



INSTALLATION AND USER MANUAL

BTR102

Bridge · Gateway · Datalogger Modbus

English · Revision 1



MODEL

BTR102

Modbus TCP ↔ RTU ↔ HTTPS gateway · Ethernet, WiFi AP/STA, MQTT, logging on SD/USB



SAFETY WARNINGS

Before installing, connecting or using the BTR102, read this chapter carefully. Failure to observe the warnings may cause damage to the device or to the connected systems or, under particular conditions, hazardous situations for people.

Icon	Peripheral	Meaning
	General warning	Indicates important information for safety or for correct operation of the device, to be read carefully before proceeding.
	Electrical hazard	Shown on the power terminals and on points with live parts: risk of electric shock if the disconnection procedures are not followed.
	Electrostatic discharge (ESD) hazard	The internal electronic components can be damaged by electrostatic discharge: take ESD precautions before handling the board with the enclosure open.
	Read the manual before use	This manual (in particular chapters 3 and 6) must be read before commissioning the device.
	Qualified personnel (illustrative symbol)	Installation, wiring and maintenance reserved for qualified technical personnel aware of electrical hazards and industrial networks.

General warnings

- **Qualified personnel:** the electrical installation and wiring of the BTR102 (power supply, RS-485 bus, Ethernet) must be carried out exclusively by qualified technical personnel, in compliance with the applicable local installation regulations.
- **Power supply:** power the device only according to the specifications (voltage, current, type of power supply) shown on the product label and in the datasheet. A non-compliant power supply may damage the device and void the warranty.
- **Disconnect before working:** always disconnect the power supply before connecting, disconnecting or modifying wiring, terminals or connectors.
- **Do not open the enclosure:** the device contains no parts that can be repaired by the end user. Opening of the enclosure by unauthorised personnel voids the warranty and may expose to electrical hazard and ESD damage.
- **ESD precautions:** any work with the enclosure open, reserved for authorised personnel, requires antistatic precautions (grounded wrist strap/work surface) to avoid damage to the electronic components.
- **Environmental conditions:** observe the temperature, humidity and protection rating (IP) ranges given in the datasheet. Avoid direct exposure to liquids, conductive dust, corrosive or potentially explosive atmospheres, unless explicitly stated otherwise by the manufacturer.
- **Not a functional safety device:** the BTR102 is a Modbus TCP↔RTU communication gateway. It is not certified for safety functions (SIL/PL) and must not be used as the only protection element in applications where a fault or a communication error could cause harm to people, plants or the environment: always provide independent safety measures at plant level.
- **Visible damage:** in case of visible damage to the enclosure, cables or connectors, do not power the device and contact technical support.
- **Network security:** before connecting the device to an Ethernet/WiFi network, especially if reachable from untrusted networks, read chapter 6 (Security) of this manual — factory credentials, TLS certificates and Modbus TCP allowlist.

NOTE This chapter supplements, without replacing them, any safety warnings and installation instructions given in the product datasheet and on the device label, which remain the prevailing reference in case of discrepancy.

REGULATORY REFERENCES AND COMPLIANCE

Icon	Peripheral	Meaning
	CE marking	The device is designed to meet the applicable requirements of the European Union legislation referred to in this chapter.
	WEEE separate collection	At the end of its life, the device must not be disposed of with household waste but handed over according to local regulations on waste electrical and electronic equipment.

Applicable regulatory framework

- **CE marking** — the device is designed to comply with the applicable requirements of the EU directives/regulations listed below. The official EU Declaration of Conformity, with the references of the directives/regulations and harmonised standards applied, is a separate document available on request from the manufacturer.
- **RED Directive 2014/53/EU** (radio equipment) — applicable to the integrated WiFi interface (STA and Access Point modes).
- **Delegated Regulation (EU) 2022/30 (RED-DA)** — cybersecurity requirements for radio equipment, mandatory from 1 August 2025; covers network protection, protection of personal data and privacy, protection from fraud.
- **Regulation (EU) 2024/2847 (Cyber Resilience Act)** — cybersecurity requirements for products with digital elements; the technical measures already implemented in the firmware (authentication, HTTPS, security log, rate limiting, allowlist, SBOM, certificate expiry monitoring) are described in chapter 6 of this manual.
- **Directive 2012/19/EU (WEEE)** — at end of life, separate collection according to local regulations; do not dispose of with household waste (see crossed-out wheellie bin symbol).
- **Directive 2011/65/EU (RoHS)** — restriction of the use of hazardous substances in electrical and electronic equipment.
- **IEC 62443-4-2** — reference standard for the cybersecurity requirements of industrial components, used in the firmware design phase. The Modbus TCP protocol has no native authentication: the residual risk is mitigated by the IP allowlist (§6.7) and must in any case be complemented by adequate network segregation at plant level.

CONTENTS

SAFETY WARNINGS	3
REGULATORY REFERENCES AND COMPLIANCE	4
1 INTRODUCTION	7
1.1 What the BTR102 is	7
1.2 Who this manual is for	7
1.3 Conventions used in this manual	7
2 DEVICE OVERVIEW	8
2.1 Available interfaces	8
2.2 Technical characteristics and connectors	8
2.3 How the gateway works	8
3 FIRST ACCESS	10
3.1 Physical connection	10
3.2 Finding the IP address of the device	10
3.3 First access to the web page	11
3.4 The two roles: User and Installer	12
4 TFT DISPLAY (FRONT PANEL)	13
4.1 Start-up sequence	13
4.2 Main screen	13
4.3 Navigating between pages	14
4.4 Physical credential reset	16
5 WEB INTERFACE — EVERYDAY USE	18
5.1 RTC configuration	18
5.2 Modbus RTU configuration	18
5.3 Ethernet configuration	19
5.4 WiFi configuration	20
5.5 RTU/TCP request configuration	21
5.6 SD card configuration	24
5.7 USB configuration	25
5.8 MQTT configuration	25
5.9 General settings	26
6 SECURITY	29
6.1 Authentication	29
6.2 HTTPS and TLS certificates	29
6.3 Access credentials	29

6.4	Physical credential reset	29
6.5	Rate limiting on access attempts	29
6.6	Security log.....	29
6.7	Allowlist Modbus TCP	30
7	FIRMWARE UPDATE (OTA).....	31
7.1	From the web page	31
7.2	From USB stick	31
7.3	Automatic rollback	31
8	MAINTENANCE.....	32
8.1	Restart.....	32
8.2	Factory reset	32
8.3	TLS certificate replacement/renewal	32
9	PROVISIONING AND TESTING (INSTALLER)	33
9.1	Machine ID and calibration	33
9.2	Automatically generated hostnames and SSIDs	33
9.3	Generating and uploading the TLS certificates	33
9.4	Factory credentials.....	33
10	TROUBLESHOOTING	34
11	APPENDIX A — WEB ENDPOINT SUMMARY	35
11.1	How to build a call.....	36
11.2	Practical examples	37
12	APPENDIX B — FORMAT OF THE GW.JSON CONFIGURATION FILE.....	38
13	APPENDIX C — COMPLETE MODBUS REGISTER MAP	40

1 INTRODUCTION

1.1 What the BTR102 is

The BTR102 is a **Bridge-Gateway-Datalogger** with multiple communication interfaces.

The device:

- exposes a **Modbus TCP server** (port 502 by default) on Ethernet and WiFi (STA and AP);
- exposes several **https routes** to retrieve the data received from the slave modules connected to it
- periodically polls up to **20 configurable requests (Modbus RTU, Modbus TCP)** to slaves connected on the RS485 bus, on WiFi STA, WiFi AP or Ethernet;
- keeps the values read in an internal register map, accessible both via Modbus TCP and via the built-in **web interface** (also via https routes);
- offers a front **TFT display** with a summary of the interface status and a **capacitive button** for navigation and emergency credential reset;
- can log the data read to an **SD card** and a **USB stick**, in CSV format, starting immediately or at a scheduled date/time.

1.2 Who this manual is for

The manual is organised for two types of reader:

- **User/operator** — whoever uses the device once the system is installed: chapters 3, 4, 5 and the everyday parts of chapter 6 (password change, log review).
- **Installer/technician** — whoever handles commissioning, network configuration, certificates and testing: the whole manual, in particular chapters 6 (advanced parts), 7, 8, 9.

1.3 Conventions used in this manual

- The names of the fields and buttons of the web interface are shown **in bold**, exactly as they appear on screen.
- The [SCREENSHOT: . . .] boxes indicate where to insert a real image (web screen, photo of the display, photo of the device).
- Blocks marked with > indicate important notes or warnings.

NOTE *The security features described in this manual are part of a compliance path towards the European **Cyber Resilience Act (CRA)**; the complete list of implemented measures is documented internally for compliance audits and is not covered by this operating manual.*

2 DEVICE OVERVIEW

2.1 Available interfaces

Interface	Description
Ethernet	RJ45 connector, static IP or DHCP
WiFi STA	Connection to an existing WiFi network as a client
WiFi AP	The device's own Access Point, for local configuration without an external network
Modbus RTU (RS485)	Serial bus to the Modbus slaves to be polled
Modbus TCP	Server on the network (configurable port, default 502) for supervision/SCADA systems
SD card	Local data logging, CSV format
Display TFT	Front panel with status summary
Capacitive button	Display navigation and physical credential reset
USB	USB stick: local data logging (like SD) and automatic firmware update

2.2 Technical characteristics and connectors

Category	Parameter	Value	Connector
Power supply	Supply voltage	10–36 Vdc	Plug-in screw terminal block, 2 poles, 3.5 mm pitch
	Maximum consumption	1,6 W	
Ethernet	Interface	RJ45, 10/100 Mb/s	RJ45
microSD card	Supported file system	FAT32	
USB	Supported file system	FAT32	
Modbus RTU (RS485)	Maximum nodes on the bus	256	Plug-in screw terminal block, 3 poles, 3.5 mm pitch
	Common-mode voltage	60 Vdc	
Environmental conditions	Operating temperature	-20...+60 °C	
	Relative humidity	Max 95%, non-condensing	

2.3 How the gateway works

The strength of the BTR102 is the number of available communication paths, which can also be used simultaneously:

- **Inbound**, the device can be reached as a Modbus TCP slave (port 502) and also via https by a supervision system on any of its active network interfaces — **Ethernet, WiFi STA or WiFi AP**.
- **Outbound**, each row of the request table (up to 20, see §5.5) polls a slave chosen independently row by row: on the RS-485 serial bus (Modbus RTU), or via Modbus TCP on any of the three network interfaces (WiFi AP, WiFi STA or Ethernet). A typical use case of the latter mode: reaching Modbus TCP "satellite" boards connected to the BTR102 WiFi AP, without them having to be visible on the SCADA Ethernet network.

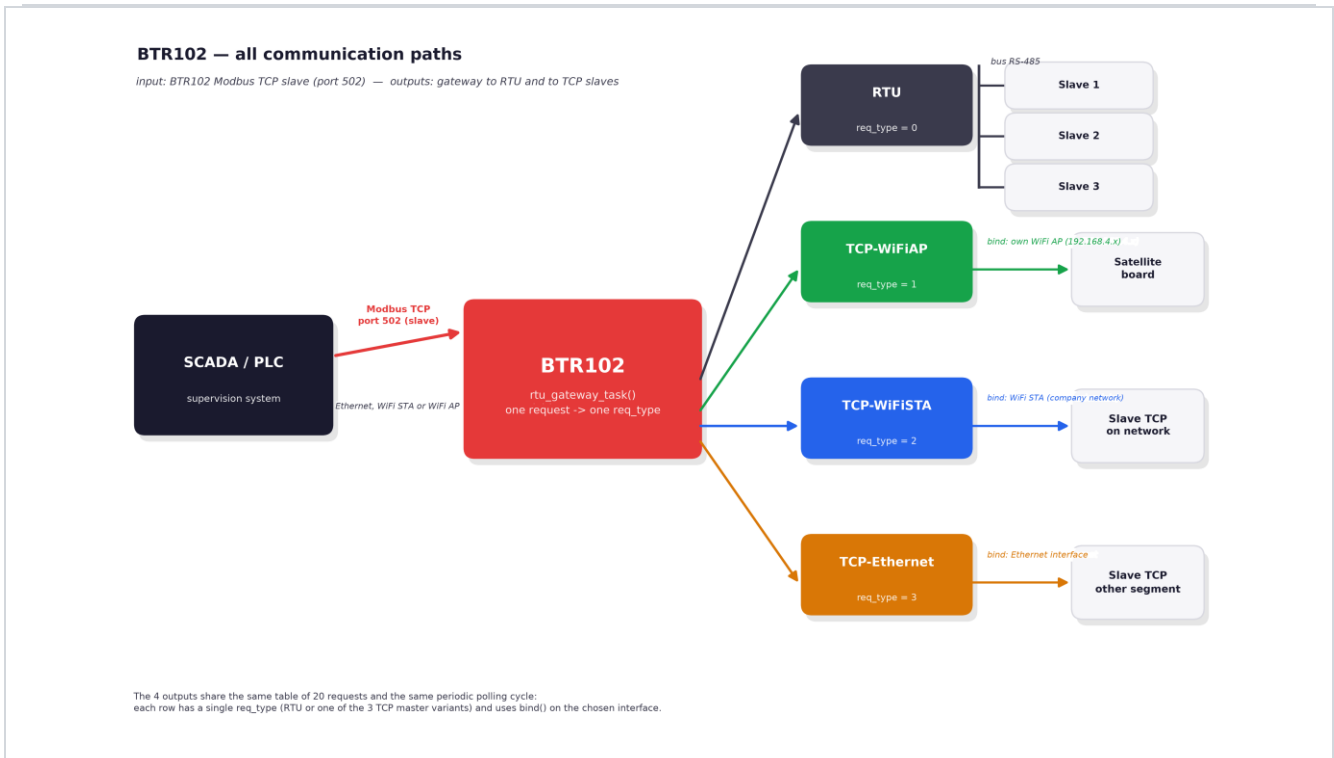


Diagram of all BTR102 communication paths: Modbus TCP input on Ethernet/WiFi STA/WiFi AP; outputs to RTU slaves on RS-485 and to TCP slaves on any of the three network interfaces

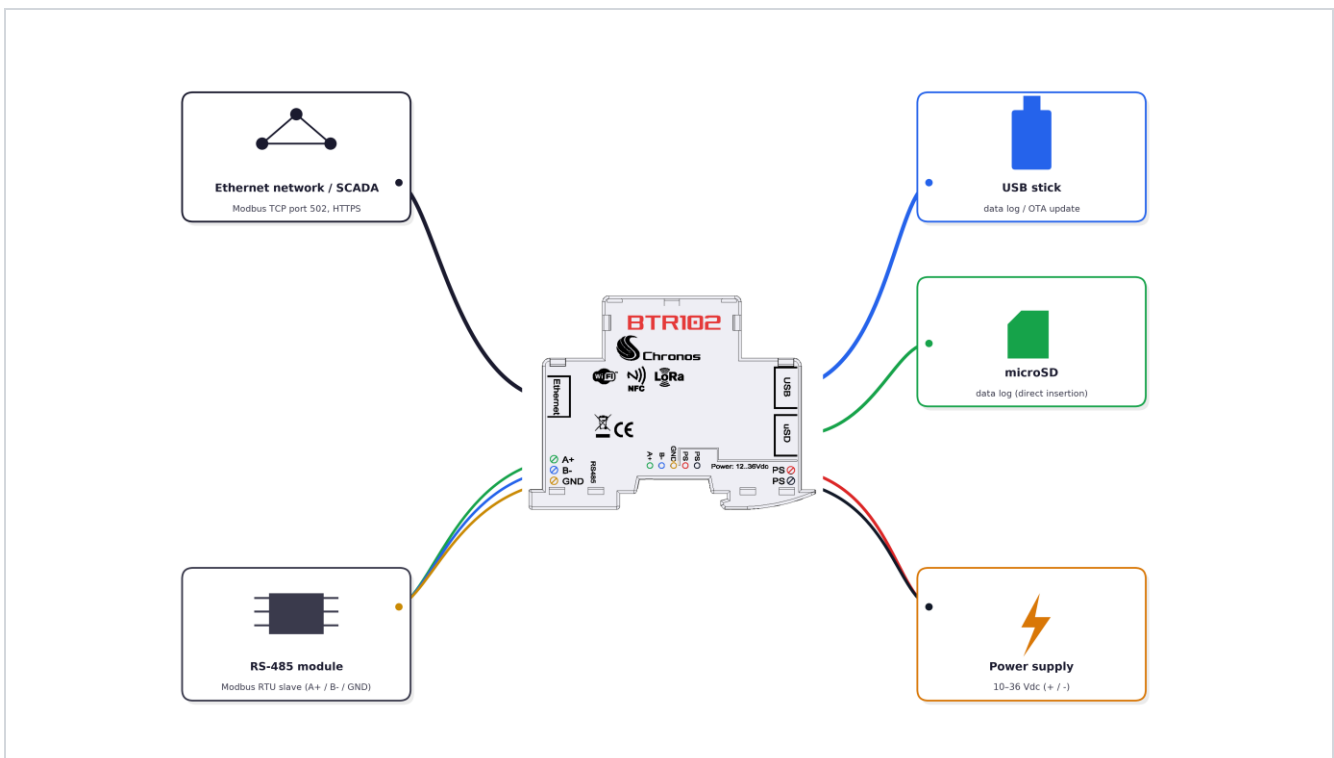
1. The device periodically reads (configurable **polling interval**) the configured slaves according to the **requests** (up to 20, each with its own connection path — RTU or TCP — address/IP and block of registers to read).
2. The values read are written into an internal register map (memory map), organised according to the classic Modbus numbering (40001 onwards).
3. A Modbus TCP client (PLC, SCADA, supervision system) can read/write this map by connecting to the Modbus TCP port of the device — the BTR102 acts as a transparent "translator" regardless of the path actually used for each slave.
4. In parallel, the same register map can be viewed and configured via the built-in web page.

The BTR102 also behaves as a **transparent Modbus bridge**: any Modbus TCP request received on port 502 — not only those matching the requests configured in §5.5 — is forwarded as is on the RTU bus to the indicated slave (by address/unit-id) and the response is returned to the TCP client just as transparently. A supervision system can therefore address a specific RTU slave directly through the BTR102, without first having to configure a dedicated request in the gateway table.

