VECTROLINK
TECHNOLOGIES

vMQT-02

Modbus to MQTT Gateway (Modbus RTU & TCP)

User Manual

Installation, Configuration & Operation Guide

Firmware version: 0.1.0

Document version: 1.0

November 18, 2026

Confidential — intended for the customer and operating personnel of this device.

Contents

1 Introduction **3**
- 1.1 About This Document . . . 3
- 1.2 What is the vMQT-02? . . . 3
- 1.3 Points . . . 4
- 1.4 Key Features at a Glance . . . 4

2 Hardware Overview **5**
- 2.1 Platform . . . 5
- 2.2 RS-485 Bus Termination . . . 5
- 2.3 Status LED Indications . . . 5
- 2.4 Push-Button Functions . . . 6
- 2.5 Power and Mounting . . . 7

3 Getting Started **8**
- 3.1 Initial Power-Up . . . 8
- 3.2 Connecting for the First Time . . . 8
- 3.3 Default Login Credentials . . . 8
- 3.4 Network Configuration . . . 9
- 3.5 Quick-Start Checklist . . . 9

4 Web Interface Guide **10**
- 4.1 How Changes Are Applied . . . 10
- 4.2 Dashboard . . . 10
  - 4.2.1 Configuration Backup . . . 10
- 4.3 Points . . . 11
  - 4.3.1 Point Fields . . . 11
  - 4.3.2 CSV Import and Export . . . 12
- 4.4 MQTT . . . 12
  - 4.4.1 Broker . . . 12
  - 4.4.2 Topics and Payload . . . 13
  - 4.4.3 Home Assistant . . . 13
- 4.5 Gateway & RS485 . . . 13
  - 4.5.1 Device . . . 13
  - 4.5.2 Modbus RTU Channels . . . 13
  - 4.5.3 Modbus TCP Devices . . . 14
- 4.6 User Management . . . 14
- 4.7 Network Settings . . . 14
  - 4.7.1 Wi-Fi Configuration . . . 14
  - 4.7.2 Ethernet Configuration . . . 14
  - 4.7.3 Automatic Firmware Updates . . . 15
  - 4.7.4 Factory Reset . . . 15

5 MQTT and Modbus Reference **16**
5.1 Topics . . . 16
5.2 Availability (Status Topic) . . . 16
5.3 Payload . . . 16
5.3.1 JSON (default) . . . 16
5.3.2 Value only . . . 17
5.4 When a Point Is Published . . . 17
5.5 Home Assistant Discovery . . . 17
5.6 Data Types . . . 18
5.6.1 Byte Order . . . 18
5.6.2 Scaling . . . 18
5.7 Polling . . . 18

6 Security Notes **20**
6.1 Device Integrity . . . 20
6.2 Encrypted Web Access (HTTPS) . . . 20
6.2.1 Why Your Browser Shows a Warning . . . 20
6.3 Accounts and Passwords . . . 20
6.4 Network Placement Recommendations . . . 21
6.5 Connection to the MQTT Broker . . . 21
6.6 Remote Access via Vectrolink Cloud . . . 21
6.7 Account Permissions . . . 22

7 Troubleshooting **23**
7.1 Common Issues . . . 23
7.2 Getting Further Help . . . 24

8 Specifications **25**
8.1 Capacity Limits . . . 25
8.2 MQTT . . . 25
8.3 Modbus . . . 26
8.4 Connectivity . . . 26
8.5 Physical and Electrical . . . 26
8.6 Account Permissions . . . 26
8.7 Security . . . 26

9 API Reference **27**
9.1 Connecting and Authentication . . . 27
9.2 Endpoints . . . 28

Chapter 1

Introduction

1.1 About This Document

This manual describes the installation, configuration, and day-to-day operation of the **Vectrolink vMQT-02**, a gateway that reads Modbus devices and publishes their values to an MQTT broker. It is intended for technicians, system integrators, and operators who install, configure, or maintain the device on site.

This document covers:

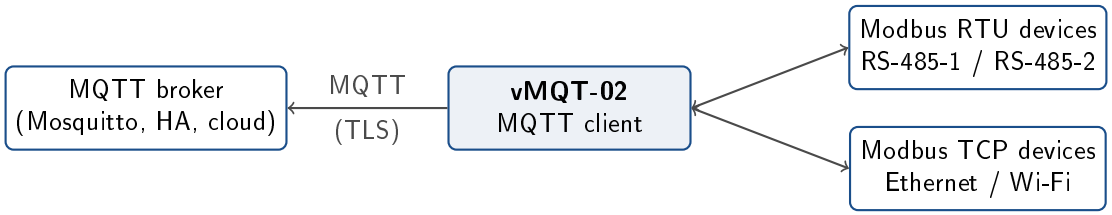
- Hardware overview and physical interfaces
- Initial setup and first boot
- Full walkthrough of the embedded web configuration interface
- How Modbus values become MQTT messages, including Home Assistant discovery
- Security notes relevant to operators and IT staff
- Troubleshooting common issues

The complete HTTP API is described in the separate *vMQT-02 API Reference*; Chapter 9 gives an overview.

1.2 What is the vMQT-02?

The vMQT-02 runs on the same hardware as the vMT-02 Modbus Translator, with firmware for a different task: it delivers the values of Modbus field devices – energy and heat meters, inverters, heat pumps, frequency drives, controllers and sensors – to systems that speak **MQTT**: Home Assistant, Node-RED, Grafana/InfluxDB (Telegraf), IoT platforms and energy management systems.

