control solutions
TERACOM



TCW260
Energy monitoring module

USER MANUAL

www.teracomsystems.com

1. Introduction

TCW260 is an energy monitoring module with Ethernet connectivity and data logging capability. All monitored parameters can be seen as numbers and as graphs.

The device has 4 digital inputs, S0+ compatible. The inputs can work in two modes – OPEN/CLOSED for a reading of “dry contact” outputs and COUNTER mode for direct connection of energy meters with pulse outputs.

The energy monitoring module has also 6 analog inputs. Every analog input can work either in voltage (0/10V) or current loop (0/20mA) modes. The mode is changed over the user interface.

All digital and analog inputs are fully isolated from the power supply ground.

TCW260 supports Modbus RTU interface for up to 24 Teracom and third-party sensors. The used RS-485 interface is fully isolated from the power supply ground.

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

The user can also arrange up to 24 independent alarms with 5 different user selectable states. Every alarm can be set up using up to 2 limits and hysteresis. The alarm can be assigned to a specific channel. In this case, in an alarm state, the assigned channel is colored on the monitoring page and graphs.

The device supports SNMP, HTTP API, Modbus TCP/IP and MQTT as machine-to-machine (M2M) communication.

2. Features

- 10/100 Mb Ethernet connectivity with Auto-MDIX;
- HTTP/HTTPS server support
- Password protected, web-based configuration and control;
- 4 digital inputs, S0-pulse interface compatible;
- 6 analog inputs with 0/10V and 0/20mA modes;
- Modbus RTU interface for up to 24 sensors (registers);
- Up to 24 channels for monitoring;
- Up to 24 independent alarms;
- SNMPv2 and SNMPv3 support;
- SNMP traps sending for alert conditions;
- SMTP with SSL/TLS security;
- TLS 1.0, TLS 1.1 and TLS 1.2 support;
- HTTP and SNMP port changing;
- HTTP API commands;
- Periodical HTTP/HTTPS POST of XML/JSON status files for client-server systems;
- Modbus TCP/IP support;
- MQTT 3.1.1 support;
- Dynamic DNS with DynDNS, No-IP and DNS-O-Matic support;
- NTP protocol support;
- Data logger for up to 70000 records;
- DIN Rail Mountable;
- Wide power supply voltage range;
- Backup/restore of settings;
- Remote firmware update.

3. Applications

The energy monitoring module TCW260 is dedicated to monitoring and recording the parameters of the measurement of resources - electricity meters, gas meters, water meters, and others. Rising resource costs require reliable analysis and optimization. Depending on the search result, this can be done at a micro level (separate machine) or macro level (company).

The monitoring module can integrate seamlessly into already working objects. With proper selection of sensors, this can happen even without interrupting the production process.

Of course, the module can be used also for general monitoring of industrial processes.

A few example applications include:

- Energy analysis and optimization;
- Water consumption analysis;
- Gas consumption optimization;
- A building management system;
- Industrial processes monitoring;
- General SCADA systems.

4. Specifications

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

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

- Power supply
  Operating voltage range (including -15/+20% according to IEC 62368-1): 10 to 28VDC
  Current consumption: 220mA @ 12VDC (without RS-485 powering)

- RS-485 interface
  Isolation: Isolated (1000VDC)
  Output voltage (pin 7 of RJ-45): 5.0 ± 0.3VDC
  Maximum output current (pin 7 of RJ-45): 0.2A

- Digital inputs
  Isolation: Isolated (1000VDC)
  Mode: OPEN/CLOSED (“Dry contact”) or COUNTER (S0-pulse interface outputs)
  Maximum input voltage: +5.5VDC
  Sampling rate: 1mS
  Digital filtering time interval: 5 to 60000mS
  Minimum pulse width (COUNTER mode): 8 ms (min.) at 5 ms filtering time
  Minimum pause between pulses (COUNTER mode): 8 ms (min.) at 5 ms filtering time
  Maximum counting frequency: 65 Hz (at 50% duty cycle and 5 ms filtering time)

- Analog inputs
  Isolation: Isolated (1000VDC)
  Type: Single ended
  Resolution: 12 bits
  Mode: Voltage or current loop
  Input range: 0/10V or 0/20mA
  Accuracy: ±1%
  Sampling rate: 600mS per channel (averaged value of 100 samples)
  Input impedance in voltage mode: 1mega-ohm (min.)
  Input impedance in current loop mode: 455ohm

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

- Lithium battery
  Type: CR1220

  ⚠ **Caution:** Replacing the battery with an incorrect type may result in an explosion.

5. LED indicators

The following indicators show the status of the controller:

- **PWR** (red) – Steady ON in normal operation; blinks in sync with the STS LED in the event of a hardware error;
- **STS** (yellow) – Flashes to indicate that the controller's main program is running;
- **NET** (orange) – Indicates network status; remains ON when a network link is established, and blinks to show network activity.

6. Installation and setup

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

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

6.1. Mounting

TCW260 must be installed in a clean, dry, and non-flammable location. Ventilation is recommended for high ambient temperature environments.

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

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

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

6.2. Connection

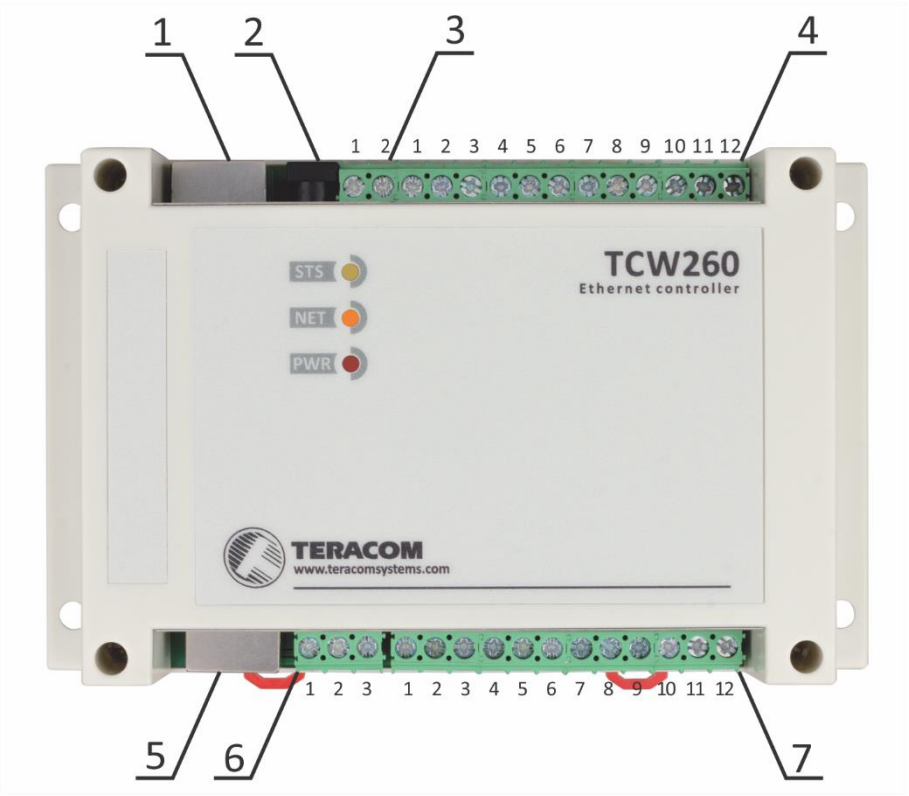
Warning! Power off before wiring.

Follow these steps for correct wiring:

- Turn off power;

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

Ensure that wires are securely attached to terminals and tightened. Improper wiring or configuration can cause permanent damage to TCW260 or connected equipment.



Connector 1 Ethernet - RJ45

Connector 2 Power - 2.1x5.5mm connector, (Central positive)

Connector 3
Pin1 – Power positive
Pin2 – Power negative (GND)

Connector 4
Pin1 – S04- (SGND)
Pin2 – not connected
Pin3 – S04+ (Digital in 4)
Pin4 – S03- (SGND)
Pin5 – not connected
Pin6 – S03+ (Digital in 3)
Pin7 – S02- (SGND)
Pin8 – not connected
Pin9 – S02+ (Digital in 2)
Pin10 – S01- (SGND)
Pin11 – not connected
Pin12 – S01+ (Digital in 1)

Connector 5
Pin1 – not connected (most left)
Pin2 – not connected
Pin3 – not connected
Pin4 – 485-
Pin5 – 485+
Pin6 – not connected
Pin7 – +VDD
Pin8 – SGND

Connector 6
Pin1 – 485+
Pin2 – SGND
Pin3 – 485-

Connector 7
Pin1 – Analog in 1
Pin2 – SGND
Pin3 – Analog in 2
Pin4 – SGND
Pin5 – Analog in 3
Pin6 – SGND
Pin7 – Analog in 4
Pin8 – SGND
Pin9 – Analog in 5
Pin10 – SGND
Pin11 – Analog in 6
Pin12 – SGND

It is recommended using grounding configuration with isolated local ground and PE.

6.2.1. Power supply connection

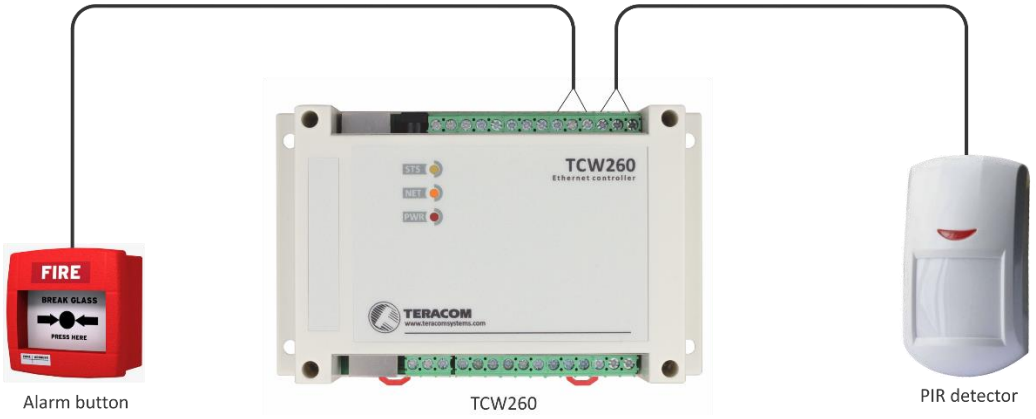
TCW260 must be powered by the adapter SYS1308(N)-2412-W2E or equivalent, suitable for overvoltage category II and certified for safety compliance. The power supply device should be able to withstand short circuits and secondary circuit overloads. Ensure the equipment is easily accessible for disconnecting from the power supply during use.

6.2.2. Digital inputs connection

All inputs are galvanic isolated from the power supply ground.

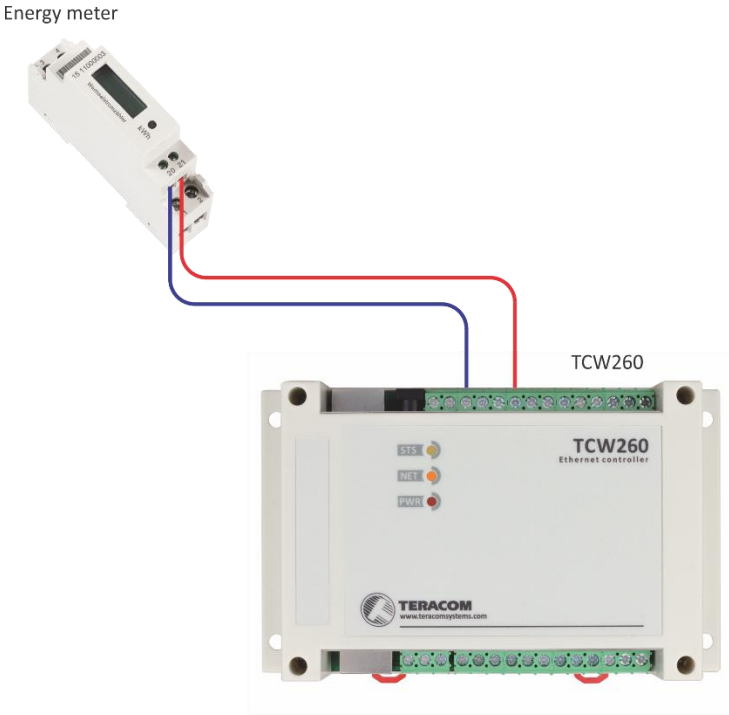
The digital inputs of the TCW260, when set to OPEN/CLOSED mode, allow monitoring of devices with “dry contact” outputs—such as door contact switches, push buttons, and PIR detectors.

The following diagram illustrates the connection of a dry contact switch to the TCW260 input: one side of the contact connects to the “S0+” terminal, while the other connects to the “S0-” terminal.



When set to COUNTER mode, the digital inputs of the TCW260 can monitor devices with an S0-pulse interface, such as energy meters and water meters.

The diagram below shows how to connect an energy meter to the TCW260 input: connect one side of the contact to the “S0+” terminal and the other side to the “S0-” terminal.



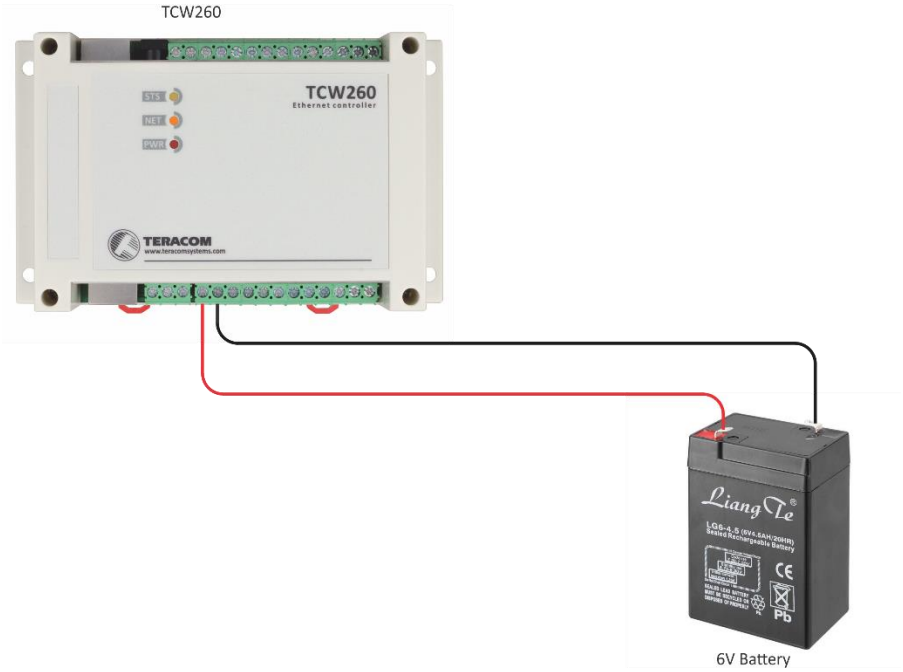
For optimal performance, the cable length should not exceed 30 meters.

6.2.3. Analog inputs connection

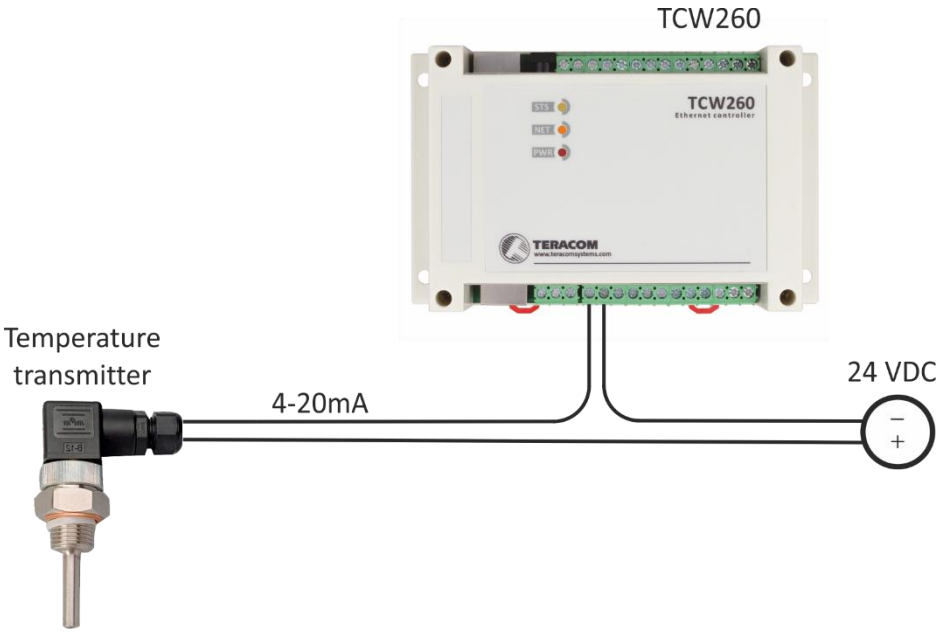
All inputs are galvanic isolated from the power supply ground.

The analog inputs of the TCW260 can monitor devices with voltage and current loop outputs, allowing direct connections to analog sensors such as temperature and humidity sensors, or current/voltage transducers.

The diagram below illustrates how to connect a battery to the analog input of the TCW260 in voltage mode: connect the positive terminal to “Analog In” and the negative terminal to “GND”.



The diagram below shows how to connect a temperature sensor with a current loop output to the analog input of the TCW260: connect the active terminal to “Analog In” and the shield terminal to “GND”.



For optimal performance, the cable length should not exceed 30 meters.

6.2.4. RS-485 connection

The RS-485 interface on the TCW260 is galvanically isolated from the power supply ground, providing additional protection.

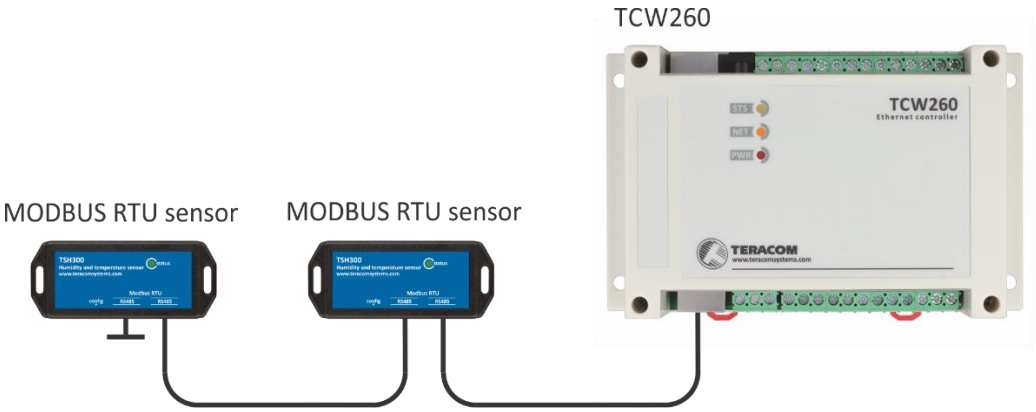
RS-485 is an industry-standard for serial communication, ideal for long-distance connections and environments with high electrical noise, such as industrial settings.

The TCW260 is compatible with Modbus RTU (over RS-485) slave devices and supports connections with up to 24 compatible devices, including those from Teracom and third-party manufacturers.

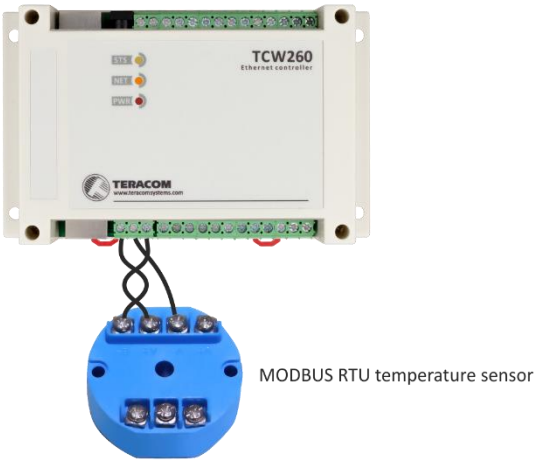
Connections are made using a standard RJ-45 connector, with the pinout specified in the "Modbus over Serial Line Specification and Implementation Guide" available at www.modbus.org.

To maintain signal integrity, it is essential to use 120-ohm line terminators at both ends of the RS-485 bus. The TCW260 includes one terminator and should be installed at one end of the line, leaving the other end for the user to terminate.

For setups with multiple devices, a daisy-chain (linear) topology is strongly recommended:



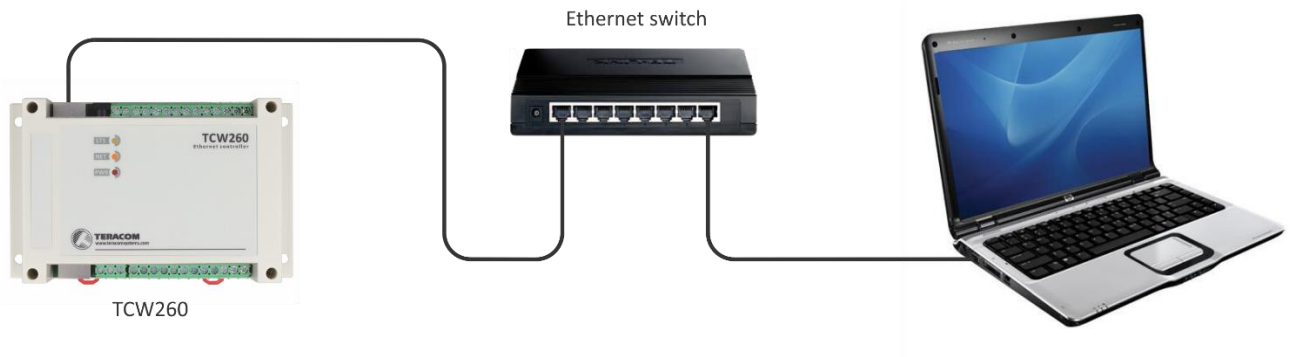
For Modbus RTU devices with screw terminals, the following connection options are available:



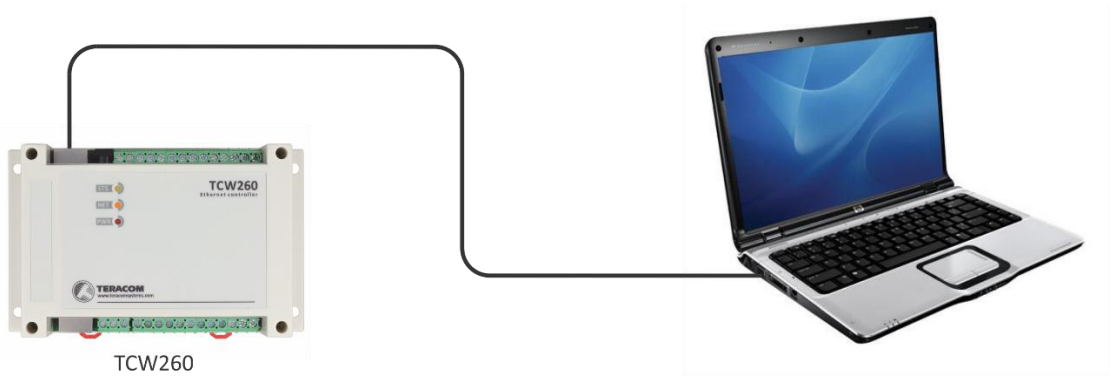
It is recommended to use only UTP or FTP cables. Keep the total cable length within 30 meters; however, functionality has been observed over longer distances.

6.2.5. Network connection

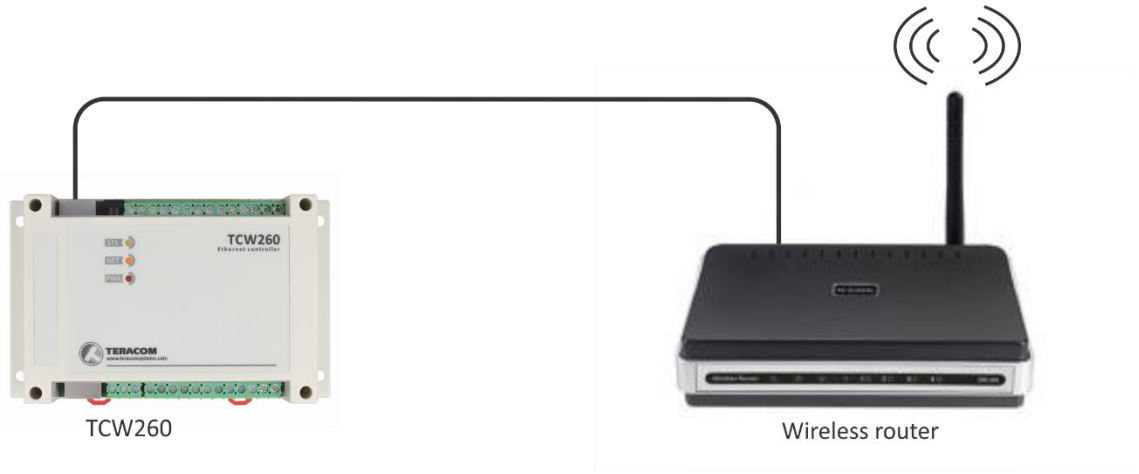
The Ethernet port of TCW260 should be connected to a 10/100 Base-T Ethernet hub, switch, or router.



For configuration purposes, the TCW260 can be directly connected to the Ethernet port of a computer. The device supports Auto-MDIX, so either a standard "straight-through" cable or a "crossover" cable can be used.



TCW260 can also be integrated into a wireless network by connecting it through a wireless router.

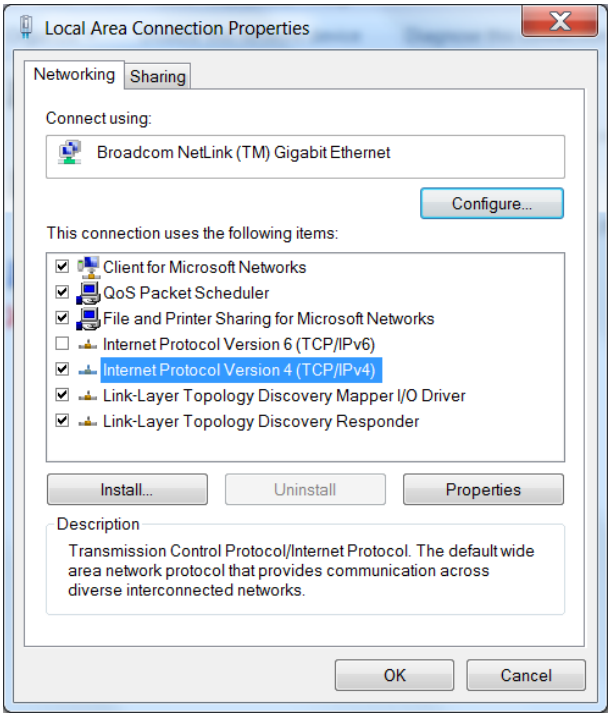


6.3. Communication setup

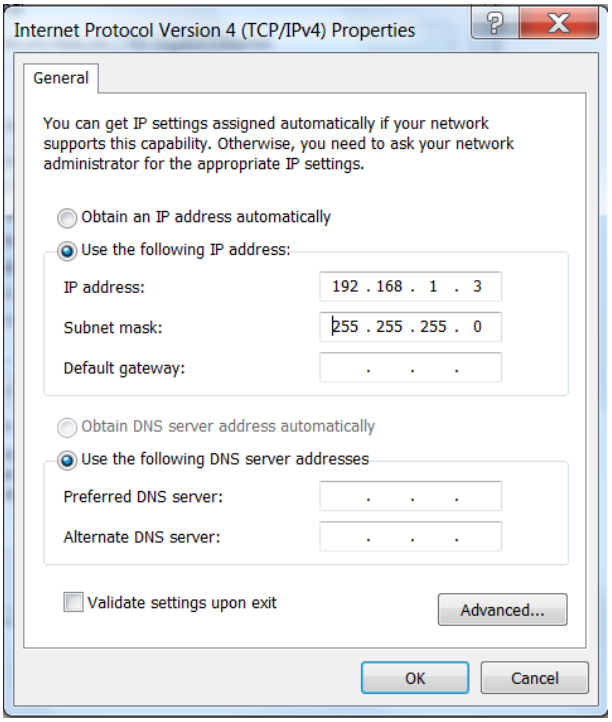
The TCW260 is shipped with the following default network settings:

- IP Address: 192.168.1.2
- Subnet Mask: 255.255.255.0
- Default Gateway: 192.168.1.1

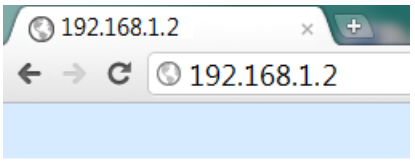
To establish communication with the TCW260, you can assign a temporary IP address to your computer. For computers running Windows OS, this can be done through the “Local Area Connection Properties”:



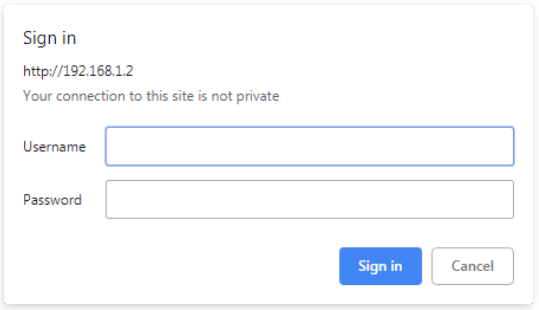
The temporary IP address should be on the same network; for example, you can use 192.168.1.3:



To access the web interface, enter http://192.168.1.2 into your browser's address bar.