3 FIRST ACCESS

3.1 Physical connection

1. Power the device according to the specifications on the label.
2. Connect the Ethernet cable (optional if only WiFi will be used) and/or the RS485 bus to the Modbus RTU slaves.
3. Wait for start-up to complete: the TFT display shows the splash screen for a few seconds, then the main screen with the status icons (see chapter 4).



Wiring diagram: power supply (2 wires), Modbus RTU RS-485 (3 wires, A+/B-/GND), Ethernet, USB and microSD to the respective external modules

3.2 Finding the IP address of the device

The BTR102 exposes three network interfaces that can be active at the same time: **Ethernet**, **WiFi STA** (if configured to connect to an existing network) and **WiFi AP** (the device's own network, always available unless explicitly disabled).

Ways to find the IP:

- **TFT display:** the Ethernet/WiFi/WiFi AP detail pages show the current IP of each interface (navigate with a short press of the capacitive button, see §4.3).
- **mDNS (hostname.local):** each interface has a unique hostname generated automatically in the factory (e.g. BTR102-eth-<data>.local) — which can be typed directly into the browser on networks that support mDNS resolution.
- **WiFi AP:** if no network is available yet, you can connect directly to the device's own WiFi network (SSID visible among the available networks, generated in the factory) — the Access Point IP is fixed and shown on the display.

3.3 First access to the web page

The webserver can be accessed via IP address or via hostname

- There are 3 hostnames to access the webserver:
- **BTR102-eth-<SN>**
- **BTR102-wifista-<SN>**
- **BTR102-wifiap-<SN>**

The username is always **admin**, for everyone, and cannot be changed.

The device label (also on the packaging box) shows two passwords and the serial number:

- **PW Web** - the password to access the device **web page** (username **admin**, see above). From the factory it is the same for the user role and for the installer role (§3.4); each of them can then change it independently (§6.3), after which they differ. **It is not the WiFi network password.**
- **PW AP** - the password of the **device's own WiFi network** (WiFi AP, its factory SSID): it is only used to join your PC/smartphone to that WiFi network, exactly as for any other WiFi network — it has nothing to do with access to the web page. **It is not the web login password.**
- **SN** – the unique serial number used to build the three hostnames (e.g.: 260915160252)

1. Open a browser and connect to the IP address (or hostname **.local**) of the device — see below the three step-by-step procedures, one for each network interface.
2. If a TLS certificate has already been loaded on the device (normally done during testing, see chapter 9), the connection is automatically made over **HTTPS**; since the certificate is self-signed (no common CA between devices, §9.3), the browser will almost always show an unrecognised certificate warning — this is normal, you can proceed ("advanced" → "proceed anyway") or install the certificate of that specific device as trusted on the PC beforehand.
3. If no certificate is present yet, the device responds over plain **HTTP**.
4. The browser asks for **user name and password** (built-in browser authentication, not a custom login page): username **admin**, password = **PW Web** shown on the label (see above — not the PW AP). For details of the two roles see §3.4, for the password change see §6.3.

Access via Ethernet

1. Connect an Ethernet cable from the RJ45 port of the device to the local network (switch/router).
2. Wait for the interface to obtain an IP address: by factory default this is via automatic DHCP; if a static IP has previously been set (§5.3), use that.
3. Find the current IP on the TFT display (Ethernet detail page, §4.3) or use the mDNS hostname **.local** (§3.2) instead of the IP.
4. From a PC connected to the same network, open a browser and type that IP address (or hostname).
5. When the browser asks for credentials: username **admin**, password = **PW Web** (see above).

Access via WiFi AP (the device's own network)

Useful when no existing Ethernet/WiFi network is available yet — typically at the very first power-on, before any configuration. The web page can be reached from the WiFi AP network even without an Ethernet cable and with WiFi STA off or not connected: the web server starts a few seconds after power-on as soon as at least one interface (Ethernet, WiFi STA or WiFi AP) is active. Caution: this procedure uses **two different passwords**, one at step 3 and one at step 5 — do not mix them up (see the reminder above).

1. On the PC or smartphone you want to connect from, open the list of available WiFi networks.
2. Select the BTR102 network: the SSID is generated in the factory and is also shown on the WiFi AP detail page of the TFT display.
3. To join that WiFi network, enter the **PW AP** (see above) — it is the WiFi password, not the web page one.

-
4. Once joined to the network, open a browser and type the IP of the Access Point: it is fixed and shown on the TFT display (§3.2).
 5. When the browser asks for the web page credentials: username `admin`, password = **PW Web** (see above — different from the PW AP just used at step 3).

Access via WiFi STA (existing WiFi network)

Requires WiFi STA to have been configured previously (panel §5.4, installer operation) — the first configuration must be done by connecting to the device via Ethernet or WiFi AP (see above); it is not available as a starting point on a device that has never been configured.

1. Check that the BTR102 is connected to the desired WiFi network (WiFi detail page on the TFT display, or panel §5.4 in the web page).
2. Find the IP assigned to the device on that network on the TFT display (§4.3) or use the mDNS hostname `.local` (§3.2).
3. From a device connected to the same WiFi network, open a browser and type that IP address (or hostname).
4. When the browser asks for credentials: username `admin`, password = **PW Web** (see above).

3.4 The two roles: User and Installer

The BTR102 distinguishes two access levels, with separate credentials:

- **User:** can view all pages and data, and change the SD logging mode. Cannot change network parameters, Modbus configuration or certificates, nor perform restarts/resets.
- **Installer:** full access, including all configurations (network, Modbus, RTU requests, certificates, allowlist), firmware update, restart and factory reset.

The browser asks for credentials according to the action requested: viewing pages requires user credentials; saving an advanced configuration requires installer credentials (if you are already authenticated as user, the browser may ask for credentials again at the first attempt of a reserved action).

4 TFT DISPLAY (FRONT PANEL)

4.1 Start-up sequence

At power-on the **splash** screen (logo) appears for a few seconds, automatically followed by the **main** screen.



Splash screen at power-on

4.2 Main screen

The main screen shows an icon for each main interface/peripheral (WiFi STA, WiFi AP, Ethernet, USB, SD, Modbus RTU), coloured according to its status:

Icon colour	Meaning
White	Peripheral disabled / not applicable
Yellow	*(SD/USB only)* Peripheral enabled but SD card / USB stick not inserted
Green	Peripheral enabled and working
Red	Peripheral enabled but in error (e.g. configured RTU requests failing, WiFi/Ethernet disconnected, SD/USB stick inserted but not readable)
Blue	*(SD/USB/MQTT only)* Peripheral enabled, mounted/connected and with logging or publishing actively in progress at this moment



Main display screen with the status icons

NOTE The WiFi AP icon never shows an error state (there is no "AP failed" condition).

Icon legend

Icon	Peripheral	Meaning
	Ethernet	Wired network interface
	WiFi STA	Connection to the existing WiFi network (client mode)
	WiFi AP	The device's own Access Point
	SD card	Card mount and logging status
	USB	USB stick mount and logging status (during an automatic firmware update, see §7.2, the icon temporarily gives way to the full-screen "OTA IN CORSO" overlay)
	Modbus RTU	Outcome of the RTU requests to the configured slaves
	MQTT	Publishing status to the configured MQTT broker (§5.8) — same colour scheme as the other logging icons

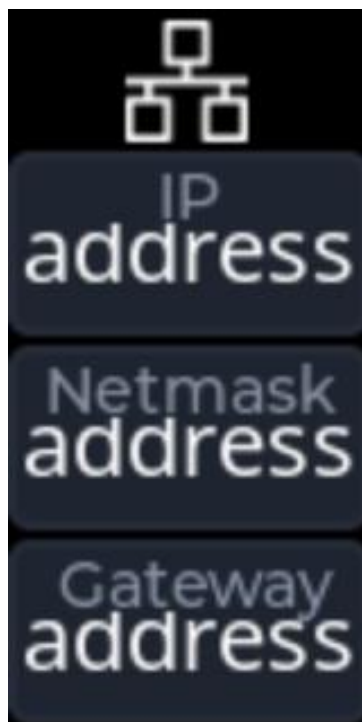
4.3 Navigating between pages

A **short press** (less than 1 second) of the capacitive button cycles to the next page:

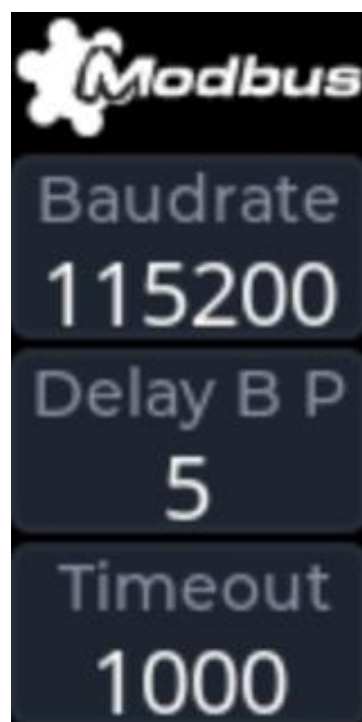
Main → Ethernet → WiFi → WiFi AP → Modbus RTU → Settings (HW/FW/Serial) → Credential reset → Main → ...

Each interface detail page shows the same essential information available in the corresponding section of the web page (IP/Netmask/Gateway for the network interfaces, communication parameters for Modbus).

The **Settings** page (icon ) shows: hardware version, firmware version, serial number of the device.



Ethernet detail page on the display



Modbus RTU detail page on the display



Settings page (HW/FW/Serial Number) on the display

4.4 Physical credential reset

On the **Credential reset** page (the last in the cycle, before returning to the main one), holding the capacitive button for **10 continuous seconds** restores to factory values: user password, installer password and WiFi AP network password. A progress bar on the screen shows the remaining press time; on completion the confirmation **"OK"** appears.



Credential reset page on the display

NOTE *This procedure is intended for emergency recovery if the credentials are lost — it requires physical access to the device and is recorded in the security log (§6.6).*

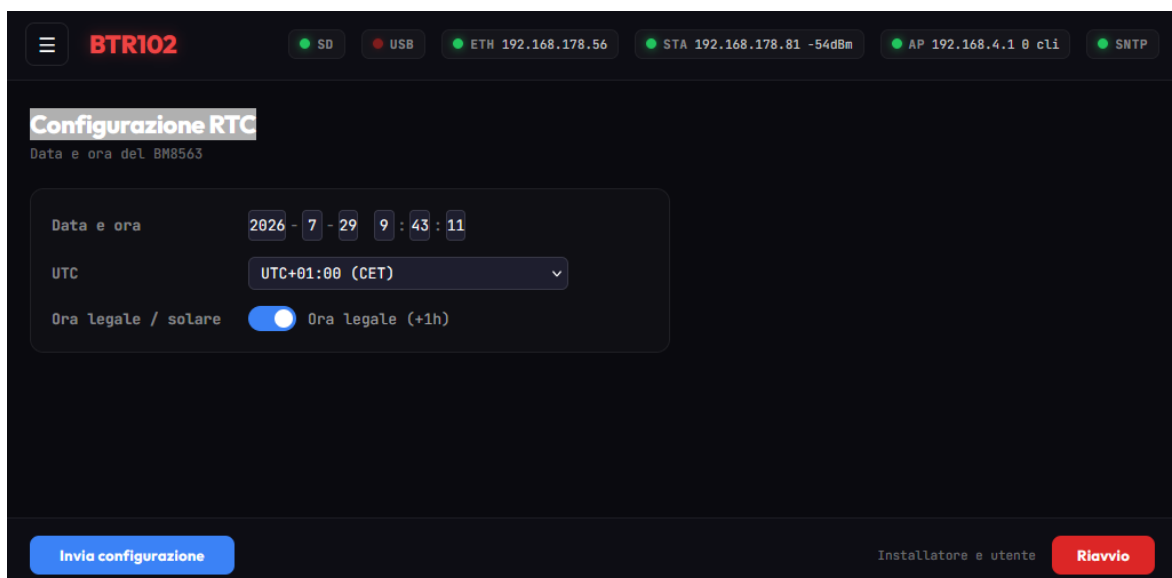
5 WEB INTERFACE — EVERYDAY USE

The web page is organised in a side menu with 9 sections. The menu opens with the ☰ button at the top left. At the top, the status bar always shows 7 dots with the summary status of SD/USB/Ethernet/WiFi STA/WiFi AP/SNTP/MQTT, regardless of the page displayed — same colour scheme (white/yellow/red/green/blue) as the TFT display icons, see §4.2.

NOTE *Interface language: selectable between Italian and English from the **General settings** panel (§5.9).*

5.1 RTC configuration

Allows reading/setting the date and time of the internal clock (RTC), the time zone (**UTC**) and daylight saving/standard time.



Web page — RTC configuration

NOTE *The RTC is the time base used for the timestamps of the security log, of the CSV files on SD and for the scheduled start of logging.*

5.2 Modbus RTU configuration

Serial communication parameters towards the RTU slaves on the RS485 bus:

- **Baud rate:** from 1200 to 115200 bps
- **Parity:** NONE / EVEN / ODD
- **Stop bits:** 1 / 1.5 / 2
- **Timeout (ms):** maximum time to wait for a slave response before considering the request failed

NOTE *Baud rate, parity and stop bits require a restart of the device to be applied (dedicated button at the bottom of the panel) — saving alone writes them to memory but the serial port keeps using the previous parameters until restart. The timeout, instead, takes effect immediately, without a restart.*

NOTE The **polling interval** (common to all RTU **and** TCP requests, not just the RTU UART) is not set on this page: it is in RTU/TCP request configuration (§5.5), next to the "Logging" selector.

Configurazione Modbus RTU
Parametri di comunicazione ModBus Master

Baud rate: 115200
Parità: NONE
Bit di stop: 1 bit
Timeout (ms): 2000

Baud rate, parità e bit di stop vengono applicati solo dopo un riavvio del dispositivo - il timeout è invece effettivo subito (il polling si imposta in pagina 5, "Configurazione richieste RTU/TCP")

Riavvia per applicare le modifiche

Web page — Modbus RTU configuration

5.3 Ethernet configuration

- **MAC, DHCP (on/off), Hostname** (generated in the factory, cannot be changed — also used in the TLS certificate)
- **Configured IP (static):** IP/Netmask/Gateway to be used if DHCP is disabled
- **Current IP (actual):** values actually in use at this moment (useful to check whether DHCP has assigned an address)
- **Modbus TCP port:** port on which the Modbus TCP server listens (default 502)

Configurazione Ethernet
DHCP / IP statico, porta Modbus TCP

INTERFACCIA ETHERNET (W5500)

MAC: 64:E8:33:67:B6:98
DHCP: ☒
Hostname: BTR102-eth-260603115422

IP CONFIGURATO (STATICO)

IP: 192.168.178.250
Netmask: 255.255.255.0
Gateway: 192.168.178.250

IP CORRENTE (REALE)

IP: 192.168.178.56
Netmask: 255.255.255.0
Gateway: 192.168.178.1

MODBUS TCP

Porta: 502

Invia configurazione

Installatore e utente Riavvio

Web page — Ethernet configuration

5.4 WiFi configuration

Two separate panels:

STA — Station Mode (connection to an existing WiFi network):

- **Enabled/DHCP, Hostname** (fixed from the factory)
- Configurable static IP or actual current IP
- **Search networks**: starts a WiFi scan and fills in the list of networks found, which can be selected to fill in the SSID automatically. The result appears next to the button:
- *N networks found* (if there are more than 20, the first 20 are shown);
- *No network found* — the search was carried out but found no networks: the BTR102 sees **only 2.4 GHz networks**; a 5 GHz-only network never appears;
- a message **in red** if the search was not possible, with the reason. The most common case is *Search failed: WiFi STA disabled*: the search requires WiFi STA to be active, so it must be enabled and the device restarted. If WiFi STA has just been enabled but the device has not yet been restarted, the message is *Search failed: WiFi STA enabled, restart the device to activate it*. Other reasons (e.g. WiFi STA busy with a connection attempt — try again after a few seconds) have a dedicated message; the technical error code is recorded in the device diagnostic log.
- **SSID/Password, Connect to network**

AP — Access Point (the device's own network):

- **Enabled, Hostname/SSID** (fixed from the factory, identical)
- **Password** editable
- IP/Netmask/Gateway of the AP network (fixed)
- List of WiFi clients currently connected to the AP

Configurazione WiFi
Configurazione STA e AP

STA - STATION MODE

MAC: 64:E8:33:67:B6:98

Abilitata: ☒

DHCP: ☒

Hostname: BTR102-wifista-260603115422

IP CONFIGURATO (STATICO)

IP: 192.168.178.200

Netmask: 255.255.255.0

Gateway: 192.168.178.200

IP CORRENTE (REALE)

IP: 192.168.178.81

Netmask: 255.255.255.0

Gateway: 192.168.178.1

CONNESSIONE

Cerca reti:

Reti trovate: - avvia scansione -

SSID: FRITZ!Box 7530 PU

Password:

AP - ACCESS POINT

MAC AP: 64:E8:33:67:B6:99

Abilitata: ☒

Hostname: BTR102-wifiap-260603115422

SSID: BTR102-wifiap-260603115422

Password:

RETE AP

IP: 192.168.4.1

Netmask: 255.255.255.0

Gateway: 192.168.4.1

CLIENTI CONNESSI

Numero: 0

Installatore e utente:

Web page — WiFi configuration (STA and AP)

5.5 RTU/TCP request configuration

This is the main page of the gateway: a table with up to **20 requests**, each to a Modbus **RTU** slave (RS485 bus) or, from the firmware with RTU/TCP gateway, to a Modbus **TCP** slave reachable on one of the device's network interfaces (WiFi AP, WiFi STA or Ethernet) — useful to let Modbus TCP "satellite" boards connected to the BTR102 WiFi AP network talk to the same SCADA on Ethernet. Each request has:

Field	Meaning
req_n	Sequential request number (1–20), fixed: shown in grey, cannot be changed
Type (req_type)	RTU (default) / TCP-WiFi AP / TCP-WiFi STA / TCP-Ethernet
req_id	RTU: slave address on the bus. TCP: Modbus Unit Identifier of the slave
IP (req_ip)	TCP types only: IP address of the slave on the chosen network, in the format 0.0.0.0 (Modbus TCP port fixed at 502, not configurable). Disabled for the RTU type
req_addr	First register to read from the slave, in absolute Modbus notation. The value also selects the read function: 30001 to 39999 input register (function 4, e.g. 30001 = first input register of the slave), 40001 to 65535 holding register (function 3, e.g. 40001 = first holding register)
req_qty	Total number of registers used by the request
valid	Outcome of the last read cycle (1 = successful, 0 = failed)
Errors	Counter of failed cycles for this request, reset (for all rows) at every device restart and every time the configuration is saved — only an indication of connection quality, not stored nor logged on SD/USB/MQTT
Reg 1...10	Up to 10 configurable registers per request, each with name, data type, unit of measurement and value read. The value is only displayed (updated at each reading) and cannot be edited by hand

NOTE RTU and TCP requests share the same 20-row table and run in sequence in the same polling cycle: too many TCP rows to unreachable slaves lengthen the cycle time (up to a few seconds per request, between timeouts and retries), also delaying the RTU requests of the same cycle. Configurable limits: up to **10 TCP-WiFi AP requests** and **10 TCP-WiFi STA** at the same time (the limit for Ethernet and the total coincide with the 20 available rows).

