxx	Set sensor **n** description (**n** is 1,2,3 or 4 for the respective sensor)

	sd1=S1-Temp – set sensor 1 description
ad**n**=xxx	Set analog input **n** description (**n** is 1 or 2 for the respective analog input) ad1=AN1 – set analog input 1 description
au**n**=xxx	Set analog input **n** unit (**n** is 1 or 2 for the respective analog input) au1=VDC – set analog input 1 unit
am**n**=x.xxx	Set analog input **n** multiplier (**n** is 1 or 2 for the respective analog input) am1=2.000 – set analog input 1 multiplier
ao**n**=x.xxxx	Set analog input **n** offset (**n** is 1 or 2 for the respective analog input) ao1=0.0005 – set analog input 1 offset
dd**n**=xxx	Set digital input **n** description (**n** is 1 or 2 for the respective digital input) dd1=DIN1 – set digital input 1 description
dl**n**=xxx	Set digital input **n** low level description (**n** is 1 or 2 for the respective digital input) dl1=CLOSED – set low level description for digital input 1
dh**n**=xxx	Set digital input **n** high level description (**n** is 1 or 2 for the respective digital input) dh1=OPEN – set high level description for digital input 1
rd**n**=xxx	Set relay **n** description (**n** is 1 or 2 for the respective relay) rd1=REL1 – set relay 1 description
pw**n**=x.x	Set relay **n** pulse time in seconds (**n** is 1 or 2 for the respective relay) pw1=0.5 – set relay 1 pulse time to 0.5 sec
rc**n**=xx	Set relay **n** activated from xx (**n** is 1 or 2 for the respective relay) Where xx: SMS/HTTP API(0), S11(1), S12(2), S21(3), S22(4), S31(5), S32(6), S41(7), S42(8), Analog input 1(9), Analog input 2(10), Digital input 1(11), Digital input 2(12), Any Alarm(13) rc1=13 – set relay 1 activated from Any alarm rc2=0 – set relay 2 activated from SMS/HTTP API
aae**n**=x	Set relay **n** action on alarm – Turn on(x=0) or Single pulse(x=2) (**n** is 1 or 2 for the respective relay) aaen1=0 – set relay 1 action on alarm Turn on
sn**pt**=30.0	Set Min of sensor to 30.0 (**p** is 1,2,3 or 4 for the respective sensor **t** is 1 or 2 for the respective parameter of sensor) sn12=30.0 will set Min for sensor 1, parameter 2
sx**pt**=40.0	Set Max of sensor to 40.0 (**p** is 1,2,3 or 4 for the respective sensor **t** is 1 or 2 for the respective parameter of sensor) sx42=40.0 will set Min for sensor 4, parameter 2
sy**pt**=2.0	Set Hysteresis of sensor to 2.0 (**p** is 1,2,3 or 4 for the respective sensor **t** is 1 or 2 for the respective parameter of sensor) sy31=2.0 will set Hys for sensor 3, parameter 1

Conditions	
s**xy**m=b	Activation (b=1)/deactivation (b=0) of Mail alarm notification for sensor x, part y (**x** is 1,2,3 or 4 for the respective sensor **y** is 1 or 2 for the respective parameter of sensor) s11m=1 – activation mail alarm notification for sensor 1, part 1
s**xy**s=b	Activation (b=1)/deactivation (b=0) of SMS alarm notification for sensor x, part y (**x** is 1,2,3 or 4 for the respective sensor **y** is 1 or 2 for the respective parameter of sensor) s21s=1 – activation sms alarm notification for sensor 2, part 1
s**xy**p=b	Activation (b=1)/deactivation (b=0) of HTTP POST alarm notification for sensor x, part y (**x** is 1,2,3 or 4 for the respective sensor **y** is 1 or 2 for the respective parameter of sensor) s12p=1 – activation of HTTP POST alarm notification for sensor 1, part 2
s**xy**q=b	Activation (b=1)/deactivation (b=0) of MQTT alarm notification for sensor x, part y (**x** is 1,2,3 or 4 for the respective sensor **y** is 1 or 2 for the respective parameter of sensor) s22q=1 – activation of MQTT alarm notification for sensor 2, part 2
sclm=x	Mail notification in case of a sensor communication lost - activation (x=1) or deactivation (x=0)
scls=x	SMS notification in case of a sensor communication lost - activation (x=1) or deactivation (x=0)
sclp=x	HTTP POST notification in case of a sensor communication lost - activation (x=1) or deactivation (x=0)
sclq=x	MQTT notification in case of a sensor communication lost - activation (x=1) or deactivation (x=0)
srn=x	Return notification for sensors - activation (x=1) or deactivation (x=0)
delsen=xxxx	Notification delay for sensors (xxxx is between 0 and 3600)
vn**f**=10.0	Set Min of analog input to 10.0 (**f** is 1 or 2 for the respective input) vn1=10.0 will set Min for analog input 1
vx**f**=20.0	Set Max of analog input to 20.0 (**f** is 1 or 2 for the respective input) vx2=20.0 will set Max for analog input 2
vy**f**=1.0	Set Hys of analog input to 1.0 (**f** is 1 or 2 for the respective input) Vy1=1.0 will set Hys for analog input 1
a**n**m=b	Mail alarm notification for analog input - activation (b=1) or deactivation (b=0) (**n** is 1 or 2 for the respective analog input) a1m=1 – mail alarm notification for analog input 1 is activated
a**n**s=b	SMS alarm notification for analog input - activation (b=1) or deactivation (b=0) (**n** is 1 or 2 for the respective analog input) a1s=1 – sms alarm notification for analog input 1 is activated