The gateway polls Modbus RTU devices on its two RS-485 channels and Modbus TCP devices on the local network, converts every configured value to an engineering value with its unit, and publishes it to the broker under a readable topic such as `vectrolink/plant-room/grid_voltage`. No PC, script or middleware is involved.



1.3 Points

The configuration of the gateway is a list of **points**. A point links one value in a Modbus device to one MQTT topic:

- **Modbus side:** channel (RS485-1, RS485-2 or one of eight Modbus TCP devices), slave address or unit ID, function (coils, discrete inputs, holding or input registers), address, data type and byte order, scale and offset.
- **MQTT side:** name (the last level of the topic), description, unit, publish period, deadband for publishing on change, and the retain flag.

Up to 500 points are supported. The point list is edited in the web interface or as a CSV file in Excel (Chapter 4).

1.4 Key Features at a Glance

Feature	Description
Modbus sources	Modbus RTU master on 2 RS-485 channels; Modbus TCP client for up to 8 devices on the local network
Values, not registers	Every point has a name, data type, byte order, scale and unit – the broker receives `230.5` with unit `V`, not raw registers
MQTT	MQTT 3.1.1 over TCP or TLS with server certificate verification; user name and password; availability status (last will)
Payload	JSON with value, unit, quality and time stamp, or the plain value
Publishing	Per point: periodically, and on change beyond a deadband
Data quality	A failed Modbus read is published as `"quality": "comm_failure"`, never as a stale value that looks fresh
Home Assistant	MQTT discovery: the points appear in Home Assistant as sensors of one device, with units and device classes
Data types	U16, S16, U32, S32, FLOAT32, single bits; four byte orders; scale and offset
Commissioning	Point list in the browser or as a CSV file; live values and per-device diagnostics
Embedded web UI	HTTPS, no software installation required – configure entirely from a browser

Chapter 2

Hardware Overview

2.1 Platform

The vMQT-02 runs on a dual-core microcontroller (240 MHz) with the following on-board interfaces:

Interface	Purpose
RS-485 – 2 independent channels (RS485-1, RS485-2)	Two separate RS-485 buses, each with its own baud rate, parity, and stop bits. Each channel polls Modbus RTU devices
10/100 Ethernet	MQTT, Modbus TCP, web interface and Vectrolink Cloud – primary recommended connectivity
Wi-Fi 802.11 b/g/n	MQTT, Modbus TCP, web interface and configuration access – secondary or backup connectivity
Status LED (RGB)	Visual indication of boot state, network activity, and Modbus traffic
Push-button	Temporary Wi-Fi activation and factory reset (see Section 2.4)

2.2 RS-485 Bus Termination

Each RS-485 channel has an on-board termination resistor that can be switched in or out using a jumper: **J5** for RS485-1, **J6** for RS485-2.

Both jumpers ship **open (termination disabled)** by default on every unit.

A correctly terminated RS-485 bus needs a termination resistor at each of its two physical ends, and nowhere else. Fitting the jumper on a device that is not at the end of the bus adds an extra resistor in parallel with the ones already there, which loads the bus and can cause intermittent communication errors that are hard to trace back to this cause.

Fit the jumper for a given channel only if the vMQT-02 is physically the last device on that RS-485 segment. Leave it open if the bus continues past the vMQT-02 to other devices, or if termination is already provided elsewhere on the segment.

2.3 Status LED Indications

The device has a single RGB status LED that communicates its current state:

The steady background colour reflects overall connectivity; brief pulses for Modbus activity and errors, and yellow/red flashes for button actions, briefly override it and then it returns to whichever background colour currently applies.

LED behaviour	Meaning
White, blinking	Device is starting up / initializing (approximately 10 seconds after power-on)
Green, steady	Normal operation: network connected (Wi-Fi or Ethernet) and Vectrolink Cloud reachable
Orange, steady	Network connected, but Vectrolink Cloud is not currently reachable (check internet access)
Red, steady	No network connection at all (neither Wi-Fi nor Ethernet)
Purple, steady	Access Point mode active – the device is broadcasting its own Wi-Fi hotspot (see Section 2.4)
Cyan, brief pulse	Modbus polling: the gateway is reading from or writing to a Modbus device
Red, brief flash	Modbus error: timeout, CRC error or exception from a Modbus device
Yellow, 1 flash	Push-button: short press recognised
Yellow, 2 flashes	Push-button: double press recognised
Amber, blinking (speeding up)	Push-button: held down, factory reset building up – starts blinking slowly around 2 seconds into the hold and blinks faster the closer it gets to 10 seconds; release before then to cancel
Yellow, 4 flashes, then red 6 flashes	Push-button: long press recognised, factory reset in progress

2.4 Push-Button Functions

The device has a single physical push-button with three distinct actions, depending on how it is pressed:

Short press (press and release quickly) Temporarily enables Wi-Fi in *station* mode for 10 minutes, then automatically reverts to the previous network configuration. Useful if Wi-Fi was disabled and you need temporary access.

Double press (two short presses within roughly half a second) Temporarily enables Wi-Fi in *access point* mode for 10 minutes – the device creates its own Wi-Fi hotspot you can connect to directly, then automatically reverts afterward. Useful if the device cannot be reached on the existing network at all.

Long press (hold for 10 seconds or more) Performs a **factory reset**: network settings, the point table, the gateway, Modbus TCP and MQTT broker settings, and user accounts are erased and reset to defaults. Starting about 2 seconds into the hold, the status LED blinks amber, speeding up the closer you get to 10 seconds – release before then to cancel without making any changes.