If the network settings are configured correctly, a login pop-up window will appear:



The default authorization credentials are as follows: username: admin and password: admin.

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

All TCWxxx controllers connected to the LAN can be easily located using the free tool “TCW Discoverer”. This tool is available for both Windows and Mac operating systems and can be downloaded from www.teracomsystems.com.

7. Set up concept

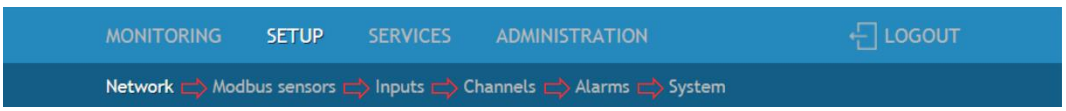
The primary setup channel for the device is the web interface, which is recommended for use, as it provides access to more settings than are available through SNMP and HTTP API commands.

The device setup begins with configuring the network settings. Next, the primary parameters—such as Modbus RTU sensors/registers, analog inputs, and digital inputs—are set.

Channels are created from the configured primary parameters. Each channel can combine up to two primary parameters and constants using mathematical operations; however, a channel can also be formed by a single primary parameter.

Once all channels are configured, alarms can be set up. It's important to note that alarms operate with channels, not with primary parameters. Each alarm can have up to two conditions, which may involve different channels. While alarms are independent of channels, they can also be assigned to a specific channel.

In summary, the correct setup should follow this order:



Once all previous settings are configured correctly, the desired services—such as the data logger, SNMP, HTTP API, etc.—can be activated.

8. Web interface

The web interface facilitates configuration and monitoring tasks, with all pages encoded in UTF-8. The controller accommodates multiple active sessions, and for the web interface, the device is compatible with both HTTP and HTTPS.

8.1. Monitoring

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

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

8.1.1. Channels

This page displays the status of all monitored channels – their values and alarm status. The information is updated on the refresh interval.

Channels	Description	Value	Unit	Status
1	Temperature	25.876	°C	Warning
2	Humidity	45.967	RH	Normal
3	Digital Input 1	OPEN		Critical
5	V05-Voltage	229.877	V	Minor
6	V06-Current	4.006	A	Normal
7	V07-Energy	6122.271	kWh	

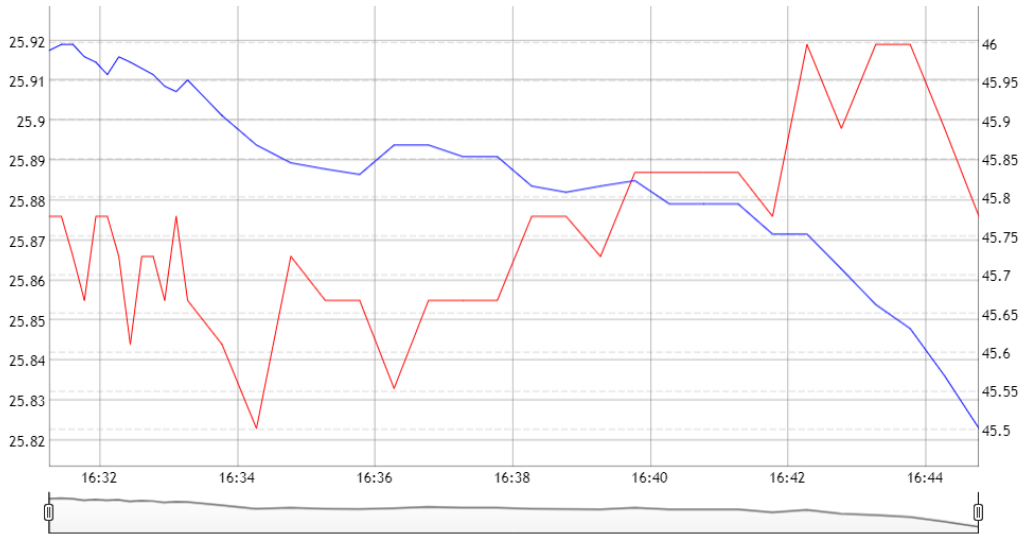
8.1.2. Alarms

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

Alarms	Description	Status
1	AL01-Temp.	Warning
2	AL02	Minor
3	AL03	Normal
4	AL04-DI1	Critical
5	AL05-Humidity	Normal

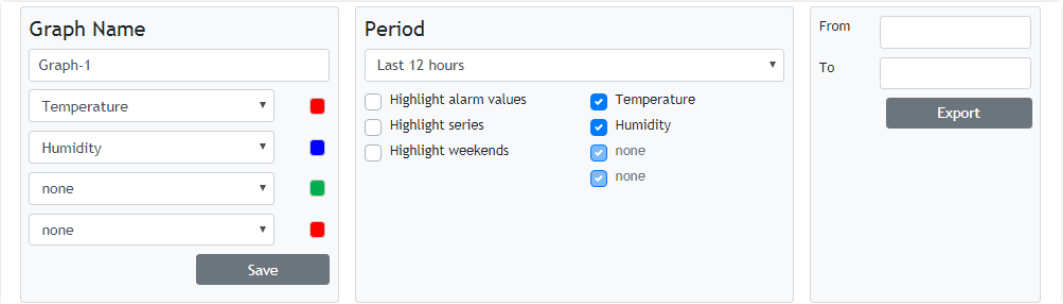
8.1.3. Graphs

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



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

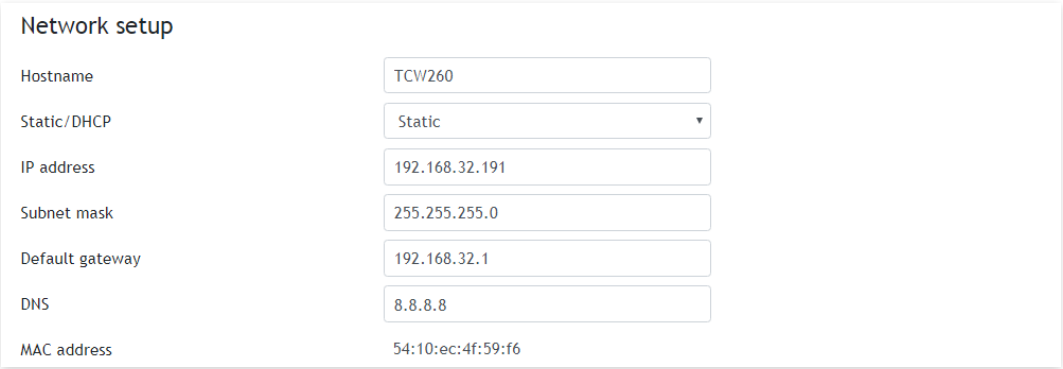
It is important to know that the information on graphs is static, it is not updated whit the newest values. If you want to see the last information, the page should be reloaded. The information can be exported in CSV file.



8.1. Setup

8.1.1. Network

The network parameters can be set on this page.



The controller supports both static and dynamic IP addresses. It is advisable to change the default IP address of the controller immediately after the first power-on to prevent potential IP address conflicts when multiple devices are connected to the same network.

Each time a new device is connected to the network, it may be necessary to clear the ARP cache. This can be done by entering arp -d in the command prompt on the computer.

The “Hostname” can be up to 15 characters long and will appear in the search results of the TCW Discoverer tool.

Public DNS servers (such as 8.8.8.8 or 8.8.4.4) can be used instead of the default gateway.

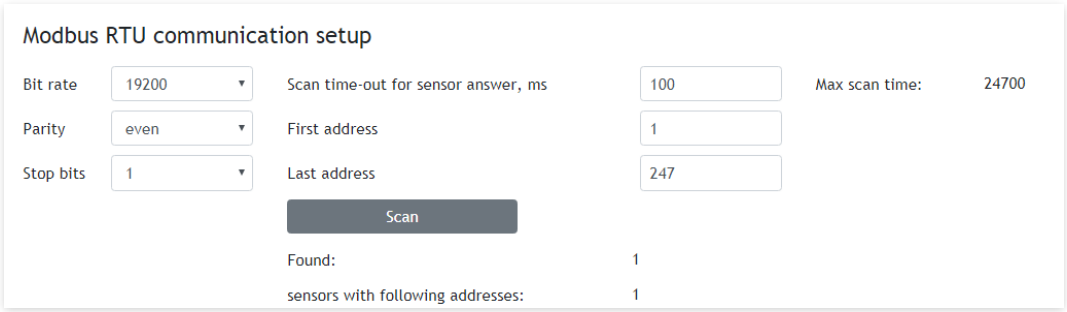
8.1.2. Modbus sensors

8.1.2.1. Modbus RTU communication setup

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

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

Playing with “Scan time-out for sensor answer” it is possible to find this parameter for an unknown sensor. The test starts with a large time-out (for example 500ms) and gradually decreases the time until the sensor stops responding. In order for the operation sustainability, the found time should be increased with for example 20%.

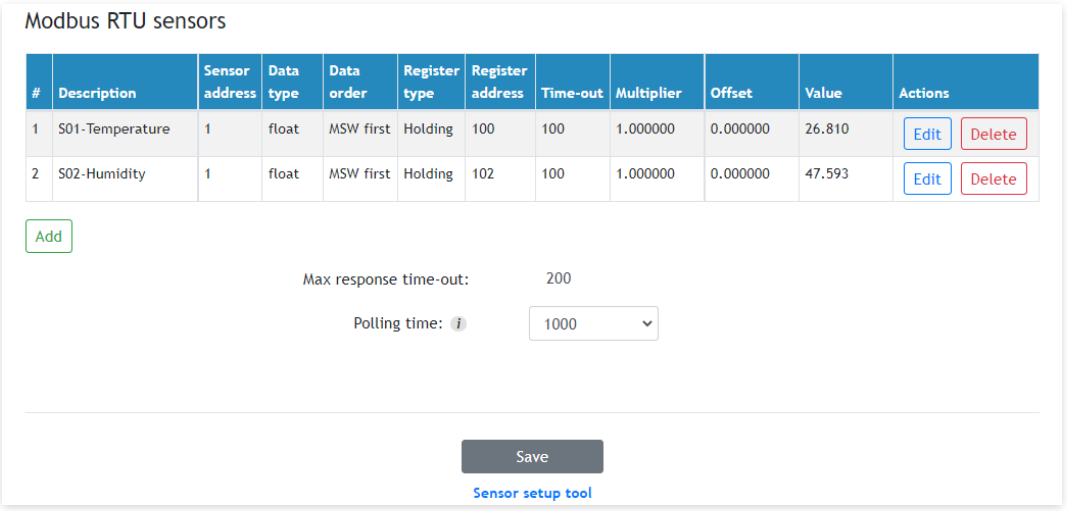


8.1.2.2. Modbus RTU sensors

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

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

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



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

The Multiplier and offset works as follows:

Value = (Raw_Value * Multiplier) + Offset.

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

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

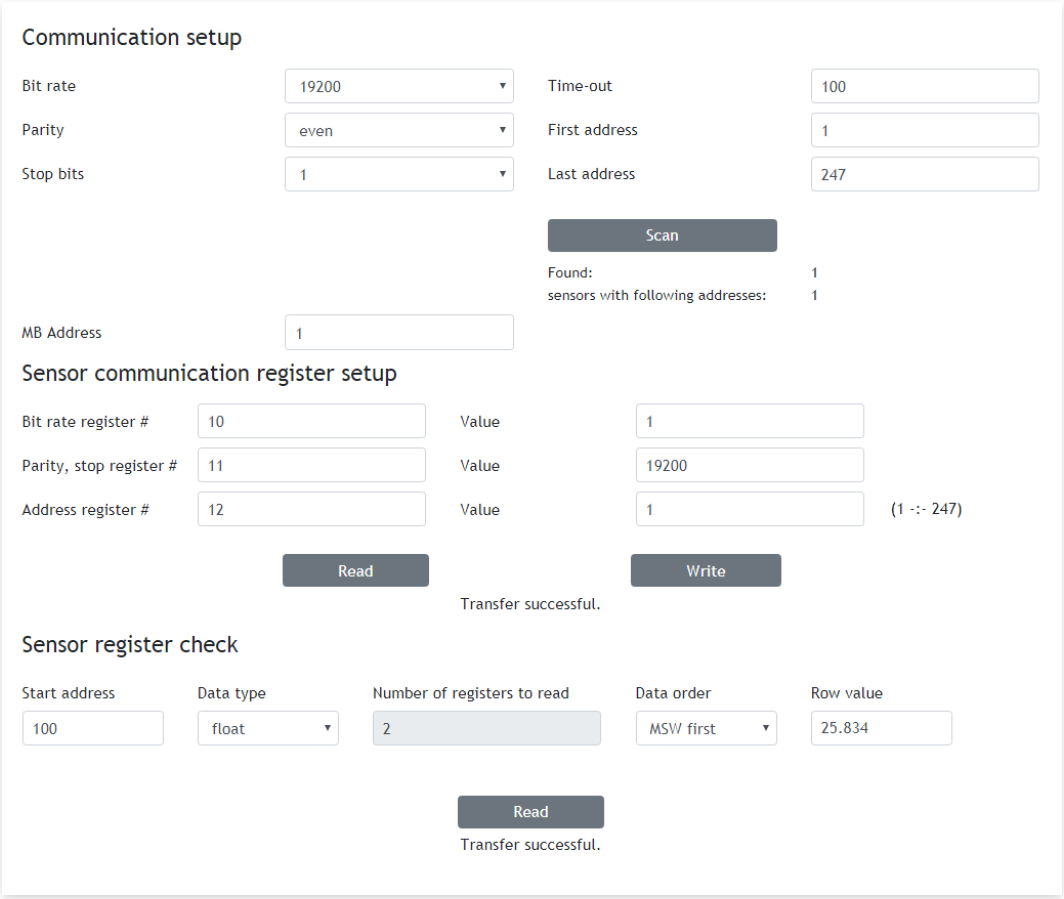
According to the above paragraph, special attention should be paid for response time-out. On the one hand, the time-out should be large enough for the sensor to respond, but on the other hand, the sum of all sensor time-outs forms the "maximum response time-out" of the whole system. The "maximum response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

For the sustainable system operation, polling of every sensor is made at a fixed time - "Polling time". It is user selectable and can be 1, 2, 3, 4, 5, 10, 15, 30, 60, 120 and 180 seconds. By default, it is 1 second.

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

8.1.2.3. Sensor setup tool

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

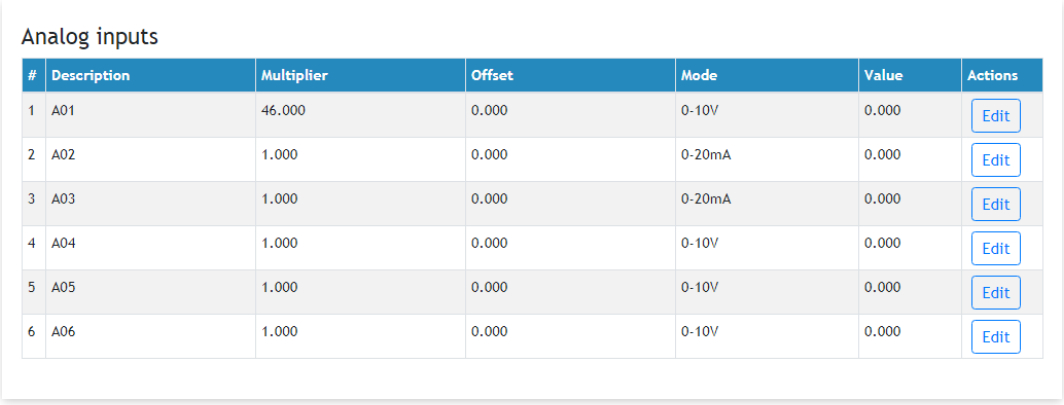


8.1.3. Inputs

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

8.1.3.1. Analog inputs

TCW260 has 6 analog inputs. All they are isolated from the power supply but use the same ground. Every analog input can work in voltage (0-10V) or current loop mode (0-20mA).



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

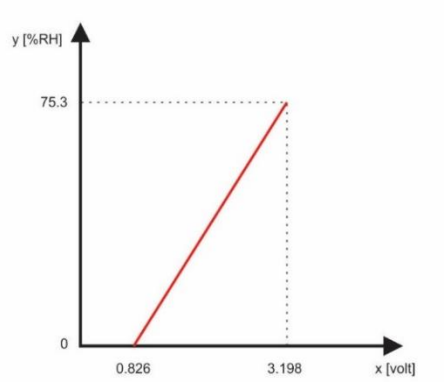
By default and after “Factory default settings” procedure: Multiplier=1.00, Offset=0.00 and Mode=0-10V.

The scaled value is calculated using the following formula:

SV = RV * MU + OF
Where:
SV – scaled (displayed) value;
RV – raw voltage from the source;
MU – multiplier;
OF – offset.

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

VOUT = 0.826 at 0% RH
VOUT = 3.198 at 75.3% RH



The sensor outputs raw voltage values, but our goal is to convert these into corresponding relative humidity (RH) values. To achieve this, we use a multiplier and an offset. These parameters enable us to calculate the relative humidity for any voltage within the sensor's operational range.

Calculation of the multiplier (MU)

The multiplier (MU) is determined by the ratio of the change in relative humidity (ΔRH%) to the change in voltage (ΔV). Geometrically, this resembles finding the slope of a line. For this sensor, the line is represented by the equation ΔRH%/ΔV. We can calculate the multiplier as follows:

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

Calculation of the offset (OF)

The offset (OF) is calculated using the multiplier and one of the known points. By substituting the scaled value (SV) and the corresponding raw value (RV) into the equation SV = RV * MU + OF, we can solve for the offset:

OF = SV - (RV * MU)

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

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

We can also calculate the offset using the other known point, where SV = 75.3 and RV = 3.198:

OF = 75.3 - (3.198 * 31.745) = 75.3 - 101.52 = -26.22

Final formula

Thus, the formula for this sensor become:

SV = RV * 31.745 - 26.22

Verification

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

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

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

8.1.3.2. Digital inputs

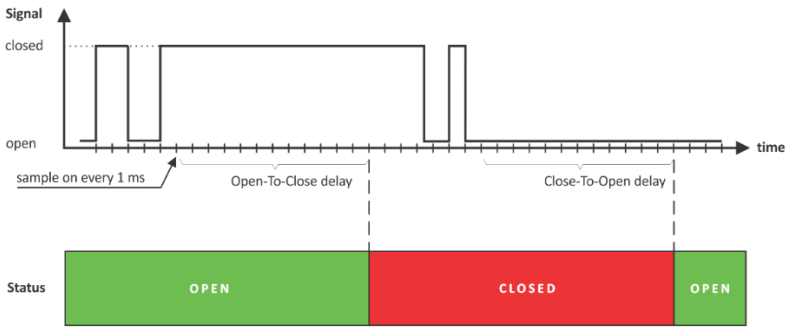
The TCW260 features four digital inputs, all of which are isolated from the power supply but share a common ground. Each digital input can operate in either OPEN/CLOSE mode or COUNTER mode. In COUNTER mode, counting can be configured to occur on rising edges, falling edges, or both.

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

Digital inputs

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

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



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

8.1.4. Channels

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

#	Description	Parameter 1	OP 1	Parameter 2	OP 2	Coefficient 1	OP3	Coefficient 2	Units	Cumulative	Actions
1	Temperature	S01-Temperature							°C		Edit Delete
2	Humidity	S02-Humidity							RH		Edit Delete
3	Digital Input 1	D1									Edit Delete
5	V05-Voltage	A01							V		Edit Delete
6	V06-Current	A02							A		Edit Delete
7	V07-Energy	A01	*	A02	*	1.000	+	0.000	kWh	✓	Edit Delete

Create

There are three types of channels:

- Discrete: Formed by a single digital input in OPEN/CLOSE mode.
- General: Formed by up to two primary parameters and constants.
- Cumulative: Similar to a general channel but with cumulative values over time. Cumulative channels are typically used for monitoring energy, volume, etc., and allow for the configuration of an initial value.

It's important to note that for both general and cumulative channels, the sequence of operations is OP1, followed by OP2, and finally OP3.

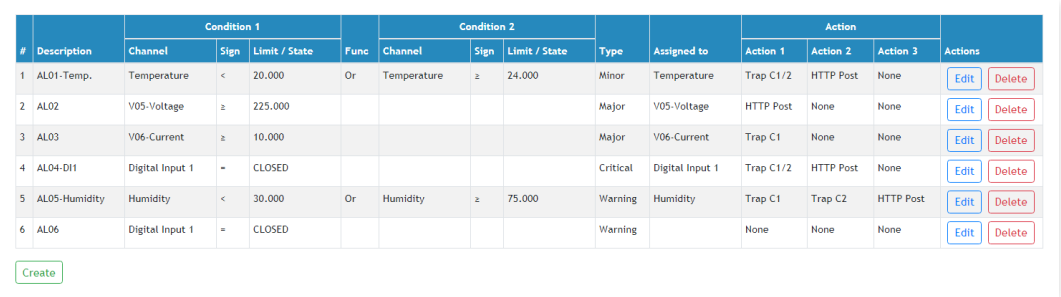
A digital input in OPEN/CLOSE mode can also be used to create a general or cumulative channel. In this case, the values for operations are 0 (for CLOSE) and 1 (for OPEN). This setup is particularly useful if you want to stop the integration for a cumulative channel using an external signal.

8.1.5. Alarms

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

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

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



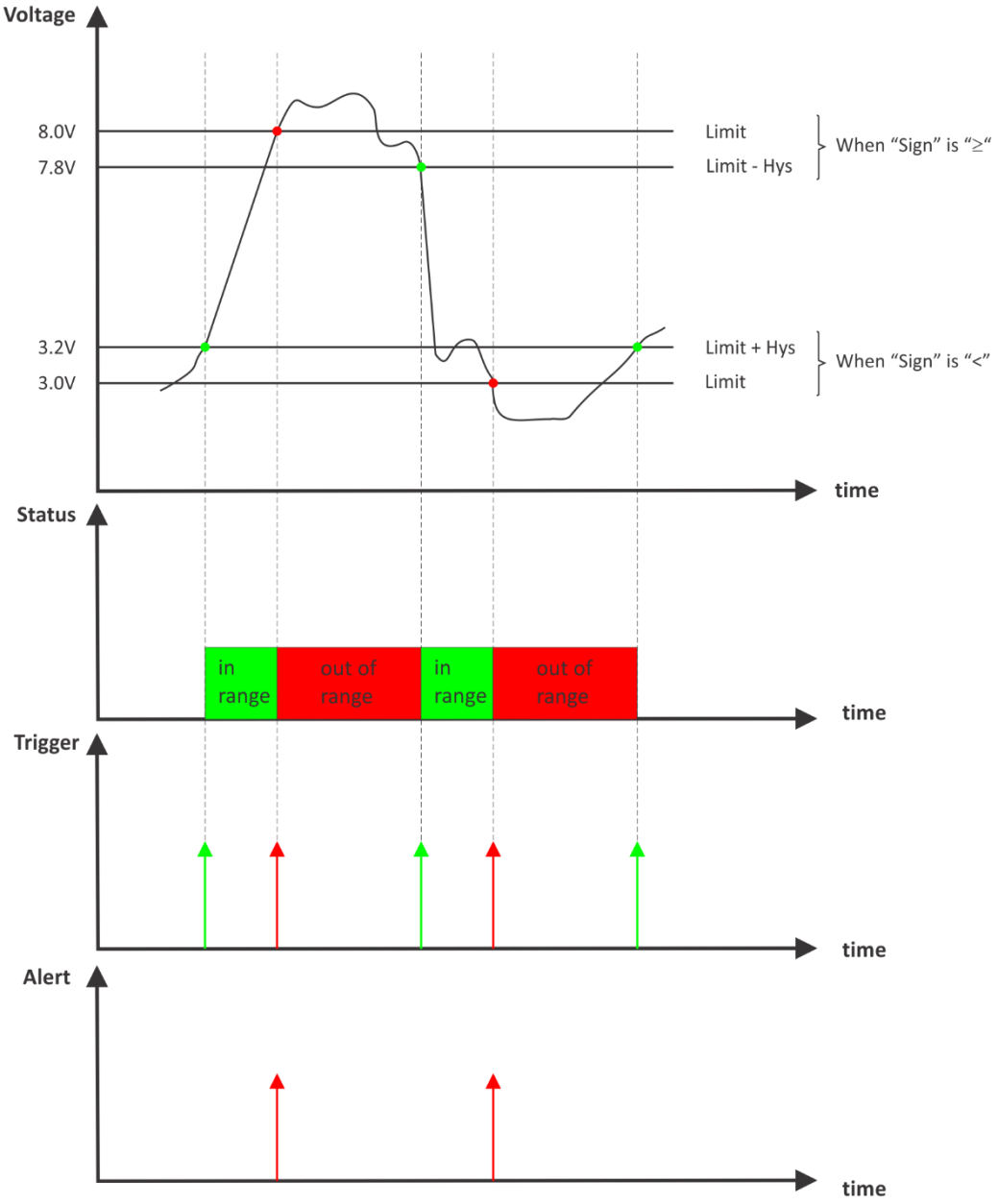
#	Description	Condition 1			Func	Condition 2			Type	Assigned to	Action			Actions
		Channel	Sign	Limit / State		Channel	Sign	Limit / State			Action 1	Action 2	Action 3	
1	AL01-Temp.	Temperature	<	20.000	Or	Temperature	≥	24.000	Minor	Temperature	Trap C1/2	HTTP Post	None	Edit Delete
2	AL02	V05-Voltage	≥	225.000					Major	V05-Voltage	HTTP Post	None	None	Edit Delete
3	AL03	V06-Current	≥	10.000					Major	V06-Current	Trap C1	None	None	Edit Delete
4	AL04-DI1	Digital Input 1	=	CLOSED					Critical	Digital Input 1	Trap C1/2	HTTP Post	None	Edit Delete
5	AL05-Humidity	Humidity	<	30.000	Or	Humidity	≥	75.000	Warning	Humidity	Trap C1	Trap C2	HTTP Post	Edit Delete
6	AL06	Digital Input 1	=	CLOSED					Warning		None	None	None	Edit Delete

Create

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

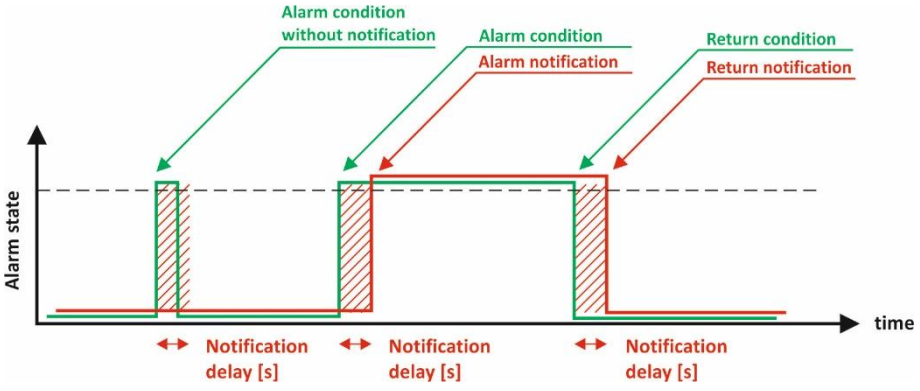
Returning to the acceptable range for the observed channel is considered when the value exceeds the Limit + Hysteresis (“Sign” is “<”) or falls below the Limit – Hysteresis (“Sign” is “≥”). Hysteresis is employed to prevent excessive triggering when the value fluctuates around the trigger point.