Each TCP request opens a dedicated connection to the slave, separate for each network interface according to the value chosen in **Type**: the request physically leaves through the selected interface (WiFi AP, WiFi STA or Ethernet), even when the subnets of these interfaces overlap (complete path diagram in §2.3).

BTR102 SD USB MQTT ETH 192.168.178.44 STA 192.168.178.43 -56dBm AP 192.168.4.1 0 cli SNTP

Configurazione richieste RTU/TCP

Ciclo n. 4975

Logging: PROGRAMMATO Polling (secondi): 2

Abilita LOG SD: ON Abilita LOG USB: ON

Data/ora avvio log: Anno - Mes - Gio Ora : Min
NA - NA - NA 17 : 7

REQ_N	REQ_ID	TIPO	IP	REQ_ADDR	REQ_QTY	VALID	REG 1	REG 2
1	1	RTU	a.b.c.d	40001	2	1	primo vint16_t A 112	secondo vint16_t B 14
2	0	RTU	a.b.c.d	null	null	0		

Web page — request row configured as TCP-WiFiAP, with the slave IP field visible

The data types supported for each register:

Type	Registers used	Byte order	Notes
uint16_t	1	AB	Unsigned integer, 16 bit
int16_t	1	AB	Signed integer, 16 bit
uint32_t	2	ABCD	Unsigned integer, 32 bit
int32_t	2	ABCD	Signed integer, 32 bit
uint32_rev	2	CDAB	As <code>uint32_t</code> , but with the two words in reverse order (low word first, used by some slaves)
int32_rev	2	CDAB	As <code>int32_t</code> , but with the two words in reverse order (low word first, used by some slaves)
float	2	ABCD	Floating-point number, 32 bit (IEEE 754)
float_rev	2	CDAB	As <code>float</code> , but with the two words in reverse order ("swapped" endianness, used by some slaves)
int64_t	4	ABCDEFGH	Signed integer, 64 bit
int64_rev	4	GHEFCBAB	As <code>int64_t</code> , but with the four words in reverse order (lowest word first, used by some slaves)

NOTE Byte order: A is the most significant byte of the value, then B, C, ... in decreasing order; the column shows in which order the bytes arrive in the registers, starting from the first. The Modbus protocol only fixes the byte order within a register (high byte first), not the register order for 32/64-bit values, which depends on the slave manufacturer. Many slave manuals and test tools (e.g. Modbus Poll) use the same notation: "ABCD" or "big-endian" corresponds to the types without suffix, "CDAB" (64-bit "GHEFCBAB") or "word swap" to the `_rev` types. The BADC and DCBA orders (bytes swapped within each register) are not supported.

NOTE req_qty must match exactly the sum of the registers used by the selected types — the device refuses to save if the count does not match, with an explicit error message.

Empty rows: a row with `req_id` = 0 is considered not configured. Setting `req_id` to 0 (or deleting it) and leaving the field clears the whole row in the page (RTU type, empty IP/address/quantity and registers); the change becomes final only with **Save**.

Checks on saving: when **Save** is pressed the device checks that every configured row (`req_id` other than 0) makes sense from the Modbus point of view:

Check	Error message
<code>req_id</code> between 1 and 247 for RTU (slave address), between 1 and 255 for TCP (Unit Identifier)	<code>req_id</code> out of range (RTU 1-247, TCP 1-255)
For TCP types, valid IP (not 0.x.x.x, not 127.x.x.x, not multicast/broadcast from 224.x.x.x upwards)	Invalid TCP slave IP address
<code>req_qty</code> between 1 and 10	<code>req_qty</code> must be at least 1 / <code>req_qty</code> exceeds the maximum allowed (10)
All registers read, from <code>req_addr</code> to <code>req_addr</code> + <code>req_qty</code> - 1, all between 30001 and 39999 (input register) or all between 40001 and 65535 (holding register). 40000 is not valid, nor is a block spanning both areas (e.g. a <code>uint32_t</code> at 39999)	REQ_ADDR out of range (30001-39999 / 40001-65535)
<code>req_qty</code> equal to the sum of the registers of the chosen types	<code>req_qty</code> does not match the register types

The error message appears **in red** at the bottom of the page for 5 seconds and indicates the row concerned (e.g. *REQ_ADDR out of range (30001-39999 / 40001-65535): req_n 1*). In case of error **nothing is saved**: neither the request table nor the settings of the "Logging and Polling" panel described below. Once the error is corrected, press **Save** again; the green save message appears only when all checks are passed.

The "**Cycle no.**" counter at the top shows how many complete read cycles have been carried out since the last start-up. It only counts cycles in which at least one request was polled: if no row is configured the counter stays still and nothing is written to SD/USB or published on MQTT, even with logging active.

NOTE Each table row also shows an **errors** counter, stacked under "valid" in the same cell: it counts the cycles in which that specific request failed (poll unsuccessful after all attempts). It is a qualitative indication of the quality of the connection to that slave — it restarts from 0 at every device restart and every time the request configuration is saved (all rows, including unchanged ones), and is never saved to flash nor included in the CSV/MQTT logs.

Logging and Polling (panel at the top of this page, above the request table): a single panel controls both the polling interval and logging on SD/USB — shared between the two channels; there are no separate selectors on the "SD card configuration"/"USB configuration" pages (§5.6/§5.7 below only describe the physical power and the files, not the logging mode).

- **Polling (seconds)**: interval between one complete read cycle and the next, valid for **all** RTU and TCP requests on this page (not just the RTU UART — for the actual serial parameters see §5.2). Effective immediately, without a restart.

NOTE The **first** cycle after power-on (or after a restart) waits for the current minute to end — up to 59 seconds — before carrying out the first reading; subsequent cycles then keep the exact set rate, without further delay. With a polling interval that divides 60 seconds (5/10/15/20/30/60) the cycles therefore always start at "round" seconds (e.g. with 10 s: :00, :10, :20...).

NOTE The alignment concerns the **read cycle**, not logging: it applies in any logging mode (OFF, ON, SCHEDULED, FALLBACK MQTT). Writes to SD/USB and MQTT publications take place at the end of each cycle and follow its rate; the time shown in each row of the CSV file is the **start** time of the cycle (e.g. 12:30:00), even if reading the slaves takes a few seconds. Several BTR102 devices with synchronised clocks (SNTP) thus stay "in phase" with each other, with the same read, write and publish instants — useful for example with the **SCHEDULED** mode below, when several gateways must start logging at the same date/time.

NOTE The alignment is kept as long as each cycle lasts less than the polling interval. If a cycle exceeds it (e.g. many requests, or unreachable TCP slaves that take a few seconds each between timeouts and retries), the next cycle starts late and from then on stays out of phase: in this case increase the polling interval. If no request is configured the cycle still runs, but it is not counted and produces no writes or publications.

- **Logging** — 4-mode selector:

Mode	Behaviour
OFF	No writing
ON	Continuous writing, one record for each complete read cycle
SCHEDULED	Writing starts automatically at the date/time set alongside (Year/Month/Day/Hour/Minute fields)
FALLBACK MQTT	Does not publish on MQTT by itself: writes a record only in the cycles in which the MQTT publication of that same cycle failed (local fallback, no automatic retry — see §5.8)

For the **SCHEDULED** mode, the Year/Month/Day fields can be left empty (they show "NA"): in that case they do not constrain the start, which will trigger the first time the set hour/minute is reached, compatibly with any other fields specified. Examples:

- Only hour/minute set → starts at the first occurrence of that time (today or tomorrow).
- Day/hour/minute set (month/year empty) → starts on the next matching day of the month, at that time.
- Full date and time → starts exactly at that moment.

NOTE If the scheduled instant is missed (device off, busy in that exact minute), the system waits for the next occurrence compatible with the set fields — it does not recover the start retroactively.

NOTE Once started, SCHEDULED logging ignores further changes to the date/time until you leave and re-enter the SCHEDULED mode (or switch the device off, which always resets the wait at restart).

- **Enable SD LOG / Enable USB LOG:** enable logging on the respective channel, also automatically switching on the physical power of the SD/USB if not already active (see §5.6/§5.7)

5.6 SD card configuration

NOTE The logging mode (OFF/ON/SCHEDULED/FALLBACK MQTT) and the polling interval are **not** set on this page, but in the "Logging and Polling" panel of **RTU/TCP request configuration** (§5.5) — this page only concerns the physical power of the card and the files already saved.

Enable SD: physical power of the card — disabling it unmounts the SD even if logging is active from §5.5. Enabled from the factory.

Status and space: shows whether the SD card is mounted correctly and the available space.

Card insertion: the device physically detects the presence of the microSD in the slot. With SD enabled but no card, the SD icon on the display and the SD dot in the web status bar turn **yellow** and no read attempt is made. When the card is inserted it is mounted automatically at the next read cycle (green); if it is removed while mounted, the device unmounts it immediately and the icon turns yellow again. **Red** instead indicates a card that is inserted but not readable (e.g. unsupported file system or damaged card).

Files present: list of the CSV files already saved on the SD, with size and timestamp of the first and last record. From here you can:

- **Download** a file (with optional date/time range filter)
- **Delete** a file (action reserved for the installer)

Configurazione SD card
Stato, logging, download file

Stato: **Montata**

Spazio: 14881.86 MB liberi / 14901.00 MB totali

Logging su SD: **PROGRAMMATO** **PROGRAMMATO**

Data/ora avvio log
Anno - Mes - Giorno Ora : Min
NA - NA - NA 16 : 54

FILE PRESENTI

NOME FILE	DIMENSIONE	PRIMO RECORD	ULTIMO RECORD	AZIONE
BTR102_20260728_171944.csv	49.3 KB	2026-07-28 17:19:44	2026-07-28 17:32:05	Scarica Cancella
BTR102_20260728_165408.csv	98.4 KB	2026-07-28 16:54:08	2026-07-28 17:19:43	Scarica Cancella
BTR102_20260728_164308.csv	37.8 KB	2026-07-28 16:43:08	2026-07-28 16:53:11	Scarica Cancella
BTR102_20260710_164747.csv	18.59 MB	2026-07-10 16:47:47	2026-07-28 16:19:23	Scarica Cancella
BTR102_20260710_101917.csv	261.9 KB	2026-07-10 10:19:17	2026-07-10 16:47:46	Scarica Cancella

[Invia configurazione](#) Installatore e utente [Riavvio](#)

Web page — SD card configuration

CSV file format: the first 4 rows are the header (field names, data type, unit of measurement of each configured register); each subsequent row is a complete RTU read cycle, with date, time (the **start** time of the cycle, e.g. 12:30:00, even if reading the slaves takes a few seconds), and outcome/value of each configured request, in the same order as the table in §5.5. A new file is created automatically every time a new RTU request configuration is saved, or at the start of a SCHEDULED session (§5.5).

5.7 USB configuration

Twin function of the SD (§5.6), for a USB stick connected to the dedicated port:

Enable USB: physical power of the port — disabling it unmounts the stick even if logging is active from §5.5, and also stops the periodic check for the automatic OTA (see below). Enabled from the factory.

Status and space: shows whether the USB stick is mounted correctly and the available space.

USB stick insertion: with USB enabled but no stick connected, the USB icon on the display and the USB dot in the web status bar are **yellow**. A stick is considered inserted when the device recognises it as mass storage; if it then cannot be mounted (e.g. unsupported file system) the icon turns **red**, if mounting succeeds **green**. A stick that is not even recognised (faulty or not compatible) stays yellow. After removal the icon turns yellow immediately if the stick was mounted, otherwise within about ten seconds.

NOTE Here too, logging mode and polling are set in §5.5, not on this page.

Files present: list of the CSV files already saved on the stick, with size and timestamp of the first and last record; from here you can download or delete a file (deletion reserved for the installer). Same CSV format as described in §5.6.

NOTE Automatic firmware update: if the stick contains an *ota* folder with a *BTR102_REV*.bin* firmware file in it, the update starts **by itself** as soon as the stick is mounted — see §7.2.

5.8 MQTT configuration

Publishing of the same "RTU requests" already logged on SD/USB to an external MQTT broker, as a third, parallel channel (mutual TLS, no user/password pair).

Broker connection:

- **Enable MQTT:** physical enable switch, independent of the mode below (mirror of "Enable SD"/"Enable USB")
- **Broker host / Port** (default 8883, standard TLS port) / **Client ID / Base topic:** broker connection parameters — unlike the other settings on this page, **they are not Modbus registers** but a separate configuration file in flash memory; they require a **restart** to be applied after saving (dedicated button at the bottom of the page). If not yet configured, **Client ID** and **Base topic** are proposed based on the board Serial Number (see "General settings"): BTR102-<SN> and BTR102/<SN> (e.g. BTR102-260603115422, BTR102/260603115422), editable; if saved empty, these same values are used anyway when connecting (the only mandatory field is the **Broker host**)
- **Mode** — 3-state selector:

Mode	Behaviour
OFF	No publishing
ON	Continuous publishing, one message for each complete RTU read cycle
SCHEDULED	Publishing starts automatically at the date/time set in "RTU request configuration" (§5.5) — same scheduled start mechanism as SD/USB, shared between the three channels

Connection status: shows whether the device is currently connected to the configured broker.

Remote commands (read only): allow a cloud system connected to the same broker to read the status, configuration and SD/USB files of the device, with the same information as the web pages. No command changes the configuration.

- **Enable commands via MQTT:** off by default

- **Command topic / Response topic:** topics on which the device receives requests and publishes responses. If left empty, <Topic base>/command and <Topic base>/response apply (shown in grey in the field). Maximum 96 characters; the characters +, # and spaces are not allowed; the two topics must differ from each other and from the data topics (<Topic base>/header, <Topic base>/data)
- Like the other connection parameters, they require a **restart** after saving

Format of a request (JSON published on the command topic) and of the response (on the response topic):

```
{ "corr": "a1", "api_request": "/usb_status" }
{ "corr": "a1", "api_request": "/usb_status", "status": "ok", "data": { ... } }
```

corr is a free identifier chosen by the sender, copied into the response to match it to the request. In case of error the response contains "status": "error" and a code in "message".

Request ()	Content of
/status	Peripheral status (like the dots in the header)
/data	Live values of the RTU/TCP requests (§5.5)
/json	RTU/TCP request configuration (Appendix B)
/sd_status, /usb_status	SD/USB stick status and list of files
/log_config	Logging mode and polling period
/general_config	General settings, temperature, connected clients
/mqtt_status	Broker connection status
/seclog	Security log (§6), in JSON format
/sd_download, /usb_download	A chunk of a file (see below)

SD/USB files are read in chunks of up to 4096 bytes:

{ "corr": "f1", "api_request": "/usb_download", "payload": { "name": "FILENAME.csv", "offset": 0, "len": 4096 } }. The response contains size (total size), offset, len, eof (last chunk) and the Base64-encoded content in the b64 field; the next chunk is requested with offset increased by len, until eof is true.

The device executes at most one request every 200 ms and queues up to 4: beyond that it responds ERR_API_BUSY. A request longer than 512 bytes is ignored.

NOTE Security: whoever can publish on the command topic can read the device data. Access to the topics must be restricted by the broker rules (policy/ACL), especially if several devices use the same client certificate.

Certificates (Mutual TLS): three files to upload — **broker CA** (to verify the identity of the broker), **client certificate** and **client private key** (for mutual authentication, provided by the broker operator or generated together with it). After uploading, you must **restart the device** to activate the new configuration.

NOTE Unlike SD/USB, the MQTT mode has no fallback mechanism of its own: if the publication of a cycle fails (broker unreachable, certificate expired, etc.), that cycle is lost unless SD and/or USB are in turn set to the **FALLBACK MQTT** value of the "Logging" selector (§5.5) — in that case they write a local CSV record only for the cycles in which MQTT publication actually failed.