Factory reset erases the point table, the gateway, Modbus TCP and MQTT broker settings, user accounts, and network configuration – this cannot be undone. Export the point table as a CSV file (Section 4.3.2) and note the gateway and MQTT broker settings (including the broker password) before performing a long press. The device's identity (name, serial number) and its connection to Vectrolink Cloud (including its TLS certificate) are *not* affected – you will not need to re-register the device afterward.

2.5 Power and Mounting

Refer to the datasheet supplied with your specific hardware revision for power supply voltage range, connector pinout, and mechanical (DIN-rail / enclosure) mounting details, as these may vary between hardware batches.

Chapter 3

Getting Started

3.1 Initial Power-Up

1. Connect the RS-485 wiring to your Modbus RTU field devices (skip this if you only read Modbus TCP devices).
2. Connect an Ethernet cable to the network on which the MQTT broker and any Modbus TCP devices can be reached.
3. Apply power to the device.
4. Observe the status LED: it will blink white for approximately 10 seconds while the device initializes, then show the network state (Chapter 2).

3.2 Connecting for the First Time

On first boot, no Wi-Fi network is configured. The device automatically falls back to creating its own access point so that it can always be reached:

1. On your laptop or phone, look for a Wi-Fi network named `vMQT-02_<SERIAL>`, where `<SERIAL>` is the 8-character serial number unique to your device (printed on the device label).
2. Connect to this network using the password `12345678`.
3. Open a web browser and navigate to:

   `https://192.168.4.1`

4. You will be presented with the login page of the embedded web interface.

If the device is connected via Ethernet with DHCP, it obtains an address on your existing network. Check your router's DHCP client list for a device named `vMQT-02_<SERIAL>` – most routers show this alongside the assigned IP address. If you cannot locate it, use the push-button (Section 2.4) to temporarily force the device into access-point mode.

3.3 Default Login Credentials

The device ships with two pre-configured accounts for day-to-day use:

Username	Password	Access
`user`	`user`	Administrative access (web + API)
`api`	`api`	API access only (no web login)

Change both default passwords immediately after your first login. They are identical on every device of this model until changed, and are publicly documented in this manual. Note in particular that `user`/`user` is a full administrative account, not a restricted one. See Section 4.6 for instructions on changing passwords and creating additional accounts.

You may notice a third account named `cloud` in the Users list. It is reserved for the Vectrolink Cloud portal's single-sign-on feature (Section 6.6) and is not meant for direct login – please do not delete or repurpose it, or the "Open Device UI" button in the portal will stop working.

3.4 Network Configuration

After logging in, open the **Network** page to configure how the device connects to your network:

- **Ethernet** – DHCP (automatic) or static IP address, subnet mask, and gateway.
- **Wi-Fi** – select a network from a scan, or enter SSID/password manually; *station* mode (joining your network), *access point* mode, or both.

The gateway connects *out* to the broker and to the Modbus TCP devices, so it does not need a fixed address for MQTT. A DNS server is needed when the broker is given by host name (for example a cloud broker); with DHCP it is provided by the router.

3.5 Quick-Start Checklist

1. Power up the device and connect to it for the first time (Section 3.3).
2. Log in and immediately change all default account passwords.
3. Configure the network connection (Ethernet and/or Wi-Fi).
4. On the **Gateway & RS485** page: set the device name (it becomes part of every topic), the baud rate, parity and stop bits of the RS-485 channels, and the address of each Modbus TCP device.
5. On the **MQTT** page: enter the broker address, port, user name and password, choose TLS, and press **Test connection**.
6. On the **Points** page: add the points, or import a prepared CSV file; save the table.
7. Restart the device to apply the configuration (the yellow banner offers a button).
8. Check the **Points** page (live values, status OK), the *Modbus Devices* table and the *MQTT* box on the Dashboard.
9. Subscribe to the topics on the broker (for example `mosquitto_sub -t 'vectrolink/#' -v`) or open Home Assistant: with discovery enabled, the device and its sensors appear under *Settings* → *Devices*.

Chapter 4

Web Interface Guide

The embedded web interface is the only configuration tool for the device – no desktop software or mobile app is required. This chapter walks through every page of the interface in the order it appears in the navigation menu.

4.1 How Changes Are Applied

The gateway builds its Modbus polling plan and its MQTT topics when it starts. Changes to the point table, the gateway and RS-485 settings, the Modbus TCP devices and the MQTT broker settings are therefore **saved immediately but applied after a restart**: after saving, a yellow banner appears with a **Restart now** button. Several changes can be saved one after another and applied with a single restart. A restart takes about 10 seconds; the broker sees the device go `offline` briefly and come back `online`.

4.2 Dashboard

The Dashboard is the landing page after login and gives an at-a-glance overview of the gateway:

Device Info Device name, serial number, firmware version, uptime, and device time.

MQTT Connection state, broker, topic base (for example `vectrolink/plant-room/`), number of points, messages published, and the last connection error, if any.

Connectivity Wi-Fi, Ethernet, and whether the Modbus devices are answering.

Modbus Polling Read requests and errors, duration of the last complete poll cycle, and the success rate.

Modbus Devices One line per Modbus device – RS-485 channel and slave address, or Modbus TCP device and unit ID: state, requests, timeouts, CRC errors, exceptions, and average/maximum response time. This is the first place to look when values do not update (Chapter 7).

