SQC485I Mesh Configurator — Windows App User Manual

Version 1.0.2 · November 2026

SQC485I Mesh Configurator sets up Siliqs LoRa Mesh nodes (SQC485I and the SQS sensor-node series). Connect a node with a USB cable (or Bluetooth) and point-and-click:

- what the node should do (sensor node / collector-gateway / RS485 tunnel)
- the radio (frequency, spreading factor, TX power) and channel (name, encryption key)
- which Modbus registers to read, and how often
- how Bluetooth pairing works

No firmware edits and no serial commands. After you save, the app reads the node's current settings back so you can confirm them.

> The Windows app runs the same configurator as the website https://mesh.siliqs.net. The difference: the Windows app works without an internet connection and does not need Chrome.

1. System requirements

Item	Requirement
Operating system	Windows 10 or Windows 11, 64-bit
Permissions	Standard user — no administrator rights needed
Disk space	About 350 MB
USB connection	A USB-C cable that carries data (charge-only cables won't work)
Bluetooth (optional)	Bluetooth 4.0 (BLE) or later on the PC

2. Installation

1. Download the installer: open https://mesh.siliqs.net and click **Download for Windows** at the bottom of the left menu, or go straight to https://mesh.siliqs.net/download/SQC485I-Mesh-Configurator-Setup-latest-x64.exe . The download box shows the installer's SHA-256 if you want to verify the file. Double-click it.
2. **Windows may show "Windows protected your PC"** (Microsoft Defender SmartScreen). This appears because the installer is **not digitally signed yet** — it does not mean the file is harmful. Click:
    1. **More info**
    2. **Run anyway**

3. In the setup wizard:
    - Choose the install folder (default `%LOCALAPPDATA%\Programs\SQC485I Mesh Configurator`, installed for the current user only).
    - Click **Install**, then **Finish**.
4. A **SQC485I Mesh Configurator** shortcut appears on the desktop and in the Start menu.

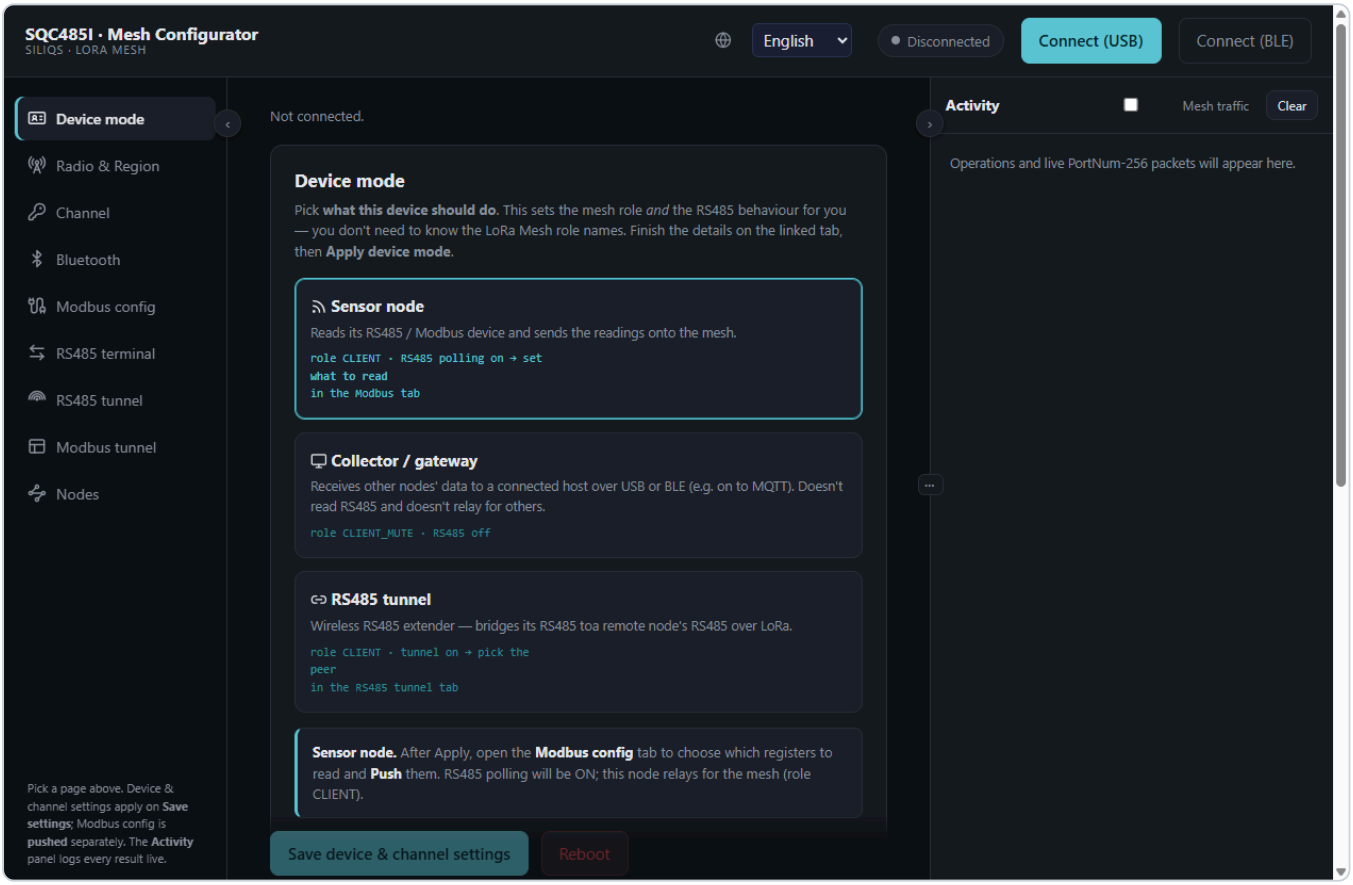
Upgrade: run the newer installer; it replaces the old version. Nothing is lost — all settings live on the node, not on the PC.