5.9 General settings

This page collects cross-cutting information and settings, described in detail in chapters 6-9:

- **CID — Device identification:** Machine ID, HW/FW versions, serial number, MAC addresses of all interfaces (read only)
- **Firmware update (OTA)** → chapter 7
- **TLS certificates, Access credentials, Security log, Modbus TCP allowlist** → chapter 6
- **Active connections:** number and IP of the clients currently connected (Modbus TCP, HTTP/HTTPS, WiFi AP)
- **Modbus TCP settings:** TCP slave address (1–247) with which the device responds to Modbus TCP requests
- **Interface language:** Italian/English, effective immediately on the whole web page without a restart
- **Reset to default** → chapter 8

BTR102 SD USB MQTT ETH 192.168.178.44 STA 192.168.178.43 -54dBm AP 192.168.4.1 0 cli SNTp

Impostazioni Generali

Identificazione dispositivo, aggiornamento firmware, ripristino

CID - IDENTIFICAZIONE DISPOSITIVO

Machine ID	BTR102
HW Version	1
FW Version	1
Variant	0
Feature	0
Serial Number	260915160252
MAC ADDRESS	
WiFi STA	64:E8:33:67:A2:E4
WiFi AP	64:E8:33:67:A2:E5
Ethernet	64:E8:33:67:A2:E7
Bluetooth	64:E8:33:67:A2:E6

AGGIORNAMENTO FIRMWARE (OTA)

File firmware Trascina qui BTR102_REV*.bin o clicca per sfogliare

[Aggiorna firmware](#)

Web page — General settings (CID, firmware update)

BTR102 SD USB MQTT ETH 192.168.178.44 STA 192.168.178.43 -55dBm AP 192.168.4.1 0 cli SNTp

Impostazioni Generali

Identificazione dispositivo, aggiornamento firmware, ripristino

CERTIFICATI TLS

Caricamento certificati generati dal collaudo (self-signed non supportato - richiede certificato dispositivo con catena CA intermedia + chiave privata)

Certificato (+ catena) Sfogliala... Nessun_onato. Carica Presente

Chiave privata Sfogliala... Nessun_onato. Carica Presente

Scadenza certificato 2028-12-18 (tra 821 giorni)

[Riavvia per attivare HTTPS](#)

CREDENZIALI DI ACCESSO

Utente admin

Nuova password (utente finale)

Nuova password (installatore)

[Aggiorna credenziali](#)

Web page — General settings (TLS certificates, access credentials)

BTR102 SD USB MQTT ETH 192.168.178.44 STA 192.168.178.43 -55dBm AP 192.168.4.1 0 cLi SNTp

Impostazioni Generali

Identificazione dispositivo, aggiornamento firmware, ripristino

CONNESSIONI ATTIVE

Socket TCP	0
Socket HTTP	1
Client WiFi AP	0

LOG DI SICUREZZA

Ultimo evento n/d - REBOOT boot: power-on

Eventi da accensione 1

Ultimi 50 eventi [Scarica log \(CSV\)](#)

Lingua interfaccia Italiano

Indirizzo slave TCP 247 (1-247)

[Reset to Default](#)

Web page — General settings (security log, Modbus TCP allowlist, reset)

BTR102 SD USB MQTT ETH 192.168.178.44 STA 192.168.178.43 -56dBm AP 192.168.4.1 0 cLi SNTp

Impostazioni Generali

Identificazione dispositivo, aggiornamento firmware, ripristino

Ultimo evento n/d - REBOOT boot: power-on

Eventi da accensione 1

Ultimi 50 eventi [Scarica log \(CSV\)](#)

Lingua interfaccia Italiano

Indirizzo slave TCP 247 (1-247)

ALLOWLIST MODBUS TCP

Abilita allowlist ☒

IP consentiti
uno per riga, max 16

192.168.1.18
192.168.1.20

[Salva allowlist](#)

Reset to Default [Reset to Default](#)

[Invia configurazione](#) Installatore e utente [Riavvia](#)

6 SECURITY

6.1 Authentication

Every page and every action of the web interface requires authentication (browser Basic Auth). As described in §3.4, there are two independent sets of credentials — **user** and **installer** — with different permissions.

6.2 HTTPS and TLS certificates

The **TLS certificates** panel (General settings) allows uploading:

- **Certificate:** the TLS certificate of the device (self-signed, generated during testing — see §9.3);
- **Private key** matching it.

After uploading both files you must **restart the device** (dedicated button in the same panel) to activate HTTPS with the new credentials.

The panel also shows the **expiry of the certificate** currently installed (date and days remaining); below 30 days remaining, a warning appears both in the panel and in the security log.

NOTE *Renewal:* as expiry approaches, generate a new self-signed device certificate (same hostnames as the device) and upload it again with the same procedure described above.

6.3 Access credentials

In the **Access credentials** panel you can set a new password for the end user and/or for the installer. The **username is fixed to admin for both roles** (user and installer) and cannot be changed: only the two passwords, separate and independent from each other, can be configured. The change requires the current installer credentials.

NOTE *It is strongly recommended to change the factory passwords at first use.*

6.4 Physical credential reset

See §4.4 — procedure from the front panel (capacitive button held for 10 seconds on the dedicated display page), useful if the credentials are lost.

6.5 Rate limiting on access attempts

After **5 consecutive failed authentication attempts** from the same IP address, the device blocks further attempts from that IP for **60 seconds** (HTTP 429 response). A correct login resets the failure counter.

6.6 Security log

The **Security log** panel shows the last event recorded and the number of events recorded since the last power-on, with a button to **download the complete history in CSV** (up to the 50 most recent events, kept across restarts).

Types of events recorded: failed login attempts, configuration changes, start/outcome of firmware updates, restarts (commanded or abnormal), credential resets, Modbus TCP connections refused because the limit was reached or by the allowlist.

6.7 Allowlist Modbus TCP

In the **Modbus TCP allowlist** panel you can restrict the accepted Modbus TCP connections to a list of authorised IPs (up to 16 addresses, one per line). With the allowlist disabled (factory setting) the device accepts connections from any IP, as is normal for Modbus TCP.

NOTE *A client whose IP is not in the list simply sees the connection close, without any application response — intentional behaviour, consistent with good network security practice.*

7 FIRMWARE UPDATE (OTA)

Two modes are available, both with the same update mechanism under the hood (see the note on rollback below).

7.1 From the web page

In the **Firmware update (OTA)** panel (General settings):

1. Drag (or select by clicking) the `BTR102_REV*.bin` firmware file supplied by the manufacturer.
2. After checking the file name, press **Update firmware**.
3. A progress bar shows the upload progress.
4. When finished, the device automatically restarts on the new firmware.

The operation requires the installer credentials.

7.2 From USB stick

A convenient alternative when the device cannot be reached over the network (e.g. first commissioning, field maintenance):

1. On the USB stick, create an `ota` folder in the root and copy into it the `BTR102_REV*.bin` firmware file supplied by the manufacturer.
2. Insert the stick into the device: if the file found is a different revision from the one currently running, the update **starts automatically**, without any action by the user.
3. During the update the TFT display shows the full-screen message **"OTA IN CORSO"** with the progress percentage — do not disconnect the stick or switch off the device during this phase.
4. When finished, the device automatically restarts on the new firmware.

NOTE *If the file on the stick is already the running revision, the update is skipped automatically (no action, no message on screen) — the same stick can stay inserted or be reused on other devices already updated without risk of repeated reflashing.*

NOTE

NOTE *If the update fails (e.g. corrupted file), the device keeps working with the previous firmware and the file on the stick is not modified: the update will be retried automatically at each subsequent insertion of the stick, until the file is replaced or removed.*

No credentials required (physical access to the device is in itself a security requirement).

7.3 Automatic rollback

The device keeps two firmware slots (A/B) with an automatic rollback mechanism: if the new firmware does not complete start-up correctly, the device reverts to the previous working version. Every update (from web or USB) is recorded in the security log (start, success or failure).

8 MAINTENANCE

8.1 Restart

Some actions (e.g. activating HTTPS after uploading certificates) require an explicit restart, available via the dedicated button in the relevant panel. The restart requires explicit confirmation and the installer credentials.

8.2 Factory reset

The **Reset to default** button (General settings) restores all configurable settings to factory values. The operation requires explicit confirmation and the installer credentials.

The reset also deletes the **RTU/TCP request configuration** (table in §5.5): all 20 rows become empty and logging on SD/USB restarts on a new file. **Not** affected: the passwords (web user/installer and WiFi AP — for these see the physical reset from the display, §4.4), the TLS and MQTT certificates, the MQTT broker connection parameters (§5.8) and the security log.

NOTE *The reset does not affect the device identification data defined during testing (Machine ID, calibration data, derived hostnames) — see chapter 9.*

8.3 TLS certificate replacement/renewal

See §6.2.

9 PROVISIONING AND TESTING (INSTALLER)

This chapter describes the one-off operations carried out during production/testing of the device, given here as a reference for those managing the commissioning of several units.

9.1 Machine ID and calibration

Each device receives during testing a unique **Machine ID** (8 characters) and a calibration date, both unchangeable after testing and visible read-only in the **CID** panel of the web page and on the **Settings** page of the TFT display (as serial number).

9.2 Automatically generated hostnames and SSIDs

Starting from the Machine ID and calibration date, at every start-up the device automatically (and deterministically, always the same over time) generates three distinct mDNS hostnames — one for Ethernet, one for WiFi STA, one for WiFi AP — and the SSID of the WiFi AP network, in the format `<machine_id>-<interface>-<calibration date>` (e.g. `BTR102-eth-260603115422`). These values cannot be changed from the web interface (they appear as read-only fields) — they are also the reference (SAN) used in the device TLS certificate.

9.3 Generating and uploading the TLS certificates

The device certificate is **self-signed**, generated offline during testing using the three hostnames of the previous point as SAN (Subject Alternative Name) and as CN. The resulting file (certificate and private key, two separate files) is then uploaded via the **TLS certificates** panel described in §6.2.

NOTE *Since there is no common CA between devices, each certificate must be approved individually on the connecting browser/client (no shared root to be installed once) — consistent with the absence of a centralised fleet management system.*

9.4 Factory credentials

The factory passwords (user/installer/WiFi AP) are set during testing and shown on the device label or in the delivery documentation provided by the installer.

10 TROUBLESHOOTING

Symptom	Possible cause / action
I cannot reach the web page	Check the current IP on the TFT display (§4.3); try the <code>.local</code> hostname or connect to the device WiFi AP network
The browser shows an untrusted certificate warning	Normal with a self-signed certificate not yet installed as trusted on the PC (§9.3) — check that the hostname you connect to matches one of the three SANs of the certificate
The browser asks for credentials for every action	Expected behaviour: some actions require the installer role even if already authenticated as user
Error 429 after a few login attempts	Rate limiting active (§6.5): wait 60 seconds and try again with the correct credentials
An RTU request is always "not valid" (<code>valid=0</code>)	Check slave address (<code>req_id</code>), RS485 wiring, serial parameters (§5.2), and that <code>req_qty</code> matches the configured register types
SD logging writes nothing	Check that the SD is mounted (SD panel), that the logging mode is not OFF and — for SCHEDULED — that the set date/time is consistent with the current RTC clock
SD or USB icon/dot yellow	Media enabled but not inserted (or USB stick not recognised): insert the microSD/stick, or disable the channel if not used (§5.6/§5.7)
SD or USB icon/dot red	Media inserted but not readable: check the file system (FAT32/exFAT) or replace the card/stick
"Search networks" does not find networks that are present	Read the message next to the button (§5.4): if red, it gives the reason (e.g. WiFi STA disabled: enable it and restart). If it says "No network found", check that the networks are 2.4 GHz (the BTR102 does not see 5 GHz) and the antenna connection
Saving the request table gives a red message	A configured row does not pass the checks of §5.5: the message shows the problem and the row (<code>req_n</code>); nothing has been saved, correct and save again
I have forgotten the credentials	Physical reset procedure from the display, §4.4
The device does not respond/seems frozen	Restart by removing and restoring power; if the problem persists after a firmware update, the automatic rollback should already have restored the previous version at the next restart

11 APPENDIX A — WEB ENDPOINT SUMMARY

Technical reference table for those integrating automation/monitoring tools with the device web interface (advanced use, not needed for everyday operation). Table checked row by row against the source code (`app_web.c`) on 2026-09-21 — the previous version was out of date (endpoints added from August 2026 onwards were missing, e.g. the whole USB/MQTT branch, and it listed an `/sd_log` that never existed in the firmware).

All endpoints share the same fixed username `admin` (§6.1) — only which of the two passwords ("user" or "installer") is accepted by the required role changes.

Endpoint	Method	Required role	Purpose
<code>/data</code>	GET	User	Live values of the configured RTU/TCP requests (same table as page 5), updated at every polling cycle
<code>/status</code>	GET	User	Peripheral status summary for the coloured dots in the web header (SD/USB/STA/MQTT)
<code>/json</code>	GET/POST	Installer (POST only)	RTU/TCP request configuration (<code>gw.json</code>)
<code>/cid</code>	GET	User	Device identification data: machine ID, HW/FW versions, calibration date, MAC addresses
<code>/rtc_config</code>	GET/POST	Installer (POST only)	RTC date/time, UTC time zone, daylight saving time
<code>/rtu_config</code>	GET/POST	Installer (POST only)	Modbus RTU serial parameters (baud rate/parity/stop bits/timeout)
<code>/log_config</code>	GET/POST	User	Polling period, logging mode (OFF/ON/SCHEDULED/MQTT-fallback) and enable per SD/USB channel
<code>/eth_config</code>	GET/POST	Installer (POST only)	Ethernet configuration
<code>/wifi_config</code>	GET/POST	Installer (POST only)	WiFi STA/AP configuration
<code>/wifi_scan</code>	POST	Installer	Starts a scan of the available WiFi networks
<code>/wifi_connect</code>	POST	Installer	Connects the device to a selected WiFi network
<code>/sd_status</code>	GET	User	SD mount status, free space, list of files
<code>/sd_enable</code>	POST	User	Physically enables/disables the SD (mount attempt)
<code>/sd_download</code>	GET	User	Downloads a CSV file from the SD (parameter <code>name</code> , optional <code>from/to</code> for a time filter)
<code>/sd_delete</code>	POST	Installer	Deletes a file from the SD
<code>/usb_status</code>	GET	User	USB stick mount status, free space, list of files
<code>/usb_enable</code>	POST	User	Physically enables/disables the USB port (MSC host)
<code>/usb_download</code>	GET	User	Downloads a CSV file from the USB stick (same parameters as <code>/sd_download</code>)
<code>/usb_delete</code>	POST	Installer	Deletes a file from the USB stick
<code>/general_config</code>	GET/POST	Installer (POST only)	Modbus TCP slave address,

Endpoint	Method	Required role	Purpose
			active connections/sockets, IPs of clients associated with the WiFi AP
/auth_config	POST	User*	Changes the end-user and/or installer password — see note below
/tls_cert	GET/POST	Installer (POST only)	Reads/uploads the device TLS certificate
/tls_key	GET/POST	Installer (POST only)	GET only checks that the private key is present (never its content); POST uploads it
/tls_cert_expiry	GET	User	Days remaining before the TLS certificate expires
/mqtt_config	GET/POST	Installer (POST only)	MQTT broker connection (host/port/client ID/base topic), remote commands (enable, command/response topics) + enable/mode status
/mqtt_status	GET	User	MQTT broker connection status
/mqtt_ca_cert, /mqtt_client_cert	GET/POST	Installer (POST only)	Read/upload the certificates for MQTT mutual TLS
/mqtt_client_key	GET/POST	Installer (POST only)	GET only checks that the key is present (never its content); POST uploads it
/modbus_allowlist	GET/POST	Installer (POST only)	IP allowlist for the Modbus TCP server (port 502)
/seclog	GET	User	Security log download (CSV)
/ota	POST	Installer	Firmware upload (manual OTA update)
/reboot	POST	Installer	Restart (requires body {"confirm":"yes"})
/reset_default	POST	Installer	Factory reset (requires body {"confirm":"yes"})

* /auth_config accepts any authenticated role to change its **own** password (field password), but requires the Installer role to also change ins_password — a User trying to change it is refused.

11.1 How to build a call

Each endpoint can only be reached via **HTTPS** (never plain HTTP) and requires **Basic Auth** authentication:

1. **Protocol and host:** `https://<device-IP-or-hostname>` — the IP is the one assigned to the device on the network (Ethernet, WiFi STA or WiFi AP depending on how it is accessed), readable from the TFT display or from the router/DHCP.
2. **Credentials:** username always `admin`; password the "user" or "installer" one set on the "General settings" page — from the factory they are in the device test document. You need the password of the role required by the table above (an installer password also works where the user one would be enough, not vice versa).
3. **Certificate:** it is self-signed (private factory CA) — a browser will show a warning to be accepted manually ("Advanced" → "Proceed anyway"); from the command line (curl) the option `-k/--insecure` is needed to skip verification.
4. **Method:** GET to read data, POST (with JSON body where applicable) to change a configuration.

From a browser (GET endpoints only, e.g. to download a CSV or view live data), the credentials can be embedded directly in the URL:

```
https://admin:PASSWORD@IP-DEVICE/data
```

From the command line (curl, works for both GET and POST):

```
curl -k -u admin:PASSWORD "https://IP-DEVICE/data"
```

NOTE *Caution on Windows PowerShell: in Windows PowerShell 5.1 (the one preinstalled on Windows 10/11, not PowerShell 7+) curl is an **alias** of Invoke-WebRequest, which has a completely different syntax — the -u flag is interpreted as an ambiguous parameter (-UseBasicParsing/-Uri/...) and the command fails with an AmbiguousParameter error, not a network error. Check with Get-Command curl: if the result is CommandType: Alias, use ``curl.exe`` instead of curl (same arguments, e.g. `curl.exe -k -u admin:PASSWORD "https://IP-DEVICE/data"`) to call the real binary bypassing the alias — or run the command from **cmd.exe** or **Git Bash**, where the alias does not exist. Verified in practice on 2026-09-21.*

11.2 Practical examples

Replace IP-DEVICE with the real address and PASSWORD with the password of the required role (see table §11).

What	Command
Live values of the RTU/TCP requests	<code>curl -k -u admin:PASSWORD "https://IP-DEVICE/data"</code>
RTU/TCP request configuration (gw.json)	<code>curl -k -u admin:PASSWORD "https://IP-DEVICE/json"</code>
List of CSV files on the SD card	<code>curl -k -u admin:PASSWORD "https://IP-DEVICE/sd_status"</code>
Download a CSV file from the SD	<code>curl -k -u admin:PASSWORD "https://IP-DEVICE/sd_download?name=BTR102_20260921_103000.csv" -o log.csv</code>
Download a CSV file from the SD, only a time range	<code>curl -k -u admin:PASSWORD "https://IP-DEVICE/sd_download?name=BTR102_20260921_103000.csv&from=20260921090000&to=20260921120000" -o log.csv</code>
Download a CSV file from the USB stick	<code>curl -k -u admin:PASSWORD "https://IP-DEVICE/usb_download?name=BTR102_20260921_103000.csv" -o log.csv</code>
Device identification data (machine ID, FW versions, MAC)	<code>curl -k -u admin:PASSWORD "https://IP-DEVICE/cid"</code>
Security log	<code>curl -k -u admin:PASSWORD "https://IP-DEVICE/seclog" -o seclog.csv</code>

NOTE *sd_download/usb_download require the **exact file name** (obtainable from /sd_status//usb_status) — wildcards are not supported.*

12 APPENDIX B — FORMAT OF THE GW.JSON CONFIGURATION FILE

The file `/gw/gw.json`, saved in internal flash memory, defines the RTU/TCP requests described in §5.5:

```
{
  "requests": [
    {
      "req_n": 1,
      "req_type": 0,
      "req_id": 1,
      "req_ip": "0.0.0.0",
      "req_addr": 40100,
      "req_qty": 3,
      "valid": 1,
      "registers": [
        { "index": 1, "name": "Temperature", "type": "float", "unit": "°C", "value": 23.5 },
        { "index": 2, "name": "Pressure", "type": "uint16_t", "unit": "Pa", "value": 101325 }
      ]
    }
  ]
}
```

- `req_n`: request index (1–20)
- `req_type`: 0 = RTU (default if the field is missing, for compatibility with files saved by firmware prior to September 2026), 1 = TCP-WiFi AP, 2 = TCP-WiFi STA, 3 = TCP-Ethernet
- `req_id`: Modbus RTU slave address (`req_type` 0) or Modbus Unit Identifier of the TCP slave (`req_type` 1-3)
- `req_ip`: IP address of the slave — present only for `req_type` 1-3, ignored for RTU (Modbus TCP port fixed at 502)
- `req_addr`: first register to read, in absolute Modbus notation — 30001..39999 = input register (function 4), 40001..65535 = holding register (function 3); same meaning for RTU and TCP
- `req_qty`: total registers used by the types listed in `registers` (`uint16_t/int16_t` = 1 register; `uint32_t/int32_t/uint32_rev/int32_rev/float/float_rev` = 2 registers; `int64_t/int64_rev` = 4 registers — see full table in §5.5)
- `registers`: list of the individual values expected in that block, with name, type and unit of measurement (also used to label the columns in the web interface and in the CSV log file)

NOTE This file is normally modified only through the web interface table (§5.5), which guarantees its consistency; a manual change to the file requires care to keep `req_qty` consistent with the declared types, otherwise the device refuses the configuration on saving (with a specific error message also when a TCP request limit is exceeded, §5.5).