4.2.1 Configuration Backup

The *Configuration Backup* card at the bottom of the Dashboard keeps the whole configuration of the device in one JSON file: device name, poll cycle, RS-485 channels, Modbus TCP devices, MQTT settings (without password and CA certificate) and the point table.

Download backup Saves the file `vMQT-02_<serial>_config_<date>.json`.

Restore Loads such a file. The device checks the whole file first – a file of another product or with an invalid value is refused and nothing changes – then asks to restart to apply it.

Not in the file: passwords and certificates, user accounts and network settings. After restoring on another device, enter these again (e.g. the broker password on the MQTT page).

> Download a backup before a factory reset and after every larger change. To set up several devices the same way, configure one, download its backup and restore it on the others; then adjust the device name and addresses.

4.3 Points

The Points page shows the point table with the live value and communication status of every point (refreshed every 2 seconds). Each row shows the name and topic, the unit, the Modbus source (channel, slave, function, address), the data type, scale/offset and the publishing settings.

Add point Opens the point editor for a new point.

Edit / Copy / × Edit a point, copy it to the next register (handy for a series of similar values), or delete it.

Save table Saves all changes made in the table; then restart to apply them. **Discard changes** reloads the saved table.

CSV / CSV (Excel PL) Downloads the saved point table as a CSV file – comma separated, or semicolon separated with a decimal comma for Excel in Polish, German and most other European locales.

Import CSV Replaces the whole point table with a CSV file (Section 4.3.2).

4.3.1 Point Fields

Field	Description
Name	Last level of the MQTT topic, max 31 characters, unique in the table, without `/`, `+` and `#`. Use letters, digits, `_` and `-` (e.g. `grid_voltage_l1`) – topics with spaces work but are awkward in most tools
Description	Optional, max 47 characters; also the friendly name of the sensor in Home Assistant
Unit	Engineering unit as text, max 11 characters, e.g. `V`, `kWh`, `°C`; sent in the JSON payload and used for Home Assistant
Publish period	Seconds between periodic publications of this point; 0 = the default period of the MQTT page
Deadband	The point is also published as soon as its value changes by at least this amount; 0 = on every change
Retain	The broker keeps the last message and gives it to new subscribers immediately
Enabled	Disabled points are neither polled nor published
Channel	RS485-1, RS485-2, or TCP-1 . . . TCP-8 (the Modbus TCP devices of the Gateway page)
Slave ID	Modbus slave address 1–247 on RS-485; unit ID 0–255 on Modbus TCP (often 1, or 255 for devices that ignore it)
Function	FC01 coils, FC02 discrete inputs, FC03 holding registers, FC04 input registers

Field	Description
Address	0-based register or bit address (holding register 40001 = address 0)
Data type	U16, S16, U32, S32, F32, or BIT (Chapter 5)
Byte order	ABCD, CDAB, BADC, DCBA – for 32-bit types
Bit	For BIT on registers: which bit (0–15)
Scale / Offset	Published value = raw value × scale + offset

4.3.2 CSV Import and Export

The CSV file has one header line naming the columns and one line per point:

```
name,description,unit,enabled,port,slave,function,address,type,order,
    bit,scale,offset,period,deadband,retain,writable
"grid_voltage_l1","Main meter L1",V,1,1,3,4,0,F32,CDAB,0,1,0,60,0.5,0,0
"pv_power","Inverter",kW,1,TCP1,1,3,30775,S32,ABCD
    ,0,0.001,0,10,0.05,0,0
```

The columns may be in any order and only `name` and `address` are required – missing columns take sensible defaults. In the `port` column, `1` and `2` are the RS-485 channels and `TCP1` … `TCP8` the Modbus TCP devices. Comma and semicolon separated files are both accepted. Before anything is saved, the whole file is checked; if a line is wrong, the import is refused with the line number and the reason (e.g. *line 7: FC01/FC02 need type BIT*), and the existing table stays unchanged.

Typical workflow: prepare the point list for one type of installation once, then for every new site export, adapt slave addresses and names in Excel, and import. Keep the exported CSV file as the backup of the point table.

4.4 MQTT

4.4.1 Broker

Field	Description
Enable MQTT	Master switch; when off, the gateway polls the Modbus devices but publishes nothing
Broker	IP address or host name of the broker
Port	1883 for MQTT, 8883 for MQTT over TLS (the usual ports)
TLS	Encrypt the connection and verify the broker's certificate. Certificates of public certificate authorities are verified with the built-in list; for a broker with its own (private) CA, paste that CA certificate in **CA certificate**
CA certificate	Optional, PEM format (`---BEGIN CERTIFICATE---`…); only needed for a private CA
User name / Password	Broker credentials. The password is never shown again after saving; leave the field empty to keep the saved one

Field	Description
Client ID	Empty = `vMQT-02_<SERIAL>`. Must be unique on the broker – two clients with the same ID disconnect each other
Keepalive	Seconds, default 60. The broker declares the gateway `offline` after about 1.5× this time without contact

Test connection connects to the broker with the values in the form (also before they are saved) and reports success or the reason of the failure, e.g. *not authorised*, *certificate verify failed* or *host not reachable*.

4.4.2 Topics and Payload

Field	Description
Topic prefix	First topic level(s), default `vectrolink`; may contain `/`, e.g. `site1/energy`
Payload format	*JSON* (value, unit, quality, time stamp) or *Value only* (Section 5.3)
Default publish period	Seconds, default 60; used by points whose own period is 0

The device name level is set on the Gateway page. Together they form the *topic base*, shown on the page: `<prefix>/<device name>/`.