It is strongly recommended not to set Hysteresis to 0.0.



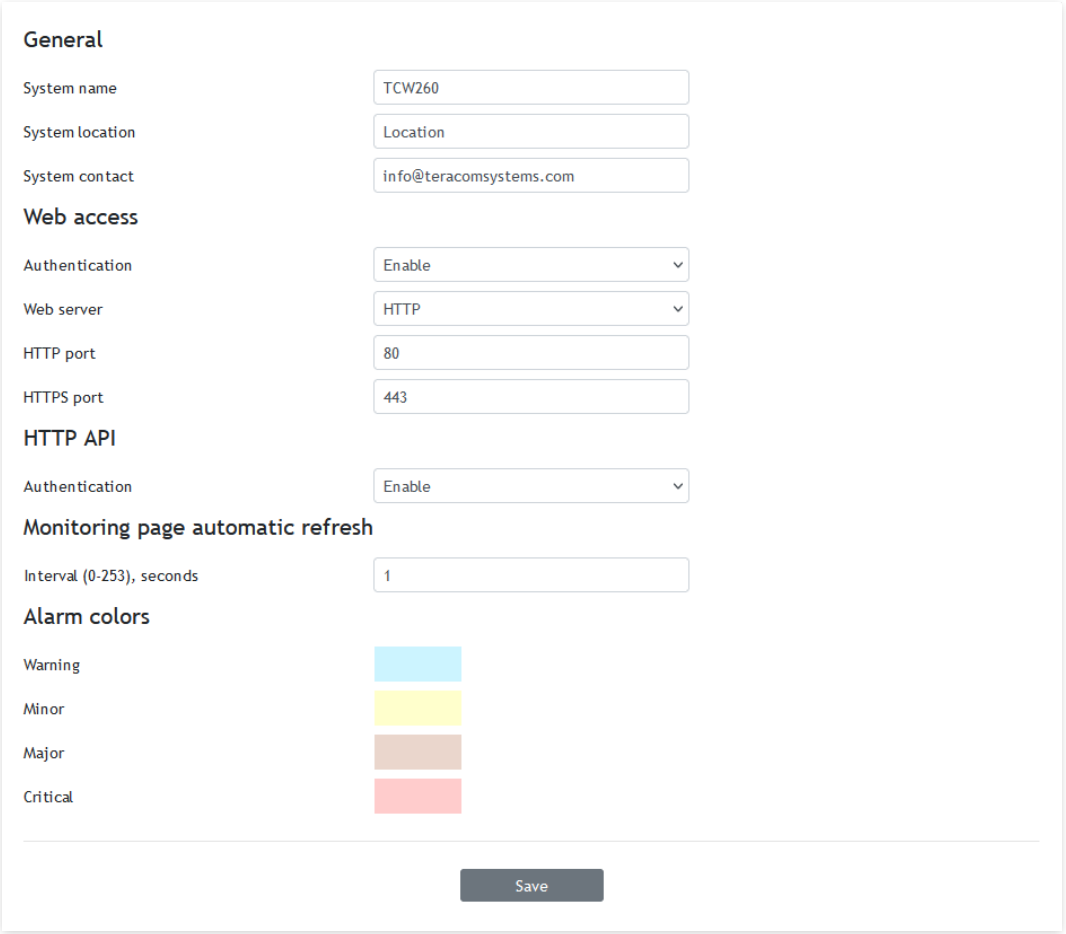
For each alarm, the "Return Notification" option is available. When this option is selected, a notification will also be sent when the parameter returns to the acceptable range.

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



8.1.6. System

This page allows you to configure various general system parameters.



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

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

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

It's important to note that the TCW260 uses a self-signed certificate for HTTPS, which may trigger a "Your connection is not private" warning in some web browsers. Additionally, the TCW260 operates with limited resources, meaning that running in HTTPS mode may be slower than in HTTP mode. For optimal performance in HTTPS mode, we recommend using Firefox due to its lower resource requirements compared to other browsers.

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

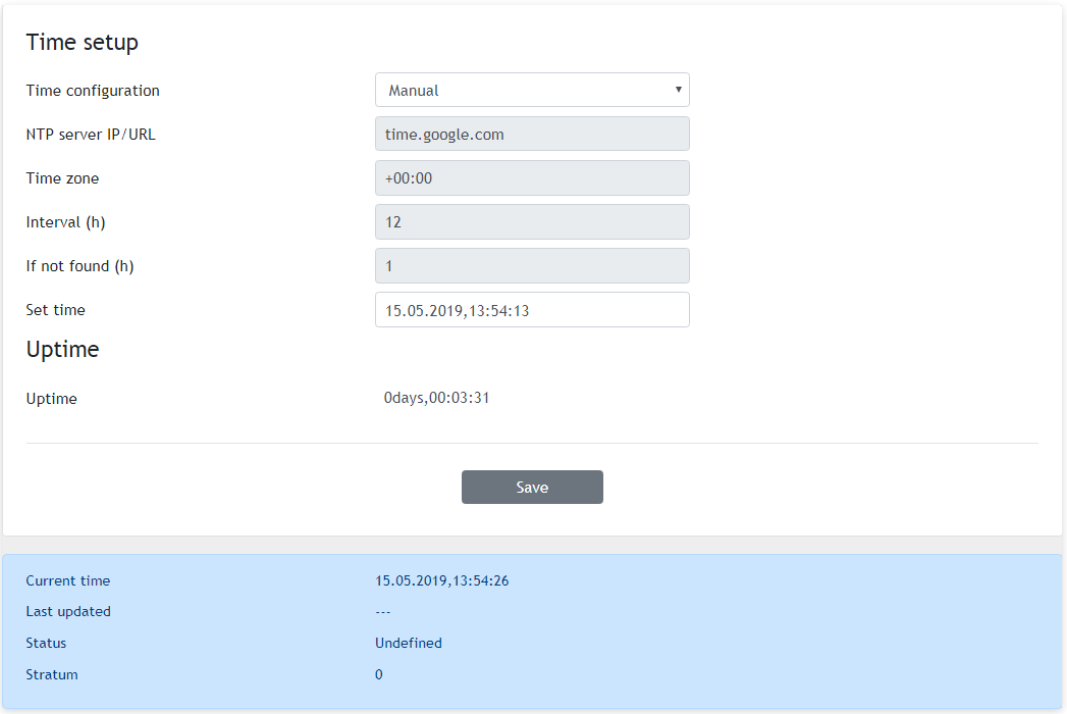
The refresh interval can be set between 0 and 253 seconds, where a setting of zero means no automatic refresh.

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

8.1. Services

8.1.1. NTP

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



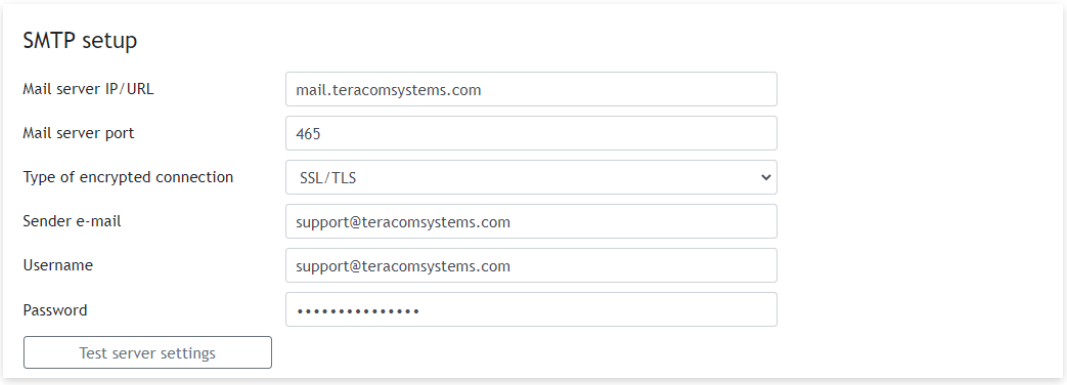
For automatic clock synchronization, the controller supports NTP (Network Time Protocol), and all necessary parameters for this synchronization are available in this section.

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

8.1.2. SMTP

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

8.1.2.1. SMTP setup



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

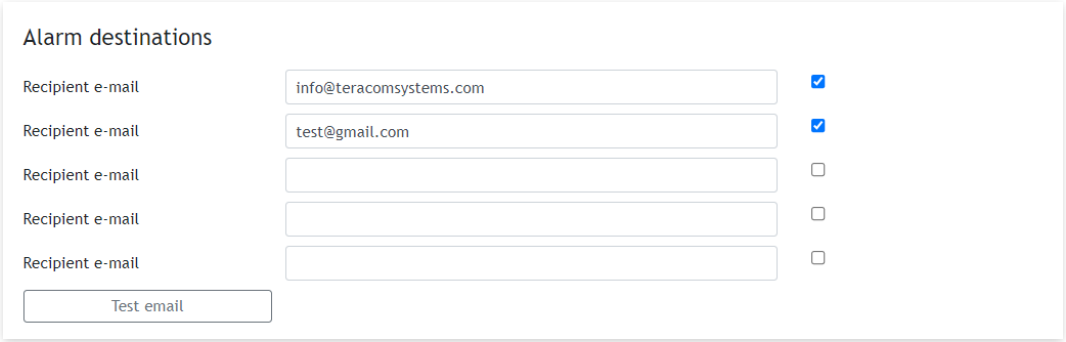
The sender's email, username, and password are standard authentication details. In the TCW260, each of these fields can accommodate up to 127 characters in length.

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

For secure communication with mail servers, the Transport Layer Security (TLS) protocol is employed. The TCW260 supports TLS versions 1.0, 1.1, and 1.2, using RSA for key exchange, agreement, and authentication. STARTTLS is also supported.

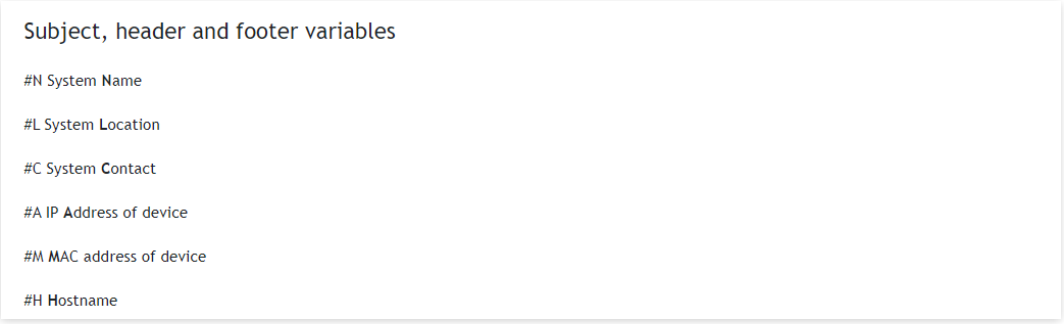
8.1.2.2. Alarm destination

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



8.1.2.3. E-mail details

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



8.1.3. SNMP

The TCW260 supports SNMPv2 and SNMPv3. The default parameters are as follows:

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

The more advanced SNMPv3 offers enhanced security management for administration, authentication, and privacy. SNMPv3 provides the following configuration options:

- No authentication and no privacy (noAuthNoPriv) - Typically used for monitoring;
- Authentication and no privacy (authNoPriv) - Usually employed for control;
- Authentication and privacy (authPriv) - Commonly used for handling sensitive data.

User-based Authentication Mechanism is based on the following:

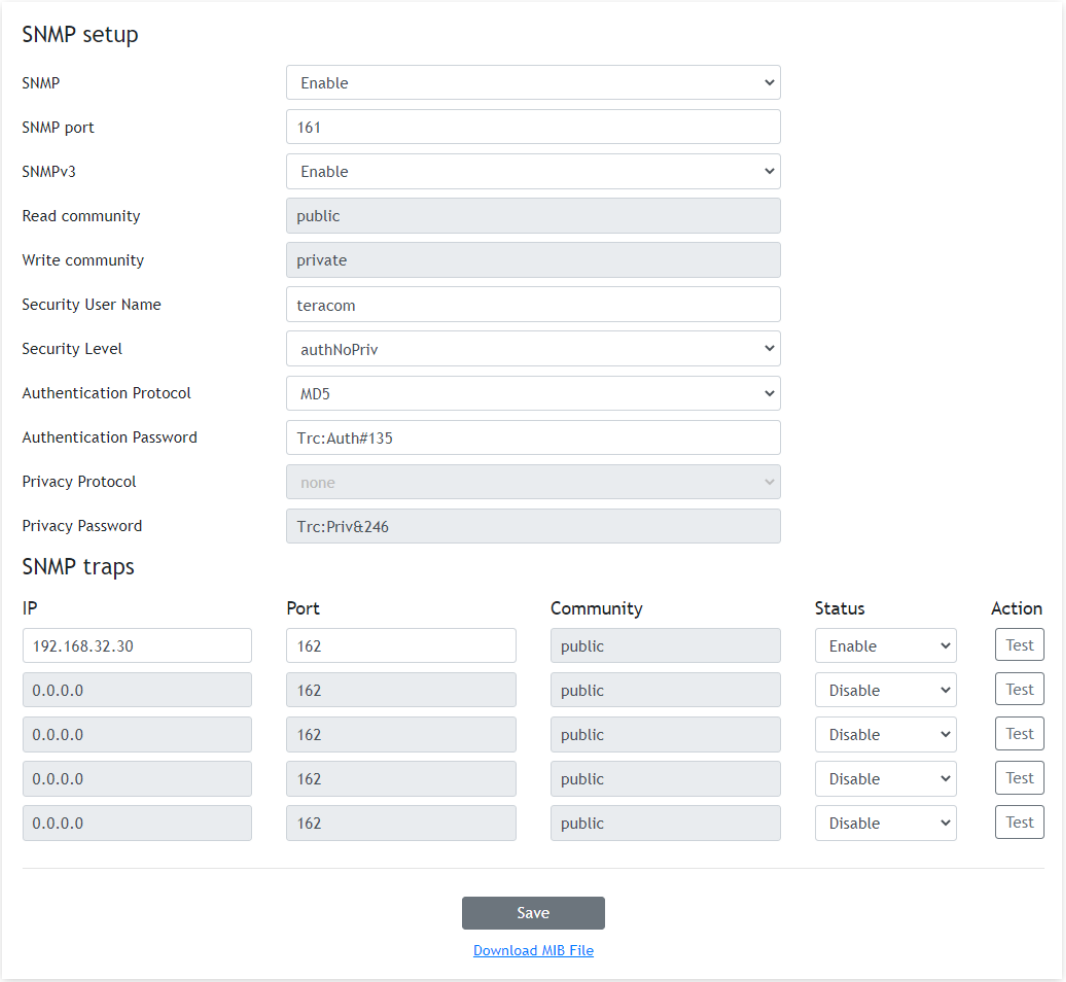
- MD5 message-digest algorithm in HMAC;
- SHA, an optional alternative algorithm;
- None authentication.

User-based Privacy Mechanism is based on the following:

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

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

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



8.1.4. Logger

The TCW260 supports data logging for all monitored channels and alarm events, allowing historical records to be stored locally and periodically uploaded to a remote server.

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

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

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

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

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

Example:

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

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

By default, the logger is disabled.

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

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

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

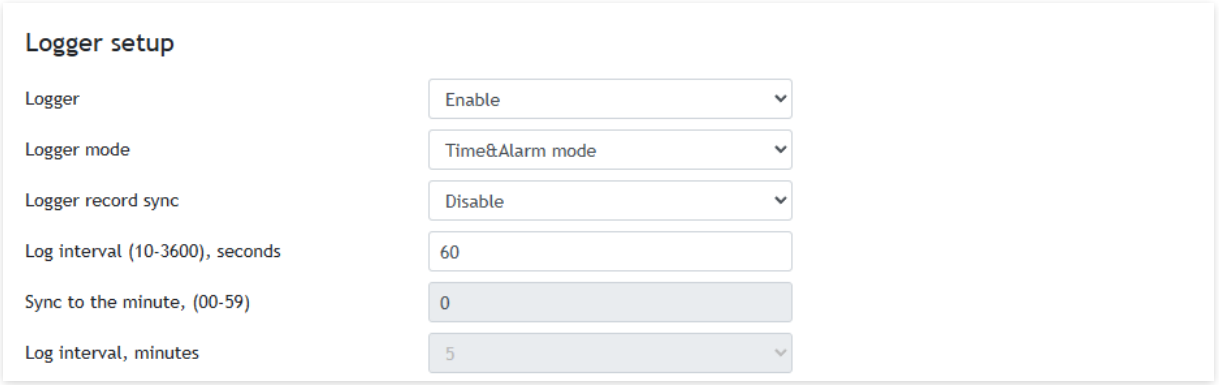
The Sync time parameter defines the exact time of day used as a reference point for periodic uploads. This ensures uploads occur at predictable intervals aligned to the configured schedule.

Example:

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

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

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



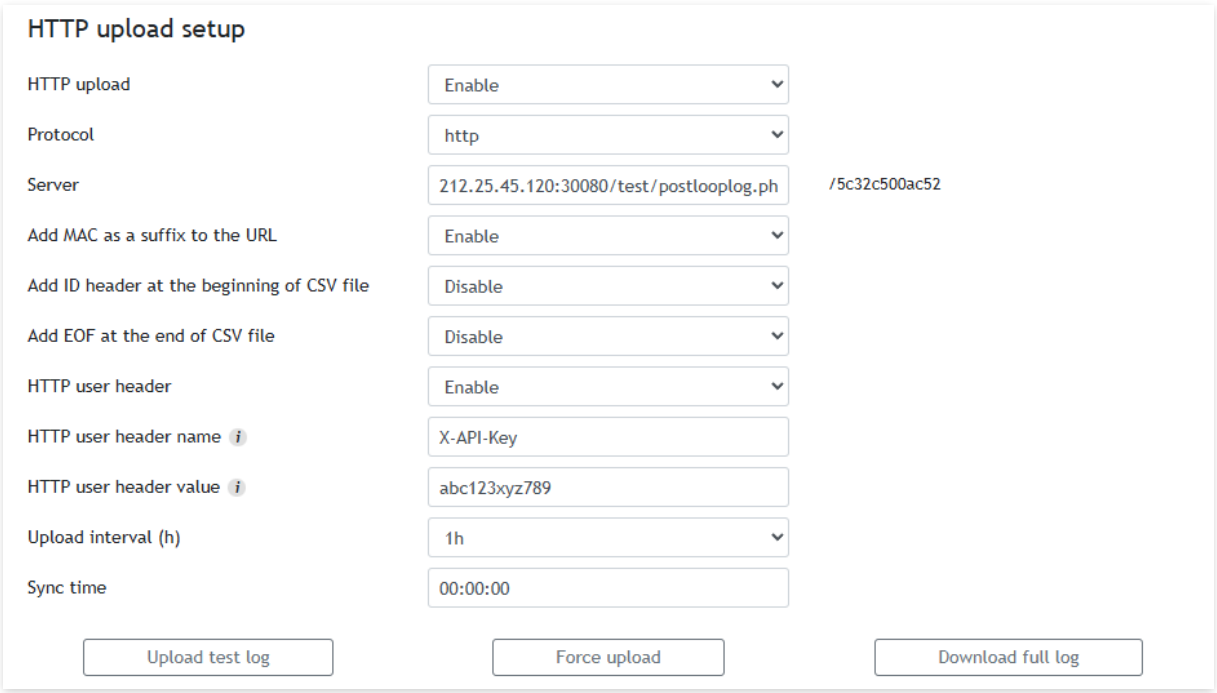
Configuration parameters:

- Logger – Enables or disables the logger functionality.
- Logger mode – Defines when data is recorded:

  Time – Records data at regular intervals.

  Alarm – Records data only when an alarm event occurs.

  Time & Alarm – Records data periodically and on alarm events.
- Logger record sync – Enables synchronization of log records to the configured time reference.
- Log interval (seconds) – Defines the interval between log records. Adjustable from 10 to 3600 seconds.



HTTP Upload configuration parameters:

- Sync to the minute – Defines the minute (00–59) used for synchronized logging.
- Log interval, minutes – Defines the interval between synchronized records when logger record sync is enabled.
- HTTP upload – Enables or disables periodic HTTP/HTTPS upload of logger records.
- Protocol – Selects the communication protocol for uploads (HTTP/HTTPS).

- Server – Specifies the destination server address (IP address, domain name, and optional port) where CSV log files will be uploaded.
- Add MAC as a suffix to the URL – Appends the device MAC address to the configured server URL for unique identification.
- Add ID header at the beginning of CSV file – Adds a header row containing parameter identifiers for easier automated processing.
- Add EOF at the end of CSV file – Appends an end-of-file marker for compatibility with server-side parsers.
- HTTP user header – Enables or disables sending a custom HTTP header with the upload request.
- HTTP user header name – Defines the custom HTTP header field name.
- HTTP user header value – Defines the custom HTTP header field value.
- Upload interval (h) – Defines how often stored records are uploaded to the server.
- Sync time – Defines the daily reference time for upload synchronization.
- Upload test log – Sends a test CSV file to verify server connectivity and upload configuration.
- Force upload – Immediately uploads all unsent logger records to the configured server.
- Download full log – Downloads the complete logger file from the device for local storage or analysis.

8.1.5. HTTP POST

The TCW260 supports automatic data transmission to remote servers through HTTP or HTTPS POST requests, enabling integration with cloud platforms, SCADA systems, monitoring software, and custom backend applications.

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

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

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

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

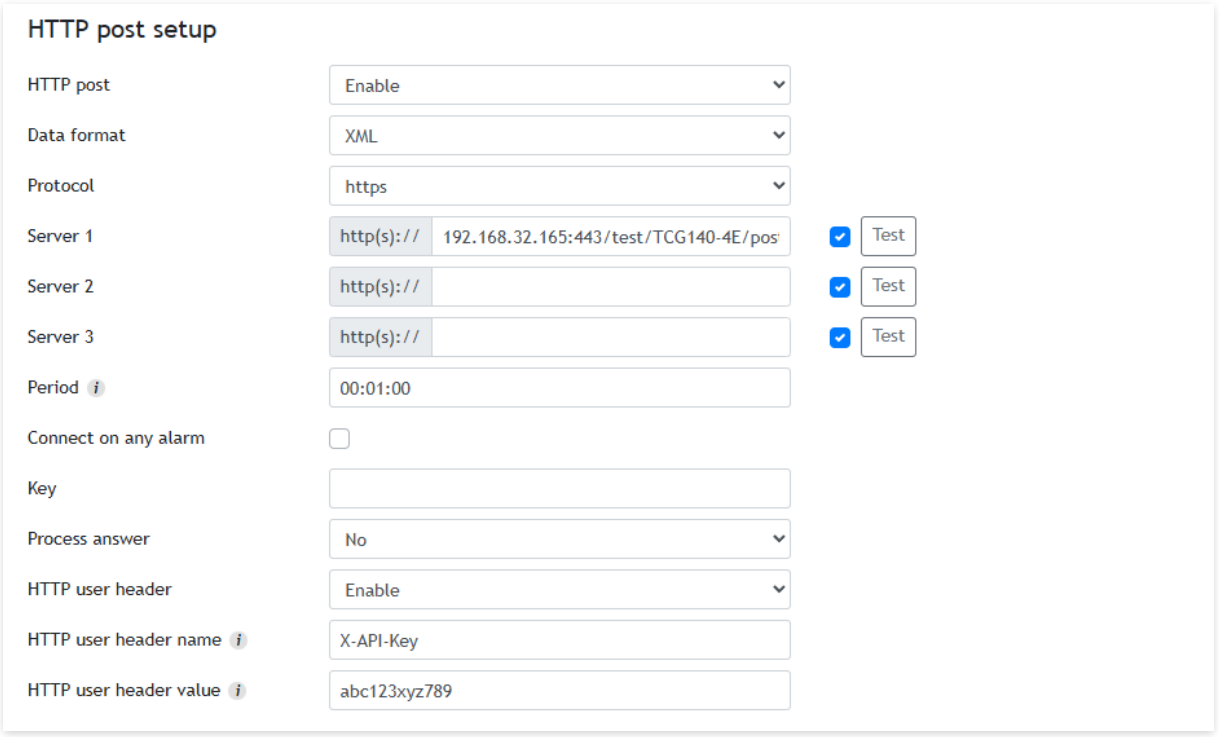
If the Connect on any alarm option is enabled, the device will immediately send an HTTP/HTTPS POST request whenever an alarm condition occurs, regardless of the configured periodic transmission interval.

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

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

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

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



Configuration parameters:

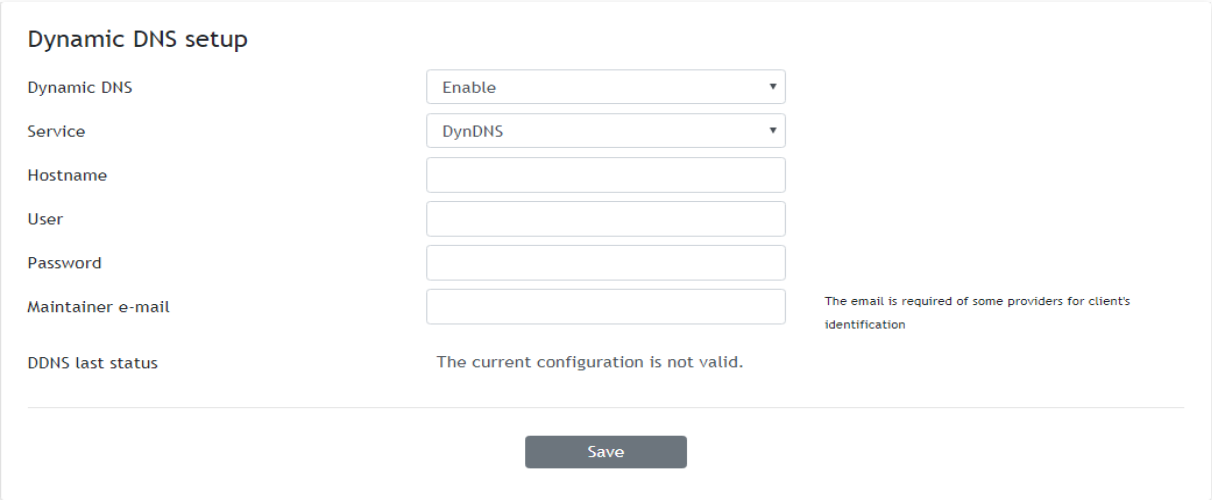
- HTTP POST – Enables or disables the HTTP POST functionality.
- Data format – Defines the payload structure sent to the server (XML or JSON).
- Protocol – Selects the communication protocol (HTTP or HTTPS).
- Server 1 – Specifies the primary destination server address (IP address, domain name, and optional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

Test – Sends a test HTTP/HTTPS POST request to the selected server to verify connectivity and configuration.