NOTE The `err_count` field visible in the web interface table (§5.5, error counter) **is not part of this file**: it is only added in the live response of `GET /data`, lives in RAM and restarts from 0 at every restart and every new saved configuration — it is never written to `gw.json`.

JSON published via MQTT (§5.8)

When MQTT publishing is active, the device publishes two distinct JSONs on the configured broker, both derived from the structures already described above — same field names, no new format to learn — but **reduced to the active requests only** (`req_id` other than 0 and `req_qty` other than 0): unlike `GET /json/GET /data` (which always include all 20 rows configurable on page 5, including empty ones), the MQTT payloads omit unconfigured rows entirely, so as not to weigh down the message published on the broker.

- `<topic base>/header` — published **retained**: same structure as the file `gw.json` (§12) and as `GET /json`, but only with the `requests[]` blocks of the active requests. Republished automatically at **every successful connection to the broker** (not only the very first after power-on: also after a reconnection, e.g. following a temporary broker outage) and after every successful save of the request configuration.
- `<topic base>/data` — published **once per read cycle** (not retained): values read in that cycle, same structure as `GET /data` (shown as a table in the web interface, which however always shows all 20 rows) but with a `requests[]` block only for the active requests:

```
{
  "cycle": 5040,
  "requests": [
    {
      "req_n": 1,
      "valid": 1,
      "err_count": 0,
      "registers": [
        { "index": 1, "value": 23.5 },
        { "index": 2, "value": null }
      ]
    }
  ]
}
```

- `cycle`: number of complete read cycles carried out since the last start-up (same "**Cycle no.**" counter as §5.5)
- `requests[].req_n`: request index, corresponds to `req_n` in `gw.json`
- `requests[].valid`: outcome of the last poll of that request (1 = successful)
- `requests[].err_count`: RAM-only error counter described above — included here because it is precisely via MQTT/web interface that it makes sense to monitor it, since it is not saved anywhere
- `requests[].registers[].value`: value read for that register index (number, or `null` if the cycle did not produce a valid value for that single register, e.g. NaN/Inf on a `float`); name/type/unit of measurement are **not** repeated here, they are already known from the header
- a request that is **not active** does not appear at all in the `requests[]` array (unlike `GET /data`, where it appears anyway with `registers` empty/`null`) — see note above on the filter applied to MQTT payloads

13 APPENDIX C — COMPLETE MODBUS REGISTER MAP

Complete list of the Modbus registers (40001 onwards) exposed by the BTR102, generated from the reference sheet BTR102_MEMORY_MAP.xlsm (source of truth also used to generate the firmware). The **Read/Write Role** columns show the minimum access level required for reading/writing via Modbus TCP (not to be confused with the User/Installer roles of the web interface, §3.4).

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40001	char	machine_id_1	Machine ID, set by Chronos. 1 char per register	B	ALWAYS READABLE	WRITABLE WITH PASS C	0	255
40002	char	machine_id_2	Machine ID, set by Chronos. 1 char per register	T	ALWAYS READABLE	WRITABLE WITH PASS C	0	255
40003	char	machine_id_3	Machine ID, set by Chronos. 1 char per register	R	ALWAYS READABLE	WRITABLE WITH PASS C	0	255
40004	char	machine_id_4	Machine ID, set by Chronos. 1 char per register	1	ALWAYS READABLE	WRITABLE WITH PASS C	0	255
40005	char	machine_id_5	Machine ID, set by Chronos. 1 char per register	0	ALWAYS READABLE	WRITABLE WITH PASS C	0	255
40006	char	machine_id_6	Machine ID, set by Chronos. 1 char per register	2	ALWAYS READABLE	WRITABLE WITH PASS C	0	255
40007	char	machine_id_7	Machine ID, set by Chronos. 1 char per register	\0	ALWAYS READABLE	WRITABLE WITH PASS C	0	255
40008	char	machine_id_8	Machine ID, set by Chronos. 1 char per register	\0	ALWAYS READABLE	WRITABLE WITH PASS C	0	255
40009	unsigned short	hw_version	Hardware version, set by Chronos.	1	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40010	unsigned short	fw_version	Firmware version, set by Chronos.	1	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40011	unsigned short	mounted_version	assembly variants	0	ALWAYS READABLE	WRITABLE WITH PASS C	0	1
40012	unsigned short	year_month	calibration year and month	0	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40013	unsigned short	day_Hour	calibration day and hour	0	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40014	unsigned short	minute_second	calibration minute and second	0	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40015	unsigned short	feature_version	0: Bridge TCP-RTU 1: Gateway TCP-RTU 2:Gateway MQTT	0	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40016	unsigned short	Mac_address_Wifi_STA_1	Mac address Wifi Station MSB	11	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40017	unsigned short	Mac_address_Wifi_STA_2	Mac address Wifi Station	22	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40018	unsigned short	Mac_address_Wifi_STA_3	Mac address Wifi Station	33	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40019	unsigned short	Mac_address_Wifi_STA_4	Mac address Wifi Station	44	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40020	unsigned short	Mac_address_Wifi_STA_5	Mac address Wifi Station	55	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40021	unsigned short	Mac_address_Wifi_STA_6	Mac address Wifi Station LSB	66	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40022	unsigned short	Mac_address_Wifi_SOFT_AP_1	Mac address Wifi software enabled access point MSB	11	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40023	unsigned short	Mac_address_Wifi_SOFT_AP_2	Mac address Wifi software enabled access point	22	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40024	unsigned short	Mac_address_Wifi_SOFT_AP_3	Mac address Wifi software enabled access point	33	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40025	unsigned short	Mac_address_Wifi_SOFT_AP_4	Mac address Wifi software enabled access point	44	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40026	unsigned short	Mac_address_Wifi_SOFT_AP_5	Mac address Wifi software enabled access point	55	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40027	unsigned short	Mac_address_Wifi_SOFT_AP_6	Mac address Wifi software enabled access point LSB	66	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40028	unsigned short	Mac_address_Ethernet_1	Mac address Ethernet MSB	31	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40029	unsigned short	Mac_address_Ethernet_2	Mac address Ethernet	32	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40030	unsigned short	Mac_address_Ethernet_3	Mac address Ethernet	33	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40031	unsigned short	Mac_address_Ethernet_4	Mac address Ethernet	34	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40032	unsigned short	Mac_address_Ethernet_5	Mac address Ethernet	35	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40033	unsigned short	Mac_address_Ethernet_6	Mac address Ethernet LSB	36	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40034	unsigned short	Mac_address_Bluetooth_1	Mac address Bluetooth MSB	96	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40035	unsigned short	Mac_address_Bluetooth_2	Mac address Bluetooth	95	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40036	unsigned short	Mac_address_Bluetooth_3	Mac address Bluetooth	94	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40037	unsigned short	Mac_address_Bluetooth_4	Mac address Bluetooth	93	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40038	unsigned short	Mac_address_Bluetooth_5	Mac address Bluetooth	92	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40039	unsigned short	Mac_address_Bluetooth_6	Mac address Bluetooth LSB	91	ALWAYS READABLE	WRITABLE WITH PASS C	0	65535
40060	unsigned short	modbus_TCP_address	Modbus TCP address (to access its registers)	247	ALWAYS	ALWAYS	0	65535

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40061	unsigned short	modbus_RTU_delay	Delay Modbus answer (in ms)	0	READABLE ALWAYS READABLE	WRITABLE ALWAYS WRITABLE	0	65535
40062	unsigned short	modbus_RTU_baudrate	Baudrate 0 → 1200 1 → 2400 2 → 4800 3 → 9600 4 → 19200 5 → 38400 6 → 57600 7 → 115200	3	ALWAYS READABLE	ALWAYS WRITABLE	0	65535
40063	unsigned short	modbus_RTU_parity	0 → NONE 1 → EVEN 2 → ODD	0	ALWAYS READABLE	ALWAYS WRITABLE	0	65535
40064	unsigned short	modbus_RTU_Stop_bit	0 → 1 Bit 1 → 1,5 Bit 2 → 2 Bit	0	ALWAYS READABLE	ALWAYS WRITABLE	0	65535
40065	unsigned short	modbus_RTU_timeout	receive timeout in ms	1000	ALWAYS READABLE	ALWAYS WRITABLE	100	65535
40066	unsigned short	Machine_Configuration_Register	#define USE_NTP 0x0001 #define BUZZER_IN_TOUCH 0x0002 #define BUZZER_IN_ALARM 0x0004 #define WIFI_STA_ENABLE 0x0008 #define WIFI_AP_ENABLE 0x0010 #define MQTT_ENABLE 0x0020 #define SD_ENABLE 0x0040 #define USB_ENABLE 0x0080	218	ALWAYS READABLE	ALWAYS WRITABLE	0	65535
40133	unsigned short	Wifi_STA_IP_0	WI-FI STA IP static (DHCP off) 0	192	ALWAYS READABLE	ALWAYS WRITABLE	0	65535
40134	unsigned short	Wifi_STA_IP_1	WI-FI STA IP static (DHCP off) 1	168	ALWAYS READABLE	ALWAYS WRITABLE	0	65535
40135	unsigned short	Wifi_STA_IP_2	WI-FI STA IP static (DHCP off) 2	178	ALWAYS READABLE	ALWAYS WRITABLE	0	65535
40136	unsigned short	Wifi_STA_IP_3	WI-FI STA IP static (DHCP off) 3	200	ALWAYS READABLE	ALWAYS WRITABLE	0	65535
40137	unsigned short	Wifi_STA_Netmask_0	WI-FI STA NETMASK static (DHCP off) 0	255	ALWAYS READABLE	ALWAYS WRITABLE	0	65535
40138	unsigned short	Wifi_STA_Netmask_1	WI-FI STA NETMASK static (DHCP off) 1	255	ALWAYS READABLE	ALWAYS WRITABLE	0	65535
40139	unsigned short	Wifi_STA_Netmask_2	WI-FI STA NETMASK static (DHCP off) 2	255	ALWAYS READABLE	ALWAYS WRITABLE	0	65535
40140	unsigned short	Wifi_STA_Netmask_3	WI-FI STA NETMASK static (DHCP off) 3	0	ALWAYS READABLE	ALWAYS WRITABLE	0	65535
40141	unsigned short	Wifi_STA_Gateway_0	WI-FI STA GATEWAY static (DHCP off) 0	192	ALWAYS	ALWAYS	0	65535

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40142	unsigned short	Wifi_STA_Gateway_1	WI-FI STA GATEWAY static (DHCP off) 1	168	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40143	unsigned short	Wifi_STA_Gateway_2	WI-FI STA GATEWAY static (DHCP off) 2	178	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40144	unsigned short	Wifi_STA_Gateway_3	WI-FI STA GATEWAY static (DHCP off) 3	1	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40177	unsigned short	Wifi_AP_IP_0	WI-FI AP IP 0	192	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40178	unsigned short	Wifi_AP_IP_1	WI-FI AP IP 1	168	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40179	unsigned short	Wifi_AP_IP_2	WI-FI AP IP 2	4	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40180	unsigned short	Wifi_AP_IP_3	WI-FI AP IP 3	1	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40181	unsigned short	Wifi_AP_Netmask_0	WI-FI AP NETMASK 0	255	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40182	unsigned short	Wifi_AP_Netmask_1	WI-FI AP NETMASK 1	255	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40183	unsigned short	Wifi_AP_Netmask_2	WI-FI AP NETMASK 2	255	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40184	unsigned short	Wifi_AP_Netmask_3	WI-FI AP NETMASK 3	0	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40185	unsigned short	Wifi_AP_Gateway_0	WI-FI AP GATEWAY 0	192	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40186	unsigned short	Wifi_AP_Gateway_1	WI-FI AP GATEWAY 1	168	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40187	unsigned short	Wifi_AP_Gateway_2	WI-FI AP GATEWAY 2	4	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40188	unsigned short	Wifi_AP_Gateway_3	WI-FI AP GATEWAY 3	1	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40189	unsigned short	Ethernet_IP_0	ETHERNET IP static (DHCP off) 0	192	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40190	unsigned short	Ethernet_IP_1	ETHERNET IP static (DHCP off) 1	168	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40191	unsigned short	Ethernet_IP_2	ETHERNET IP static (DHCP off) 2	178	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40192	unsigned short	Ethernet_IP_3	ETHERNET IP static (DHCP off) 3	250	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40193	unsigned short	Ethernet_Netmask_0	ETHERNET NETMASK static (DHCP off) 0	255	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40194	unsigned short	Ethernet_Netmask_1	ETHERNET NETMASK static (DHCP off) 1	255	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40195	unsigned short	Ethernet_Netmask_2	ETHERNET NETMASK static (DHCP off) 2	255	READABLE ALWAYS	WRITABLE ALWAYS	0	65535

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40196	unsigned short	Ethernet_Netmask_3	ETHERNET NETMASK static (DHCP off) 3	0	ALWAYS READABLE	ALWAYS WRITABLE	0	65535
40197	unsigned short	Ethernet_Gateway_0	ETHERNET GATEWAY static (DHCP off) 0	192	ALWAYS READABLE	ALWAYS WRITABLE	0	65535
40198	unsigned short	Ethernet_Gateway_1	ETHERNET GATEWAY static (DHCP off) 1	168	ALWAYS READABLE	ALWAYS WRITABLE	0	65535
40199	unsigned short	Ethernet_Gateway_2	ETHERNET GATEWAY static (DHCP off) 2	178	ALWAYS READABLE	ALWAYS WRITABLE	0	65535
40200	unsigned short	Ethernet_Gateway_3	ETHERNET GATEWAY static (DHCP off) 3	1	ALWAYS READABLE	ALWAYS WRITABLE	0	65535
40201	unsigned long	Communications_settings	#define WIFI_STA_DHCP 0X0001 #define ETHERNET_DHCP 0X0002	3	ALWAYS READABLE	ALWAYS WRITABLE	0	4294967295
40203	signed long	UTC_Time_zone_value	The value of this register sets the Time Zone out of a list of 461 elements. The entire list is in the documentation, the default value is 390 which is for 'Europe/Rome'	390	ALWAYS READABLE	ALWAYS WRITABLE	-2147483648	2147483647
40205	unsigned short	interface_language	#define ITALIANO 0X0000 #define ENGLISH 0X0001	0	ALWAYS READABLE	ALWAYS WRITABLE	0	65535
40206	unsigned short	ora_solare_legale	0: standard time1: daylight saving time	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40209	unsigned short	logging_register_address_1	Modbus Address to do logging in position 1	40000	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40210	unsigned short	logging_register_address_2	Modbus Address to do logging in position 2	40000	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40211	unsigned short	logging_register_address_3	Modbus Address to do logging in position 3	40000	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40212	unsigned short	logging_register_address_4	Modbus Address to do logging in position 4	40000	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40213	unsigned short	logging_register_address_5	Modbus Address to do logging in position 5	40000	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40214	unsigned short	logging_register_address_6	Modbus Address to do logging in position 6	40000	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40215	unsigned short	logging_register_address_7	Modbus Address to do logging in position 7	40000	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40216	unsigned short	logging_register_address_8	Modbus Address to do logging in position 8	40000	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40217	unsigned short	logging_register_address_9	Modbus Address to do logging in position 9	40000	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40218	unsigned short	logging_register_address_10	Modbus Address to do logging in position 10	40000	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40219	unsigned short	logging_register_address_11	Modbus Address to do logging in position 11	40000	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40220	unsigned short	logging_register_address_12	Modbus Address to do logging in position 12	40000	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40221	unsigned short	logging_register_address_13	Modbus Address to do logging in position 13	40000	ALWAYS	ALWAYS	40000	49653