4.4.3 Home Assistant

Home Assistant discovery (on by default) publishes a configuration message for every point, so that Home Assistant creates the device and its sensors automatically (Section 5.5). **Discovery prefix** is `homeassistant` unless it was changed in Home Assistant's MQTT integration.

4.5 Gateway & RS485

4.5.1 Device

Field	Description
Device name	Topic level after the prefix, max 31 characters, without `/`, `+` and `#`; empty = `vMQT-02_<SERIAL>`. Use the location or function, e.g. `plant-room`
Topic base	Read-only: the beginning of every topic of this gateway

4.5.2 Modbus RTU Channels

For each RS-485 channel: baud rate (1200–115200 bps), parity (None/Even/Odd), stop bits (1 or 2), and the response timeout. The **poll cycle target** sets how often all points are read; if reading all points takes longer, the next cycle starts immediately.

4.5.3 Modbus TCP Devices

Up to eight Modbus TCP devices (TCP-1 … TCP-8) on the local network: inverters, meters with an Ethernet port, PLCs, or RS-485/Ethernet converters with several RTU devices behind them (the unit ID of a point then selects the RTU device).

Field	Description
Host	IPv4 address or host name; empty = slot not used
Port	Modbus TCP port, normally 502
Timeout	Connection and response timeout, 100–10000 ms (default 1000)

The gateway keeps one connection open to each device and reuses it for all requests. If a device cannot be reached, its points show a communication failure and the gateway tries again every 5 seconds, without slowing down the other devices.

4.6 User Management

Any account can see and use this page, but only `superadmin` can grant admin privileges or edit/delete the built-in `user`/`api`/`cloud` accounts. Every other account can create and manage additional accounts (up to 5).

Create User Creates a new account with a username and password, and independently toggleable *Web access* and *API access* permissions.

Users table Lists existing accounts and provides per-account actions (edit, delete) for the accounts you're allowed to manage.

4.7 Network Settings

4.7.1 Wi-Fi Configuration

Field	Description
Mode	AP Only (device hosts its own network), STA Only (device joins an existing network), or AP + STA
Enable Wi-Fi	Master on/off switch for the Wi-Fi radio
Scan Networks	Lists nearby Wi-Fi networks for easy selection
Saved Networks	Up to 5 networks with a priority each (0 = highest)
IP Configuration	DHCP (automatic) or static IP / netmask / gateway / DNS

4.7.2 Ethernet Configuration

Ethernet is always enabled. IP addressing can be set to DHCP or a static IP / netmask / gateway / DNS configuration. After changing any network setting, click **Save Network Settings**; the device may briefly become unreachable at its previous address.

4.7.3 Automatic Firmware Updates

The checkbox **Allow firmware update** in the sidebar (enabled by default) lets the device download and install firmware updates published by Vectrolink through the Cloud update service. When disabled, the device keeps working unchanged but does not install updates; a firmware file can still be installed manually. We recommend leaving it enabled to receive security fixes and new features.

4.7.4 Factory Reset

A **Factory Reset** button is available in the sidebar, next to Restart. It has the same effect as the push-button procedure (Section 2.4) and asks for confirmation first.

Chapter 5

MQTT and Modbus Reference

5.1 Topics

Every topic of the gateway starts with the *topic base*, made of the topic prefix (MQTT page) and the device name (Gateway page):

Topic	Content
`<prefix>/<device>/<point>`	Value of a point (Section 5.3)
`<prefix>/<device>/status`	`online` / `offline` – availability of the gateway, retained

Example with prefix `vectrolink`, device name `plant-room` and a point `grid_voltage_l1`:

```
vectrolink/plant-room/status                online
vectrolink/plant-room/grid_voltage_l1   {"value":230.5,"unit":"V","
   quality":"good","ts":1794573479}
```

To receive everything from one gateway, subscribe to `vectrolink/plant-room/#`; from all gateways of a site, to `vectrolink/+/#`.

5.2 Availability (Status Topic)

When the gateway connects to the broker it publishes `online` to the status topic, retained. It also registers `offline` as its *last will*: if the gateway loses power or the network, the broker publishes `offline` itself after the keepalive time runs out. Before a planned restart the gateway publishes `offline` on its own. Dashboards and Home Assistant use this topic to show all values of the gateway as unavailable.

5.3 Payload

5.3.1 JSON (default)

```
{"value":230.5,"unit":"V","quality":"good","ts":1794573479}
```

Key	Meaning
`value`	Engineering value: raw × scale + offset. `0`/`1` for bits. `null` when the read failed.
`unit`	Unit of the point; left out when the point has none.
`quality`	`good` – read successfully in the last poll cycle; `comm_failure` – the Modbus device did not answer or answered with an error.
`ts`	Time of the reading, Unix epoch seconds (UTC). Left out while the device time is not synchronised (no internet access).

5.3.2 Value only

The payload is the number alone, e.g. `230.5` – for simple consumers and tools that expect a bare value. In this format a failed read cannot be shown in the payload, so nothing is published for a point while its device does not answer; the status topic and the web interface show the failure. Choose JSON when the consumer must be able to tell a stale value from a fresh one.

5.4 When a Point Is Published

- **Periodically**, every *publish period* seconds (the point's own period, or the default period of the MQTT page) – also when the value has not changed, so the consumer sees that the value is current.
- **On change**, right after the poll cycle in which the value moved by at least the point's *deadband* since the last publication (deadband 0: on every change). Bits are published on every change.
- **On a change of quality** – when a device stops or starts answering.
- **After connecting** to the broker, all points once.