8.1.6. Dynamic DNS

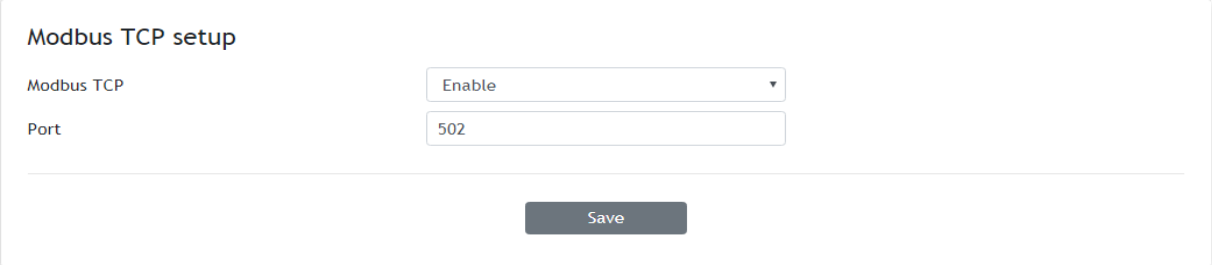
With dynamic DNS, the TCW260 can be accessed from the public Internet without the need for a broadband account with a static IP address.

TCW260 supports the following DNS services: DynDNS, No-IP, and DNS-O-Matric.



8.1.7. Modbus TCP/IP

TCW260 supports Modbus TCP/IP via the Ethernet interface.



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

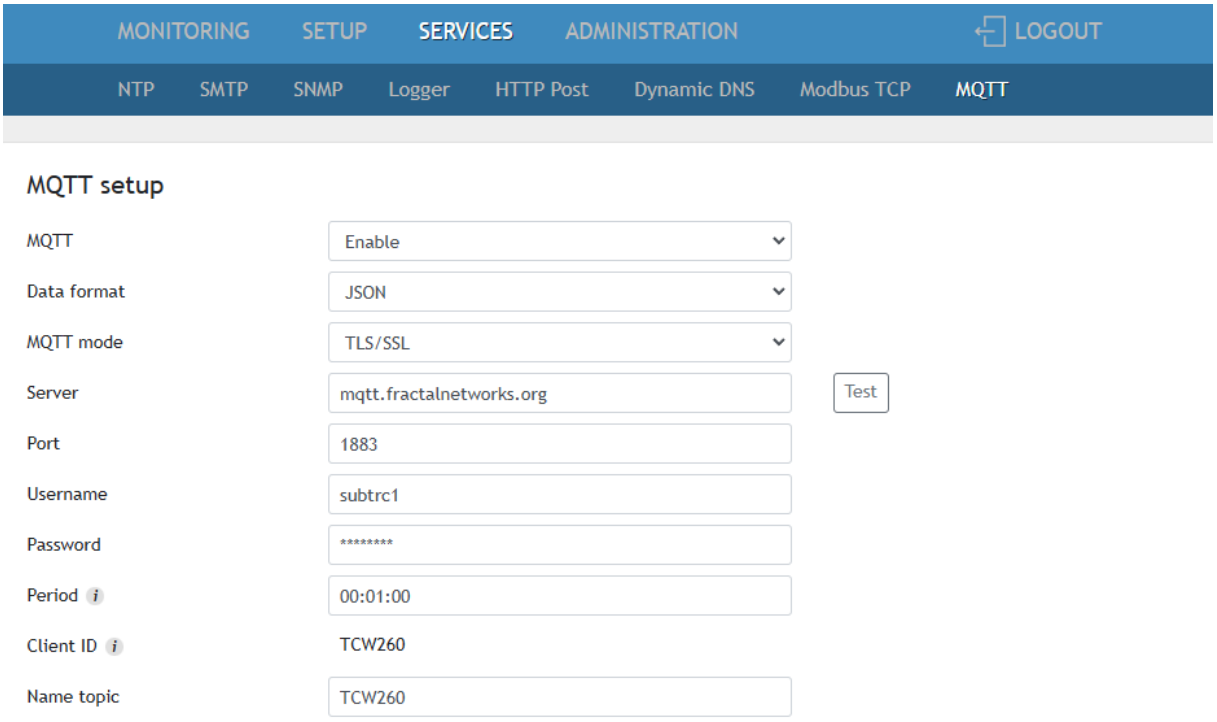
A table containing the register addresses can be found in section " Modbus TCP/IP”.

8.1.8. MQTT

The TCW260 supports MQTT version 3.1.1 exclusively in "Publisher" mode. This section provides the settings for configuring the MQTT protocol on the device.

8.1.8.1. MQTT General setup

This section allows configuration of the general parameters for MQTT operation. By default, the MQTT service is disabled. After enabling it, the controller establishes a connection with the MQTT broker, whose parameters are configured in the fields “Server”, “Port”, “Username”, and “Password”. The “Username”, “Password”, and “Server name” field supports up to 127 characters. The publishing interval ranges from 1 minute to 48 hours.



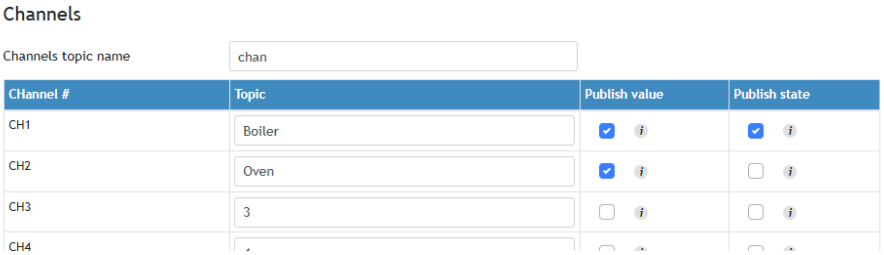
Messages can be sent to the MQTT broker in JSON or plain text format, with an option for an encrypted (TLS) or unencrypted connection.

In the "Topic name" field, the root topic is set, in which the MQTT publish service will operate.

Note: The System ID is used as the Client ID.

8.1.8.2. MQTT Channels topics

In this section, the publishing parameters for each channel are configured.

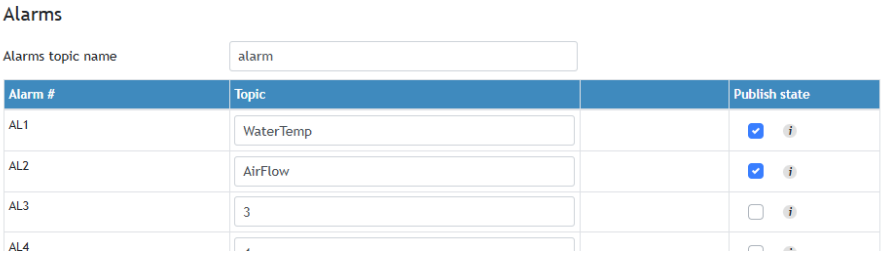


It is possible to monitor all 24 channels, with each channel's value and status available for publishing. With these settings, the exact topics for values and statuses will be:

- TCW260/chan/Boiler/value
- TCW260/chan/Boiler/state
- TCW260/chan/Oven/value

8.1.8.3. MQTT Alarms topics

In this section, the publishing parameters for each alarm are configured.



With these settings, the exact topics for alarm statuses will be:

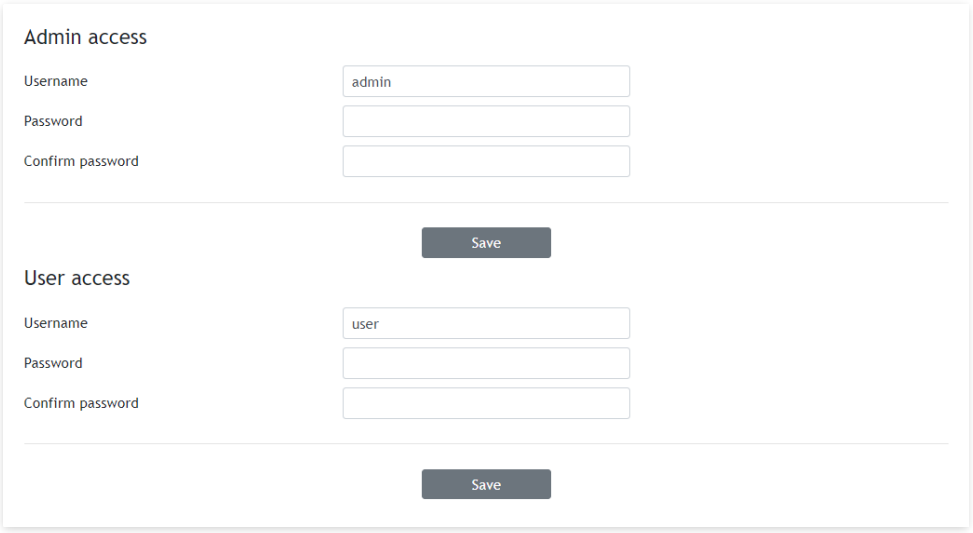
- TCW260/alarm/WaterTemp/state
- TCW260/alarm/AirFlow/state

It is possible to monitor all 24 alarms, with the status of each alarm available for publishing.

8.1. Administration

8.1.1. User/Pass

The TCW260 supports two user roles: "Admin" and "User".



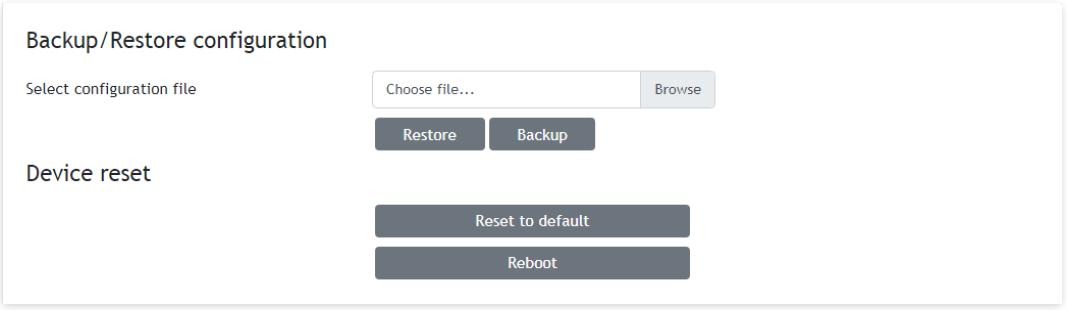
The Admin role has full administrative rights and can modify all settings.

The User role has limited access, allowing only viewing of the Monitoring page and preventing any modifications to settings.

Both usernames and passwords can be up to 31 characters in length.

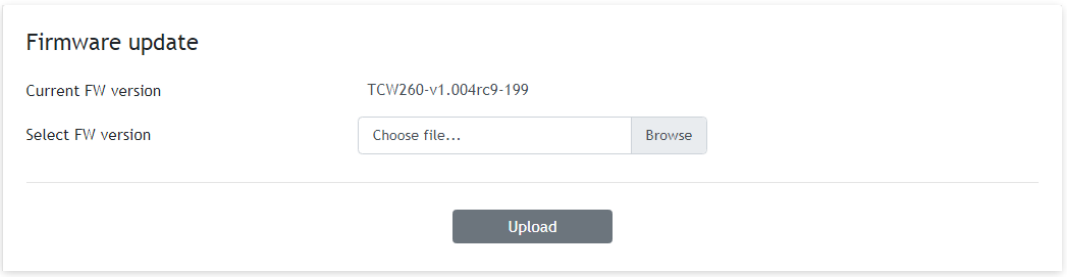
8.1.2. Backup/Restore

The TCW260 allows for the backup and restoration of all user settings. These settings are saved in an XML backup file, which can be utilized for restoring configurations on multiple devices. This feature is particularly useful for applying similar settings across a batch of controllers.



8.1.3. FW update

The TCW260 can be updated through the web interface.



To update the device, follow these steps:

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

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

8.2. Logout

The TCW260 supports multi-session access; however, it is recommended to log out after completing your tasks for security and best practices.

9. Protocols and API

9.1. SNMP

The TCW260 supports SNMP (Simple Network Management Protocol), a standard protocol for managing IP network devices, enabling administrative computers (managers) to monitor and control these devices. The device can be configured and monitored using an SNMP v.2 or v.3 compatible program, with parameters organized by function. To obtain a valid OID number, replace "!" with "1.3.6.1.4.1.38783". To save configuration changes, set configurationSaved (OID !.8.6.3.0) to "1".

product

OID	Name	Access	Description	Syntax
!.6.1.1.0	name	read-only	Device name	DisplayString
!.6.1.2.0	version	read-only	Firmware version	DisplayString
!.6.1.3.0	date	read-only	Release date	DisplayString

setup -> network

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

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

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

OID	Name	Access	Description	Syntax
!.6.2.2.1.1.1.2.?.0	mbSenDescription.?	read-write	Sensor description	DisplayString
!.6.2.2.1.1.1.3.?.0	mbSenMult.?	read-write	Sensor multiplier x1000 in Integer format	Integer32
!.6.2.2.1.1.1.4.?.0	mbSenOffset.?	read-write	Sensor offset x1000 in Integer format	Integer32
!.6.2.2.1.1.1.5.?.0	mbSenVal.?	read-only	Sensor value x1000 in Integer format	Integer32
!.6.2.2.1.1.1.6.?.0	mbSenCounter.?	read-only	Sensor as 32-bit Counter	Counter32

setup -> parameters -> analogInputs -> analogInpTable -> analogInpEntry -> analogInpIndex 1 to 6

replace "**?**" with a number from 1 to 6 in the table below

OID	Name	Access	Description	Syntax
!.6.2.2.2.1.1.2.?.0	analogInpDescription.?	read-write	Analog input description	DisplayString
!.6.2.2.2.1.1.3.?.0	analogInpMult.?	read-write	Analog input multiplier x1000 in Integer format	Integer32
!.6.2.2.2.1.1.4.?.0	analogInpOffset.?	read-write	Analog input offset x1000 in Integer format	Integer32
!.6.2.2.2.1.1.5.?.0	analogInpMode.?	read-write	Analog input mode - 0-10V or 4-20mA	Integer32
!.6.2.2.2.1.1.6.?.0	analogInpValue.?	read-only	Analog input value x1000 in Integer format	Integer32

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

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

OID	Name	Access	Description	Syntax
!.6.2.2.3.1.1.2.?.0	digInpDescription.?	read-write	Digital Input description	DisplayString
!.6.2.2.3.1.1.3.?.0	digInplowLevel.?	read-write	Digital Input closed state	DisplayString
!.6.2.2.3.1.1.4.?.0	digInphighLevel.?	read-write	Digital Input open state	DisplayString
!.6.2.2.3.1.1.5.?.0	digInpMode.?	read-write	Digital Input mode - Discrete or Counter	INTEGER { openclosed(0), risingEdge(1), fallingEdge(2), bothEdges(3) }
!.6.2.2.3.1.1.6.?.0	digInpCloseToOpenDelay.?	read-write	Digital input Close To Open delay	Integer32(0..60000)
!.6.2.2.3.1.1.7.?.0	digInpOpenToCloseDelay.?	read-write	Digital input Open To Close delay	Integer32(0..60000)
!.6.2.2.3.1.1.8.?.0	digInpCounterInitValue.?	read-only	Digital input counter initial value	Integer32
!.6.2.2.3.1.1.9.?.0	digInpValue.?	read-only	Digital input value	Unsigned32

monitorNcontrol -> channels -> chanTable -> chanEntry -> chIndex 1 to 24

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

OID	Name	Access	Description	Syntax
!.6.3.1.1.1.2.?.0	chType.?	read-write	Channel type	INTEGER {general(0), cumulative(1), discrete(2), counter(3)}
!.6.3.1.1.1.3.?.0	chdescription.?	read-write	Channel description	DisplayString
!.6.3.1.1.1.4.?.0	chParam1.?	read-write	Channel parameter 1	INTEGER {none(0), s01(3), s02(4), s03(5), s04(6), s05(7), s06(8), s07(9), s08(10),s09(11), s10(12), s11(13), s12(14), s13(15), s14(16), s15(17), s16(18), s17(19), s18(20), s19(21), s20(22), s21(23), s22(24), s23(25), s24(26), a01(27), a02(28), a03(29), a04(30), a05(31), a06(32), d01(33), d02(34), d03(35), d04(36)}
!.6.3.1.1.1.5.?.0	chOP1.?	read-write	Channel operand 1	INTEGER{none(0), multiplication(1), division(2), sum(3), subtract(4)}
!.6.3.1.1.1.6.?.0	chParam2.?	read-write	Channel parameter 2	INTEGER {none(0), one(1), null(2), s01(3), s02(4), s03(5), s04(6), s05(7), s06(8), s07(9), s08(10),s09(11), s10(12), s11(13), s12(14), s13(15), s14(16), s15(17), s16(18), s17(19), s18(20), s19(21), s20(22), s21(23), s22(24), s23(25), s24(26),a01(27), a02(28), a03(29), a04(30), a05(31), a06(32),d01(33), d02(34), d03(35), d04(36)}
!.6.3.1.1.1.7.?.0	chOP2.?	read-write	Channel operand 2	INTEGER {none(0), multiplication(1), division(2), sum(3), subtract(4)}

!.6.3.1.1.1.8.?.0	chCoef1.?	read-write	Channel coefficient 1 x1000 in Integer format	Integer32
!.6.3.1.1.1.9.?.0	chOP3.?	read-write	Channel operand 3	INTEGER {none(0), multiplication(1), division(2), sum(3), subtract(4)}
!.6.3.1.1.1.10.?.0	chCoef2.?	read-write	Channel coefficient 2 x1000 in Integer format	Integer32
!.6.3.1.1.1.11.?.0	chUnit.?	read-write	Channel unit	DisplayString
!.6.3.1.1.1.12.?.0	chCumulInitValue.?	read-write	Channel cumulative initial value	Integer32
!.6.3.1.1.1.13.?.0	chValue.?	read-only	Channel value x1000 in Integer format	Integer32
!.6.3.1.1.1.14.?.0	chCounter.?	read-only	Channel as 32-bit counter	Counter32
!.6.3.1.1.1.15.?.0	chAlarmStatus.?	read-only	Channel alarm status	INTEGER {undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)}

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

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

OID	Name	Access	Description	Syntax
!.6.3.2.1.1.2.?.0	alDescription.?	read-write	Alarm description	DisplayString
!.6.3.2.1.1.3.?.0	alCond1Channel.?	read-write	Alarm condition 1 channel	INTEGER {none(0), v01(1), v02(2), v03(3), v04(4), v05(5), v06(6), v07(7), v08(8), v09(9), v10(10), v11(11), v12(12), v13(13), v14(14), v15(15), v16(16), v17(17), v18(18), v19(19), v20(20), v21(21), v22(22), v23(23), v24(24)}
!.6.3.2.1.1.4.?.0	alCond1Operand.?	read-write	Alarm condition 1 operand	INTEGER{larger(1), less(2)}
!.6.3.2.1.1.5.?.0	alCond1Limit.?	read-write	Alarm condition 1 limit x1000 in Integer format	Integer32
!.6.3.2.1.1.6.?.0	alCond1Hys.?	read-write	Alarm condition 1 hysteresis x1000 in Integer format	Integer32
!.6.3.2.1.1.7.?.0	alCond1AlarmState.?	read-write	Alarm condition 1 discrete alarm state	INTEGER {open(0), closed(1)}
!.6.3.2.1.1.8.?.0	alCondLogic.?	read-write	Alarm conditions logic	INTEGER{none(0), and(1), or(2)}
!.6.3.2.1.1.9.?.0	alCond2Channel.?	read-write	Alarm condition 2 channel	INTEGER {none(0), v01(1), v02(2), v03(3), v04(4), v05(5), v06(6), v07(7), v08(8), v09(9), v10(10), v11(11), v12(12), v13(13), v14(14), v15(15), v16(16), v17(17), v18(18), v19(19), v20(20), v21(21), v22(22), v23(23), v24(24)}
!.6.3.2.1.1.10.?.0	alCond2Operand.?	read-write	Alarm condition 2 operand	INTEGER{larger(1), less(2)}
!.6.3.2.1.1.11.?.0	alCond2Limit.?	read-write	Alarm condition 2 limit x1000 in Integer format	Integer32
!.6.3.2.1.1.12.?.0	alCond2Hys.?	read-write	Alarm condition 2 hysteresis x1000 in Integer format	Integer32
!.6.3.2.1.1.13.?.0	alCond2AlarmState.?	read-write	Alarm condition 2 discrete alarm state	INTEGER {open(0), closed(1)}
!.6.3.2.1.1.14.?.0	alType.?	read-write	Alarm type	INTEGER {warning(3), minor(4), major(5), critical(6)}
!.6.3.2.1.1.15.?.0	alAssigned.?	read-write	Alarm assigned to	INTEGER {none(0), v01(1), v02(2), v03(3), v04(4), v05(5), v06(6), v07(7), v08(8),v09(9), v10(10), v11(11), v12(12), v13(13), v14(14), v15(15),

				v16(16),v17(17), v18(18), v19(19), v20(20), v21(21), v22(22), v23(23), v24(24)}
!.6.3.2.1.1.16.?.0	alActionDelay.?	read-write	Alarm action delay	Integer32
!.6.3.2.1.1.17.?.0	alActionOnReturn.?	read-write	Alarm action on return	INTEGER {no(0), yes(1)}
!.6.3.2.1.1.18.?.0	alAction1.?	read-write	Alarm action 1	INTEGER {none(0), trapcond1(1), trapcond2(2), trapcond1and2(3), postiostate(4) , mqttpublish(6)}
!.6.3.2.1.1.19.?.0	alAction2.?	read-write	Alarm action 2	INTEGER {none(0), trapcond1(1), trapcond2(2), trapcond1and2(3), postiostate(4), mqttpublish(6)}

!.6.3.2.1.1.20.?.0	alAction3.?	read-write	Alarm action 3	INTEGER {none(0), trapcond1(1), trapcond2(2), trapcond1and2(3), postiostate(4) , mqttpublish(6)}
!.6.3.2.1.1.21.?.0	alStatus.?	read-write	Alarm status	INTEGER {undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)}

monitorNcontrol

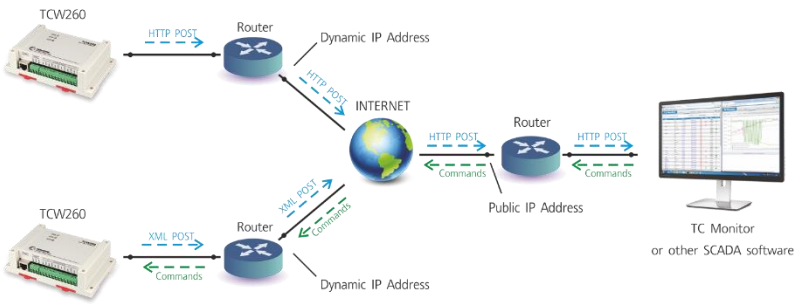
OID	Name	Access	Description	Syntax
!.6.3.3.0	configurationSaved	read-write	Configuration save status SAVED/UNSAVED	INTEGER { unsaved(0), saved(1) }
!.6.3.4.0	restartDevice	read-write	Restart Device	INTEGER { cancel(0), restart(1) }
!.6.3.5.0	hardwareErr	read-only	Hardware Error	INTEGER { noErr(0), hwErr(1)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

9.2. HTTP API

9.2.1. HTTP POST

TCW260 can execute HTTP/HTTPS POST requests to upload XML or JSON files to a dedicated server. This functionality is particularly useful when the controller is situated behind a router without a public IP address or when the user lacks access to the router's configuration. It is important to note that the server should possess a public IP address.

The typical monitoring application is shown in the picture below:



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

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

- Save following code as post.php:

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

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

- Upload the post.php file to a public web server that supports PHP. To ensure the script is functioning correctly, enter the URL (for example, www.yourserverURL.com/post.php) in your web browser. If successful, a webpage displaying “The PHP script is working!” will appear.
- Configure the TCW260 controller to send HTTP/HTTPS POST requests to your web server. Input the URL (yourserverURL.com/post.php) in the URL field, and click the “Test HTTP POST” button.
- If the HTTP/HTTPS POST request is received and processed successfully, “OK” will be displayed next to the button. Additionally, an XML file will be created in the same directory where post.php is located. The filename will include a timestamp, formatted as 20190420103318_status.xml.

9.2.2. HTTP/HTTPS GET

HTTP/HTTPS GET can be utilized to monitor the TCW260 by retrieving XML or JSON files. The format for accessing the status is as follows:

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

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

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

HTTP/HTTPS GET requests can be sent at any time to the TCW260 if it is on the same network or has appropriate routing.

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

9.2.2.1. Commands

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

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

Where:

yyy is the command;

xxx is the parameter.

Example:

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

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

9.2.2.2. HTTP/HTTPS GET authentication

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

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

Example:

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

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

9.2.3. List of HTTP API commands

Command	Description
dataf=x	Data format XML/JSON for HHTP POST – 0 XML, 1 JSON
pushtls=x	HTTP/HTTPS protocol, where x is 0 for HTTP and 1 for HTTPS
purl1=yyy	URL for HTTP POST to Server 1, where yyy is a full path to php file. Example: purl=212.25.45.120:30181/xampp/test/posttest1.php
purl2=yyy	URL for HTTP POST to Server 1, where yyy is a full path to php file. Example: purl=212.25.45.120:30181/xampp/test/posttest2.php
purl3=yyy	URL for HTTP POST to Server 1, where yyy is a full path to php file. Example: purl=212.25.45.120:30181/xampp/test/posttest3.php
pper=x	HTTP POST period in seconds (x is between 10 and 14400)
dk=xxx	HTTP POST key – xxx is up to 127 characters
mdata=x	Data format JSON/Plain text for MQTT Publish – 0 JSON, 1 Plain text
mmode=x	Publish protocol, where x is 0 for unsecure and 1 for TLS/SSL
muser=xxxx	Username authentication for MQTT, where xxxx is a username
mpass=xxxx	Password authentication for MQTT, where xxxx is a password
murl=yyy	URL for MQTT publish, where yyy is a path murl=212.25.45.120
mport=yyyy	Port for MQTT publish, where yyyy is a port mport=1883

mper=x	MQTT publish period in seconds (x is between 60 and 172800) mper=600 – will set MQTT publish period to 600 seconds
save	Save all previous changes (except relays' one) in theFLASH memory. **As every save reflects the FLASH cycles (endurance), this command should be used very carefully.** pper=120&save – will set POST period to 120 seconds and save it
FIN	Terminate session (it works with HTTP/HTTPS POST, but not with HTTP GET)

9.3. Modbus TCP/IP

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

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

9.3.1. Codes and answers

9.3.1.1. Read Discrete Inputs (FC=02)

Request
This command is requesting the content of 4 discrete inputs

02 0064 0004
02: The Function Code 2 (read Discrete Inputs)
0064: The Data Address of the first register requested (0064 hex = 100)
0004: The total number of discrete inputs requested

Response
02 01 0C
02: The Function Code 2 (read Discrete Inputs)
01: The number of data bytes to follow
0C: Inputs status (0x0000 1100)
In the example above, the value of DI4=1, DI3=1, DI2=0, DI1=0.