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
					READABLE	WRITABLE		
40222	unsigned short	logging_register_address_14	Modbus Address to do logging in position 14	40000	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40223	unsigned short	logging_register_address_15	Modbus Address to do logging in position 15	40000	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40224	unsigned short	logging_register_address_16	Modbus Address to do logging in position 16	40000	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40225	unsigned short	logging_register_address_17	Modbus Address to do logging in position 17	40000	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40226	unsigned short	logging_register_address_18	Modbus Address to do logging in position 18	40000	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40227	unsigned short	logging_register_address_19	Modbus Address to do logging in position 19	40000	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40228	unsigned short	logging_register_address_20	Modbus Address to do logging in position 20	40000	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40229	unsigned short	logging_period_s_flash	logging period in seconds for internal flash	5	ALWAYS READABLE	ALWAYS WRITABLE	5	65535
40230	unsigned short	mqtt_logging_register_address_1	Modbus Address used to log data with MQTT 1	40737	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40231	unsigned short	mqtt_logging_register_address_2	Modbus Address used to log data with MQTT 2	40741	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40232	unsigned short	mqtt_logging_register_address_3	Modbus Address used to log data with MQTT 3	40745	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40233	unsigned short	mqtt_logging_register_address_4	Modbus Address used to log data with MQTT 4	40753	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40234	unsigned short	mqtt_logging_register_address_5	Modbus Address used to log data with MQTT 5	40757	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40235	unsigned short	mqtt_logging_register_address_6	Modbus Address used to log data with MQTT 6	40761	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40236	unsigned short	mqtt_logging_register_address_7	Modbus Address used to log data with MQTT 7	40907	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40237	unsigned short	mqtt_logging_register_address_8	Modbus Address used to log data with MQTT 8	40909	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40238	unsigned short	mqtt_logging_register_address_9	Modbus Address used to log data with MQTT 9	40911	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40239	unsigned short	mqtt_logging_register_address_10	Modbus Address used to log data with MQTT 10	40923	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40240	unsigned short	mqtt_logging_register_address_11	Modbus Address used to log data with MQTT 11	40925	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40241	unsigned short	mqtt_logging_register_address_12	Modbus Address used to log data with MQTT 12	40927	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40242	unsigned short	mqtt_logging_register_address_13	Modbus Address used to log data with MQTT 13	40933	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40243	unsigned short	mqtt_logging_register_address_14	Modbus Address used to log data with MQTT 14	40935	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40244	unsigned short	mqtt_logging_register_address_15	Modbus Address used to log data with MQTT 15	40937	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40245	unsigned short	mqtt_logging_register_address_16	Modbus Address used to log data with MQTT 16	40941	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40246	unsigned short	mqtt_logging_register_address_17	Modbus Address used to log data with MQTT 17	40943	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40247	unsigned short	mqtt_logging_register_address_18	Modbus Address used to log data with MQTT 18	40945	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40248	unsigned short	mqtt_logging_register_address_19	Modbus Address used to log data with MQTT 19	40973	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40249	unsigned short	mqtt_logging_register_address_20	Modbus Address used to log data with MQTT 20	40000	ALWAYS READABLE	ALWAYS WRITABLE	40000	49653
40250	unsigned short	mqtt_logging_period_s	mqtt publish period	20	ALWAYS READABLE	ALWAYS WRITABLE	20	65535
40252	char	webserver_password_0	webserver authentication password char 0	a	NEVER READABLE	ALWAYS WRITABLE	-128	255
40253	char	webserver_password_1	webserver authentication password char 1	d	NEVER READABLE	ALWAYS WRITABLE	-128	255
40254	char	webserver_password_2	webserver authentication password char 2	m	NEVER READABLE	ALWAYS WRITABLE	-128	255
40255	char	webserver_password_3	webserver authentication password char 3	i	NEVER READABLE	ALWAYS WRITABLE	-128	255
40256	char	webserver_password_4	webserver authentication password char 4	n	NEVER READABLE	ALWAYS WRITABLE	-128	255
40257	char	webserver_password_5	webserver authentication password char 5	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40258	char	webserver_password_6	webserver authentication password char 6	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40259	char	webserver_password_7	webserver authentication password char 7	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40260	char	webserver_password_8	webserver authentication password char 8	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40261	char	webserver_password_9	webserver authentication password char 9	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40262	char	webserver_password_10	webserver authentication password char 10	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40263	char	webserver_password_11	webserver authentication password char 11	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40264	char	webserver_password_12	webserver authentication password char 12	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40265	char	webserver_password_13	webserver authentication password char 13	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40266	char	webserver_password_14	webserver authentication password char 14	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40267	char	webserver_password_15	webserver authentication password char 15	\0	NEVER	ALWAYS	-128	255

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40268	char	webserver_password_default_0	webserver authentication password char 0	a	NEVER READABLE	ALWAYS WRITABLE	-128	255
40269	char	webserver_password_default_1	webserver authentication password char 1	d	NEVER READABLE	ALWAYS WRITABLE	-128	255
40270	char	webserver_password_default_2	webserver authentication password char 2	m	NEVER READABLE	ALWAYS WRITABLE	-128	255
40271	char	webserver_password_default_3	webserver authentication password char 3	i	NEVER READABLE	ALWAYS WRITABLE	-128	255
40272	char	webserver_password_default_4	webserver authentication password char 4	n	NEVER READABLE	ALWAYS WRITABLE	-128	255
40273	char	webserver_password_default_5	webserver authentication password char 5	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40274	char	webserver_password_default_6	webserver authentication password char 6	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40275	char	webserver_password_default_7	webserver authentication password char 7	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40276	char	webserver_password_default_8	webserver authentication password char 8	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40277	char	webserver_password_default_9	webserver authentication password char 9	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40278	char	webserver_password_default_10	webserver authentication password char 10	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40279	char	webserver_password_default_11	webserver authentication password char 11	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40280	char	webserver_password_default_12	webserver authentication password char 12	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40281	char	webserver_password_default_13	webserver authentication password char 13	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40282	char	webserver_password_default_14	webserver authentication password char 14	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40283	char	webserver_password_default_15	webserver authentication password char 15	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40284	char	webserver_ins_password_0	webserver authentication password char 0	a	NEVER READABLE	ALWAYS WRITABLE	-128	255
40285	char	webserver_ins_password_1	webserver authentication password char 1	d	NEVER READABLE	ALWAYS WRITABLE	-128	255
40286	char	webserver_ins_password_2	webserver authentication password char 2	m	NEVER READABLE	ALWAYS WRITABLE	-128	255
40287	char	webserver_ins_password_3	webserver authentication password char 3	i	NEVER READABLE	ALWAYS WRITABLE	-128	255
40288	char	webserver_ins_password_4	webserver authentication password char 4	n	NEVER READABLE	ALWAYS WRITABLE	-128	255
40289	char	webserver_ins_password_5	webserver authentication password char 5	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40290	char	webserver_ins_password_6	webserver authentication password char 6	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40291	char	webserver_ins_password_7	webserver authentication password char 7	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40292	char	webserver_ins_password_8	webserver authentication password char 8	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40293	char	webserver_ins_password_9	webserver authentication password char 9	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40294	char	webserver_ins_password_10	webserver authentication password char 10	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40295	char	webserver_ins_password_11	webserver authentication password char 11	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40296	char	webserver_ins_password_12	webserver authentication password char 12	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40297	char	webserver_ins_password_13	webserver authentication password char 13	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40298	char	webserver_ins_password_14	webserver authentication password char 14	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40299	char	webserver_ins_password_15	webserver authentication password char 15	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40300	char	webserver_ins_password_default_0	webserver authentication password char 0	a	NEVER READABLE	ALWAYS WRITABLE	-128	255
40301	char	webserver_ins_password_default_1	webserver authentication password char 1	d	NEVER READABLE	ALWAYS WRITABLE	-128	255
40302	char	webserver_ins_password_default_2	webserver authentication password char 2	m	NEVER READABLE	ALWAYS WRITABLE	-128	255
40303	char	webserver_ins_password_default_3	webserver authentication password char 3	i	NEVER READABLE	ALWAYS WRITABLE	-128	255
40304	char	webserver_ins_password_default_4	webserver authentication password char 4	n	NEVER READABLE	ALWAYS WRITABLE	-128	255
40305	char	webserver_ins_password_default_5	webserver authentication password char 5	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40306	char	webserver_ins_password_default_6	webserver authentication password char 6	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40307	char	webserver_ins_password_default_7	webserver authentication password char 7	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40308	char	webserver_ins_password_default_8	webserver authentication password char 8	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40309	char	webserver_ins_password_default_9	webserver authentication password char 9	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40310	char	webserver_ins_password_default_10	webserver authentication password char 10	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40311	char	webserver_ins_password_default_11	webserver authentication password char 11	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40312	char	webserver_ins_password_default_12	webserver authentication password char 12	\0	NEVER	ALWAYS	-128	255

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40313	char	webserver_ins_password_default_13	webserver authentication password char 13	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40314	char	webserver_ins_password_default_14	webserver authentication password char 14	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40315	char	webserver_ins_password_default_15	webserver authentication password char 15	\0	NEVER READABLE	ALWAYS WRITABLE	-128	255
40325	signed short	wifi_tx_power	Wifi Strength of our antenna in dBm	20	ALWAYS READABLE	ALWAYS WRITABLE	-32768	32767
40326	unsigned short	periodo_polling	polling period in tenths of a second	10	ALWAYS READABLE	ALWAYS WRITABLE	10	65535
40327	unsigned short	delay_between_polls	delay before a new request in ms	10	ALWAYS READABLE	ALWAYS WRITABLE	0	65535
40328	unsigned short	max_n_record	maximum number of records on USB or µSD	10000	ALWAYS READABLE	ALWAYS WRITABLE	1	65535
40329	unsigned short	conf_logging	#define LOG_SD_MASK 0x0003 #define LOG_SD_OFF 0x0000 #define LOG_SD_ON 0x0001 #define LOG_SD_PROGRAMMATO 0x0002 #define LOG_SD_MQTT 0x0003 #define LOG_USB_MASK 0x000C #define LOG_USB_OFF 0x0000 #define LOG_USB_ON 0x0004 #define LOG_USB_PROGRAMMATO 0x0008 #define LOG_USB_MQTT 0x000C #define LOG_MQTT_MASK 0x0030 #define LOG_MQTT_OFF 0x0000 #define LOG_MQTT_ON 0x0010 #define LOG_MQTT_PROGRAMMATO 0x0020	0	ALWAYS READABLE	ALWAYS WRITABLE	0	65535
40330	unsigned short	year_month_log	start log year and month	65535	ALWAYS READABLE	ALWAYS WRITABLE	0	65535
40331	unsigned short	day_Hour_log	start log day and hour	65280	ALWAYS READABLE	ALWAYS WRITABLE	0	65535
40332	unsigned short	minute_second_log	start log minute and second	0	ALWAYS READABLE	ALWAYS WRITABLE	0	65535
40339	unsigned short	Ethernet_IP_assigned_by_dhcp_0	ETHERNET IP ASSIGNED (DHCP on) 0	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40340	unsigned short	Ethernet_IP_assigned_by_dhcp_1	ETHERNET IP ASSIGNED (DHCP on) 1	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40341	unsigned short	Ethernet_IP_assigned_by_dhcp_2	ETHERNET IP ASSIGNED (DHCP on) 2	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40342	unsigned short	Ethernet_IP_assigned_by_dhcp_3	ETHERNET IP ASSIGNED (DHCP on) 3	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40343	unsigned short	Ethernet_Netmask_assigned_by_dhcp_0	ETHERNET NETMASK ASSIGNED (DHCP on) 0	0	ALWAYS READABLE	NEVER WRITABLE	0	65535

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40344	unsigned short	Ethernet_Netmask_assigned_by_dhcp_1	ETHERNET NETMASK ASSIGNED (DHCP on) 1	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40345	unsigned short	Ethernet_Netmask_assigned_by_dhcp_2	ETHERNET NETMASK ASSIGNED (DHCP on) 2	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40346	unsigned short	Ethernet_Netmask_assigned_by_dhcp_3	ETHERNET NETMASK ASSIGNED (DHCP on) 3	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40347	unsigned short	Ethernet_Gateway_assigned_by_dhcp_0	ETHERNET GATEWAY ASSIGNED (DHCP on) 0	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40348	unsigned short	Ethernet_Gateway_assigned_by_dhcp_1	ETHERNET GATEWAY ASSIGNED (DHCP on) 1	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40349	unsigned short	Ethernet_Gateway_assigned_by_dhcp_2	ETHERNET GATEWAY ASSIGNED (DHCP on) 2	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40350	unsigned short	Ethernet_Gateway_assigned_by_dhcp_3	ETHERNET GATEWAY ASSIGNED (DHCP on) 3	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40353	unsigned long	Status_1	#define FLASH_SETTINGS_ERROR 0x00000001 #define FLASH_CALIBRATION_ERROR 0x00000002 #define FLASH_JSON_ERROR 0x00000004 #define BUZZER_ON 0x00000008 #define RTC_CLOCK_NOT_UPDATED 0x00000010	0	ALWAYS READABLE	NEVER WRITABLE	0	4294967295
40355	unsigned long	Status_2	#define NTP_SET 0x00002000 #define TOUCH_BOTTOM_PRESSED 0x00040000 #define ERROR_TEMP_SENSOR 0x00080000 #define ERROR_TOUCH_SENSOR 0x00100000 #define REQUEST_RESET_SETTINGS 0x00200000 #define RTC_WRITING_IN_PROGRESS 0x00400000 #define REQUEST_RSSI 0x00800000	0	ALWAYS READABLE	NEVER WRITABLE	0	4294967295
40359	unsigned long	Communication_status	#define WIFI_STA_CONNECTED 0x0001 #define ETHERNET_CONNECTED 0x0002 #define CLOUD_CONNECTED 0x0004 #define SD_CARD_MOUNTED 0x0008 #define USB_MOUNTED 0x0010 #define MQTT_CONNECTED 0x0020 #define SD_CARD_INSERTED 0x0040 #define USB_INSERTED 0x0080	0	ALWAYS READABLE	NEVER WRITABLE	0	4294967295
40361	unsigned short	socket_TCP	number of TCP sockets	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40362	unsigned short	socket_HTTP	number of HTTP sockets	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40363	unsigned short	socket_AP	number of AP sockets	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40364	signed short	temperatura_int	internal temperature in tenths of a degree	0	ALWAYS READABLE	NEVER WRITABLE	-32768	32767
40365	unsigned long	soglia_touch		0	ALWAYS READABLE	NEVER WRITABLE	0	4294967295
40367	unsigned short	Wifi_STA_IP_assigned_by_dhcp_0	WI-FI STA IP SET (DHCP on) 0	0	ALWAYS	NEVER	0	65535

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40368	unsigned short	Wifi_STA_IP_assigned_by_dhcp_1	WI-FI STA IP SET (DHCP on) 1	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40369	unsigned short	Wifi_STA_IP_assigned_by_dhcp_2	WI-FI STA IP SET (DHCP on) 2	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40370	unsigned short	Wifi_STA_IP_assigned_by_dhcp_3	WI-FI STA IP SET (DHCP on) 3	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40371	unsigned short	Wifi_STA_Netmask_assigned_by_dhcp_0	WI-FI STA NETMASK SET (DHCP on) 0	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40372	unsigned short	Wifi_STA_Netmask_assigned_by_dhcp_1	WI-FI STA NETMASK SET (DHCP on) 1	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40373	unsigned short	Wifi_STA_Netmask_assigned_by_dhcp_2	WI-FI STA NETMASK SET (DHCP on) 2	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40374	unsigned short	Wifi_STA_Netmask_assigned_by_dhcp_3	WI-FI STA NETMASK SET (DHCP on) 3	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40375	unsigned short	Wifi_STA_Gateway_assigned_by_dhcp_0	WI-FI STA GATEWAY SET (DHCP on) 0	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40376	unsigned short	Wifi_STA_Gateway_assigned_by_dhcp_1	WI-FI STA GATEWAY SET (DHCP on) 1	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40377	unsigned short	Wifi_STA_Gateway_assigned_by_dhcp_2	WI-FI STA GATEWAY SET (DHCP on) 2	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40378	unsigned short	Wifi_STA_Gateway_assigned_by_dhcp_3	WI-FI STA GATEWAY SET (DHCP on) 3	0	ALWAYS READABLE	NEVER WRITABLE	0	65535
40379	signed short	RSSI	Wifi Strength (of the connected AP)	0	ALWAYS READABLE	NEVER WRITABLE	-32768	32767
40383	unsigned short	Command	#define COMANDO_SALVATAGGIO_FLASH_TARATURE 0xC1B0U #define COMANDO_SALVATAGGIO_FLASH_SOLO_SETTAGGI 0xF1C0U #define COMANDO_SALVATAGGIO_FLASH_SETTAGGI_E_NFC 0xC1C0U #define COMANDO_RESET_SETTINGS_DEFAULT 0XD1C0U #define COMANDO_LEGGI_DIP 0xD166U #define COMANDO_RESET 0xC1A0U #define COMANDO_BOOTLOADER 0xB000U #define COMANDO_BUZZER_ON 0xDABAU #define COMANDO_BUZZER_OFF 0xDABBU #define COMANDO_SET_RTC 0xB123U #define COMANDO_RESET_SOLO_CREDENZIALI 0xC3EDU #define COMANDO_TARA_TASTO 0xCA22U	0	ALWAYS READABLE	ALWAYS WRITABLE	0	65535

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40384	unsigned short	req_id1	ID of the first request	1	ALWAYS READABLE	NEVER WRITABLE	1	65535
40385	unsigned short	req_type1	request type	0	ALWAYS READABLE	NEVER WRITABLE	0	3
40386	unsigned short	req_ip1_0	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40387	unsigned short	req_ip1_1	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40388	unsigned short	req_ip1_2	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40389	unsigned short	req_ip1_3	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40390	unsigned short	req_addr1	Address of the first register to read	40001	ALWAYS READABLE	NEVER WRITABLE	40001	65535
40391	unsigned short	req_qty1	number of registers to read	1	ALWAYS READABLE	NEVER WRITABLE	1	10
40392	unsigned short	req_id2	ID of the first request	1	ALWAYS READABLE	NEVER WRITABLE	1	65535
40393	unsigned short	req_type2	request type	0	ALWAYS READABLE	NEVER WRITABLE	0	3
40394	unsigned short	req_ip2_0	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40395	unsigned short	req_ip2_1	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40396	unsigned short	req_ip2_2	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40397	unsigned short	req_ip2_3	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40398	unsigned short	req_addr2	Address of the first register to read	40001	ALWAYS READABLE	NEVER WRITABLE	40001	65535
40399	unsigned short	req_qty2	number of registers to read	1	ALWAYS READABLE	NEVER WRITABLE	1	10
40400	unsigned short	req_id3	ID of the first request	1	ALWAYS READABLE	NEVER WRITABLE	1	65535
40401	unsigned short	req_type3	request type	0	ALWAYS READABLE	NEVER WRITABLE	0	3
40402	unsigned short	req_ip3_0	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40403	unsigned short	req_ip3_1	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40404	unsigned short	req_ip3_2	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40405	unsigned short	req_ip3_3	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40406	unsigned short	req_addr3	Address of the first register to read	40001	ALWAYS	NEVER	40001	65535