a**n**p=b	HTTP POST alarm notification for analog input - activation (b=1) or deactivation (b=0) (**n** is 1 or 2 for the respective analog input) a2p=1 – HTTP POST alarm notification for analog input 2 is activated
a**n**q=b	MQTT alarm notification for analog input - activation (b=1) or deactivation (b=0) (**n** is 1 or 2 for the respective analog input) a2q=1 – MQTT alarm notification for analog input 2 is activated
arn=x	Return notification for analog inputs - activation (x=1) or deactivation (x=0)
delanl=xxxx	Notification delay for analog inputs (xxxx is between 0 and 3600)
ds**n**=x	Select digital alarm state to Low level(x=0) or High level(x=1) (**n** is 1 or 2 for the respective digital input)
dda**n**=xxxx	Low to high delay for digital input **n** (**n** is 1 or 2 for the respective digital input) (xxxx is between 0 and 3600)
ddd**n**=xxxx	High to low delay for digital input **n** (**n** is 1 or 2 for the respective digital input) (xxxx is between 0 and 3600)
d**n**m=b	Mail alarm notification for digital input - activation (b=1) or deactivation (b=0) (**n** is 1 or 2 for the respective digital input) d1m=1 – mail alarm notification for digital input 1 is activated
d**n**s=b	SMS alarm notification for digital input - activation (b=1) or deactivation (b=0) (**n** is 1 or 2 for the respective digital input) d1s=1 – SMS alarm notification for digital input 1 is activated
d**n**p=b	HTTP POST alarm notification for digital input - activation (b=1) or deactivation (b=0) (**n** is 1 or 2 for the respective digital input) d2p=0 – HTTP POST alarm notification for digital input 2 is deactivated
d**n**q=b	MQTT alarm notification for digital input - activation (b=1) or deactivation (b=0) (**n** is 1 or 2 for the respective digital input) d2q=1 – MQTT alarm notification for digital input 2 is activated
drn=x	Return notification for digital inputs - activation (x=1) or deactivation (x=0)
deldig=xxxx	Notification delay for digital inputs (xxxx is between 0 and 3600)
System	
APN=xxx	Set APN for data connection
authmod =x	Set authentication mode for data connection – None(x=0), PAP(x=1), CHAP(x=2), PAP or CHAP(x=3)
APNU=xxx	Set username authentication for data connection
APNP=xxx	Set password authentication for data connection
aren=x	Enable(x=1)/disable(x=0) automatic network connection reset
timers=hh:mm:ss	Time of automatic network connection reset (hh:mm:ss)