9.3.1.2. Read Holding Registers (FC=03)

Request
This command is requesting the content of holding registers 19300 – Channel 1 value.
03 4B64 0002
03: The Function Code 3 (read Holding Registers)
4B64: The Data Address of the first register requested (4B64 hex = 19300)
0002: The total number of registers requested (read 2 registers each 2 bytes = 4 bytes)

Response

03 04 41DD 4210

03: The Function Code 3 (read Channel 1 Holding Registers)
04: The number of data bytes to follow (2 registers x 2 bytes each = 4 bytes)
41DD 4210: 4 bytes value
All holding registers with float value are sent in big-endian.
In the example above, the value of 27.6572571 is sent.

Request

This command is requesting the content of holding registers 18100 – Channel 1 description.

03 46B4 0008

03: The Function Code 3 (read Holding Registers)
46B4: The Data Address of the first register requested (46B4 hex = 18100)
0008: The total number of registers requested (read 8 registers each 2 bytes = 16 bytes)

Response

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

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

9.3.1.3. Write Single Register (FC=06)

Request

This command is write value in a single register with address 18300 – Channel 1 parameter 1, where s01(sensor 1) is selected.

06 477C 0003

06: The Function Code 6 (write Single register)
477C: The Data Address of the register (477C hex = 18300)
0003: The value to be write – 3(s01).

Response

06 477C 0003

06: The Function Code 6 (write Single register)
477C: The Data Address of the register (477C hex = 18300)
0003: The value to be write – 3(s01).

9.3.1.4. Write Multiple Registers (FC=16)

Request

This command is write a value in contiguous registers starting at address 18300 – Channel 1 parameter 1 and Channel 2 parameter 1.

10 477C 0002 04 0003 0004

10: The Function Code 16(10 hex)
477C: Starting Address of the firsrt register(477C hex = 18300)
0002: Quantity of registers
04: Byte count
0003: The value to be write 3(s01) in first register, address 18300
0004: The value to be write 4(s02) in second register, address 18301

Response
10 477C 0002

10: The Function Code 16(10 hex)
477C: Starting Address of the fisrt register (477C hex = 18300)
0002: Quantity of registers

9.3.1.5. Exception codes

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

01 Illegal function

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

02 Illegal data address

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

9.3.2. Main registers address table

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

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

Parameter	FC	PDU decimal address	Data size	Data
Digital input 1	02	100	Discrete	
Digital input 2	02	101	Discrete	
Digital input 3	02	102	Discrete	
Digital input 4	02	103	Discrete	
Analog input 1 value	03	14400	32-bit Float	
Analog input 2 value	03	14402	32-bit Float	
Analog input 3 value	03	14404	32-bit Float	
Analog input 4 value	03	14406	32-bit Float	
Analog input 5 value	03	14408	32-bit Float	
Analog input 6 value	03	14410	32-bit Float	
Channel 1 value	03	19300	32-bit Float	
Channel 2 value	03	19302	32-bit Float	
Channel 3 value	03	19304	32-bit Float	
Channel 4 value	03	19306	32-bit Float	
Channel 5 value	03	19308	32-bit Float	
Channel 6 value	03	19310	32-bit Float	
Channel 7 value	03	19312	32-bit Float	
Channel 8 value	03	19314	32-bit Float	
Channel 9 value	03	19316	32-bit Float	
Channel 10 value	03	19318	32-bit Float	

Channel 11 value	03	19320	32-bit Float	
Channel 12 value	03	19322	32-bit Float	
Channel 13 value	03	19324	32-bit Float	
Channel 14 value	03	19326	32-bit Float	
Channel 15 value	03	19328	32-bit Float	
Channel 16 value	03	19330	32-bit Float	
Channel 17 value	03	19332	32-bit Float	
Channel 18 value	03	19334	32-bit Float	
Channel 19 value	03	19336	32-bit Float	
Channel 20 value	03	19338	32-bit Float	
Channel 21 value	03	19340	32-bit Float	
Channel 22 value	03	19342	32-bit Float	
Channel 23 value	03	19344	32-bit Float	
Channel 24 value	03	19346	32-bit Float	
Alarm 1 status	03	22000	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 2 status	03	22001	16-bit unsign int	—"—
Alarm 3 status	03	22002	16-bit unsign int	—"—
Alarm 4 status	03	22003	16-bit unsign int	—"—
Alarm 5 status	03	22004	16-bit unsign int	—"—
Alarm 6 status	03	22005	16-bit unsign int	—"—
Alarm 7 status	03	22006	16-bit unsign int	—"—
Alarm 8 status	03	22007	16-bit unsign int	—"—
Alarm 9 status	03	22008	16-bit unsign int	—"—
Alarm 10 status	03	22009	16-bit unsign int	—"—
Alarm 11 status	03	22010	16-bit unsign int	—"—
Alarm 12 status	03	22011	16-bit unsign int	—"—
Alarm 13 status	03	22012	16-bit unsign int	—"—
Alarm 14 status	03	22013	16-bit unsign int	—"—
Alarm 15 status	03	22014	16-bit unsign int	—"—
Alarm 16 status	03	22015	16-bit unsign int	—"—
Alarm 17 status	03	22016	16-bit unsign int	—"—
Alarm 18 status	03	22017	16-bit unsign int	—"—
Alarm 19 status	03	22018	16-bit unsign int	—"—
Alarm 20 status	03	22019	16-bit unsign int	—"—
Alarm 21 status	03	22020	16-bit unsign int	—"—
Alarm 22 status	03	22021	16-bit unsign int	—"—
Alarm 23 status	03	22022	16-bit unsign int	—"—
Alarm 24 status	03	22023	16-bit unsign int	—"—
Save configuration	03,06	24000	16-bit unsign int	unsaved(0), saved(1)

9.4. Modbus RTU

9.4.1. Communication parameters

For Modbus RTU, TCW260 supports the following communication parameters:

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

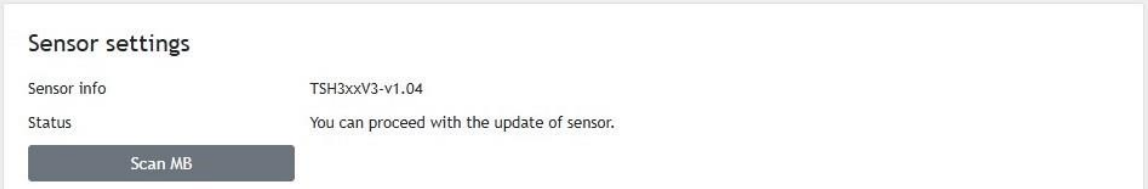
The factory default communication parameters for the device are the standard ones for Modbus RTU:

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

9.4.2. Teracom sensors update tool

The TCW260 supports the Teracom sensor firmware update tool, which is available at http://device.ip.address/teracom485.htm.
Attention! To make any changes to a Modbus RTU sensor, it should be the only device on the RS-485 bus.

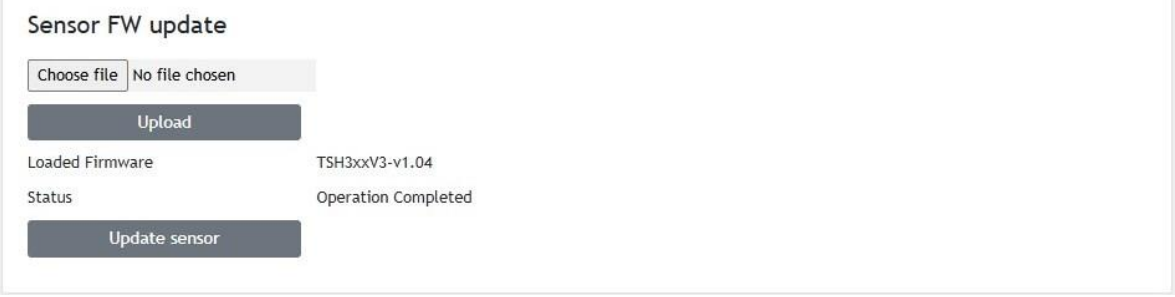
9.4.2.1. Sensor settings



The tool operates using the current Modbus RTU communication parameters. To avoid communication collisions, it is advisable to configure both the TCW260 and the sensor with the factory default Modbus RTU communication parameters. This practice ensures smooth operation. The default Modbus RTU communication parameters for the TCW260 are detailed in Section 9.4.1.

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

9.4.2.2. Sensor FW update



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

9.5. MQTT

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

The TCW260 supports “Publish” mode only.

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

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

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

9.6. Data Logger

The logger utilizes a circular buffer in FLASH memory. When the buffer is full, new data overwrites the oldest entries, ensuring that the FLASH memory continuously stores a complete log. There is no command to clear the log, but a copy of the full log is always available for download.
The number of records that can be stored depends on the length of the descriptions and the types of characters used. In the worst-case scenario—using 15-byte descriptions with characters from the upper range of UTF-8—the total number of records is approximately 52,371. This capacity is sufficient for 36 days of logging at one-minute intervals.
In most cases, the data logger can retain around 71,400 records, which allows for 49 days of logging at one-minute intervals.
New records can be periodically uploaded as a CSV file to a designated HTTP server at intervals of 1, 2, 3, 4, 6, 8, 12, and 24 hours. A semicolon is used as the delimiter in the CSV format.
The first row of the log file always contains the header. All rows, including the header, begin with the record ID and timestamp.

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

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

ID	32-bit unique number for every row (record).
Time	a time stamp of record, in format dd.mm.yyyy, hh:mm:ss.
Type of record	following types of records are available: "Time" for periodical record; "Event" for record initiated by alarm condition; "Type" for header record; "Start" after power-up condition; "Restart" after reset condition; "Power Down" after power-down condition; "Bad" for a problematic record
Channels - values/units	Channels 1 to 24 values/units
Channels - states/units	Channels 1 to 24 states/units For chanel types General, Cumulative and Discrete following states are available: 0 – “Undefined” 1 – “Normal” 2 – “Indeterminate” 3 – “Warning”

4 – “Minor”
5 – “Major”
6 – “Critical”
For chanel type Counter following states are available:
8 – “Undefined”
9 – “Normal”
10 – “Indeterminate”
11 – “Warning”
12 – “Minor”
13 – “Major”
14 – “Critical”

Alarms - values/descriptions

Alarms 1 to 24 values/descriptions
Following alarm values are available:
0 – “Undefined”
1 – “Normal”
2 – “Indeterminate”
3 – “Warning”
4 – “Minor”
5 – “Major”
6 – “Critical”

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

```
ID;Time;Type;Ch1/°C;Ch2;Ch3;Ch4;Ch5;Ch6;Ch7;Ch8;Ch9;Ch10;Ch11;Ch12;Ch13;Ch14;Ch15;Ch16;Ch17;Ch18;Ch19;Ch20;Ch21;Ch22;Ch23;Ch24....
25114;14.05.2019,16:49:49; Time;25.319;1.000;118.833;229.877;0.000;6587.396;;;;;;;;;;;;;;;;;;;;....
```

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

```
ID;Time;Type;....Ch1/°C;Ch2;Ch3;Ch4;Ch5;Ch6;Ch7;Ch8;Ch9;Ch10;Ch11;Ch12;Ch13;Ch14;Ch15;Ch16;Ch17;Ch18;Ch19;Ch20;Ch21;Ch22;Ch23;Ch24....
25114;14.05.2019,16:49:49;Time;....1;1;3;6;10;11;;;;;;;;;;;;;;;;;;;;....
```

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

```
ID;Time;Type;...AL1/AL1-temp;AL2;AL3;AL4;AL5;AL6;AL7;AL8;AL9;AL10;AL11;AL12;AL13;AL14;AL15;AL16;AL17;AL18;AL19;AL20;AL21;AL22;AL23;AL24....
25114;14.05.2019,16:49:49;Time;....1;1;1;3;4;5;;;;;;;;;;;;;;;;;;
```

10. Factory default settings

TCW260 can be restored to its original factory default settings in three different ways.

10.1. Factory reset without network settings

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

10.2. Factory reset for network settings only

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

10.3. Full factory reset

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

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



The factory default settings are:

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

11. Environment information

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

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

12. Safety

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

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

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

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

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

To ensure that the device works correctly follow these steps:

- Ensure that the device is installed correctly by referring to this user manual;
- Log into the device using a web browser;
- Configure the necessary settings;
- Navigate to SETUP → INPUTS and set up Digital Input 1 as “Discrete OPEN/CLOSED;
- Short the “S01+ (Digital In 1)” and “S01- (SGND)” terminals;
- Verify that the correct value for Digital Input 1 is displayed in the corresponding field;
- Observe that the flashing “STS” LED indicates proper operation.

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

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

13. Maintenance

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

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

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

XML file structure

```
<Monitor>
  <DeviceInfo>
    <DeviceName>TCW260</DeviceName>
    <HostName>TCW260 </HostName>
    <ID>54:10:ec:4f:59:f6</ID>
    <FwVer>TCW260-v1.004</FwVer>
    <MnfInfo>www.teracomsystems.com</MnfInfo>
    <SysContact>info@teracomsystems.com</SysContact>
    <SysName>TCW260</SysName>
    <SysLocation>Location</SysLocation>
  </DeviceInfo>
  <CH>
    <CH1>
      <type>0</type>
      <description>Temperature</description>
      <value>24.386</value>
      <valuebin/>
      <unit>°C</unit>
      <alarmbin>4</alarmbin>
      <alarm>Minor</alarm>
      <selch>3</selch>
    </CH1>
    <CH2>
      <type>0</type>
      <description>Humidity</description>
      <value>51.323</value>
      <valuebin/>
      <unit>RH</unit>
      <alarmbin>1</alarmbin>
      <alarm>Normal</alarm>
      <selch>4</selch>
    </CH2>
    <CH3>
      <type>2</type>
      <description>Digital Input 1</description>
      <value>OPEN</value>
      <valuebin>1</valuebin>
      <unit/>
      <alarmbin>1</alarmbin>
      <alarm>Normal</alarm>
      <selch>33</selch>
    </CH3>
    <CH4>
      <type>0</type>
      <description>V04</description>
      <value>---</value>
      <valuebin/>
      <unit/>
      <alarmbin>0</alarmbin>
      <alarm/>
      <selch>0</selch>
    </CH4>
    <CH5>
      <type>0</type>
      <description>V05-Voltage</description>
      <value>0.000</value>
      <valuebin/>
      <unit>V</unit>
      <alarmbin>1</alarmbin>
      <alarm>Normal</alarm>
      <selch>27</selch>
    </CH5>
    <CH6>
      <type>0</type>
      <description>V06-Current</description>
      <value>0.000</value>
      <valuebin/>
      <unit>A</unit>
      <alarmbin>1</alarmbin>
      <alarm>Normal</alarm>
      <selch>28</selch>
    </CH6>
```

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

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

```
      <valuebin/>
      <unit/>
      <alarmbin>0</alarmbin>
      <alarm/>
      <selch>0</selch>
    </CH22>
    <CH23>
    <type>0</type>
      <description>V23</description>
      <value>---</value>
      <valuebin/>
      <unit/>
      <alarmbin>0</alarmbin>
      <alarm/>
      <selch>0</selch>
    </CH23>
    <CH24>
      <type>0</type>
      <description>V24</description>
      <value>---</value>
      <valuebin/>
      <unit/>
      <alarmbin>0</alarmbin>
      <alarm/>
      <selch>0</selch>
    </CH24>
  </CH>
  <AL>
    <AL1>
      <description>AL01-Temp.</description>
      <alarmbin>4</alarmbin>
      <alarm>Minor</alarm>
      <assign>1</assign>
    </AL1>
    <AL2>
      <description>AL02</description>
      <alarmbin>1</alarmbin>
      <alarm>Normal</alarm>
      <assign>5</assign>
    </AL2>
    <AL3>
      <description>AL03</description>
      <alarmbin>1</alarmbin>
      <alarm>Normal</alarm>
      <assign>6</assign>
    </AL3>
    <AL4>
      <description>AL04-DI1</description>
      <alarmbin>1</alarmbin>
      <alarm>Normal</alarm>
      <assign>3</assign>
    </AL4>
    <AL5>
      <description>AL05-Humidity</description>
      <alarmbin>1</alarmbin>
      <alarm>Normal</alarm>
      <assign>2</assign>
    </AL5>
    <AL6>
      <description>AL06</description>
      <alarmbin>1</alarmbin>
      <alarm>Normal</alarm>
      <assign>3</assign>
    </AL6>
    <AL7>
      <description>AL07</description>
      <alarmbin>0</alarmbin>
      <alarm/>
      <assign>0</assign>
    </AL7>
    <AL8>
      <description>AL08</description>
      <alarmbin>0</alarmbin>
      <alarm/>
      <assign>0</assign>
    </AL8>
    <AL9>
```

<description>AL09</description>
<alarmbin>0</alarmbin>
<alarm/>
<assign>0</assign>
</AL9>
<AL10>
<description>AL10</description>
<alarmbin>0</alarmbin>
<alarm/>
<assign>0</assign>
</AL10>
<AL11>
<description>AL11</description>
<alarmbin>0</alarmbin>
<alarm/>
<assign>0</assign>
</AL11>
<AL12>
<description>AL12</description>
<alarmbin>0</alarmbin>
<alarm/>
<assign>0</assign>
</AL12>
<AL13>
<description>AL13</description>
<alarmbin>0</alarmbin>
<alarm/>
<assign>0</assign>
</AL13>
<AL14>
<description>AL14</description>
<alarmbin>0</alarmbin>
<alarm/>
<assign>0</assign>
</AL14>
<AL15>
<description>AL15</description>
<alarmbin>0</alarmbin>
<alarm/>
<assign>0</assign>
</AL15>
<AL16>
<description>AL16</description>
<alarmbin>0</alarmbin>
<alarm/>
<assign>0</assign>
</AL16>
<AL17>
<description>AL17</description>
<alarmbin>0</alarmbin>
<alarm/>
<assign>0</assign>
</AL17>
<AL18>
<description>AL18</description>
<alarmbin>0</alarmbin>
<alarm/>
<assign>0</assign>
</AL18>
<AL19>
<description>AL19</description>
<alarmbin>0</alarmbin>
<alarm/>
<assign>0</assign>
</AL19>
<AL20>
<description>AL20</description>
<alarmbin>0</alarmbin>
<alarm/>
<assign>0</assign>
</AL20>
<AL21>
<description>AL21</description>
<alarmbin>0</alarmbin>
<alarm/>
<assign>0</assign>
</AL21>

```
    <AL22>
      <description>AL22</description>
      <alarmbin>0</alarmbin>
      <alarm/>
      <assign>0</assign>
    </AL22>
    <AL23>
      <description>AL23</description>
      <alarmbin>0</alarmbin>
      <alarm/>
      <assign>0</assign>
    </AL23>
    <AL24>
      <description>AL24</description>
      <alarmbin>0</alarmbin>
      <alarm/>
      <assign>0</assign>
    </AL24>
  </AL>
  <HTTPPost>
    <Key/>
    <PostPeriod>300</PostPeriod>
  </HTTPPost>
  <MQTT>
     <Period>300</Period>
  </MQTT>
  <Sys>
    <hwerr/>
    <HighAlarmbin>4</HighAlarmbin>
    <HighAlarm>Minor</HighAlarm>
  </Sys>
  <Time>
    <Date>15.05.2019</Date>
    <Time>10:32:44</Time>
  </Time>
</Monitor>
```

Where:

<CH1>… <CH24> - channels;

<AL1> … <AL24> - alarms;

<alarmbin> - number values from 0 to 6;

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

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

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

JSON file structure


```
{
  "Monitor": {
    "DeviceInfo": {
      "DeviceName": "TCW260",
      "HostName": "TCW260       ",
      "ID": "54:10:ec:4f:59:f6",
      "FwVer": "TCW260-v1.004",
      "MnfInfo": "www.teracomsystems.com",
      "SysContact": "info@teracomsystems.com",
      "SysName": "TCW260",
      "SysLocation": "Location"
    },
    "CH": {
      "CH1": {
        "type": "0",
        "description": "Temperature",
        "value": "24.268",
        "valuebin": "",
        "unit": "°C",
        “alarmbin”: "4",
        “alarm”: "Minor",
        "selch":"3"
      },
      "CH2": {
        "type": "0",
        "description": "Humidity",
        "value": "52.490",
       "valuebin": "",
        "unit": "RH",
        “alarmbin”: "1",
        “alarm”: "Normal",
        "selch":"4"
      },
      "CH3": {
        "type": "2",
        "description": "Digital Input 1",
        "value": "OPEN",
        "valuebin": "1",
        "unit": "",
        “alarmbin”: "1",
        “alarm”: "Normal",
        "selch":"33"
      },
      "CH4": {
        "type": "0",
        "description": "V04",
        "value": "---",
        "valuebin": "",
        "unit": "",
        “alarmbin”: "0",
        “alarm”: "",
        "selch":"0"
      },
      "CH5": {
        "type": "0",
        "description": "V05-Voltage",
        "value": "0.000",
        "valuebin": "",
        "unit": "V",
        “alarmbin”: "1",
        “alarm”: "Normal",
        "selch":"27"
      },
      "CH6": {
        "type": "0",
        "description": "V06-Current",
        "value": "0.000",
        "valuebin": "",
        "unit": "A",
        “alarmbin”: "1",
        “alarm”: "Normal",
        "selch":"28"
      },
```

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

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

```
      "valuebin": "",
      "unit": "",
      “alarmbin”: "0",
      “alarm”: "",
      "selch":"0"
    },
    "CH23": {
      "type": "0",
      "description": "V23",
      "value": "---",
      "valuebin": "",
      "unit": "",
      “alarmbin”: "0",
      “alarm”: "",
      "selch":"0"
    },
    "CH24": {
      "type": "0",
      "description": "V24",
      "value": "---",
      "valuebin": "",
      "unit": "",
      “alarmbin”: "0",
      “alarm”: "",
      "selch":"0"
    }
  },
  "AL": {
    "AL1": {
      "description": "AL01-Temp.",
      “alarmbin”: "aa(0)",
      “alarm”: "Minor",
      "assign":"1"
    },
    "AL2": {
      "description": "AL02",
      “alarmbin”: "aa(1)",
      “alarm”: "Normal",
      "assign":"5"
    },
    "AL3": {
      "description": "AL03",
      “alarmbin”: "aa(2)",
      “alarm”: "Normal",
      "assign":"6"
    },
    "AL4": {
      "description": "AL04-DI1",
      “alarmbin”: "aa(3)",
      “alarm”: "Normal",
      "assign":"3"
    },
    "AL5": {
      "description": "AL05-Humidity",
      “alarmbin”: "aa(4)",
      “alarm”: "Normal",
      "assign":"2"
    },
    "AL6": {
      "description": "AL06",
      “alarmbin”: "aa(5)",
      “alarm”: "Normal",
      "assign":"3"
    },
    "AL7": {
      "description": "AL07",
      “alarmbin”: "aa(6)",
      “alarm”: "",
      "assign":"0"
    },
    "AL8": {
      "description": "AL08",
      “alarmbin”: "aa(7)",
      “alarm”: "",
      "assign":"0"
    },
    "AL9": {
```

```
    "description": "AL09",
    “alarmbin”: "aa(8)",
    “alarm”: "",
    "assign":"0"
  },
  "AL10": {
    "description": "AL10",
    “alarmbin”: "aa(9)",
    “alarm”: "",
    "assign":"0"
  },
  "AL11": {
    "description": "AL11",
    “alarmbin”: "aa(10)",
    “alarm”: "",
    "assign":"0"
  },
  "AL12": {
    "description": "AL12",
    “alarmbin”: "aa(11)",
    “alarm”: "",
    "assign":"0"
  },
  "AL13": {
    "description": "AL13",
    “alarmbin”: "aa(12)",
    “alarm”: "",
    "assign":"0"
  },
  "AL14": {
    "description": "AL14",
    “alarmbin”: "aa(13)",
    “alarm”: "",
    "assign":"0"
  },
  "AL15": {
    "description": "AL15",
    “alarmbin”: "aa(14)",
    “alarm”: "",
    "assign":"0"
  },
  "AL16": {
    "description": "AL16",
    “alarmbin”: "aa(15)",
    “alarm”: "",
    "assign":"0"
  },
  "AL17": {
    "description": "AL17",
    “alarmbin”: "aa(16)",
    “alarm”: "",
    "assign":"0"
  },
  "AL18": {
    "description": "AL18",
    “alarmbin”: "aa(17)",
    “alarm”: "",
    "assign":"0"
  },
  "AL19": {
    "description": "AL19",
    “alarmbin”: "aa(18)",
    “alarm”: "",
    "assign":"0"
  },
  "AL20": {
    "description": "AL20",
    “alarmbin”: "aa(19)",
    “alarm”: "",
    "assign":"0"
  },
  "AL21": {
    "description": "AL21",
    “alarmbin”: "aa(20)",
    “alarm”: "",
    "assign":"0"
  },
```

```
      "AL22": {
        "description": "AL22",
        “alarmbin”: "aa(21)",
        “alarm”: "",
        "assign":"0"
      },
      "AL23": {
        "description": "AL23",
        “alarmbin”: "aa(22)",
        “alarm”: "",
        "assign":"0"
      },
      "AL24": {
        "description": "AL24",
        “alarmbin”: "aa(23)",
        “alarm”: "",
        "assign":"0"
      }
    },
    "HTTPPost": {
      "Key": "",
      "PostPeriod": "300"
    },
    “MQTT”: {
       “Period”: “300”
    },
    "Sys": {
      "hwerr": "",
      "HighAlarmbin": "4",
      "HighAlarm": "Minor"
    },
    "Time": {
      "Date": "15.05.2019",
      "Time": "10:30:00"
    }
  }
}
```

Modbus TCP/IP full address table

Parameter	FC	PDU decimal address	Data size	Data
Digital input 1	02	100	Discrete	
Digital input 2	02	101	Discrete	
Digital input 3	02	102	Discrete	
Digital input 4	02	103	Discrete	
Digital input 1 Description	03,16	12000	16 bytes UTF-8	
Digital input 2 Description	03,16	12008	16 bytes UTF-8	
Digital input 3 Description	03,16	12016	16 bytes UTF-8	
Digital input 4 Description	03,16	12024	16 bytes UTF-8	
Digital input 1 closed state description	03,16	12100	16 bytes UTF-8	
Digital input 2 closed state description	03,16	12108	16 bytes UTF-8	
Digital input 3 closed state description	03,16	12116	16 bytes UTF-8	
Digital input 4 closed state description	03,16	12124	16 bytes UTF-8	
Digital input 1 open state description	03,16	12200	16 bytes UTF-8	
Digital input 2 open state description	03,16	12208	16 bytes UTF-8	
Digital input 3 open state description	03,16	12216	16 bytes UTF-8	
Digital input 4 open state description	03,16	12224	16 bytes UTF-8	
Digital input 1 mode	03,06,16	12300	16-bit unsign int	openclosed(0), risingEdge(1), fallingEdge(2), bothEdges(3)
Digital input 2 mode	03,06,16	12301	16-bit unsign int	—"—
Digital input 3 mode	03,06,16	12302	16-bit unsign int	—"—
Digital input 4 mode	03,06,16	12303	16-bit unsign int	—"—
Digital input 1 close to open delay	03,16	12400	32-bit unsign int	
Digital input 2 close to open delay	03,16	12402	32-bit unsign int	
Digital input 3 close to open delay	03,16	12404	32-bit unsign int	
Digital input 4 close to open delay	03,16	12406	32-bit unsign int	
Digital input 1 open to close delay	03,16	12500	32-bit unsign int	
Digital input 2 open to close delay	03,16	12502	32-bit unsign int	