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40407	unsigned short	req_qty3	number of registers to read	1	READABLE ALWAYS	WRITABLE NEVER	1	10
40408	unsigned short	req_id4	ID of the first request	1	READABLE ALWAYS	WRITABLE NEVER	1	65535
40409	unsigned short	req_type4	request type	0	READABLE ALWAYS	WRITABLE NEVER	0	3
40410	unsigned short	req_ip4_0	IP	0	READABLE ALWAYS	WRITABLE NEVER	0	255
40411	unsigned short	req_ip4_1	IP	0	READABLE ALWAYS	WRITABLE NEVER	0	255
40412	unsigned short	req_ip4_2	IP	0	READABLE ALWAYS	WRITABLE NEVER	0	255
40413	unsigned short	req_ip4_3	IP	0	READABLE ALWAYS	WRITABLE NEVER	0	255
40414	unsigned short	req_addr4	Address of the first register to read	40001	READABLE ALWAYS	WRITABLE NEVER	40001	65535
40415	unsigned short	req_qty4	number of registers to read	1	READABLE ALWAYS	WRITABLE NEVER	1	10
40416	unsigned short	req_id5	ID of the first request	1	READABLE ALWAYS	WRITABLE NEVER	1	65535
40417	unsigned short	req_type5	request type	0	READABLE ALWAYS	WRITABLE NEVER	0	3
40418	unsigned short	req_ip5_0	IP	0	READABLE ALWAYS	WRITABLE NEVER	0	255
40419	unsigned short	req_ip5_1	IP	0	READABLE ALWAYS	WRITABLE NEVER	0	255
40420	unsigned short	req_ip5_2	IP	0	READABLE ALWAYS	WRITABLE NEVER	0	255
40421	unsigned short	req_ip5_3	IP	0	READABLE ALWAYS	WRITABLE NEVER	0	255
40422	unsigned short	req_addr5	Address of the first register to read	40001	READABLE ALWAYS	WRITABLE NEVER	40001	65535
40423	unsigned short	req_qty5	number of registers to read	1	READABLE ALWAYS	WRITABLE NEVER	1	10
40424	unsigned short	req_id6	ID of the first request	1	READABLE ALWAYS	WRITABLE NEVER	1	65535
40425	unsigned short	req_type6	request type	0	READABLE ALWAYS	WRITABLE NEVER	0	3
40426	unsigned short	req_ip6_0	IP	0	READABLE ALWAYS	WRITABLE NEVER	0	255
40427	unsigned short	req_ip6_1	IP	0	READABLE ALWAYS	WRITABLE NEVER	0	255
40428	unsigned short	req_ip6_2	IP	0	READABLE ALWAYS	WRITABLE NEVER	0	255

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40429	unsigned short	req_ip6_3	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40430	unsigned short	req_addr6	Address of the first register to read	40001	ALWAYS READABLE	NEVER WRITABLE	40001	65535
40431	unsigned short	req_qty6	number of registers to read	1	ALWAYS READABLE	NEVER WRITABLE	1	10
40432	unsigned short	req_id7	ID of the first request	1	ALWAYS READABLE	NEVER WRITABLE	1	65535
40433	unsigned short	req_type7	request type	0	ALWAYS READABLE	NEVER WRITABLE	0	3
40434	unsigned short	req_ip7_0	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40435	unsigned short	req_ip7_1	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40436	unsigned short	req_ip7_2	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40437	unsigned short	req_ip7_3	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40438	unsigned short	req_addr7	Address of the first register to read	40001	ALWAYS READABLE	NEVER WRITABLE	40001	65535
40439	unsigned short	req_qty7	number of registers to read	1	ALWAYS READABLE	NEVER WRITABLE	1	10
40440	unsigned short	req_id8	ID of the first request	1	ALWAYS READABLE	NEVER WRITABLE	1	65535
40441	unsigned short	req_type8	request type	0	ALWAYS READABLE	NEVER WRITABLE	0	3
40442	unsigned short	req_ip8_0	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40443	unsigned short	req_ip8_1	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40444	unsigned short	req_ip8_2	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40445	unsigned short	req_ip8_3	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40446	unsigned short	req_addr8	Address of the first register to read	40001	ALWAYS READABLE	NEVER WRITABLE	40001	65535
40447	unsigned short	req_qty8	number of registers to read	1	ALWAYS READABLE	NEVER WRITABLE	1	10
40448	unsigned short	req_id9	ID of the first request	1	ALWAYS READABLE	NEVER WRITABLE	1	65535
40449	unsigned short	req_type9	request type	0	ALWAYS READABLE	NEVER WRITABLE	0	3
40450	unsigned short	req_ip9_0	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40451	unsigned short	req_ip9_1	IP	0	ALWAYS	NEVER	0	255

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40452	unsigned short	req_ip9_2	IP	0	READABLE ALWAYS	WRITABLE NEVER	0	255
40453	unsigned short	req_ip9_3	IP	0	READABLE ALWAYS	WRITABLE NEVER	0	255
40454	unsigned short	req_addr9	Address of the first register to read	40001	READABLE ALWAYS	WRITABLE NEVER	40001	65535
40455	unsigned short	req_qty9	number of registers to read	1	READABLE ALWAYS	WRITABLE NEVER	1	10
40456	unsigned short	req_id10	ID of the first request	1	READABLE ALWAYS	WRITABLE NEVER	1	65535
40457	unsigned short	req_type10	request type	0	READABLE ALWAYS	WRITABLE NEVER	0	3
40458	unsigned short	req_ip10_0	IP	0	READABLE ALWAYS	WRITABLE NEVER	0	255
40459	unsigned short	req_ip10_1	IP	0	READABLE ALWAYS	WRITABLE NEVER	0	255
40460	unsigned short	req_ip10_2	IP	0	READABLE ALWAYS	WRITABLE NEVER	0	255
40461	unsigned short	req_ip10_3	IP	0	READABLE ALWAYS	WRITABLE NEVER	0	255
40462	unsigned short	req_addr10	Address of the first register to read	40001	READABLE ALWAYS	WRITABLE NEVER	40001	65535
40463	unsigned short	req_qty10	number of registers to read	1	READABLE ALWAYS	WRITABLE NEVER	1	10
40464	unsigned short	req_id11	ID of the first request	1	READABLE ALWAYS	WRITABLE NEVER	1	65535
40465	unsigned short	req_type11	request type	0	READABLE ALWAYS	WRITABLE NEVER	0	3
40466	unsigned short	req_ip11_0	IP	0	READABLE ALWAYS	WRITABLE NEVER	0	255
40467	unsigned short	req_ip11_1	IP	0	READABLE ALWAYS	WRITABLE NEVER	0	255
40468	unsigned short	req_ip11_2	IP	0	READABLE ALWAYS	WRITABLE NEVER	0	255
40469	unsigned short	req_ip11_3	IP	0	READABLE ALWAYS	WRITABLE NEVER	0	255
40470	unsigned short	req_addr11	Address of the first register to read	40001	READABLE ALWAYS	WRITABLE NEVER	40001	65535
40471	unsigned short	req_qty11	number of registers to read	1	READABLE ALWAYS	WRITABLE NEVER	1	10
40472	unsigned short	req_id12	ID of the first request	1	READABLE ALWAYS	WRITABLE NEVER	1	65535
40473	unsigned short	req_type12	request type	0	READABLE ALWAYS	WRITABLE NEVER	0	3

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40474	unsigned short	req_ip12_0	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40475	unsigned short	req_ip12_1	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40476	unsigned short	req_ip12_2	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40477	unsigned short	req_ip12_3	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40478	unsigned short	req_addr12	Address of the first register to read	40001	ALWAYS READABLE	NEVER WRITABLE	40001	65535
40479	unsigned short	req_qty12	number of registers to read	1	ALWAYS READABLE	NEVER WRITABLE	1	10
40480	unsigned short	req_id13	ID of the first request	1	ALWAYS READABLE	NEVER WRITABLE	1	65535
40481	unsigned short	req_type13	request type	0	ALWAYS READABLE	NEVER WRITABLE	0	3
40482	unsigned short	req_ip13_0	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40483	unsigned short	req_ip13_1	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40484	unsigned short	req_ip13_2	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40485	unsigned short	req_ip13_3	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40486	unsigned short	req_addr13	Address of the first register to read	40001	ALWAYS READABLE	NEVER WRITABLE	40001	65535
40487	unsigned short	req_qty13	number of registers to read	1	ALWAYS READABLE	NEVER WRITABLE	1	10
40488	unsigned short	req_id14	ID of the first request	1	ALWAYS READABLE	NEVER WRITABLE	1	65535
40489	unsigned short	req_type14	request type	0	ALWAYS READABLE	NEVER WRITABLE	0	3
40490	unsigned short	req_ip14_0	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40491	unsigned short	req_ip14_1	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40492	unsigned short	req_ip14_2	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40493	unsigned short	req_ip14_3	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40494	unsigned short	req_addr14	Address of the first register to read	40001	ALWAYS READABLE	NEVER WRITABLE	40001	65535
40495	unsigned short	req_qty14	number of registers to read	1	ALWAYS READABLE	NEVER WRITABLE	1	10
40496	unsigned short	req_id15	ID of the first request	1	ALWAYS	NEVER	1	65535

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
					READABLE	WRITABLE		
40497	unsigned short	req_type15	request type	0	ALWAYS READABLE	NEVER WRITABLE	0	3
40498	unsigned short	req_ip15_0	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40499	unsigned short	req_ip15_1	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40500	unsigned short	req_ip15_2	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40501	unsigned short	req_ip15_3	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40502	unsigned short	req_addr15	Address of the first register to read	40001	ALWAYS READABLE	NEVER WRITABLE	40001	65535
40503	unsigned short	req_qty15	number of registers to read	1	ALWAYS READABLE	NEVER WRITABLE	1	10
40504	unsigned short	req_id16	ID of the first request	1	ALWAYS READABLE	NEVER WRITABLE	1	65535
40505	unsigned short	req_type16	request type	0	ALWAYS READABLE	NEVER WRITABLE	0	3
40506	unsigned short	req_ip16_0	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40507	unsigned short	req_ip16_1	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40508	unsigned short	req_ip16_2	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40509	unsigned short	req_ip16_3	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40510	unsigned short	req_addr16	Address of the first register to read	40001	ALWAYS READABLE	NEVER WRITABLE	40001	65535
40511	unsigned short	req_qty16	number of registers to read	1	ALWAYS READABLE	NEVER WRITABLE	1	10
40512	unsigned short	req_id17	ID of the first request	1	ALWAYS READABLE	NEVER WRITABLE	1	65535
40513	unsigned short	req_type17	request type	0	ALWAYS READABLE	NEVER WRITABLE	0	3
40514	unsigned short	req_ip17_0	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40515	unsigned short	req_ip17_1	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40516	unsigned short	req_ip17_2	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40517	unsigned short	req_ip17_3	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40518	unsigned short	req_addr17	Address of the first register to read	40001	ALWAYS READABLE	NEVER WRITABLE	40001	65535

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40519	unsigned short	req_qty17	number of registers to read	1	ALWAYS READABLE	NEVER WRITABLE	1	10
40520	unsigned short	req_id18	ID of the first request	1	ALWAYS READABLE	NEVER WRITABLE	1	65535
40521	unsigned short	req_type18	request type	0	ALWAYS READABLE	NEVER WRITABLE	0	3
40522	unsigned short	req_ip18_0	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40523	unsigned short	req_ip18_1	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40524	unsigned short	req_ip18_2	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40525	unsigned short	req_ip18_3	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40526	unsigned short	req_addr18	Address of the first register to read	40001	ALWAYS READABLE	NEVER WRITABLE	40001	65535
40527	unsigned short	req_qty18	number of registers to read	1	ALWAYS READABLE	NEVER WRITABLE	1	10
40528	unsigned short	req_id19	ID of the first request	1	ALWAYS READABLE	NEVER WRITABLE	1	65535
40529	unsigned short	req_type19	request type	0	ALWAYS READABLE	NEVER WRITABLE	0	3
40530	unsigned short	req_ip19_0	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40531	unsigned short	req_ip19_1	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40532	unsigned short	req_ip19_2	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40533	unsigned short	req_ip19_3	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40534	unsigned short	req_addr19	Address of the first register to read	40001	ALWAYS READABLE	NEVER WRITABLE	40001	65535
40535	unsigned short	req_qty19	number of registers to read	1	ALWAYS READABLE	NEVER WRITABLE	1	10
40536	unsigned short	req_id20	ID of the first request	1	ALWAYS READABLE	NEVER WRITABLE	1	65535
40537	unsigned short	req_type20	request type	0	ALWAYS READABLE	NEVER WRITABLE	0	3
40538	unsigned short	req_ip20_0	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40539	unsigned short	req_ip20_1	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40540	unsigned short	req_ip20_2	IP	0	ALWAYS READABLE	NEVER WRITABLE	0	255
40541	unsigned short	req_ip20_3	IP	0	ALWAYS	NEVER	0	255

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40542	unsigned short	req_addr20	Address of the first register to read	40001	READABLE ALWAYS	WRITABLE NEVER	40001	65535
40543	unsigned short	req_qty20	number of registers to read	1	READABLE ALWAYS	WRITABLE NEVER	1	10
40544	unsigned short	Year	RTC : year (2000-2099)	2025	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40545	unsigned short	Month	RTC : month (1-12)	10	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40546	unsigned short	Day	RTC : day month (1-31)	6	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40547	unsigned short	Hour	RTC : hour (0-23)	11	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40548	unsigned short	Minute	RTC : minute (0-59)	7	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40549	unsigned short	Seconds	RTC : second (0-59)	45	READABLE ALWAYS	WRITABLE ALWAYS	0	65535
40550	unsigned short	log_1	logged register 1	0	READABLE ALWAYS	WRITABLE NEVER	0	65535
40551	unsigned short	log_2	logged register 2	0	READABLE ALWAYS	WRITABLE NEVER	0	65535
40552	unsigned short	log_3	logged register 3	0	READABLE ALWAYS	WRITABLE NEVER	0	65535
40553	unsigned short	log_4	logged register 4	0	READABLE ALWAYS	WRITABLE NEVER	0	65535
40554	unsigned short	log_5	logged register 5	0	READABLE ALWAYS	WRITABLE NEVER	0	65535
40555	unsigned short	log_6	logged register 6	0	READABLE ALWAYS	WRITABLE NEVER	0	65535
40556	unsigned short	log_7	logged register 7	1	READABLE ALWAYS	WRITABLE NEVER	0	65535
40557	unsigned short	log_8	logged register 8	2	READABLE ALWAYS	WRITABLE NEVER	0	65535
40558	unsigned short	log_9	logged register 9	3	READABLE ALWAYS	WRITABLE NEVER	0	65535
40559	unsigned short	log_10	logged register 10	4	READABLE ALWAYS	WRITABLE NEVER	0	65535
40560	unsigned short	log_11	logged register 11	5	READABLE ALWAYS	WRITABLE NEVER	0	65535
40561	unsigned short	log_12	logged register 12	6	READABLE ALWAYS	WRITABLE NEVER	0	65535
40562	unsigned short	log_13	logged register 13	7	READABLE ALWAYS	WRITABLE NEVER	0	65535
40563	unsigned short	log_14	logged register 14	8	READABLE ALWAYS	WRITABLE NEVER	0	65535

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40564	unsigned short	log_15	logged register 15	9	ALWAYS READABLE	NEVER WRITABLE	0	65535
40565	unsigned short	log_16	logged register 16	10	ALWAYS READABLE	NEVER WRITABLE	0	65535
40566	unsigned short	log_17	logged register 17	11	ALWAYS READABLE	NEVER WRITABLE	0	65535
40567	unsigned short	log_18	logged register 18	12	ALWAYS READABLE	NEVER WRITABLE	0	65535
40568	unsigned short	log_19	logged register 19	13	ALWAYS READABLE	NEVER WRITABLE	0	65535
40569	unsigned short	log_20	logged register 20	14	ALWAYS READABLE	NEVER WRITABLE	0	65535
40570	unsigned short	log_21	logged register 21	15	ALWAYS READABLE	NEVER WRITABLE	0	65535
40571	unsigned short	log_22	logged register 22	16	ALWAYS READABLE	NEVER WRITABLE	0	65535
40572	unsigned short	log_23	logged register 23	17	ALWAYS READABLE	NEVER WRITABLE	0	65535
40573	unsigned short	log_24	logged register 24	18	ALWAYS READABLE	NEVER WRITABLE	0	65535
40574	unsigned short	log_25	logged register 25	19	ALWAYS READABLE	NEVER WRITABLE	0	65535
40575	unsigned short	log_26	logged register 26	20	ALWAYS READABLE	NEVER WRITABLE	0	65535
40576	unsigned short	log_27	logged register 27	21	ALWAYS READABLE	NEVER WRITABLE	0	65535
40577	unsigned short	log_28	logged register 28	22	ALWAYS READABLE	NEVER WRITABLE	0	65535
40578	unsigned short	log_29	logged register 29	23	ALWAYS READABLE	NEVER WRITABLE	0	65535
40579	unsigned short	log_30	logged register 30	24	ALWAYS READABLE	NEVER WRITABLE	0	65535
40580	unsigned short	log_31	logged register 31	25	ALWAYS READABLE	NEVER WRITABLE	0	65535
40581	unsigned short	log_32	logged register 32	26	ALWAYS READABLE	NEVER WRITABLE	0	65535
40582	unsigned short	log_33	logged register 33	27	ALWAYS READABLE	NEVER WRITABLE	0	65535
40583	unsigned short	log_34	logged register 34	28	ALWAYS READABLE	NEVER WRITABLE	0	65535
40584	unsigned short	log_35	logged register 35	29	ALWAYS READABLE	NEVER WRITABLE	0	65535
40585	unsigned short	log_36	logged register 36	30	ALWAYS READABLE	NEVER WRITABLE	0	65535
40586	unsigned short	log_37	logged register 37	31	ALWAYS	NEVER	0	65535