Uninstall: Windows *Settings* → *Apps* → *Installed apps* → SQC485I Mesh Configurator → Uninstall.

Mass deployment (IT): silent install with `SQC485I-Mesh-Configurator-Setup-1.0.2-x64.exe /S`.

---

3. The main window



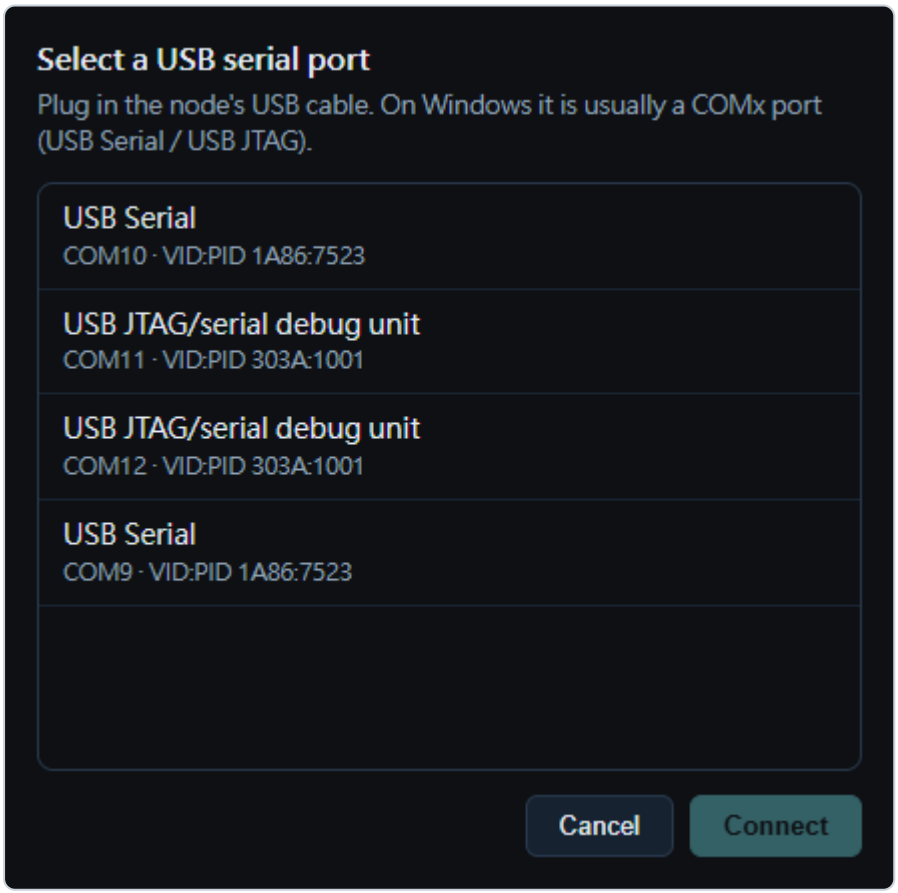
Area	What it does
Top right	Language (English / 繁體中文), connection status, **Connect (USB)**, **Connect (BLE)**, **Disconnect**
Left menu	The settings pages. Once connected, pages that don't apply to the product are hidden (e.g. SQS sensor nodes have no RS485 pages)
Centre	Settings for the current page
Bottom	**Save device & channel settings**, **Reboot**
Right, "Activity"	The result of every action — what was read, what was saved, any error. Look here first when something seems wrong