Digital input 3 open to close delay	03,16	12504	32-bit unsign int	
Digital input 4 open to close delay	03,16	12506	32-bit unsign int	
Digital input 1 counter init value	03,16	12600	32-bit unsign int	
Digital input 2 counter init value	03,16	12602	32-bit unsign int	
Digital input 3 counter init value	03,16	12604	32-bit unsign int	
Digital input 4 counter init value	03,16	12606	32-bit unsign int	
Digital input 1 counter value	03	12700	32-bit unsign int	
Digital input 2 counter value	03	12702	32-bit unsign int	
Digital input 3 counter value	03	12704	32-bit unsign int	
Digital input 4 counter value	03	12706	32-bit unsign int	
Analog input 1 description	03,16	14000	16 bytes UTF-8	
Analog input 2 description	03,16	14008	16 bytes UTF-8	
Analog input 3 description	03,16	14016	16 bytes UTF-8	
Analog input 4 description	03,16	14024	16 bytes UTF-8	
Analog input 5 description	03,16	14032	16 bytes UTF-8	
Analog input 6 description	03,16	14040	16 bytes UTF-8	
Analog input 1 multiplier	03,16	14100	32-bit Float	
Analog input 2 multiplier	03,16	14102	32-bit Float	
Analog input 3 multiplier	03,16	14104	32-bit Float	
Analog input 4 multiplier	03,16	14106	32-bit Float	
Analog input 5 multiplier	03,16	14108	32-bit Float	
Analog input 6 multiplier	03,16	14100	32-bit Float	
Analog input 1 offset	03,16	14200	32-bit Float	
Analog input 2 offset	03,16	14202	32-bit Float	
Analog input 3 offset	03,16	14204	32-bit Float	
Analog input 4 offset	03,16	14206	32-bit Float	
Analog input 5 offset	03,16	14208	32-bit Float	
Analog input 6 offset	03,16	14200	32-bit Float	
Analog input 1 mode(V/mA)	03,06,16	14300	16-bit unsign int	0-10V(0), 4-20mA(1)
Analog input 2 mode(V/mA)	03,06,16	14301	16-bit unsign int	0-10V(0), 4-20mA(1)
Analog input 3 mode(V/mA)	03,06,16	14302	16-bit unsign int	0-10V(0), 4-20mA(1)
Analog input 4 mode(V/mA)	03,06,16	14303	16-bit unsign int	0-10V(0), 4-20mA(1)
Analog input 5 mode(V/mA)	03,06,16	14304	16-bit unsign int	0-10V(0), 4-20mA(1)
Analog input 6 mode(V/mA)	03,06,16	14305	16-bit unsign int	0-10V(0), 4-20mA(1)
Analog input 1 value	03	14400	32-bit Float	
Analog input 2 value	03	14402	32-bit Float	
Analog input 3 value	03	14404	32-bit Float	

Analog input 4 value	03	14406	32-bit Float	
Analog input 5 value	03	14408	32-bit Float	
Analog input 6 value	03	14410	32-bit Float	
MB sensor 1 description	03,06,16	16000	16 bytes UTF-8	
MB sensor 2 description	03,06,16	16008	16 bytes UTF-8	
MB sensor 3 description	03,06,16	16016	16 bytes UTF-8	
MB sensor 4 description	03,06,16	16024	16 bytes UTF-8	
MB sensor 5 description	03,06,16	16032	16 bytes UTF-8	
MB sensor 6 description	03,06,16	16040	16 bytes UTF-8	
MB sensor 7 description	03,06,16	16048	16 bytes UTF-8	
MB sensor 8 description	03,06,16	16056	16 bytes UTF-8	
MB sensor 9 description	03,06,16	16064	16 bytes UTF-8	
MB sensor 10 description	03,06,16	16072	16 bytes UTF-8	
MB sensor 11 description	03,06,16	16080	16 bytes UTF-8	
MB sensor 12 description	03,06,16	16088	16 bytes UTF-8	
MB sensor 13 description	03,06,16	16096	16 bytes UTF-8	
MB sensor 14 description	03,06,16	16104	16 bytes UTF-8	
MB sensor 15 description	03,06,16	16112	16 bytes UTF-8	
MB sensor 16 description	03,06,16	16120	16 bytes UTF-8	
MB sensor 17 description	03,06,16	16128	16 bytes UTF-8	
MB sensor 18 description	03,06,16	16136	16 bytes UTF-8	
MB sensor 19 description	03,06,16	16144	16 bytes UTF-8	
MB sensor 20 description	03,06,16	16152	16 bytes UTF-8	
MB sensor 21 description	03,06,16	16160	16 bytes UTF-8	
MB sensor 22 description	03,06,16	16168	16 bytes UTF-8	
MB sensor 23 description	03,06,16	16176	16 bytes UTF-8	
MB sensor 24 description	03,06,16	16184	16 bytes UTF-8	
MB sensor 1 multiplier	03,16	16200	32-bit Float	
MB sensor 2 multiplier	03,16	16202	32-bit Float	
MB sensor 3 multiplier	03,16	16204	32-bit Float	
MB sensor 4 multiplier	03,16	16206	32-bit Float	
MB sensor 5 multiplier	03,16	16208	32-bit Float	
MB sensor 6 multiplier	03,16	16210	32-bit Float	
MB sensor 7 multiplier	03,16	16212	32-bit Float	
MB sensor 8 multiplier	03,16	16214	32-bit Float	
MB sensor 9 multiplier	03,16	16216	32-bit Float	
MB sensor 10 multiplier	03,16	16218	32-bit Float	
MB sensor 11 multiplier	03,16	16220	32-bit Float	
MB sensor 12 multiplier	03,16	16222	32-bit Float	
MB sensor 13 multiplier	03,16	16224	32-bit Float	
MB sensor 14 multiplier	03,16	16226	32-bit Float	
MB sensor 15 multiplier	03,16	16228	32-bit Float	
MB sensor 16 multiplier	03,16	16230	32-bit Float	
MB sensor 17 multiplier	03,16	16232	32-bit Float	
MB sensor 18 multiplier	03,16	16234	32-bit Float	
MB sensor 19 multiplier	03,16	16236	32-bit Float	

MB sensor 20 multiplier	03,16	16238	32-bit Float	
MB sensor 21 multiplier	03,16	16240	32-bit Float	
MB sensor 22 multiplier	03,16	16242	32-bit Float	
MB sensor 23 multiplier	03,16	16244	32-bit Float	
MB sensor 24 multiplier	03,16	16246	32-bit Float	
MB sensor 1 offset	03,16	16300	32-bit Float	
MB sensor 2 offset	03,16	16302	32-bit Float	
MB sensor 3 offset	03,16	16304	32-bit Float	
MB sensor 4 offset	03,16	16306	32-bit Float	
MB sensor 5 offset	03,16	16308	32-bit Float	
MB sensor 6 offset	03,16	16310	32-bit Float	
MB sensor 7 offset	03,16	16312	32-bit Float	
MB sensor 8 offset	03,16	16314	32-bit Float	
MB sensor 9 offset	03,16	16316	32-bit Float	
MB sensor 10 offset	03,16	16318	32-bit Float	
MB sensor 11 offset	03,16	16320	32-bit Float	
MB sensor 12 offset	03,16	16322	32-bit Float	
MB sensor 13 offset	03,16	16324	32-bit Float	
MB sensor 14 offset	03,16	16326	32-bit Float	
MB sensor 15 offset	03,16	16328	32-bit Float	
MB sensor 16 offset	03,16	16330	32-bit Float	
MB sensor 17 offset	03,16	16332	32-bit Float	
MB sensor 18 offset	03,16	16334	32-bit Float	
MB sensor 19 offset	03,16	16336	32-bit Float	
MB sensor 20 offset	03,16	16338	32-bit Float	
MB sensor 21 offset	03,16	16340	32-bit Float	
MB sensor 22 offset	03,16	16342	32-bit Float	
MB sensor 23 offset	03,16	16344	32-bit Float	
MB sensor 24 offset	03,16	16346	32-bit Float	
MB sensor 1 value	03	16400	32-bit Float	
MB sensor 2 value	03	16402	32-bit Float	
MB sensor 3 value	03	16404	32-bit Float	
MB sensor 4 value	03	16406	32-bit Float	
MB sensor 5 value	03	16408	32-bit Float	
MB sensor 6 value	03	16410	32-bit Float	
MB sensor 7 value	03	16412	32-bit Float	
MB sensor 8 value	03	16414	32-bit Float	
MB sensor 9 value	03	16416	32-bit Float	
MB sensor 10 value	03	16418	32-bit Float	
MB sensor 11 value	03	16420	32-bit Float	
MB sensor 12 value	03	16422	32-bit Float	
MB sensor 13 value	03	16424	32-bit Float	
MB sensor 14 value	03	16426	32-bit Float	
MB sensor 15 value	03	16428	32-bit Float	
MB sensor 16 value	03	16430	32-bit Float	
MB sensor 17 value	03	16432	32-bit Float	

MB sensor 18 value	03	16434	32-bit Float	
MB sensor 19 value	03	16436	32-bit Float	
MB sensor 20 value	03	16438	32-bit Float	
MB sensor 21 value	03	16440	32-bit Float	
MB sensor 22 value	03	16442	32-bit Float	
MB sensor 23 value	03	16444	32-bit Float	
MB sensor 24 value	03	16446	32-bit Float	
MB sensor 1 counter	03	16500	32-bit unsign int	
MB sensor 2 counter	03	16502	32-bit unsign int	
MB sensor 3 counter	03	16504	32-bit unsign int	
MB sensor 4 counter	03	16506	32-bit unsign int	
MB sensor 5 counter	03	16508	32-bit unsign int	
MB sensor 6 counter	03	16510	32-bit unsign int	
MB sensor 7 counter	03	16512	32-bit unsign int	
MB sensor 8 counter	03	16514	32-bit unsign int	
MB sensor 9 counter	03	16516	32-bit unsign int	
MB sensor 10 counter	03	16518	32-bit unsign int	
MB sensor 11 counter	03	16520	32-bit unsign int	
MB sensor 12 counter	03	16522	32-bit unsign int	
MB sensor 13 counter	03	16524	32-bit unsign int	
MB sensor 14 counter	03	16526	32-bit unsign int	
MB sensor 15 counter	03	16528	32-bit unsign int	
MB sensor 16 counter	03	16530	32-bit unsign int	
MB sensor 17 counter	03	16532	32-bit unsign int	
MB sensor 18 counter	03	16534	32-bit unsign int	
MB sensor 19 counter	03	16536	32-bit unsign int	
MB sensor 20 counter	03	16538	32-bit unsign int	
MB sensor 21 counter	03	16540	32-bit unsign int	
MB sensor 22 counter	03	16542	32-bit unsign int	
MB sensor 23 counter	03	16544	32-bit unsign int	
MB sensor 24 counter	03	16546	32-bit unsign int	
Channel 1 type	03,06,16	18000	16-bit unsign int	general(0), discrete(2) , counter(3)
Channel 2 type	03,06,16	18001	16-bit unsign int	—"—
Channel 3 type	03,06,16	18002	16-bit unsign int	—"—
Channel 4 type	03,06,16	18003	16-bit unsign int	—"—
Channel 5 type	03,06,16	18004	16-bit unsign int	—"—
Channel 6 type	03,06,16	18005	16-bit unsign int	—"—
Channel 7 type	03,06,16	18006	16-bit unsign int	—"—
Channel 8 type	03,06,16	18007	16-bit unsign int	—"—
Channel 9 type	03,06,16	18008	16-bit unsign int	—"—
Channel 10 type	03,06,16	18009	16-bit unsign int	—"—
Channel 11 type	03,06,16	18010	16-bit unsign int	—"—
Channel 12 type	03,06,16	18011	16-bit unsign int	—"—
Channel 13 type	03,06,16	18012	16-bit unsign int	—"—
Channel 14 type	03,06,16	18013	16-bit unsign int	—"—
Channel 15 type	03,06,16	18014	16-bit unsign int	—"—

Channel 16 type	03,06,16	18015	16-bit unsign int	—"—
Channel 17 type	03,06,16	18016	16-bit unsign int	—"—
Channel 18 type	03,06,16	18017	16-bit unsign int	—"—
Channel 19 type	03,06,16	18018	16-bit unsign int	—"—
Channel 20 type	03,06,16	18019	16-bit unsign int	—"—
Channel 21 type	03,06,16	18020	16-bit unsign int	—"—
Channel 22 type	03,06,16	18021	16-bit unsign int	—"—
Channel 23 type	03,06,16	18022	16-bit unsign int	—"—
Channel 24 type	03,06,16	18023	16-bit unsign int	—"—
Channel 1 description	03,16	18100	16 bytes UTF-8	
Channel 2 description	03,16	18108	16 bytes UTF-8	
Channel 3 description	03,16	18116	16 bytes UTF-8	
Channel 4 description	03,16	18124	16 bytes UTF-8	
Channel 5 description	03,16	18132	16 bytes UTF-8	
Channel 6 description	03,16	18140	16 bytes UTF-8	
Channel 7 description	03,16	18148	16 bytes UTF-8	
Channel 8 description	03,16	18156	16 bytes UTF-8	
Channel 9 description	03,16	18164	16 bytes UTF-8	
Channel 10 description	03,16	18172	16 bytes UTF-8	
Channel 11 description	03,16	18180	16 bytes UTF-8	
Channel 12 description	03,16	18188	16 bytes UTF-8	
Channel 13 description	03,16	18196	16 bytes UTF-8	
Channel 14 description	03,16	18204	16 bytes UTF-8	
Channel 15 description	03,16	18212	16 bytes UTF-8	
Channel 16 description	03,16	18220	16 bytes UTF-8	
Channel 17 description	03,16	18228	16 bytes UTF-8	
Channel 18 description	03,16	18236	16 bytes UTF-8	
Channel 19 description	03,16	18244	16 bytes UTF-8	
Channel 20 description	03,16	18252	16 bytes UTF-8	
Channel 21 description	03,16	18260	16 bytes UTF-8	
Channel 22 description	03,16	18268	16 bytes UTF-8	
Channel 23 description	03,16	18276	16 bytes UTF-8	
Channel 24 description	03,16	18284	16 bytes UTF-8	
Channel 1 parameter 1	03,06,16	18300	16-bit unsign int	none(0), s01(3), s02(4), s03(5), s04(6), s05(7), s06(8), s07(9), s08(10), s09(11), s10(12), s11(13), s12(14), s13(15), s14(16), s15(17), s16(18), s17(19), s18(20), s19(21), s20(22), s21(23), s22(24), s23(25), s24(26), a01(27), a02(28), a03(29), a04(30), d01(31), d02(32), d03(33), d04(34)
Channel 2 parameter 1	03,06,16	18301	16-bit unsign int	—"—
Channel 3 parameter 1	03,06,16	18302	16-bit unsign int	—"—
Channel 4 parameter 1	03,06,16	18303	16-bit unsign int	—"—
Channel 5 parameter 1	03,06,16	18304	16-bit unsign int	—"—

Channel 6 parameter 1	03,06,16	18305	16-bit unsign int	—"—
Channel 7 parameter 1	03,06,16	18306	16-bit unsign int	—"—
Channel 8 parameter 1	03,06,16	18307	16-bit unsign int	—"—
Channel 9 parameter 1	03,06,16	18308	16-bit unsign int	—"—
Channel 10 parameter 1	03,06,16	18309	16-bit unsign int	—"—
Channel 11 parameter 1	03,06,16	18310	16-bit unsign int	—"—
Channel 12 parameter 1	03,06,16	18311	16-bit unsign int	—"—
Channel 13 parameter 1	03,06,16	18312	16-bit unsign int	—"—
Channel 14 parameter 1	03,06,16	18313	16-bit unsign int	—"—
Channel 15 parameter 1	03,06,16	18314	16-bit unsign int	—"—
Channel 16 parameter 1	03,06,16	18315	16-bit unsign int	—"—
Channel 17 parameter 1	03,06,16	18316	16-bit unsign int	—"—
Channel 18 parameter 1	03,06,16	18317	16-bit unsign int	—"—
Channel 19 parameter 1	03,06,16	18318	16-bit unsign int	—"—
Channel 20 parameter 1	03,06,16	18319	16-bit unsign int	—"—
Channel 21 parameter 1	03,06,16	18320	16-bit unsign int	—"—
Channel 22 parameter 1	03,06,16	18321	16-bit unsign int	—"—
Channel 23 parameter 1	03,06,16	18322	16-bit unsign int	—"—
Channel 24 parameter 1	03,06,16	18323	16-bit unsign int	—"—
Channel 1 op 1	03,06,16	18400	16-bit unsign int	none(0), multiplication(1), division(2), sum(3), subtract(4)
Channel 2 op 1	03,06,16	18401	16-bit unsign int	—"—
Channel 3 op 1	03,06,16	18402	16-bit unsign int	—"—
Channel 4 op 1	03,06,16	18403	16-bit unsign int	—"—
Channel 5 op 1	03,06,16	18404	16-bit unsign int	—"—
Channel 6 op 1	03,06,16	18405	16-bit unsign int	—"—
Channel 7 op 1	03,06,16	18406	16-bit unsign int	—"—
Channel 8 op 1	03,06,16	18407	16-bit unsign int	—"—
Channel 9 op 1	03,06,16	18408	16-bit unsign int	—"—
Channel 10 op 1	03,06,16	18409	16-bit unsign int	—"—
Channel 11 op 1	03,06,16	18410	16-bit unsign int	—"—
Channel 12 op 1	03,06,16	18411	16-bit unsign int	—"—
Channel 13 op 1	03,06,16	18412	16-bit unsign int	—"—
Channel 14 op 1	03,06,16	18413	16-bit unsign int	—"—
Channel 15 op 1	03,06,16	18414	16-bit unsign int	—"—
Channel 16 op 1	03,06,16	18415	16-bit unsign int	—"—
Channel 17 op 1	03,06,16	18416	16-bit unsign int	—"—
Channel 18 op 1	03,06,16	18417	16-bit unsign int	—"—
Channel 19 op 1	03,06,16	18418	16-bit unsign int	—"—
Channel 20 op 1	03,06,16	18419	16-bit unsign int	—"—
Channel 21 op 1	03,06,16	18420	16-bit unsign int	—"—
Channel 22 op 1	03,06,16	18421	16-bit unsign int	—"—
Channel 23 op 1	03,06,16	18422	16-bit unsign int	—"—
Channel 24 op 1	03,06,16	18423	16-bit unsign int	—"—
Channel 1 parameter 2	03,06,16	18500	16-bit unsign int	none(0), s01(3), s02(4), s03(5), s04(6), s05(7),

				s06(8), s07(9), s08(10), s09(11), s10(12), s11(13), s12(14), s13(15), s14(16), s15(17), s16(18), s17(19), s18(20), s19(21), s20(22), s21(23), s22(24), s23(25), s24(26), a01(27), a02(28), a03(29), a04(30), d01(31), d02(32), d03(33), d04(34)
Channel 2 parameter 2	03,06,16	18501	16-bit unsign int	—"—
Channel 3 parameter 2	03,06,16	18502	16-bit unsign int	—"—
Channel 4 parameter 2	03,06,16	18503	16-bit unsign int	—"—
Channel 5 parameter 2	03,06,16	18504	16-bit unsign int	—"—
Channel 6 parameter 2	03,06,16	18505	16-bit unsign int	—"—
Channel 7 parameter 2	03,06,16	18506	16-bit unsign int	—"—
Channel 8 parameter 2	03,06,16	18507	16-bit unsign int	—"—
Channel 9 parameter 2	03,06,16	18508	16-bit unsign int	—"—
Channel 10 parameter 2	03,06,16	18509	16-bit unsign int	—"—
Channel 11 parameter 2	03,06,16	18510	16-bit unsign int	—"—
Channel 12 parameter 2	03,06,16	18511	16-bit unsign int	—"—
Channel 13 parameter 2	03,06,16	18512	16-bit unsign int	—"—
Channel 14 parameter 2	03,06,16	18513	16-bit unsign int	—"—
Channel 15 parameter 2	03,06,16	18514	16-bit unsign int	—"—
Channel 16 parameter 2	03,06,16	18515	16-bit unsign int	—"—
Channel 17 parameter 2	03,06,16	18516	16-bit unsign int	—"—
Channel 18 parameter 2	03,06,16	18517	16-bit unsign int	—"—
Channel 19 parameter 2	03,06,16	18518	16-bit unsign int	—"—
Channel 20 parameter 2	03,06,16	18519	16-bit unsign int	—"—
Channel 21 parameter 2	03,06,16	18520	16-bit unsign int	—"—
Channel 22 parameter 2	03,06,16	18521	16-bit unsign int	—"—
Channel 23 parameter 2	03,06,16	18522	16-bit unsign int	—"—
Channel 24 parameter 2	03,06,16	18523	16-bit unsign int	—"—
Channel 1 op 2	03,06,16	18600	16-bit unsign int	none(0), multiplication(1), division(2), sum(3), subtract(4)
Channel 2 op 2	03,06,16	18601	16-bit unsign int	—"—
Channel 3 op 2	03,06,16	18602	16-bit unsign int	—"—
Channel 4 op 2	03,06,16	18603	16-bit unsign int	—"—
Channel 5 op 2	03,06,16	18604	16-bit unsign int	—"—
Channel 6 op 2	03,06,16	18605	16-bit unsign int	—"—
Channel 7 op 2	03,06,16	18606	16-bit unsign int	—"—
Channel 8 op 2	03,06,16	18607	16-bit unsign int	—"—
Channel 9 op 2	03,06,16	18608	16-bit unsign int	—"—
Channel 10 op 2	03,06,16	18609	16-bit unsign int	—"—
Channel 11 op 2	03,06,16	18610	16-bit unsign int	—"—
Channel 12 op 2	03,06,16	18611	16-bit unsign int	—"—
Channel 13 op 2	03,06,16	18612	16-bit unsign int	—"—
Channel 14 op 2	03,06,16	18613	16-bit unsign int	—"—
Channel 15 op 2	03,06,16	18614	16-bit unsign int	—"—

Channel 16 op 2	03,06,16	18615	16-bit unsign int	—"—
Channel 17 op 2	03,06,16	18616	16-bit unsign int	—"—
Channel 18 op 2	03,06,16	18617	16-bit unsign int	—"—
Channel 19 op 2	03,06,16	18618	16-bit unsign int	—"—
Channel 20 op 2	03,06,16	18619	16-bit unsign int	—"—
Channel 21 op 2	03,06,16	18620	16-bit unsign int	—"—
Channel 22 op 2	03,06,16	18621	16-bit unsign int	—"—
Channel 23 op 2	03,06,16	18622	16-bit unsign int	—"—
Channel 24 op 2	03,06,16	18623	16-bit unsign int	—"—
Channel 1 coeff 1	03,16	18700	32-bit Float	
Channel 2 coeff 1	03,16	18702	32-bit Float	
Channel 3 coeff 1	03,16	18704	32-bit Float	
Channel 4 coeff 1	03,16	18706	32-bit Float	
Channel 5 coeff 1	03,16	18708	32-bit Float	
Channel 6 coeff 1	03,16	18710	32-bit Float	
Channel 7 coeff 1	03,16	18712	32-bit Float	
Channel 8 coeff 1	03,16	18714	32-bit Float	
Channel 9 coeff 1	03,16	18716	32-bit Float	
Channel 10 coeff 1	03,16	18718	32-bit Float	
Channel 11 coeff 1	03,16	18720	32-bit Float	
Channel 12 coeff 1	03,16	18722	32-bit Float	
Channel 13 coeff 1	03,16	18724	32-bit Float	
Channel 14 coeff 1	03,16	18726	32-bit Float	
Channel 15 coeff 1	03,16	18728	32-bit Float	
Channel 16 coeff 1	03,16	18730	32-bit Float	
Channel 17 coeff 1	03,16	18732	32-bit Float	
Channel 18 coeff 1	03,16	18734	32-bit Float	
Channel 19 coeff 1	03,16	18736	32-bit Float	
Channel 20 coeff 1	03,16	18738	32-bit Float	
Channel 21 coeff 1	03,16	18740	32-bit Float	
Channel 22 coeff 1	03,16	18742	32-bit Float	
Channel 23 coeff 1	03,16	18744	32-bit Float	
Channel 24 coeff 1	03,16	18746	32-bit Float	
Channel 1 op 3	03,06,16	18800	16-bit unsign int	none(0), multiplication(1), division(2), sum(3), subtract(4)
Channel 2 op 3	03,06,16	18801	16-bit unsign int	—"—
Channel 3 op 3	03,06,16	18802	16-bit unsign int	—"—
Channel 4 op 3	03,06,16	18803	16-bit unsign int	—"—
Channel 5 op 3	03,06,16	18804	16-bit unsign int	—"—
Channel 6 op 3	03,06,16	18805	16-bit unsign int	—"—
Channel 7 op 3	03,06,16	18806	16-bit unsign int	—"—
Channel 8 op 3	03,06,16	18807	16-bit unsign int	—"—
Channel 9 op 3	03,06,16	18808	16-bit unsign int	—"—
Channel 10 op 3	03,06,16	18809	16-bit unsign int	—"—
Channel 11 op 3	03,06,16	18810	16-bit unsign int	—"—
Channel 12 op 3	03,06,16	18811	16-bit unsign int	—"—

Channel 13 op 3	03,06,16	18812	16-bit unsign int	—"—
Channel 14 op 3	03,06,16	18813	16-bit unsign int	—"—
Channel 15 op 3	03,06,16	18814	16-bit unsign int	—"—
Channel 16 op 3	03,06,16	18815	16-bit unsign int	—"—
Channel 17 op 3	03,06,16	18816	16-bit unsign int	—"—
Channel 18 op 3	03,06,16	18817	16-bit unsign int	—"—
Channel 19 op 3	03,06,16	18818	16-bit unsign int	—"—
Channel 20 op 3	03,06,16	18819	16-bit unsign int	—"—
Channel 21 op 3	03,06,16	18820	16-bit unsign int	—"—
Channel 22 op 3	03,06,16	18821	16-bit unsign int	—"—
Channel 23 op 3	03,06,16	18822	16-bit unsign int	—"—
Channel 24 op 3	03,06,16	18823	16-bit unsign int	—"—
Channel 1 coeff 2	03,16	18900	32-bit Float	
Channel 2 coeff 2	03,16	18902	32-bit Float	
Channel 3 coeff 2	03,16	18904	32-bit Float	
Channel 4 coeff 2	03,16	18906	32-bit Float	
Channel 5 coeff 2	03,16	18908	32-bit Float	
Channel 6 coeff 2	03,16	18910	32-bit Float	
Channel 7 coeff 2	03,16	18912	32-bit Float	
Channel 8 coeff 2	03,16	18914	32-bit Float	
Channel 9 coeff 2	03,16	18916	32-bit Float	
Channel 10 coeff 2	03,16	18918	32-bit Float	
Channel 11 coeff 2	03,16	18920	32-bit Float	
Channel 12 coeff 2	03,16	18922	32-bit Float	
Channel 13 coeff 2	03,16	18924	32-bit Float	
Channel 14 coeff 2	03,16	18926	32-bit Float	
Channel 15 coeff 2	03,16	18928	32-bit Float	
Channel 16 coeff 2	03,16	18930	32-bit Float	
Channel 17 coeff 2	03,16	18932	32-bit Float	
Channel 18 coeff 2	03,16	18934	32-bit Float	
Channel 29 coeff 2	03,16	18936	32-bit Float	
Channel 20 coeff 2	03,16	18938	32-bit Float	
Channel 21 coeff 2	03,16	18940	32-bit Float	
Channel 22 coeff 2	03,16	18942	32-bit Float	
Channel 23 coeff 2	03,16	18944	32-bit Float	
Channel 24 coeff 2	03,16	18946	32-bit Float	
Channel 1 unit	03,16	19000	16 bytes UTF-8	
Channel 2 unit	03,16	19008	16 bytes UTF-8	
Channel 3 unit	03,16	19016	16 bytes UTF-8	
Channel 4 unit	03,16	19024	16 bytes UTF-8	
Channel 5 unit	03,16	19032	16 bytes UTF-8	
Channel 6 unit	03,16	19040	16 bytes UTF-8	
Channel 7 unit	03,16	19048	16 bytes UTF-8	
Channel 8 unit	03,16	19056	16 bytes UTF-8	
Channel 9 unit	03,16	19064	16 bytes UTF-8	
Channel 10 unit	03,16	19072	16 bytes UTF-8	