Choose the deadband to match what the consumer needs: a temperature with deadband 0.1 and a period of 300 s gives prompt updates without flooding the broker; an energy counter with deadband 0 changes with every pulse. Values are published with QoS 0; the status topic and the discovery messages with QoS 1 and retained.

5.5 Home Assistant Discovery

With *Home Assistant discovery* on, the gateway publishes one retained configuration message per point under

```
homeassistant/<sensor|binary_sensor>/vmqt02_<serial>/<point>/config
```

and Home Assistant creates the entities by itself:

- One **device** per gateway: name = device name, manufacturer Vectrolink, model vMQT-02, firmware version, serial number.
- One **sensor** per numeric point, one **binary sensor** per BIT point. The entity name is the description of the point (or its name if the description is empty).
- The unit is passed on, and common units also set the *device class* and *state class*, so that values show up in the right place – for example in the Energy dashboard:

Unit of the point	Device class	State class
`V`, `mV`, `kV`	voltage	measurement
`A`, `mA`	current	measurement
`W`, `kW`	power	measurement
`Wh`, `kWh`, `MWh`	energy	total_increasing
`°C`, `°F`	temperature	measurement
`%RH` (sent to Home Assistant as `%`)	humidity	measurement
`Hz`	frequency	measurement
`Pa`, `hPa`, `kPa`, `bar`	pressure	measurement
`m³`	volume	total_increasing
other / none	–	measurement

- Availability is taken from the status topic: when the gateway is offline, all its entities are *unavailable*. With the JSON payload, a point with `comm_failure` is also shown as unavailable.
- Deleting a point from the table removes its entity from Home Assistant after the next restart of the gateway (an empty configuration message is published). Entities of other gateways are not touched. Switching discovery off removes all entities of the gateway.

Home Assistant setup: install the *Mosquitto broker* add-on (or use your own broker), add the *MQTT* integration, create a Home Assistant user for the gateway and enter it on the MQTT page. Discovery is on by default in Home Assistant.

5.6 Data Types

Type	Registers	Description
U16	1	Unsigned 16-bit integer, 0–65535
S16	1	Signed 16-bit integer, −32768–32767
U32	2	Unsigned 32-bit integer
S32	2	Signed 32-bit integer
F32	2	32-bit IEEE 754 floating point
BIT	–	A coil or discrete input (FC01/FC02), or one bit (0–15) of a register (FC03/FC04)

5.6.1 Byte Order

32-bit values occupy two consecutive registers. Device manufacturers order the four bytes differently; with the bytes of the value named A B C D (A = most significant):

Setting	Registers	Typical use
ABCD	AB, CD	Big-endian – the Modbus standard, most meters
CDAB	CD, AB	Word swap – low word first, common for floats
BADC	BA, DC	Byte swap within each register
DCBA	DC, BA	Little-endian

If a 32-bit value looks absurd – for example 10^{-38} or 10^{30} instead of 230.5 – the byte order is wrong. Try CDAB first, then the others.

5.6.2 Scaling

The published value is $raw \times scale + offset$. Example: a meter reports voltage in 0.1 V steps (raw 2305) – with scale 0.1 the broker receives 230.5.

5.7 Polling

All enabled points are read in a cycle. Points of the same device with neighbouring addresses are read together in one request (up to 125 registers), so a meter whose values lie close together costs one request per cycle, not one per value. If a device rejects such a combined request, the gateway automatically reads its points one by one.

The RS-485 channels and the Modbus TCP devices are read in the same cycle. A Modbus TCP device is reached over one permanent connection; a device that does not answer is retried

every 5 seconds and does not delay the others. Several RTU devices behind one RS-485/Ethernet converter are configured as one TCP device, with the unit ID of each point selecting the RTU device.

Chapter 6

Security Notes

6.1 Device Integrity

The device runs only firmware signed by Vectrolink (Secure Boot): modified or foreign firmware is rejected, also when it is uploaded through the web interface or delivered as an update. The configuration and the stored passwords are kept in encrypted memory.

6.2 Encrypted Web Access (HTTPS)

The web interface is served over HTTPS. The device generates its own unique, self-signed TLS certificate the first time it starts – there is no central certificate authority involved, and no internet connection is required for this.

6.2.1 Why Your Browser Shows a Warning

Because the certificate is self-signed rather than issued by a publicly trusted certificate authority, your browser will show a warning similar to *"Your connection is not private"* or *"This site is not secure"* the first time you connect to the device. **This is expected behaviour for this class of device** – the same thing happens with most routers, managed switches, and industrial IP devices that you configure over your local network.

To proceed:

1. Click **Advanced** (or equivalent, depending on your browser).
2. Click **Proceed to <device IP> (unsafe)** or **Continue**.

The connection is still encrypted – the warning relates only to the browser being unable to verify the certificate's issuer, not to the strength of the encryption itself.

> If you want to remove this warning permanently across your organisation's computers, your IT department can import the device's certificate into the trusted certificate store of the browsers/computers that need access. Ask your system integrator for the procedure, as this requires access to the device's certificate file and is typically a one-time setup per device.

6.3 Accounts and Passwords

- Change **all** default account passwords immediately after first login (Section 3.3). They are the same on every unit until changed.
- Use strong, unique passwords – they are not subject to any complexity policy enforced by the firmware, so this is your responsibility.