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40587	unsigned short	log_38	logged register 38	32	READABLE	WRITABLE	0	65535
40588	unsigned short	log_39	logged register 39	33	ALWAYS READABLE	NEVER WRITABLE	0	65535
40589	unsigned short	log_40	logged register 40	34	ALWAYS READABLE	NEVER WRITABLE	0	65535
40590	unsigned short	log_41	logged register 41	35	ALWAYS READABLE	NEVER WRITABLE	0	65535
40591	unsigned short	log_42	logged register 42	36	ALWAYS READABLE	NEVER WRITABLE	0	65535
40592	unsigned short	log_43	logged register 43	37	ALWAYS READABLE	NEVER WRITABLE	0	65535
40593	unsigned short	log_44	logged register 44	38	ALWAYS READABLE	NEVER WRITABLE	0	65535
40594	unsigned short	log_45	logged register 45	39	ALWAYS READABLE	NEVER WRITABLE	0	65535
40595	unsigned short	log_46	logged register 46	40	ALWAYS READABLE	NEVER WRITABLE	0	65535
40596	unsigned short	log_47	logged register 47	41	ALWAYS READABLE	NEVER WRITABLE	0	65535
40597	unsigned short	log_48	logged register 48	42	ALWAYS READABLE	NEVER WRITABLE	0	65535
40598	unsigned short	log_49	logged register 49	43	ALWAYS READABLE	NEVER WRITABLE	0	65535
40599	unsigned short	log_50	logged register 50	44	ALWAYS READABLE	NEVER WRITABLE	0	65535
40600	unsigned short	log_51	logged register 51	45	ALWAYS READABLE	NEVER WRITABLE	0	65535
40601	unsigned short	log_52	logged register 52	46	ALWAYS READABLE	NEVER WRITABLE	0	65535
40602	unsigned short	log_53	logged register 53	47	ALWAYS READABLE	NEVER WRITABLE	0	65535
40603	unsigned short	log_54	logged register 54	48	ALWAYS READABLE	NEVER WRITABLE	0	65535
40604	unsigned short	log_55	logged register 55	49	ALWAYS READABLE	NEVER WRITABLE	0	65535
40605	unsigned short	log_56	logged register 56	50	ALWAYS READABLE	NEVER WRITABLE	0	65535
40606	unsigned short	log_57	logged register 57	51	ALWAYS READABLE	NEVER WRITABLE	0	65535
40607	unsigned short	log_58	logged register 58	52	ALWAYS READABLE	NEVER WRITABLE	0	65535
40608	unsigned short	log_59	logged register 59	53	ALWAYS READABLE	NEVER WRITABLE	0	65535

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40609	unsigned short	log_60	logged register 60	54	ALWAYS READABLE	NEVER WRITABLE	0	65535
40610	unsigned short	log_61	logged register 61	55	ALWAYS READABLE	NEVER WRITABLE	0	65535
40611	unsigned short	log_62	logged register 62	56	ALWAYS READABLE	NEVER WRITABLE	0	65535
40612	unsigned short	log_63	logged register 63	57	ALWAYS READABLE	NEVER WRITABLE	0	65535
40613	unsigned short	log_64	logged register 64	58	ALWAYS READABLE	NEVER WRITABLE	0	65535
40614	unsigned short	log_65	logged register 65	59	ALWAYS READABLE	NEVER WRITABLE	0	65535
40615	unsigned short	log_66	logged register 66	60	ALWAYS READABLE	NEVER WRITABLE	0	65535
40616	unsigned short	log_67	logged register 67	61	ALWAYS READABLE	NEVER WRITABLE	0	65535
40617	unsigned short	log_68	logged register 68	62	ALWAYS READABLE	NEVER WRITABLE	0	65535
40618	unsigned short	log_69	logged register 69	63	ALWAYS READABLE	NEVER WRITABLE	0	65535
40619	unsigned short	log_70	logged register 70	64	ALWAYS READABLE	NEVER WRITABLE	0	65535
40620	unsigned short	log_71	logged register 71	65	ALWAYS READABLE	NEVER WRITABLE	0	65535
40621	unsigned short	log_72	logged register 72	66	ALWAYS READABLE	NEVER WRITABLE	0	65535
40622	unsigned short	log_73	logged register 73	67	ALWAYS READABLE	NEVER WRITABLE	0	65535
40623	unsigned short	log_74	logged register 74	68	ALWAYS READABLE	NEVER WRITABLE	0	65535
40624	unsigned short	log_75	logged register 75	69	ALWAYS READABLE	NEVER WRITABLE	0	65535
40625	unsigned short	log_76	logged register 76	70	ALWAYS READABLE	NEVER WRITABLE	0	65535
40626	unsigned short	log_77	logged register 77	71	ALWAYS READABLE	NEVER WRITABLE	0	65535
40627	unsigned short	log_78	logged register 78	72	ALWAYS READABLE	NEVER WRITABLE	0	65535
40628	unsigned short	log_79	logged register 79	73	ALWAYS READABLE	NEVER WRITABLE	0	65535
40629	unsigned short	log_80	logged register 80	74	ALWAYS READABLE	NEVER WRITABLE	0	65535
40630	unsigned short	log_81	logged register 81	75	ALWAYS READABLE	NEVER WRITABLE	0	65535
40631	unsigned short	log_82	logged register 82	76	ALWAYS	NEVER	0	65535

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40632	unsigned short	log_83	logged register 83	77	READABLE	WRITABLE	0	65535
40633	unsigned short	log_84	logged register 84	78	ALWAYS READABLE	NEVER WRITABLE	0	65535
40634	unsigned short	log_85	logged register 85	79	ALWAYS READABLE	NEVER WRITABLE	0	65535
40635	unsigned short	log_86	logged register 86	80	ALWAYS READABLE	NEVER WRITABLE	0	65535
40636	unsigned short	log_87	logged register 87	81	ALWAYS READABLE	NEVER WRITABLE	0	65535
40637	unsigned short	log_88	logged register 88	82	ALWAYS READABLE	NEVER WRITABLE	0	65535
40638	unsigned short	log_89	logged register 89	83	ALWAYS READABLE	NEVER WRITABLE	0	65535
40639	unsigned short	log_90	logged register 90	84	ALWAYS READABLE	NEVER WRITABLE	0	65535
40640	unsigned short	log_91	logged register 91	85	ALWAYS READABLE	NEVER WRITABLE	0	65535
40641	unsigned short	log_92	logged register 92	86	ALWAYS READABLE	NEVER WRITABLE	0	65535
40642	unsigned short	log_93	logged register 93	87	ALWAYS READABLE	NEVER WRITABLE	0	65535
40643	unsigned short	log_94	logged register 94	88	ALWAYS READABLE	NEVER WRITABLE	0	65535
40644	unsigned short	log_95	logged register 95	89	ALWAYS READABLE	NEVER WRITABLE	0	65535
40645	unsigned short	log_96	logged register 96	90	ALWAYS READABLE	NEVER WRITABLE	0	65535
40646	unsigned short	log_97	logged register 97	91	ALWAYS READABLE	NEVER WRITABLE	0	65535
40647	unsigned short	log_98	logged register 98	92	ALWAYS READABLE	NEVER WRITABLE	0	65535
40648	unsigned short	log_99	logged register 99	93	ALWAYS READABLE	NEVER WRITABLE	0	65535
40649	unsigned short	log_100	logged register 100	94	ALWAYS READABLE	NEVER WRITABLE	0	65535
40650	unsigned short	log_101	logged register 101	95	ALWAYS READABLE	NEVER WRITABLE	0	65535
40651	unsigned short	log_102	logged register 102	96	ALWAYS READABLE	NEVER WRITABLE	0	65535
40652	unsigned short	log_103	logged register 103	97	ALWAYS READABLE	NEVER WRITABLE	0	65535
40653	unsigned short	log_104	logged register 104	98	ALWAYS READABLE	NEVER WRITABLE	0	65535

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40654	unsigned short	log_105	logged register 105	99	ALWAYS READABLE	NEVER WRITABLE	0	65535
40655	unsigned short	log_106	logged register 106	100	ALWAYS READABLE	NEVER WRITABLE	0	65535
40656	unsigned short	log_107	logged register 107	101	ALWAYS READABLE	NEVER WRITABLE	0	65535
40657	unsigned short	log_108	logged register 108	102	ALWAYS READABLE	NEVER WRITABLE	0	65535
40658	unsigned short	log_109	logged register 109	103	ALWAYS READABLE	NEVER WRITABLE	0	65535
40659	unsigned short	log_110	logged register 110	104	ALWAYS READABLE	NEVER WRITABLE	0	65535
40660	unsigned short	log_111	logged register 111	105	ALWAYS READABLE	NEVER WRITABLE	0	65535
40661	unsigned short	log_112	logged register 112	106	ALWAYS READABLE	NEVER WRITABLE	0	65535
40662	unsigned short	log_113	logged register 113	107	ALWAYS READABLE	NEVER WRITABLE	0	65535
40663	unsigned short	log_114	logged register 114	108	ALWAYS READABLE	NEVER WRITABLE	0	65535
40664	unsigned short	log_115	logged register 115	109	ALWAYS READABLE	NEVER WRITABLE	0	65535
40665	unsigned short	log_116	logged register 116	110	ALWAYS READABLE	NEVER WRITABLE	0	65535
40666	unsigned short	log_117	logged register 117	111	ALWAYS READABLE	NEVER WRITABLE	0	65535
40667	unsigned short	log_118	logged register 118	112	ALWAYS READABLE	NEVER WRITABLE	0	65535
40668	unsigned short	log_119	logged register 119	113	ALWAYS READABLE	NEVER WRITABLE	0	65535
40669	unsigned short	log_120	logged register 120	114	ALWAYS READABLE	NEVER WRITABLE	0	65535
40670	unsigned short	log_121	logged register 121	115	ALWAYS READABLE	NEVER WRITABLE	0	65535
40671	unsigned short	log_122	logged register 122	116	ALWAYS READABLE	NEVER WRITABLE	0	65535
40672	unsigned short	log_123	logged register 123	117	ALWAYS READABLE	NEVER WRITABLE	0	65535
40673	unsigned short	log_124	logged register 124	118	ALWAYS READABLE	NEVER WRITABLE	0	65535
40674	unsigned short	log_125	logged register 125	119	ALWAYS READABLE	NEVER WRITABLE	0	65535
40675	unsigned short	log_126	logged register 126	120	ALWAYS READABLE	NEVER WRITABLE	0	65535
40676	unsigned short	log_127	logged register 127	121	ALWAYS	NEVER	0	65535

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40677	unsigned short	log_128	logged register 128	122	READABLE ALWAYS	WRITABLE NEVER	0	65535
40678	unsigned short	log_129	logged register 129	123	READABLE ALWAYS	WRITABLE NEVER	0	65535
40679	unsigned short	log_130	logged register 130	124	READABLE ALWAYS	WRITABLE NEVER	0	65535
40680	unsigned short	log_131	logged register 131	125	READABLE ALWAYS	WRITABLE NEVER	0	65535
40681	unsigned short	log_132	logged register 132	126	READABLE ALWAYS	WRITABLE NEVER	0	65535
40682	unsigned short	log_133	logged register 133	127	READABLE ALWAYS	WRITABLE NEVER	0	65535
40683	unsigned short	log_134	logged register 134	128	READABLE ALWAYS	WRITABLE NEVER	0	65535
40684	unsigned short	log_135	logged register 135	129	READABLE ALWAYS	WRITABLE NEVER	0	65535
40685	unsigned short	log_136	logged register 136	130	READABLE ALWAYS	WRITABLE NEVER	0	65535
40686	unsigned short	log_137	logged register 137	131	READABLE ALWAYS	WRITABLE NEVER	0	65535
40687	unsigned short	log_138	logged register 138	132	READABLE ALWAYS	WRITABLE NEVER	0	65535
40688	unsigned short	log_139	logged register 139	133	READABLE ALWAYS	WRITABLE NEVER	0	65535
40689	unsigned short	log_140	logged register 140	134	READABLE ALWAYS	WRITABLE NEVER	0	65535
40690	unsigned short	log_141	logged register 141	135	READABLE ALWAYS	WRITABLE NEVER	0	65535
40691	unsigned short	log_142	logged register 142	136	READABLE ALWAYS	WRITABLE NEVER	0	65535
40692	unsigned short	log_143	logged register 143	137	READABLE ALWAYS	WRITABLE NEVER	0	65535
40693	unsigned short	log_144	logged register 144	138	READABLE ALWAYS	WRITABLE NEVER	0	65535
40694	unsigned short	log_145	logged register 145	139	READABLE ALWAYS	WRITABLE NEVER	0	65535
40695	unsigned short	log_146	logged register 146	140	READABLE ALWAYS	WRITABLE NEVER	0	65535
40696	unsigned short	log_147	logged register 147	141	READABLE ALWAYS	WRITABLE NEVER	0	65535
40697	unsigned short	log_148	logged register 148	142	READABLE ALWAYS	WRITABLE NEVER	0	65535
40698	unsigned short	log_149	logged register 149	143	READABLE ALWAYS	WRITABLE NEVER	0	65535

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40699	unsigned short	log_150	logged register 150	144	ALWAYS READABLE	NEVER WRITABLE	0	65535
40700	unsigned short	log_151	logged register 151	145	ALWAYS READABLE	NEVER WRITABLE	0	65535
40701	unsigned short	log_152	logged register 152	146	ALWAYS READABLE	NEVER WRITABLE	0	65535
40702	unsigned short	log_153	logged register 153	147	ALWAYS READABLE	NEVER WRITABLE	0	65535
40703	unsigned short	log_154	logged register 154	148	ALWAYS READABLE	NEVER WRITABLE	0	65535
40704	unsigned short	log_155	logged register 155	149	ALWAYS READABLE	NEVER WRITABLE	0	65535
40705	unsigned short	log_156	logged register 156	150	ALWAYS READABLE	NEVER WRITABLE	0	65535
40706	unsigned short	log_157	logged register 157	151	ALWAYS READABLE	NEVER WRITABLE	0	65535
40707	unsigned short	log_158	logged register 158	152	ALWAYS READABLE	NEVER WRITABLE	0	65535
40708	unsigned short	log_159	logged register 159	153	ALWAYS READABLE	NEVER WRITABLE	0	65535
40709	unsigned short	log_160	logged register 160	154	ALWAYS READABLE	NEVER WRITABLE	0	65535
40710	unsigned short	log_161	logged register 161	155	ALWAYS READABLE	NEVER WRITABLE	0	65535
40711	unsigned short	log_162	logged register 162	156	ALWAYS READABLE	NEVER WRITABLE	0	65535
40712	unsigned short	log_163	logged register 163	157	ALWAYS READABLE	NEVER WRITABLE	0	65535
40713	unsigned short	log_164	logged register 164	158	ALWAYS READABLE	NEVER WRITABLE	0	65535
40714	unsigned short	log_165	logged register 165	159	ALWAYS READABLE	NEVER WRITABLE	0	65535
40715	unsigned short	log_166	logged register 166	160	ALWAYS READABLE	NEVER WRITABLE	0	65535
40716	unsigned short	log_167	logged register 167	161	ALWAYS READABLE	NEVER WRITABLE	0	65535
40717	unsigned short	log_168	logged register 168	162	ALWAYS READABLE	NEVER WRITABLE	0	65535
40718	unsigned short	log_169	logged register 169	163	ALWAYS READABLE	NEVER WRITABLE	0	65535
40719	unsigned short	log_170	logged register 170	164	ALWAYS READABLE	NEVER WRITABLE	0	65535
40720	unsigned short	log_171	logged register 171	165	ALWAYS READABLE	NEVER WRITABLE	0	65535
40721	unsigned short	log_172	logged register 172	166	ALWAYS	NEVER	0	65535

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40722	unsigned short	log_173	logged register 173	167	READABLE ALWAYS READABLE	WRITABLE NEVER WRITABLE	0	65535
40723	unsigned short	log_174	logged register 174	168	READABLE ALWAYS READABLE	WRITABLE NEVER WRITABLE	0	65535
40724	unsigned short	log_175	logged register 175	169	READABLE ALWAYS READABLE	WRITABLE NEVER WRITABLE	0	65535
40725	unsigned short	log_176	logged register 176	170	READABLE ALWAYS READABLE	WRITABLE NEVER WRITABLE	0	65535
40726	unsigned short	log_177	logged register 177	171	READABLE ALWAYS READABLE	WRITABLE NEVER WRITABLE	0	65535
40727	unsigned short	log_178	logged register 178	172	READABLE ALWAYS READABLE	WRITABLE NEVER WRITABLE	0	65535
40728	unsigned short	log_179	logged register 179	173	READABLE ALWAYS READABLE	WRITABLE NEVER WRITABLE	0	65535
40729	unsigned short	log_180	logged register 180	174	READABLE ALWAYS READABLE	WRITABLE NEVER WRITABLE	0	65535
40730	unsigned short	log_181	logged register 181	175	READABLE ALWAYS READABLE	WRITABLE NEVER WRITABLE	0	65535
40731	unsigned short	log_182	logged register 182	176	READABLE ALWAYS READABLE	WRITABLE NEVER WRITABLE	0	65535
40732	unsigned short	log_183	logged register 183	177	READABLE ALWAYS READABLE	WRITABLE NEVER WRITABLE	0	65535
40733	unsigned short	log_184	logged register 184	178	READABLE ALWAYS READABLE	WRITABLE NEVER WRITABLE	0	65535
40734	unsigned short	log_185	logged register 185	179	READABLE ALWAYS READABLE	WRITABLE NEVER WRITABLE	0	65535
40735	unsigned short	log_186	logged register 186	180	READABLE ALWAYS READABLE	WRITABLE NEVER WRITABLE	0	65535
40736	unsigned short	log_187	logged register 187	181	READABLE ALWAYS READABLE	WRITABLE NEVER WRITABLE	0	65535
40737	unsigned short	log_188	logged register 188	182	READABLE ALWAYS READABLE	WRITABLE NEVER WRITABLE	0	65535
40738	unsigned short	log_189	logged register 189	183	READABLE ALWAYS READABLE	WRITABLE NEVER WRITABLE	0	65535
40739	unsigned short	log_190	logged register 190	184	READABLE ALWAYS READABLE	WRITABLE NEVER WRITABLE	0	65535
40740	unsigned short	log_191	logged register 191	185	READABLE ALWAYS READABLE	WRITABLE NEVER WRITABLE	0	65535
40741	unsigned short	log_192	logged register 192	186	READABLE ALWAYS READABLE	WRITABLE NEVER WRITABLE	0	65535
40742	unsigned short	log_193	logged register 193	187	READABLE ALWAYS READABLE	WRITABLE NEVER WRITABLE	0	65535
40743	unsigned short	log_194	logged register 194	188	READABLE ALWAYS READABLE	WRITABLE NEVER WRITABLE	0	65535

Register	Type	Name	Description	Default	Read Role	Write Role	Min	Max
40744	unsigned short	log_195	logged register 195	189	ALWAYS READABLE	NEVER WRITABLE	0	65535
40745	unsigned short	log_196	logged register 196	190	ALWAYS READABLE	NEVER WRITABLE	0	65535
40746	unsigned short	log_197	logged register 197	191	ALWAYS READABLE	NEVER WRITABLE	0	65535
40747	unsigned short	log_198	logged register 198	192	ALWAYS READABLE	NEVER WRITABLE	0	65535
40748	unsigned short	log_199	logged register 199	193	ALWAYS READABLE	NEVER WRITABLE	0	65535
40749	unsigned short	log_200	logged register 200	194	ALWAYS READABLE	NEVER WRITABLE	0	65535