tu=x	Set temperature units – Celsius(x=0)/ Fahrenheit(x=1)
pu=x	Set pressure units – hPa(x=0), mbar(x=1) or mmhg(x=2)
hn=xxx	Set host name - xxx, max 31 symbols
wm=x	Set writing mode - left-to-right(x=0) or right-to-left(x=1)
sn=xxx	Set system name – xxx, max 31 symbols
sl=xxx	Set system location – xxx, max 31 symbols
sc=xxx	Set system contact – xxx, max 31 symbols
Time	
TS=xxx	Time server URL – xxx=IP/URL:port, max 63 symbols
TZ=hh:mm	Time zone – hh:mm TZ=+03:00
interval1=x	NTP time synchronization interval, hours interval1=12 - NTP time synchronization interval 12 hours
interval2=x	Repeat NTP time synchronization if not found, hours Interval2=1 – repeat NTP synchronization if not found after 1 hour
HTTP POST	
pmet=x	HTTP POST disable (x=0) or enable (x=1)
pDrv=x	HTTP POST mode – http (x=2) or https (x=3)
dataf=x	Data format XML/JSON for HHTP POST – 0 XML, 1 JSON
purl=yyy	URL for XML/JSON HTTP POST to Server 1, where yyy is a full path to php file purl=212.25.45.120:30181/xampp/test/pushtest.php
purl2=yyy	URL for XML/JSON HTTP POST to Server 2, where yyy is a full path to php file purl2=212.25.45.120:30181/xampp/test/pushtest.php
pper=x	XML/JSON HTTP POST period in seconds (x is between 60 and 172800) pper=600 – will set POST period to 600 seconds
pperh=hh:mm:ss	XML/JSON HTTP POST period in format hh:mm:ss pperh=00:30:00 – will set POST period on 30 minutes
dk=xxx	XML/JSON HTTP POST key – xxx is up to 127 characters
pans=x	Process answer - No (x=0) or Yes (x=1)
MQTT	
mqtten=x	MQTT service disable (x=0) or enable (x=1)
mdata=x	Data format JSON/Plain text for MQTT Publish – JSON (x=0) or Plain text (x=1)
mmode=x	Publish protocol, where x is 0 for unsecure and 1 for TLS/SSL
murl=xxx	URL for MQTT publish – xxx=IP/URL, max 127 symbols
mport=xxxx	Port for MQTT publish, where xxxx is a port
muser=xxxx	Username authentication for MQTT, where xxxx is a username
mpass=xxxx	Password authentication for MQTT, where xxxx is a password
mper=xxx	MQTT publish period in seconds (x is between 60 and 172800) mper=600 – will set MQTT publish period to 600 seconds
nametopic=xxx	MQTT main topic name
senstopic=xxx	MQTT sensors topic name
std**ny**=xxx	MQTT sensor **n** part **y** topic name, where **n** is a sensor number and **y** is a sensor part std21=21, set topic name for sensor 2 part 1
antopic=xxx	MQTT analog inputs topic name

atd**n**=xxx	MQTT analog input **n** topic name atd1=1, set analog input 1 topic name
digtopic=xxx	MQTT digital inputs topic name
dtd**n**=xxx	MQTT digital input **n** topic name dtd1=1, set digital input 1 topic name
Geolocation	
glocen=x	Geolocation service disable (x=0) or enable (x=1)
glpr=x	Geolocation provider – Unwiredlabs (x=0) or Google (x=2)
glurl=xxx	Geolocation server URL
token=xxx	Token / API key
glper=xx	Geolocation period request, where **xx** are seconds from 60 to 14400
save	Save all previous changes (except relays' one) in the FLASH memory. **As every save reflects the FLASH cycles (endurance), this command should be used very carefully.** pper=120&save – will set POST period to 120 seconds and save it
FIN	Terminate the session

Multiple commands are sent concatenated with "&".

The commands are sent in the answer on XML/JSON or CSV HTTP POSTs. They are executed if "Process Answer" is enabled.