- Create individual accounts for each technician or operator who needs access, rather than sharing one login, so that access can be revoked individually if needed.
- After 5 failed login attempts from the same source, the device temporarily locks out further attempts from it for 5 minutes. This slows down automated guessing but is not a substitute for strong, unique passwords – avoid placing the device's web interface directly on the public internet regardless.

6.4 Network Placement Recommendations

- Place the device on an isolated operational-technology (OT) network segment where practical, separate from general office IT traffic.
- Do not expose the device's web interface directly to the public internet with port forwarding. If remote access is required, use **Vectrolink Cloud** (Section 6.6) or a VPN provided by your IT department.
- Restrict network access to the device to the systems and personnel that genuinely need it (the MQTT broker, maintenance technicians).

6.5 Connection to the MQTT Broker

- Use **TLS** (port 8883) whenever the broker is outside the local network, and preferably also inside it. With TLS, the gateway verifies the broker's certificate: a connection to a broker with an unknown or wrong certificate is refused, so the broker password cannot be sent to an impostor.
- Give every gateway its **own broker account** with rights limited to its own topics (for Mosquitto: an ACL allowing `vectrolink/<device>/#` and, with discovery, `homeassistant/#`). A gateway that publishes only needs write access.
- The broker password is stored encrypted on the device and is never returned by the web interface or the API – it can only be replaced.
- Without TLS, the broker password and all values travel as plain text on the network. This is acceptable only on an isolated network segment.

> Modbus RTU and Modbus TCP have no authentication: any device on the bus or network segment can read and write the field devices. Keep Modbus TCP devices on the operational-technology network, separated from guest and office networks.

6.6 Remote Access via Vectrolink Cloud

When the device has internet access, its Web UI and REST API can be reached remotely through the Vectrolink Cloud portal, without a VPN and without forwarding any port on your router or firewall. The device only makes an outbound connection to the Cloud relay; no inbound port is ever opened on your network.

From the portal, clicking **Open Device UI** logs you into the device's web interface automatically – the portal signs you in on your behalf using a dedicated, restricted service account, so you are not prompted for the device's own login a second time. This does not replace or bypass individual user accounts on the device itself (Section 3.3); it only skips a redundant login step for someone who has already authenticated to the portal.

Vectrolink Cloud access is a separate service from the device hardware itself. Ask your Vectrolink representative about enabling it for your account if it is not already active.

6.7 Account Permissions

Accounts can be granted *Web access*, *API access*, or both, independently. Any logged-in account can change network settings, perform a factory reset, and create/manage additional accounts of its own. **Granting admin privileges, and editing or deleting the built-in `superadmin`/`cloud`/`user`/`api` accounts, is restricted to the `superadmin` account only.** See Section 4.6 for details.

Chapter 7

Troubleshooting

7.1 Common Issues

The gateway does not connect to the broker. Press **Test connection** on the MQTT page; the message names the cause:

- *host not reachable / connection refused* – wrong address or port, the broker is not running, or a firewall blocks the port (1883 or 8883) on the broker's computer. Check that the broker listens on the network and not only on `localhost` (Mosquitto 2.x does that unless a `listener` is configured).
- *not authorised* – wrong user name or password, or the broker does not allow this client.
- *certificate verify failed* – TLS: the broker's certificate is not issued by a public CA and no CA certificate was entered, the certificate has expired, or it was issued for a different host name or IP address than the one entered as broker.
- *cannot resolve host* – the broker is given by name, but the gateway has no DNS server (static IP without DNS).

The gateway connects and disconnects repeatedly. Another client uses the same client ID – for example two gateways with a manually entered identical ID. Leave the client ID empty to use the unique default.

A point shows a communication failure / its value does not update. Look at the *Modbus Devices* table on the Dashboard:

- *Timeouts* increasing – the device does not answer: check (1) the slave address, (2) baud rate, parity and stop bits of the channel, (3) the wiring polarity (A/B) and that the device is powered, (4) the termination jumper if the vMQT-02 is at the end of the bus (Chapter 2).
- *CRC errors* increasing – the answer is corrupted: termination, cable quality, two devices with the same address, or a baud rate at the limit of the cable length.
- *Exceptions* increasing – the device answers but rejects the request: wrong function code or a register that does not exist; consult the device's register map.

A Modbus TCP device does not answer. Check that the host and port are right and that the device is reachable from the gateway's network. Many devices accept only a few simultaneous Modbus TCP connections – close other programs that hold a connection. Some devices answer only to a particular unit ID (often 1, sometimes 255 or 0); converters to RS-485 need the RTU slave address as unit ID.

The value updates, but the number looks wrong. For 32-bit values, try another byte order (Chapter 5). Check scale and offset, and whether the register address is 0-based (holding register 40001 = address 0).

Values arrive too often or too rarely. Adjust the deadband and the publish period of the point (Section 4.3.1). A deadband of 0 publishes every change of the value, which for a noisy measurement means every poll cycle.

Home Assistant does not show the device. Check that discovery is on, that the discovery prefix matches the one of Home Assistant's MQTT integration (default `homeassistant`), and that the broker account of the gateway may publish under it. Subscribing to `homeassistant/#` on the broker shows whether the configuration messages arrive.

My changes have no effect. Changes to points, gateway and MQTT settings are applied after a restart – use **Restart now** in the yellow banner.

I cannot reach the web interface after changing network settings. Use the push-button (Section 2.4): a short press re-enables Wi-Fi in station mode for 10 minutes, a double press creates a Wi-Fi access point for 10 minutes.