The `‹` `›` buttons collapse the menu and the activity panel.

---

4. Connecting over USB (recommended)

1. Plug the node into the PC with a USB-C cable. **Use a USB port on the PC itself**, not a USB hub (see section 8 for why).
2. Click **Connect (USB)** at the top right.
3. The "Select a USB serial port" window opens. SQC485I / SQS nodes appear as **USB JTAG/serial debug unit**, VID:PID **303A:1001**. Select it and click **Connect**.



> If several COM ports are listed, unplug the node and see which one disappears.

4. After about 1–5 seconds the status turns **Connected**, and the Activity panel shows:
   - `Node !xxxxxxxx` (node ID)
   - `read channel …`, `read role …`, `read region …` (settings read from the node)
   - `Node capability: … fw x.y.z` (product and firmware version)
   - `read config → …` (current device mode)

   The node ID and firmware version are also shown at the top of the page.
5. When done, click **Disconnect** before unplugging.

> On connect, the app sets the node's clock from the PC (the node has no clock battery); the Activity panel shows "Synced device time from this browser". No settings are changed.

---

5. Connecting over Bluetooth

Bluetooth is handy when the node is already installed and hard to reach with a cable. On Windows, **pair the node in Windows Settings once first**; after that the app can connect.

Before you start: the node's Bluetooth must be on. From firmware 1.2.5 the factory default is **Bluetooth disabled** — connect over USB first and, on the Bluetooth page, choose "Fixed PIN" and set a 6-digit PIN (see 6.4).

First time: pair in Windows

1. Open Windows **Settings** → **Bluetooth & devices** → **Add device** → **Bluetooth**.
2. Pick the node, named `Meshtastic_xxxx` (xxxx = last 4 characters of the node ID).
3. Enter the node's 6-digit PIN and click **Connect**; wait for "Paired" / "Your device is ready to go".

> Without this step the app connects for a few seconds and then drops, with "Connect failed: Device connection lost" in the Activity panel.

Every time after that

1. Click **Connect (BLE)**.
2. The "Select a Bluetooth node" window keeps scanning for nearby nodes. Select yours and click **Connect**.
3. From here on, everything works as over USB.

Changing the PIN or retiring a PC: *Settings → Bluetooth & devices → Devices →* `Meshtastic_xxxx` → **Remove device**. After the node's PIN changes, remove the old pairing and pair again.