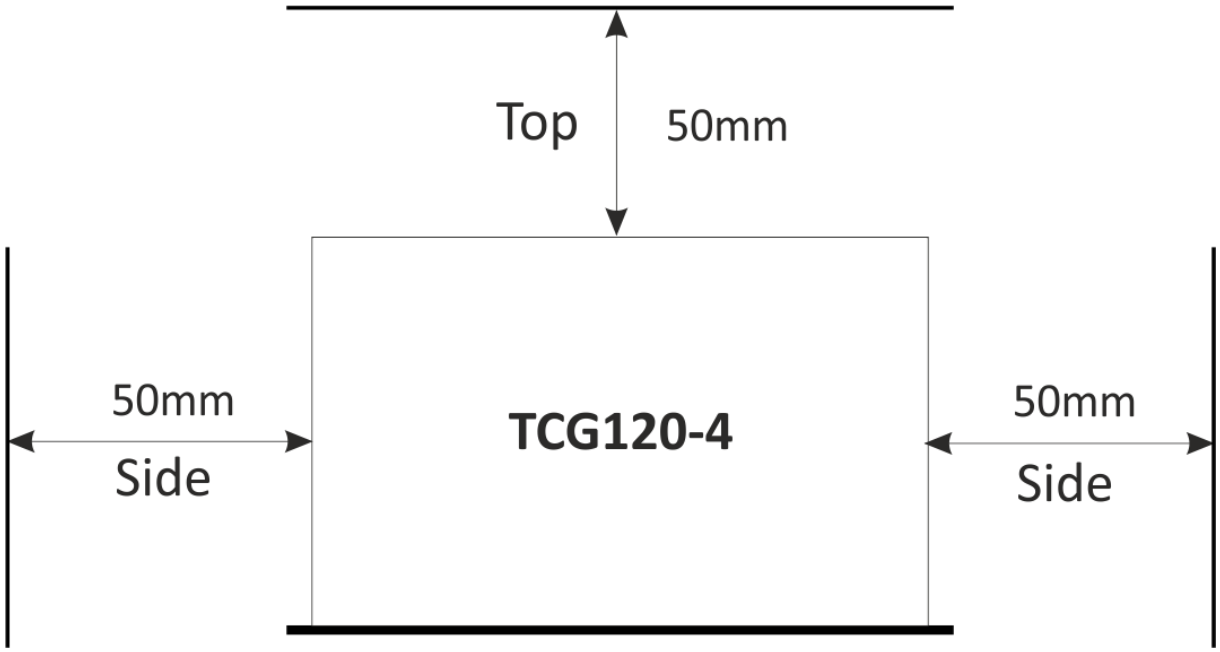


Fig.1

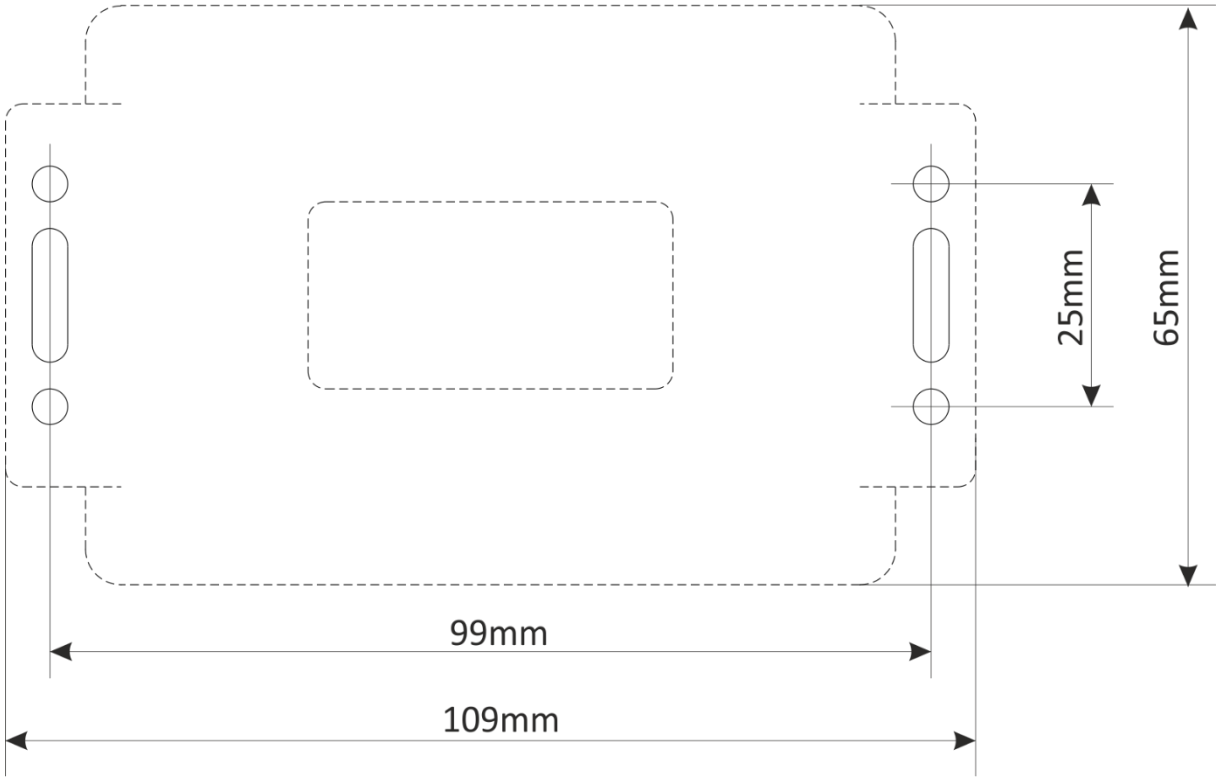


Fig.2