My browser shows a security warning when opening the device's web page. This is expected – see Chapter 6, Section "Encrypted Web Access (HTTPS)".

I forgot my account password and cannot log in with any account. The only way to regain access is a factory reset via a long press of the push-button (10 seconds or more), which erases the configuration and restores the default accounts (Section 3.3). Import your saved CSV point table and enter the broker settings again afterwards.

Firmware upload fails with "Invalid firmware file". Select the plain application `.bin` file for the vMQT-02, not a merged flash image – and not a firmware file of another product, which the device refuses.

7.2 Getting Further Help

If an issue persists after the steps above, collect the following before contacting support:

- Firmware version and serial number (Dashboard).
- The point table as a CSV file.
- A screenshot of the *MQTT* box and of the *Modbus Devices* table on the Dashboard.
- The broker used (product and version) and the name and model of the Modbus devices concerned.

Chapter 8

Specifications

8.1 Capacity Limits

Resource	Maximum	Note
Points (Modbus values published to MQTT)	500	
RS-485 channels	2	Modbus RTU
Modbus TCP devices	8	each with any number of unit IDs
Registers per Modbus read request	125	
Additional user accounts (beyond defaults)	5	–
Saved Wi-Fi networks	5	–

8.2 MQTT

Property	Specification
Role	MQTT client (publisher)
Protocol	MQTT 3.1.1 over TCP or TLS 1.2
TLS	Server certificate verification with a built-in list of public CAs or a customer CA certificate (PEM)
Authentication	User name and password
Topics	`<prefix>/<device>/<point>`, availability `<prefix>/<device>/status` (last will)
Payload	JSON (value, unit, quality, time stamp) or plain value
QoS	0 for values, 1 for status and discovery; retain per point
Publishing	Periodic (1–65535 s per point) and on change beyond a deadband
Home Assistant	MQTT discovery (sensor, binary_sensor) with device, unit, device and state class

8.3 Modbus

Property	Specification
Role	Modbus RTU master on RS-485-1 and/or RS-485-2; Modbus TCP client
Read	FC01, FC02, FC03, FC04
Data types	U16, S16, U32, S32, FLOAT32, bits; byte orders ABCD, CDAB, BADC, DCBA
Modbus TCP	Up to 8 devices (IPv4 address or host name, port, timeout); one permanent connection per device; unit ID 0–255

8.4 Connectivity

Interface	Specification
RS-485	2 independent channels; 1200–115200 bps, parity None/Even/Odd, 1 or 2 stop bits, 8 data bits
Ethernet	10/100 Mbps, DHCP or static IP
Wi-Fi	802.11 b/g/n 2.4 GHz, built-in antenna, station and/or access point mode, DHCP or static IP

8.5 Physical and Electrical

Property	Specification
Power supply	10–24 V DC, less than 3 W
Protection rating	IP20, reverse-polarity protected
Operating temperature	−20 °C to +70 °C
Housing	DIN-rail (35 mm TS), 2-module width
Terminal wire gauge	24–12 AWG
Indicators	1 multicolour status LED – see Chapter 2, Section "Status LED Indications"

8.6 Account Permissions

Administrator and standard user accounts, with independently toggleable Web and API access – see Chapter 6.

8.7 Security

- Secure Boot: only firmware signed by Vectrolink runs on the device; configuration stored encrypted.
- HTTPS web interface with a per-device, self-signed TLS certificate generated on first boot.
- MQTT over TLS with server certificate verification; broker password write-only.
- Password-based authentication with session tokens; temporary lock-out after repeated failed logins.

Chapter 9

API Reference

Everything available in the web interface is also available as a JSON HTTP API, which you can use to integrate the device with your own systems – for example a commissioning script that loads the point tables of many gateways. This chapter gives an overview; every endpoint with its exact requests and responses is described in the separate **vMQT-02 API Reference**.

9.1 Connecting and Authentication

All endpoints are served over HTTPS. Because the device uses a self-signed certificate (Chapter 6), your HTTP client needs to trust that certificate or skip its verification for this host.

1. Obtain a session token by logging in:

```
POST /api/auth/login
{"username": "<username>", "password": "<password>"}
```

2. Include the token on every subsequent request:

```
Authorization: Bearer <token>
```

For machine-to-machine access, create a dedicated account with *API access* and without *Web access* (Section 4.6).

9.2 Endpoints

Endpoint	Description
`GET /api/status`	Firmware, uptime, network, gateway summary, MQTT state, Modbus counters
`GET /api/gateway/config`	Device name, poll cycle, RS-485 channels, Modbus TCP devices
`POST /api/gateway/config`	Change them (applied after restart)
`GET /api/mqtt/config`	Broker, TLS, topics, payload, Home Assistant
`POST /api/mqtt/config`	Change them (applied after restart)
`POST /api/mqtt/test`	Test a broker connection
`GET /api/mqtt/status`	Connection state and counters
`GET /api/points`	Point table with topics and live values
`GET /api/points/values`	Live values only, for frequent polling
`GET /api/points/csv`	Point table as CSV (`?sep=semicolon` for Excel)
`POST /api/points/csv`	Replace the point table (applied after restart)
`GET /api/gateway/stats`	Modbus counters per device
`GET/POST /api/backup`	Configuration backup: download / restore one JSON file
`POST /api/system/restart`	Restart the device
`GET/POST /api/network`	Network settings
`/api/auth/...`	Login, logout, sessions, user accounts, passwords