Channel 11 unit	03,16	19080	16 bytes UTF-8	
Channel 12 unit	03,16	19088	16 bytes UTF-8	
Channel 13 unit	03,16	19096	16 bytes UTF-8	
Channel 14 unit	03,16	19104	16 bytes UTF-8	
Channel 15 unit	03,16	19112	16 bytes UTF-8	
Channel 16 unit	03,16	19120	16 bytes UTF-8	
Channel 17 unit	03,16	19128	16 bytes UTF-8	
Channel 18 unit	03,16	19136	16 bytes UTF-8	
Channel 19 unit	03,16	19144	16 bytes UTF-8	
Channel 20 unit	03,16	19152	16 bytes UTF-8	
Channel 21 unit	03,16	19160	16 bytes UTF-8	
Channel 22 unit	03,16	19168	16 bytes UTF-8	
Channel 23 unit	03,16	19176	16 bytes UTF-8	
Channel 24 unit	03,16	19184	16 bytes UTF-8	
Channel 1 comulative value	03,16	19200	32-bit Float	
Channel 2 comulative value	03,16	19202	32-bit Float	
Channel 3 comulative value	03,16	19204	32-bit Float	
Channel 4 comulative value	03,16	19206	32-bit Float	
Channel 5 comulative value	03,16	19208	32-bit Float	
Channel 6 comulative value	03,16	19210	32-bit Float	
Channel 7 comulative value	03,16	19212	32-bit Float	
Channel 8 comulative value	03,16	19214	32-bit Float	
Channel 9 comulative value	03,16	19216	32-bit Float	
Channel 10 comulative value	03,16	19218	32-bit Float	
Channel 11 comulative value	03,16	19220	32-bit Float	
Channel 12 comulative value	03,16	19222	32-bit Float	
Channel 13 comulative value	03,16	19224	32-bit Float	
Channel 14 comulative value	03,16	19226	32-bit Float	
Channel 15 comulative value	03,16	19228	32-bit Float	
Channel 16 comulative value	03,16	19230	32-bit Float	
Channel 17 comulative value	03,16	19232	32-bit Float	
Channel 18 comulative value	03,16	19234	32-bit Float	
Channel 19 comulative value	03,16	19236	32-bit Float	
Channel 20 comulative value	03,16	19238	32-bit Float	
Channel 21 comulative value	03,16	19240	32-bit Float	
Channel 22 comulative value	03,16	19242	32-bit Float	
Channel 23 comulative value	03,16	19244	32-bit Float	
Channel 24 comulative value	03,16	19246	32-bit Float	
Channel 1 value	03	19300	32-bit Float	
Channel 2 value	03	19302	32-bit Float	
Channel 3 value	03	19304	32-bit Float	
Channel 4 value	03	19306	32-bit Float	
Channel 5 value	03	19308	32-bit Float	
Channel 6 value	03	19310	32-bit Float	
Channel 7 value	03	19312	32-bit Float	
Channel 8 value	03	19314	32-bit Float	

Channel 9 value	03	19316	32-bit Float	
Channel 10 value	03	19318	32-bit Float	
Channel 11 value	03	19320	32-bit Float	
Channel 12 value	03	19322	32-bit Float	
Channel 13 value	03	19324	32-bit Float	
Channel 14 value	03	19326	32-bit Float	
Channel 15 value	03	19328	32-bit Float	
Channel 16 value	03	19330	32-bit Float	
Channel 17 value	03	19332	32-bit Float	
Channel 18 value	03	19334	32-bit Float	
Channel 19 value	03	19336	32-bit Float	
Channel 20 value	03	19338	32-bit Float	
Channel 21 value	03	19340	32-bit Float	
Channel 22 value	03	19342	32-bit Float	
Channel 23 value	03	19344	32-bit Float	
Channel 24 value	03	19346	32-bit Float	
Channel 1 counter	03	19400	32-bit unsign int	
Channel 2 counter	03	19402	32-bit unsign int	
Channel 3 counter	03	19404	32-bit unsign int	
Channel 4 counter	03	19406	32-bit unsign int	
Channel 5 counter	03	19408	32-bit unsign int	
Channel 6 counter	03	19410	32-bit unsign int	
Channel 7 counter	03	19412	32-bit unsign int	
Channel 8 counter	03	19414	32-bit unsign int	
Channel 9 counter	03	19416	32-bit unsign int	
Channel 10 counter	03	19418	32-bit unsign int	
Channel 11 counter	03	19420	32-bit unsign int	
Channel 12 counter	03	19422	32-bit unsign int	
Channel 13 counter	03	19424	32-bit unsign int	
Channel 14 counter	03	19426	32-bit unsign int	
Channel 15 counter	03	19428	32-bit unsign int	
Channel 16 counter	03	19430	32-bit unsign int	
Channel 17 counter	03	19432	32-bit unsign int	
Channel 18 counter	03	19434	32-bit unsign int	
Channel 19 counter	03	19436	32-bit unsign int	
Channel 20 counter	03	19438	32-bit unsign int	
Channel 21 counter	03	19440	32-bit unsign int	
Channel 22 counter	03	19442	32-bit unsign int	
Channel 23 counter	03	19444	32-bit unsign int	
Channel 24 counter	03	19446	32-bit unsign int	
Channel 1 alarm status	03	19500	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Channel 2 alarm status	03	19501	16-bit unsign int	—"—
Channel 3 alarm status	03	19502	16-bit unsign int	—"—
Channel 4 alarm status	03	19503	16-bit unsign int	—"—

Channel 5 alarm status	03	19504	16-bit unsign int	—"—
Channel 6 alarm status	03	19505	16-bit unsign int	—"—
Channel 7 alarm status	03	19506	16-bit unsign int	—"—
Channel 8 alarm status	03	19507	16-bit unsign int	—"—
Channel 9 alarm status	03	19508	16-bit unsign int	—"—
Channel 10 alarm status	03	19509	16-bit unsign int	—"—
Channel 11 alarm status	03	19510	16-bit unsign int	—"—
Channel 12 alarm status	03	19511	16-bit unsign int	—"—
Channel 13 alarm status	03	19512	16-bit unsign int	—"—
Channel 14 alarm status	03	19513	16-bit unsign int	—"—
Channel 15 alarm status	03	19514	16-bit unsign int	—"—
Channel 16 alarm status	03	19515	16-bit unsign int	—"—
Channel 17 alarm status	03	19516	16-bit unsign int	—"—
Channel 18 alarm status	03	19517	16-bit unsign int	—"—
Channel 19 alarm status	03	19518	16-bit unsign int	—"—
Channel 20 alarm status	03	19519	16-bit unsign int	—"—
Channel 21 alarm status	03	19520	16-bit unsign int	—"—
Channel 22 alarm status	03	19521	16-bit unsign int	—"—
Channel 23 alarm status	03	19522	16-bit unsign int	—"—
Channel 24 alarm status	03	19523	16-bit unsign int	—"—
Alarm 1 description	03,16	20000	16 bytes UTF-8	
Alarm 2 description	03,16	20008	16 bytes UTF-8	
Alarm 3 description	03,16	20016	16 bytes UTF-8	
Alarm 4 description	03,16	20024	16 bytes UTF-8	
Alarm 5 description	03,16	20032	16 bytes UTF-8	
Alarm 6 description	03,16	20040	16 bytes UTF-8	
Alarm 7 description	03,16	20048	16 bytes UTF-8	
Alarm 8 description	03,16	20056	16 bytes UTF-8	
Alarm 9 description	03,16	20064	16 bytes UTF-8	
Alarm 10 description	03,16	20072	16 bytes UTF-8	
Alarm 11 description	03,16	20080	16 bytes UTF-8	
Alarm 12 description	03,16	20088	16 bytes UTF-8	
Alarm 13 description	03,16	20096	16 bytes UTF-8	
Alarm 14 description	03,16	20104	16 bytes UTF-8	
Alarm 15 description	03,16	20112	16 bytes UTF-8	
Alarm 16 description	03,16	20120	16 bytes UTF-8	
Alarm 17 description	03,16	20128	16 bytes UTF-8	
Alarm 18 description	03,16	20136	16 bytes UTF-8	
Alarm 19 description	03,16	20144	16 bytes UTF-8	
Alarm 20 description	03,16	20152	16 bytes UTF-8	
Alarm 21 description	03,16	20160	16 bytes UTF-8	
Alarm 22 description	03,16	20168	16 bytes UTF-8	
Alarm 23 description	03,16	20176	16 bytes UTF-8	
Alarm 24 description	03,16	20184	16 bytes UTF-8	
Alarm 1 condition 1 channel	03,06,16	20200	16-bit unsign int	none(0), v01(1), v02(2), v03(3), v04(4), v05(5), v06(6), v07(7), v08(8),

				v09(9), v10(10), v11(11), v12(12), v13(13), v14(14), v15(15), v16(16), v17(17), v18(18), v19(19), v20(20), v21(21), v22(22), v23(23), v24(24)
Alarm 2 condition 1 channel	03,06,16	20201	16-bit unsign int	—"—
Alarm 3 condition 1 channel	03,06,16	20202	16-bit unsign int	—"—
Alarm 4 condition 1 channel	03,06,16	20203	16-bit unsign int	—"—
Alarm 5 condition 1 channel	03,06,16	20204	16-bit unsign int	—"—
Alarm 6 condition 1 channel	03,06,16	20205	16-bit unsign int	—"—
Alarm 7 condition 1 channel	03,06,16	20206	16-bit unsign int	—"—
Alarm 8 condition 1 channel	03,06,16	20207	16-bit unsign int	—"—
Alarm 9 condition 1 channel	03,06,16	20208	16-bit unsign int	—"—
Alarm 10 condition 1 channel	03,06,16	20209	16-bit unsign int	—"—
Alarm 11 condition 1 channel	03,06,16	20210	16-bit unsign int	—"—
Alarm 12 condition 1 channel	03,06,16	20211	16-bit unsign int	—"—
Alarm 13 condition 1 channel	03,06,16	20212	16-bit unsign int	—"—
Alarm 14 condition 1 channel	03,06,16	20213	16-bit unsign int	—"—
Alarm 15 condition 1 channel	03,06,16	20214	16-bit unsign int	—"—
Alarm 16 condition 1 channel	03,06,16	20215	16-bit unsign int	—"—
Alarm 17 condition 1 channel	03,06,16	20216	16-bit unsign int	—"—
Alarm 18 condition 1 channel	03,06,16	20217	16-bit unsign int	—"—
Alarm 19 condition 1 channel	03,06,16	20218	16-bit unsign int	—"—
Alarm 20 condition 1 channel	03,06,16	20219	16-bit unsign int	—"—
Alarm 21 condition 1 channel	03,06,16	20220	16-bit unsign int	—"—
Alarm 22 condition 1 channel	03,06,16	20221	16-bit unsign int	—"—
Alarm 23 condition 1 channel	03,06,16	20222	16-bit unsign int	—"—
Alarm 24 condition 1 channel	03,06,16	20223	16-bit unsign int	—"—
Alarm 1 condition 1 operand	03,06,16	20300	16-bit unsign int	larger(0), less(1)
Alarm 2 condition 1 operand	03,06,16	20301	16-bit unsign int	larger(0), less(1)
Alarm 3 condition 1 operand	03,06,16	20302	16-bit unsign int	larger(0), less(1)
Alarm 4 condition 1 operand	03,06,16	20303	16-bit unsign int	larger(0), less(1)
Alarm 5 condition 1 operand	03,06,16	20304	16-bit unsign int	larger(0), less(1)
Alarm 6 condition 1 operand	03,06,16	20305	16-bit unsign int	larger(0), less(1)
Alarm 7 condition 1 operand	03,06,16	20306	16-bit unsign int	larger(0), less(1)
Alarm 8 condition 1 operand	03,06,16	20307	16-bit unsign int	larger(0), less(1)
Alarm 9 condition 1 operand	03,06,16	20308	16-bit unsign int	larger(0), less(1)
Alarm 10 condition 1 operand	03,06,16	20309	16-bit unsign int	larger(0), less(1)
Alarm 11 condition 1 operand	03,06,16	20310	16-bit unsign int	larger(0), less(1)
Alarm 12 condition 1 operand	03,06,16	20311	16-bit unsign int	larger(0), less(1)
Alarm 13 condition 1 operand	03,06,16	20312	16-bit unsign int	larger(0), less(1)
Alarm 14 condition 1 operand	03,06,16	20313	16-bit unsign int	larger(0), less(1)
Alarm 15 condition 1 operand	03,06,16	20314	16-bit unsign int	larger(0), less(1)
Alarm 16 condition 1 operand	03,06,16	20315	16-bit unsign int	larger(0), less(1)
Alarm 17 condition 1 operand	03,06,16	20316	16-bit unsign int	larger(0), less(1)
Alarm 18 condition 1 operand	03,06,16	20317	16-bit unsign int	larger(0), less(1)
Alarm 19 condition 1 operand	03,06,16	20318	16-bit unsign int	larger(0), less(1)

Alarm 20 condition 1 operand	03,06,16	20319	16-bit unsign int	larger(0), less(1)
Alarm 21 condition 1 operand	03,06,16	20320	16-bit unsign int	larger(0), less(1)
Alarm 22 condition 1 operand	03,06,16	20321	16-bit unsign int	larger(0), less(1)
Alarm 23 condition 1 operand	03,06,16	20322	16-bit unsign int	larger(0), less(1)
Alarm 24 condition 1 operand	03,06,16	20323	16-bit unsign int	larger(0), less(1)
Alarm 1 condition 1 limit	03,16	20400	32-bit Float	
Alarm 2 condition 1 limit	03,16	20402	32-bit Float	
Alarm 3 condition 1 limit	03,16	20404	32-bit Float	
Alarm 4 condition 1 limit	03,16	20406	32-bit Float	
Alarm 5 condition 1 limit	03,16	20408	32-bit Float	
Alarm 6 condition 1 limit	03,16	20410	32-bit Float	
Alarm 7 condition 1 limit	03,16	20412	32-bit Float	
Alarm 8 condition 1 limit	03,16	20414	32-bit Float	
Alarm 9 condition 1 limit	03,16	20416	32-bit Float	
Alarm 10 condition 1 limit	03,16	20418	32-bit Float	
Alarm 11 condition 1 limit	03,16	20420	32-bit Float	
Alarm 12 condition 1 limit	03,16	20422	32-bit Float	
Alarm 13 condition 1 limit	03,16	20424	32-bit Float	
Alarm 14 condition 1 limit	03,16	20426	32-bit Float	
Alarm 15 condition 1 limit	03,16	20428	32-bit Float	
Alarm 16 condition 1 limit	03,16	20430	32-bit Float	
Alarm 17 condition 1 limit	03,16	20432	32-bit Float	
Alarm 18 condition 1 limit	03,16	20434	32-bit Float	
Alarm 19 condition 1 limit	03,16	20436	32-bit Float	
Alarm 20 condition 1 limit	03,16	20438	32-bit Float	
Alarm 21 condition 1 limit	03,16	20440	32-bit Float	
Alarm 22 condition 1 limit	03,16	20442	32-bit Float	
Alarm 23 condition 1 limit	03,16	20444	32-bit Float	
Alarm 23 condition 1 limit	03,16	20446	32-bit Float	
Alarm 1 condition 1 hysteresis	03,16	20500	32-bit Float	
Alarm 2 condition 1 hysteresis	03,16	20502	32-bit Float	
Alarm 3 condition 1 hysteresis	03,16	20504	32-bit Float	
Alarm 4 condition 1 hysteresis	03,16	20506	32-bit Float	
Alarm 5 condition 1 hysteresis	03,16	20508	32-bit Float	
Alarm 6 condition 1 hysteresis	03,16	20510	32-bit Float	
Alarm 7 condition 1 hysteresis	03,16	20512	32-bit Float	
Alarm 8 condition 1 hysteresis	03,16	20514	32-bit Float	
Alarm 9 condition 1 hysteresis	03,16	20516	32-bit Float	
Alarm 10 condition 1 hysteresis	03,16	20518	32-bit Float	
Alarm 11 condition 1 hysteresis	03,16	20520	32-bit Float	
Alarm 12 condition 1 hysteresis	03,16	20522	32-bit Float	
Alarm 13 condition 1 hysteresis	03,16	20524	32-bit Float	

Alarm 14 condition 1 hysteresis	03,16	20526	32-bit Float	
Alarm 15 condition 1 hysteresis	03,16	20528	32-bit Float	
Alarm 16 condition 1 hysteresis	03,16	20530	32-bit Float	
Alarm 17 condition 1 hysteresis	03,16	20532	32-bit Float	
Alarm 18 condition 1 hysteresis	03,16	20534	32-bit Float	
Alarm 19 condition 1 hysteresis	03,16	20536	32-bit Float	
Alarm 20 condition 1 hysteresis	03,16	20538	32-bit Float	
Alarm 21 condition 1 hysteresis	03,16	20540	32-bit Float	
Alarm 22 condition 1 hysteresis	03,16	20542	32-bit Float	
Alarm 23 condition 1 hysteresis	03,16	20544	32-bit Float	
Alarm 24 condition 1 hysteresis	03,16	20546	32-bit Float	
Alarm 1 condition 1 discrete al. state	03,06,16	20600	16-bit unsign int	open(0), closed(1)
Alarm 2 condition 1 discrete al. state	03,06,16	20601	16-bit unsign int	open(0), closed(1)
Alarm 3 condition 1 discrete al. state	03,06,16	20602	16-bit unsign int	open(0), closed(1)
Alarm 4 condition 1 discrete al. state	03,06,16	20603	16-bit unsign int	open(0), closed(1)
Alarm 5 condition 1 discrete al. state	03,06,16	20604	16-bit unsign int	open(0), closed(1)
Alarm 6 condition 1 discrete al. state	03,06,16	20605	16-bit unsign int	open(0), closed(1)
Alarm 7 condition 1 discrete al. state	03,06,16	20606	16-bit unsign int	open(0), closed(1)
Alarm 8 condition 1 discrete al. state	03,06,16	20607	16-bit unsign int	open(0), closed(1)
Alarm 9 condition 1 discrete al. state	03,06,16	20608	16-bit unsign int	open(0), closed(1)
Alarm 10 condition 1 discrete al. state	03,06,16	20609	16-bit unsign int	open(0), closed(1)
Alarm 11 condition 1 discrete al. state	03,06,16	20610	16-bit unsign int	open(0), closed(1)
Alarm 12 condition 1 discrete al. state	03,06,16	20611	16-bit unsign int	open(0), closed(1)
Alarm 13 condition 1 discrete al. state	03,06,16	20612	16-bit unsign int	open(0), closed(1)
Alarm 14 condition 1 discrete al. state	03,06,16	20613	16-bit unsign int	open(0), closed(1)
Alarm 15 condition 1 discrete al. state	03,06,16	20614	16-bit unsign int	open(0), closed(1)
Alarm 16 condition 1 discrete al. state	03,06,16	20615	16-bit unsign int	open(0), closed(1)

Alarm 17 condition 1 discrete al. state	03,06,16	20616	16-bit unsign int	open(0), closed(1)
Alarm 18 condition 1 discrete al. state	03,06,16	20617	16-bit unsign int	open(0), closed(1)
Alarm 19 condition 1 discrete al. state	03,06,16	20618	16-bit unsign int	open(0), closed(1)
Alarm 20 condition 1 discrete al. state	03,06,16	20619	16-bit unsign int	open(0), closed(1)
Alarm 21 condition 1 discrete al. state	03,06,16	20620	16-bit unsign int	open(0), closed(1)
Alarm 22 condition 1 discrete al. state	03,06,16	20621	16-bit unsign int	open(0), closed(1)
Alarm 23 condition 1 discrete al. state	03,06,16	20622	16-bit unsign int	open(0), closed(1)
Alarm 24 condition 1 discrete al. state	03,06,16	20623	16-bit unsign int	open(0), closed(1)
Alarm 1 function	03,06,16	20700	16-bit unsign int	none(0}, and(1), or(2)
Alarm 2 function	03,06,16	20701	16-bit unsign int	none(0}, and(1), or(2)
Alarm 3 function	03,06,16	20702	16-bit unsign int	none(0}, and(1), or(2)
Alarm 4 function	03,06,16	20703	16-bit unsign int	none(0}, and(1), or(2)
Alarm 5 function	03,06,16	20704	16-bit unsign int	none(0}, and(1), or(2)
Alarm 6 function	03,06,16	20705	16-bit unsign int	none(0}, and(1), or(2)
Alarm 7 function	03,06,16	20706	16-bit unsign int	none(0}, and(1), or(2)
Alarm 8 function	03,06,16	20707	16-bit unsign int	none(0}, and(1), or(2)
Alarm 9 function	03,06,16	20708	16-bit unsign int	none(0}, and(1), or(2)
Alarm 10 function	03,06,16	20709	16-bit unsign int	none(0}, and(1), or(2)
Alarm 11 function	03,06,16	20710	16-bit unsign int	none(0}, and(1), or(2)
Alarm 12 function	03,06,16	20711	16-bit unsign int	none(0}, and(1), or(2)
Alarm 13 function	03,06,16	20712	16-bit unsign int	none(0}, and(1), or(2)
Alarm 14 function	03,06,16	20713	16-bit unsign int	none(0}, and(1), or(2)
Alarm 15 function	03,06,16	20714	16-bit unsign int	none(0}, and(1), or(2)
Alarm 16 function	03,06,16	20715	16-bit unsign int	none(0}, and(1), or(2)
Alarm 17 function	03,06,16	20716	16-bit unsign int	none(0}, and(1), or(2)
Alarm 18 function	03,06,16	20717	16-bit unsign int	none(0}, and(1), or(2)
Alarm 19 function	03,06,16	20718	16-bit unsign int	none(0}, and(1), or(2)
Alarm 20 function	03,06,16	20719	16-bit unsign int	none(0}, and(1), or(2)
Alarm 21 function	03,06,16	20720	16-bit unsign int	none(0}, and(1), or(2)
Alarm 22 function	03,06,16	20721	16-bit unsign int	none(0}, and(1), or(2)
Alarm 23 function	03,06,16	20722	16-bit unsign int	none(0}, and(1), or(2)
Alarm 24 function	03,06,16	20723	16-bit unsign int	none(0}, and(1), or(2)
Alarm 1 condition 2 channel	03,06,16	20800	16-bit unsign int	none(0), v01(1), v02(2), v03(3), v04(4), v05(5), v06(6), v07(7), v08(8), v09(9), v10(10), v11(11), v12(12), v13(13), v14(14), v15(15), v16(16), v17(17), v18(18), v19(19), v20(20), v21(21), v22(22), v23(23), v24(24)

Alarm 2 condition 2 channel	03,06,16	20801	16-bit unsign int	—"—
Alarm 3 condition 2 channel	03,06,16	20802	16-bit unsign int	—"—
Alarm 4 condition 2 channel	03,06,16	20803	16-bit unsign int	—"—
Alarm 5 condition 2 channel	03,06,16	20804	16-bit unsign int	—"—
Alarm 6 condition 2 channel	03,06,16	20805	16-bit unsign int	—"—
Alarm 7 condition 2 channel	03,06,16	20806	16-bit unsign int	—"—
Alarm 8 condition 2 channel	03,06,16	20807	16-bit unsign int	—"—
Alarm 9 condition 2 channel	03,06,16	20808	16-bit unsign int	—"—
Alarm 10 condition 2 channel	03,06,16	20809	16-bit unsign int	—"—
Alarm 11 condition 2 channel	03,06,16	20810	16-bit unsign int	—"—
Alarm 12 condition 2 channel	03,06,16	20811	16-bit unsign int	—"—
Alarm 13 condition 2 channel	03,06,16	20812	16-bit unsign int	—"—
Alarm 14 condition 2 channel	03,06,16	20813	16-bit unsign int	—"—
Alarm 15 condition 2 channel	03,06,16	20814	16-bit unsign int	—"—
Alarm 16 condition 2 channel	03,06,16	20815	16-bit unsign int	—"—
Alarm 17 condition 2 channel	03,06,16	20816	16-bit unsign int	—"—
Alarm 18 condition 2 channel	03,06,16	20817	16-bit unsign int	—"—
Alarm 19 condition 2 channel	03,06,16	20818	16-bit unsign int	—"—
Alarm 20 condition 2 channel	03,06,16	20819	16-bit unsign int	—"—
Alarm 21 condition 2 channel	03,06,16	20820	16-bit unsign int	—"—
Alarm 22 condition 2 channel	03,06,16	20821	16-bit unsign int	—"—
Alarm 23 condition 2 channel	03,06,16	20822	16-bit unsign int	—"—
Alarm 24 condition 2 channel	03,06,16	20823	16-bit unsign int	—"—
Alarm 1 condition 2 operand	03,06,16	20900	16-bit unsign int	larger(0), less(1)
Alarm 2 condition 2 operand	03,06,16	20901	16-bit unsign int	larger(0), less(1)
Alarm 3 condition 2 operand	03,06,16	20902	16-bit unsign int	larger(0), less(1)
Alarm 4 condition 2 operand	03,06,16	20903	16-bit unsign int	larger(0), less(1)
Alarm 5 condition 2 operand	03,06,16	20904	16-bit unsign int	larger(0), less(1)
Alarm 6 condition 2 operand	03,06,16	20905	16-bit unsign int	larger(0), less(1)
Alarm 7 condition 2 operand	03,06,16	20906	16-bit unsign int	larger(0), less(1)
Alarm 8 condition 2 operand	03,06,16	20907	16-bit unsign int	larger(0), less(1)
Alarm 9 condition 2 operand	03,06,16	20908	16-bit unsign int	larger(0), less(1)
Alarm 10 condition 2 operand	03,06,16	20909	16-bit unsign int	larger(0), less(1)
Alarm 11 condition 2 operand	03,06,16	20910	16-bit unsign int	larger(0), less(1)
Alarm 12 condition 2 operand	03,06,16	20911	16-bit unsign int	larger(0), less(1)
Alarm 13 condition 2 operand	03,06,16	20912	16-bit unsign int	larger(0), less(1)
Alarm 14 condition 2 operand	03,06,16	20913	16-bit unsign int	larger(0), less(1)
Alarm 15 condition 2 operand	03,06,16	20914	16-bit unsign int	larger(0), less(1)
Alarm 16 condition 2 operand	03,06,16	20915	16-bit unsign int	larger(0), less(1)
Alarm 17 condition 2 operand	03,06,16	20916	16-bit unsign int	larger(0), less(1)
Alarm 18 condition 2 operand	03,06,16	20917	16-bit unsign int	larger(0), less(1)
Alarm 19 condition 2 operand	03,06,16	20918	16-bit unsign int	larger(0), less(1)
Alarm 20 condition 2 operand	03,06,16	20919	16-bit unsign int	larger(0), less(1)
Alarm 21 condition 2 operand	03,06,16	20920	16-bit unsign int	larger(0), less(1)
Alarm 22 condition 2 operand	03,06,16	20921	16-bit unsign int	larger(0), less(1)
Alarm 23 condition 2 operand	03,06,16	20922	16-bit unsign int	larger(0), less(1)
Alarm 24 condition 2 operand	03,06,16	20923	16-bit unsign int	larger(0), less(1)

Alarm 1 condition 2 limit	03,16	21000	32-bit Float	
Alarm 2 condition 2 limit	03,16	21002	32-bit Float	
Alarm 3 condition 2 limit	03,16	21004	32-bit Float	
Alarm 4 condition 2 limit	03,16	21006	32-bit Float	
Alarm 5 condition 2 limit	03,16	21008	32-bit Float	
Alarm 6 condition 2 limit	03,16	21010	32-bit Float	
Alarm 7 condition 2 limit	03,16	21012	32-bit Float	
Alarm 8 condition 2 limit	03,16	21014	32-bit Float	
Alarm 9 condition 2 limit	03,16	21016	32-bit Float	
Alarm 10 condition 2 limit	03,16	21018	32-bit Float	
Alarm 11 condition 2 limit	03,16	21020	32-bit Float	
Alarm 12 condition 2 limit	03,16	21022	32-bit Float	
Alarm 13 condition 2 limit	03,16	21024	32-bit Float	
Alarm 14 condition 2 limit	03,16	21026	32-bit Float	
Alarm 15 condition 2 limit	03,16	21028	32-bit Float	
Alarm 16 condition 2 limit	03,16	21030	32-bit Float	
Alarm 17 condition 2 limit	03,16	21032	32-bit Float	
Alarm 18 condition 2 limit	03,16	21034	32-bit Float	
Alarm 19 condition 2 limit	03,16	21036	32-bit Float	
Alarm 20 condition 2 limit	03,16	21038	32-bit Float	
Alarm 21 condition 2 limit	03,16	21040	32-bit Float	
Alarm 22 condition 2 limit	03,16	21042	32-bit Float	
Alarm 23 condition 2 limit	03,16	21044	32-bit Float	
Alarm 24 condition 2 limit	03,16	21046	32-bit Float	
Alarm 1 condition 2 hysteresis	03,16	21100	32-bit Float	
Alarm 2 condition 2 hysteresis	03,16	21102	32-bit Float	
Alarm 3 condition 2 hysteresis	03,16	21104	32-bit Float	
Alarm 4 condition 2 hysteresis	03,16	21106	32-bit Float	
Alarm 5 condition 2 hysteresis	03,16	21108	32-bit Float	
Alarm 6 condition 2 hysteresis	03,16	21110	32-bit Float	
Alarm 7 condition 2 hysteresis	03,16	21112	32-bit Float	
Alarm 8 condition 2 hysteresis	03,16	21114	32-bit Float	
Alarm 9 condition 2 hysteresis	03,16	21116	32-bit Float	
Alarm 10 condition 2 hysteresis	03,16	21118	32-bit Float	
Alarm 11 condition 2 hysteresis	03,16	21120	32-bit Float	
Alarm 12 condition 2 hysteresis	03,16	21122	32-bit Float	
Alarm 13 condition 2 hysteresis	03,16	21124	32-bit Float	
Alarm 14 condition 2 hysteresis	03,16	21126	32-bit Float	
Alarm 15 condition 2 hysteresis	03,16	21128	32-bit Float	
Alarm 16 condition 2 hysteresis	03,16	21130	32-bit Float	
Alarm 17 condition 2 hysteresis	03,16	21132	32-bit Float	

Alarm 18 condition 2 hysteresis	03,16	21134	32-bit Float	
Alarm 19 condition 2 hysteresis	03,16	21136	32-bit Float	
Alarm 20 condition 2 hysteresis	03,16	21138	32-bit Float	
Alarm 21 condition 2 hysteresis	03,16	21140	32-bit Float	
Alarm 22 condition 2 hysteresis	03,16	21142	32-bit Float	
Alarm 23 condition 2 hysteresis	03,16	21144	32-bit Float	
Alarm 24 condition 2 hysteresis	03,16	21146	32-bit Float	
Alarm 1 condition 2 discrete al. state	03,06,16	21200	16-bit unsign int	open(0), closed(1)
Alarm 2 condition 2 discrete al. state	03,06,16	21201	16-bit unsign int	open(0), closed(1)
Alarm 3 condition 2 discrete al. state	03,06,16	21202	16-bit unsign int	open(0), closed(1)
Alarm 4 condition 2 discrete al. state	03,06,16	21203	16-bit unsign int	open(0), closed(1)
Alarm 5 condition 2 discrete al. state	03,06,16	21204	16-bit unsign int	open(0), closed(1)
Alarm 6 condition 2 discrete al. state	03,06,16	21205	16-bit unsign int	open(0), closed(1)
Alarm 7 condition 2 discrete al. state	03,06,16	21206	16-bit unsign int	open(0), closed(1)
Alarm 8 condition 2 discrete al. state	03,06,16	21207	16-bit unsign int	open(0), closed(1)
Alarm 9 condition 2 discrete al. state	03,06,16	21208	16-bit unsign int	open(0), closed(1)
Alarm 10 condition 2 discrete al. state	03,06,16	21209	16-bit unsign int	open(0), closed(1)
Alarm 11 condition 2 discrete al. state	03,06,16	21210	16-bit unsign int	open(0), closed(1)
Alarm 12 condition 2 discrete al. state	03,06,16	21211	16-bit unsign int	open(0), closed(1)
Alarm 13 condition 2 discrete al. state	03,06,16	21212	16-bit unsign int	open(0), closed(1)
Alarm 14 condition 2 discrete al. state	03,06,16	21213	16-bit unsign int	open(0), closed(1)
Alarm 15 condition 2 discrete al. state	03,06,16	21214	16-bit unsign int	open(0), closed(1)
Alarm 16 condition 2 discrete al. state	03,06,16	21215	16-bit unsign int	open(0), closed(1)
Alarm 17 condition 2 discrete al. state	03,06,16	21216	16-bit unsign int	open(0), closed(1)
Alarm 18 condition 2 discrete al. state	03,06,16	21217	16-bit unsign int	open(0), closed(1)
Alarm 19 condition 2 discrete al. state	03,06,16	21218	16-bit unsign int	open(0), closed(1)
Alarm 20 condition 2 discrete al. state	03,06,16	21219	16-bit unsign int	open(0), closed(1)

Alarm 21 condition 2 discrete al. state	03,06,16	21220	16-bit unsign int	open(0), closed(1)
Alarm 22 condition 2 discrete al. state	03,06,16	21221	16-bit unsign int	open(0), closed(1)
Alarm 23 condition 2 discrete al. state	03,06,16	21222	16-bit unsign int	open(0), closed(1)
Alarm 24 condition 2 discrete al. state	03,06,16	21223	16-bit unsign int	open(0), closed(1)
Alarm 1 type	03,06,16	21300	16-bit unsign int	warning(3), minor(4), major(5), critical(6)
Alarm 2 type	03,06,16	21301	16-bit unsign int	—"—
Alarm 3 type	03,06,16	21302	16-bit unsign int	—"—
Alarm 4 type	03,06,16	21303	16-bit unsign int	—"—
Alarm 5 type	03,06,16	21304	16-bit unsign int	—"—
Alarm 6 type	03,06,16	21305	16-bit unsign int	—"—
Alarm 7 type	03,06,16	21306	16-bit unsign int	—"—
Alarm 8 type	03,06,16	21307	16-bit unsign int	—"—
Alarm 9 type	03,06,16	21308	16-bit unsign int	—"—
Alarm 10 type	03,06,16	21309	16-bit unsign int	—"—
Alarm 11 type	03,06,16	21310	16-bit unsign int	—"—
Alarm 12 type	03,06,16	21311	16-bit unsign int	—"—
Alarm 13 type	03,06,16	21312	16-bit unsign int	—"—
Alarm 14 type	03,06,16	21313	16-bit unsign int	—"—
Alarm 15 type	03,06,16	21314	16-bit unsign int	—"—
Alarm 16 type	03,06,16	21315	16-bit unsign int	—"—
Alarm 17 type	03,06,16	21316	16-bit unsign int	—"—
Alarm 18 type	03,06,16	21317	16-bit unsign int	—"—
Alarm 19 type	03,06,16	21318	16-bit unsign int	—"—
Alarm 20 type	03,06,16	21319	16-bit unsign int	—"—
Alarm 21 type	03,06,16	21320	16-bit unsign int	—"—
Alarm 22 type	03,06,16	21321	16-bit unsign int	—"—
Alarm 23 type	03,06,16	21322	16-bit unsign int	—"—
Alarm 24 type	03,06,16	21323	16-bit unsign int	—"—
Alarm 1 assigned to channel	03,06,16	21400	16-bit unsign int	none(0), v01(1), v02(2), v03(3), v04(4), v05(5), v06(6), v07(7), v08(8),v09(9), v10(10), v11(11), v12(12), v13(13), v14(14), v15(15), v16(16),v17(17), v18(18), v19(19), v20(20), v21(21), v22(22), v23(23), v24(24)
Alarm 2 assigned to channel	03,06,16	21401	16-bit unsign int	—"—
Alarm 3 assigned to channel	03,06,16	21402	16-bit unsign int	—"—
Alarm 4 assigned to channel	03,06,16	21403	16-bit unsign int	—"—
Alarm 5 assigned to channel	03,06,16	21404	16-bit unsign int	—"—
Alarm 6 assigned to channel	03,06,16	21405	16-bit unsign int	—"—
Alarm 7 assigned to channel	03,06,16	21406	16-bit unsign int	—"—
Alarm 8 assigned to channel	03,06,16	21407	16-bit unsign int	—"—

Alarm 9 assigned to channel	03,06,16	21408	16-bit unsign int	—"—
Alarm 10 assigned to channel	03,06,16	21409	16-bit unsign int	—"—
Alarm 11 assigned to channel	03,06,16	21410	16-bit unsign int	—"—
Alarm 12 assigned to channel	03,06,16	21411	16-bit unsign int	—"—
Alarm 13 assigned to channel	03,06,16	21412	16-bit unsign int	—"—
Alarm 14 assigned to channel	03,06,16	21413	16-bit unsign int	—"—
Alarm 15 assigned to channel	03,06,16	21414	16-bit unsign int	—"—
Alarm 16 assigned to channel	03,06,16	21415	16-bit unsign int	—"—
Alarm 17 assigned to channel	03,06,16	21416	16-bit unsign int	—"—
Alarm 18 assigned to channel	03,06,16	21417	16-bit unsign int	—"—
Alarm 19 assigned to channel	03,06,16	21418	16-bit unsign int	—"—
Alarm 20 assigned to channel	03,06,16	21419	16-bit unsign int	—"—
Alarm 21 assigned to channel	03,06,16	21420	16-bit unsign int	—"—
Alarm 22 assigned to channel	03,06,16	21421	16-bit unsign int	—"—
Alarm 23 assigned to channel	03,06,16	21422	16-bit unsign int	—"—
Alarm 24 assigned to channel	03,06,16	21423	16-bit unsign int	—"—
Alarm 1 delay	03,16	21500	32-bit Float	
Alarm 2 delay	03,16	21502	32-bit Float	
Alarm 3 delay	03,16	21504	32-bit Float	
Alarm 4 delay	03,16	21506	32-bit Float	
Alarm 5 delay	03,16	21508	32-bit Float	
Alarm 6 delay	03,16	21510	32-bit Float	
Alarm 7 delay	03,16	21512	32-bit Float	
Alarm 8 delay	03,16	21514	32-bit Float	
Alarm 9 delay	03,16	21516	32-bit Float	
Alarm 10 delay	03,16	21518	32-bit Float	
Alarm 11 delay	03,16	21520	32-bit Float	
Alarm 12 delay	03,16	21522	32-bit Float	
Alarm 13 delay	03,16	21524	32-bit Float	
Alarm 14 delay	03,16	21526	32-bit Float	
Alarm 15 delay	03,16	21528	32-bit Float	
Alarm 16 delay	03,16	21530	32-bit Float	
Alarm 17 delay	03,16	21532	32-bit Float	
Alarm 18 delay	03,16	21534	32-bit Float	
Alarm 19 delay	03,16	21536	32-bit Float	
Alarm 20 delay	03,16	21538	32-bit Float	
Alarm 21 delay	03,16	21540	32-bit Float	
Alarm 22 delay	03,16	21542	32-bit Float	
Alarm 23 delay	03,16	21544	32-bit Float	
Alarm 24 delay	03,16	21546	32-bit Float	
Alarm 1 action on return	03,06,16	21600	16-bit unsign int	no(0), yes(1)
Alarm 2 action on return	03,06,16	21601	16-bit unsign int	no(0), yes(1)
Alarm 3 action on return	03,06,16	21602	16-bit unsign int	no(0), yes(1)
Alarm 4 action on return	03,06,16	21603	16-bit unsign int	no(0), yes(1)
Alarm 5 action on return	03,06,16	21604	16-bit unsign int	no(0), yes(1)
Alarm 6 action on return	03,06,16	21605	16-bit unsign int	no(0), yes(1)

Alarm 7 action on return	03,06,16	21606	16-bit unsign int	no(0), yes(1)
Alarm 8 action on return	03,06,16	21607	16-bit unsign int	no(0), yes(1)
Alarm 9 action on return	03,06,16	21608	16-bit unsign int	no(0), yes(1)
Alarm 10 action on return	03,06,16	21609	16-bit unsign int	no(0), yes(1)
Alarm 11 action on return	03,06,16	21610	16-bit unsign int	no(0), yes(1)
Alarm 12 action on return	03,06,16	21611	16-bit unsign int	no(0), yes(1)
Alarm 13 action on return	03,06,16	21612	16-bit unsign int	no(0), yes(1)
Alarm 14 action on return	03,06,16	21613	16-bit unsign int	no(0), yes(1)
Alarm 15 action on return	03,06,16	21614	16-bit unsign int	no(0), yes(1)
Alarm 16 action on return	03,06,16	21615	16-bit unsign int	no(0), yes(1)
Alarm 17 action on return	03,06,16	21616	16-bit unsign int	no(0), yes(1)
Alarm 18 action on return	03,06,16	21617	16-bit unsign int	no(0), yes(1)
Alarm 19 action on return	03,06,16	21618	16-bit unsign int	no(0), yes(1)
Alarm 20 action on return	03,06,16	21619	16-bit unsign int	no(0), yes(1)
Alarm 21 action on return	03,06,16	21620	16-bit unsign int	no(0), yes(1)
Alarm 22 action on return	03,06,16	21621	16-bit unsign int	no(0), yes(1)
Alarm 23 action on return	03,06,16	21622	16-bit unsign int	no(0), yes(1)
Alarm 24 action on return	03,06,16	21623	16-bit unsign int	no(0), yes(1)
Alarm 1 action 1	03,06,16	21700	16-bit unsign int	none(0), trapcond1(1), trapcond2(2), trapcond1and2(3), postiostate(4), mqttpublish(6)
Alarm 2 action 1	03,06,16	21701	16-bit unsign int	—"—
Alarm 3 action 1	03,06,16	21702	16-bit unsign int	—"—
Alarm 4 action 1	03,06,16	21703	16-bit unsign int	—"—
Alarm 5 action 1	03,06,16	21704	16-bit unsign int	—"—
Alarm 6 action 1	03,06,16	21705	16-bit unsign int	—"—
Alarm 7 action 1	03,06,16	21706	16-bit unsign int	—"—
Alarm 8 action 1	03,06,16	21707	16-bit unsign int	—"—
Alarm 9 action 1	03,06,16	21708	16-bit unsign int	—"—
Alarm 10 action 1	03,06,16	21709	16-bit unsign int	—"—
Alarm 11 action 1	03,06,16	21710	16-bit unsign int	—"—
Alarm 12 action 1	03,06,16	21711	16-bit unsign int	—"—
Alarm 13 action 1	03,06,16	21712	16-bit unsign int	—"—
Alarm 14 action 1	03,06,16	21713	16-bit unsign int	—"—
Alarm 15 action 1	03,06,16	21714	16-bit unsign int	—"—
Alarm 16 action 1	03,06,16	21715	16-bit unsign int	—"—
Alarm 17 action 1	03,06,16	21716	16-bit unsign int	—"—
Alarm 18 action 1	03,06,16	21717	16-bit unsign int	—"—
Alarm 19 action 1	03,06,16	21718	16-bit unsign int	—"—
Alarm 20 action 1	03,06,16	21719	16-bit unsign int	—"—
Alarm 21 action 1	03,06,16	21720	16-bit unsign int	—"—
Alarm 22 action 1	03,06,16	21721	16-bit unsign int	—"—
Alarm 23 action 1	03,06,16	21722	16-bit unsign int	—"—
Alarm 24 action 1	03,06,16	21723	16-bit unsign int	—"—

Alarm 1 action 2	03,06,16	21800	16-bit unsign int	none(0), trapcond1(1), trapcond2(2), trapcond1and2(3), postiostate(4), mqttpublish(6)
Alarm 2 action 2	03,06,16	21801	16-bit unsign int	—"—
Alarm 3 action 2	03,06,16	21802	16-bit unsign int	—"—
Alarm 4 action 2	03,06,16	21803	16-bit unsign int	—"—
Alarm 5 action 2	03,06,16	21804	16-bit unsign int	—"—
Alarm 6 action 2	03,06,16	21805	16-bit unsign int	—"—
Alarm 7 action 2	03,06,16	21806	16-bit unsign int	—"—
Alarm 8 action 2	03,06,16	21807	16-bit unsign int	—"—
Alarm 9 action 2	03,06,16	21808	16-bit unsign int	—"—
Alarm 10 action 2	03,06,16	21809	16-bit unsign int	—"—
Alarm 11 action 2	03,06,16	21810	16-bit unsign int	—"—
Alarm 12 action 2	03,06,16	21811	16-bit unsign int	—"—
Alarm 13 action 2	03,06,16	21812	16-bit unsign int	—"—
Alarm 14 action 2	03,06,16	21813	16-bit unsign int	—"—
Alarm 15 action 2	03,06,16	21814	16-bit unsign int	—"—
Alarm 16 action 2	03,06,16	21815	16-bit unsign int	—"—
Alarm 17 action 2	03,06,16	21816	16-bit unsign int	—"—
Alarm 18 action 2	03,06,16	21817	16-bit unsign int	—"—
Alarm 19 action 2	03,06,16	21818	16-bit unsign int	—"—
Alarm 20 action 2	03,06,16	21819	16-bit unsign int	—"—
Alarm 21 action 2	03,06,16	21820	16-bit unsign int	—"—
Alarm 22 action 2	03,06,16	21821	16-bit unsign int	—"—
Alarm 23 action 2	03,06,16	21822	16-bit unsign int	—"—
Alarm 24 action 2	03,06,16	21823	16-bit unsign int	—"—
Alarm 1 action 3	03,06,16	21900	16-bit unsign int	none(0), trapcond1(1), trapcond2(2), trapcond1and2(3), postiostate(4), mqttpublish(6)
Alarm 2 action 3	03,06,16	21901	16-bit unsign int	—"—
Alarm 3 action 3	03,06,16	21902	16-bit unsign int	—"—
Alarm 4 action 3	03,06,16	21903	16-bit unsign int	—"—
Alarm 5 action 3	03,06,16	21904	16-bit unsign int	—"—
Alarm 6 action 3	03,06,16	21905	16-bit unsign int	—"—
Alarm 7 action 3	03,06,16	21906	16-bit unsign int	—"—
Alarm 8 action 3	03,06,16	21907	16-bit unsign int	—"—
Alarm 9 action 3	03,06,16	21908	16-bit unsign int	—"—
Alarm 10 action 3	03,06,16	21909	16-bit unsign int	—"—
Alarm 11 action 3	03,06,16	21910	16-bit unsign int	—"—
Alarm 12 action 3	03,06,16	21911	16-bit unsign int	—"—
Alarm 13 action 3	03,06,16	21912	16-bit unsign int	—"—
Alarm 14 action 3	03,06,16	21913	16-bit unsign int	—"—
Alarm 15 action 3	03,06,16	21914	16-bit unsign int	—"—
Alarm 16 action 3	03,06,16	21915	16-bit unsign int	—"—
Alarm 17 action 3	03,06,16	21916	16-bit unsign int	—"—

Alarm 18 action 3	03,06,16	21917	16-bit unsign int	—"—
Alarm 19 action 3	03,06,16	21918	16-bit unsign int	—"—
Alarm 20 action 3	03,06,16	21919	16-bit unsign int	—"—
Alarm 21 action 3	03,06,16	21920	16-bit unsign int	—"—
Alarm 22 action 3	03,06,16	21921	16-bit unsign int	—"—
Alarm 23 action 3	03,06,16	21922	16-bit unsign int	—"—
Alarm 24 action 3	03,06,16	21923	16-bit unsign int	—"—
Alarm 1 status	03	22000	16-bit unsign int	undefined(0), normal(1), indeterminate(2), warning(3), minor(4), major(5), critical(6)
Alarm 2 status	03	22001	16-bit unsign int	—"—
Alarm 3 status	03	22002	16-bit unsign int	—"—
Alarm 4 status	03	22003	16-bit unsign int	—"—
Alarm 5 status	03	22004	16-bit unsign int	—"—
Alarm 6 status	03	22005	16-bit unsign int	—"—
Alarm 7 status	03	22006	16-bit unsign int	—"—
Alarm 8 status	03	22007	16-bit unsign int	—"—
Alarm 9 status	03	22008	16-bit unsign int	—"—
Alarm 10 status	03	22009	16-bit unsign int	—"—
Alarm 11 status	03	22010	16-bit unsign int	—"—
Alarm 12 status	03	22011	16-bit unsign int	—"—
Alarm 13 status	03	22012	16-bit unsign int	—"—
Alarm 14 status	03	22013	16-bit unsign int	—"—
Alarm 15 status	03	22014	16-bit unsign int	—"—
Alarm 16 status	03	22015	16-bit unsign int	—"—
Alarm 17 status	03	22016	16-bit unsign int	—"—
Alarm 18 status	03	22017	16-bit unsign int	—"—
Alarm 19 status	03	22018	16-bit unsign int	—"—
Alarm 20 status	03	22019	16-bit unsign int	—"—
Alarm 21 status	03	22020	16-bit unsign int	—"—
Alarm 22 status	03	22021	16-bit unsign int	—"—
Alarm 23 status	03	22022	16-bit unsign int	—"—
Alarm 24 status	03	22023	16-bit unsign int	—"—
Save configuration	03,06	24000	16-bit unsign int	unsaved(0), saved(1)
Restart device	03,06	24001	16-bit unsign int	cancel(0), restart(1)
HW error	03	24002	16-bit unsign int	noErr(0), hwErr(1)
Device ID	03	24100	18 bytes UTF-8	Example: 5c:32:c5:00:ac:52
Hostname	03	24200	16 bytes UTF-8	
Device IP	03	24300	16 bytes UTF-8	Example: 192.168.1.2

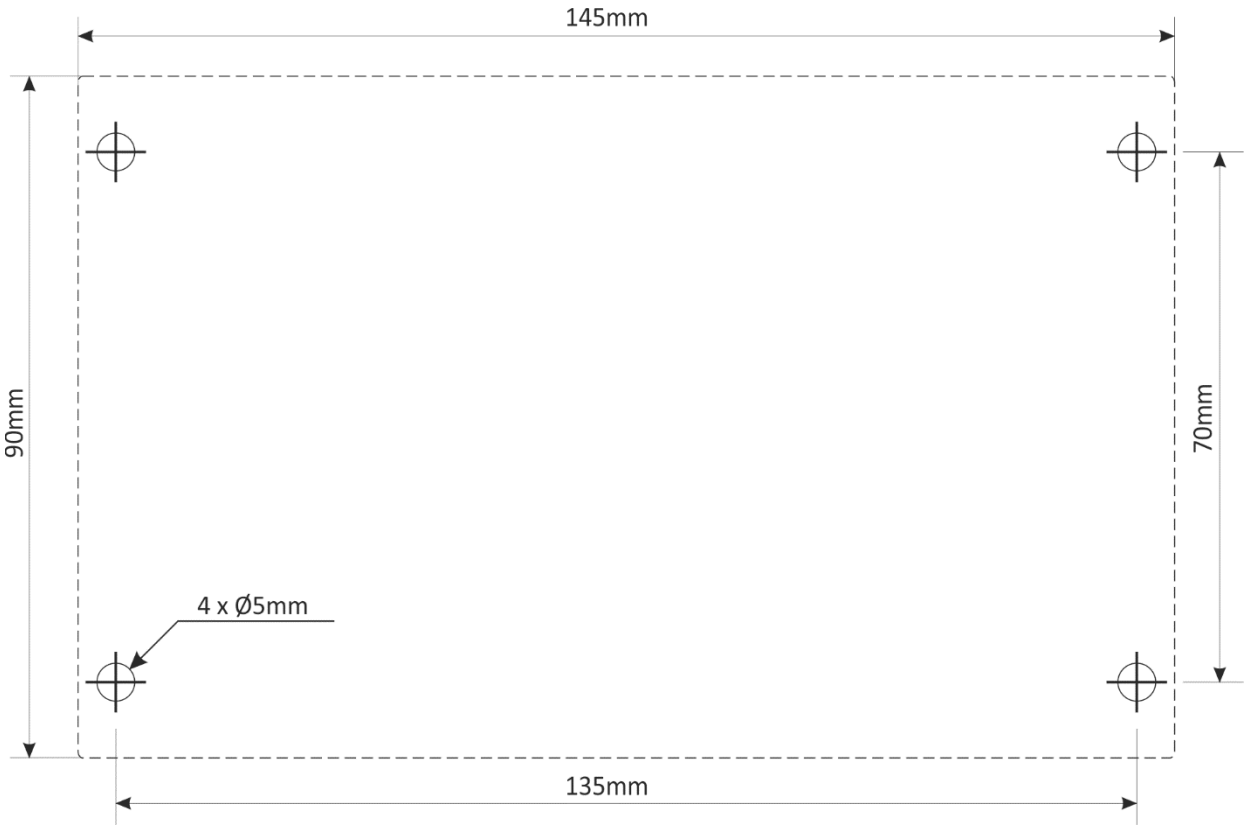

145mm
90mm
70mm
4 x Ø5mm
135mm

Fig.1

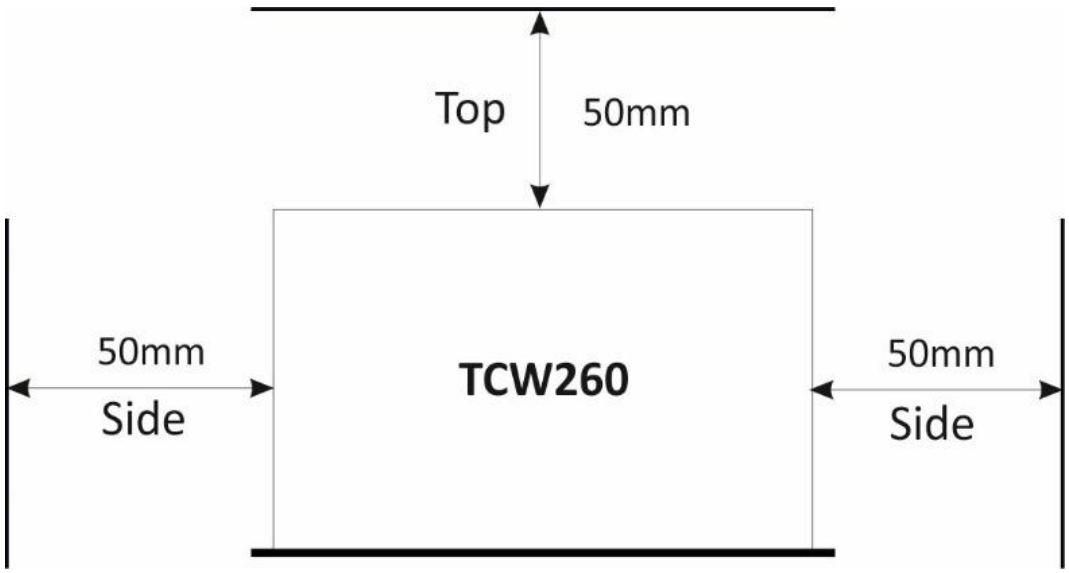

Top
50mm
50mm
Side
TCW260
50mm
Side

Fig.2