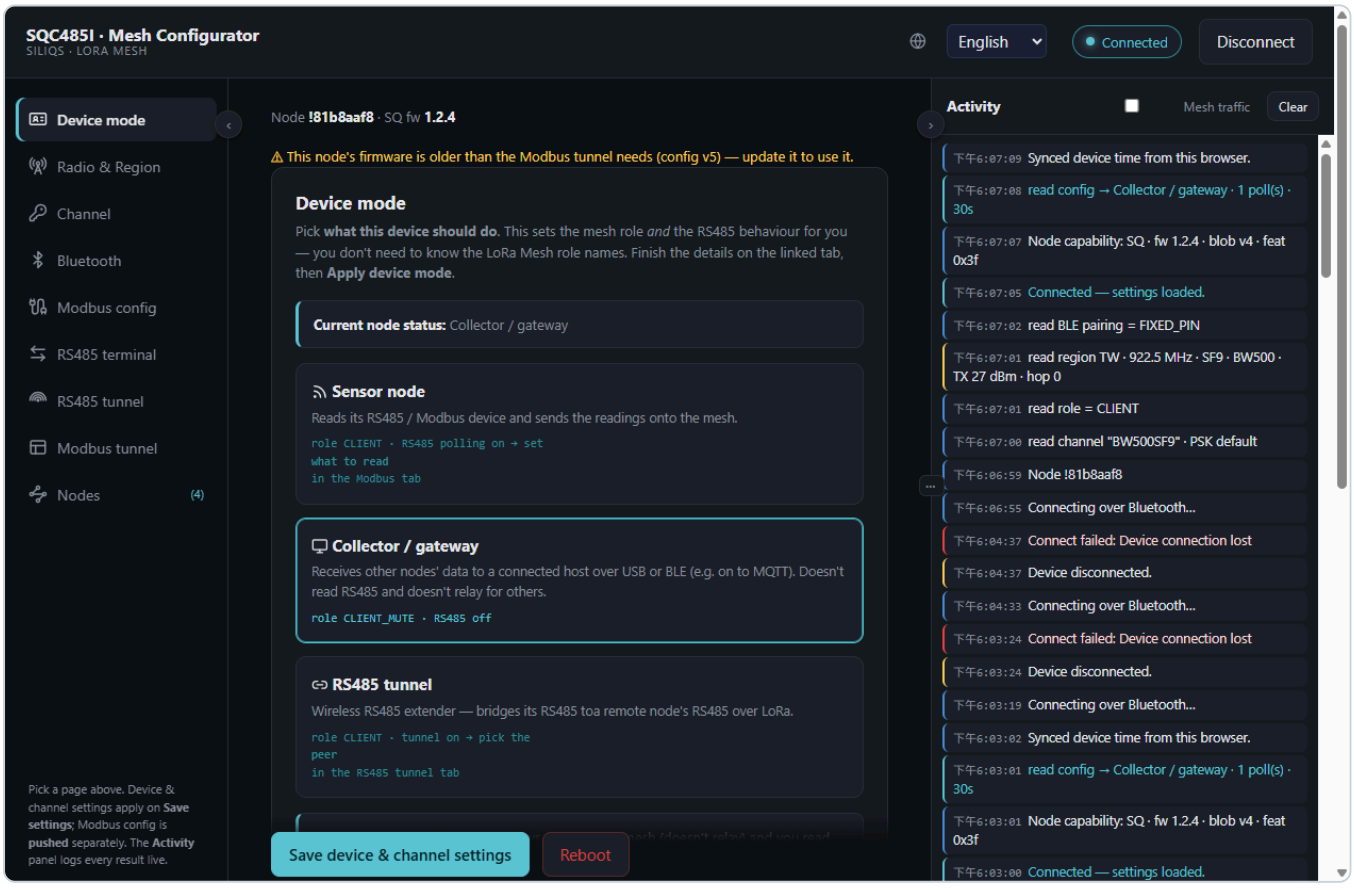
> **Note (ESP32-C3 models):** starting Bluetooth at boot draws extra power; on a weak supply the node may keep rebooting. Prefer USB for configuration.

6. The pages

Each page has its own apply button, separate from **Save device & channel settings** at the bottom:

Page	Saved with
Device mode	**Apply device mode** on that page
Radio & region, Channel	**Save device & channel settings** at the bottom
Bluetooth	**Apply Bluetooth settings** / **Set BLE power**
Modbus config	**Push Modbus config to device**
RS485 tunnel	**Apply & save to device**
Modbus tunnel	**Save Modbus tunnel config to device**
Telemetry (sensor nodes)	**Set interval on device**

6.1 Device mode

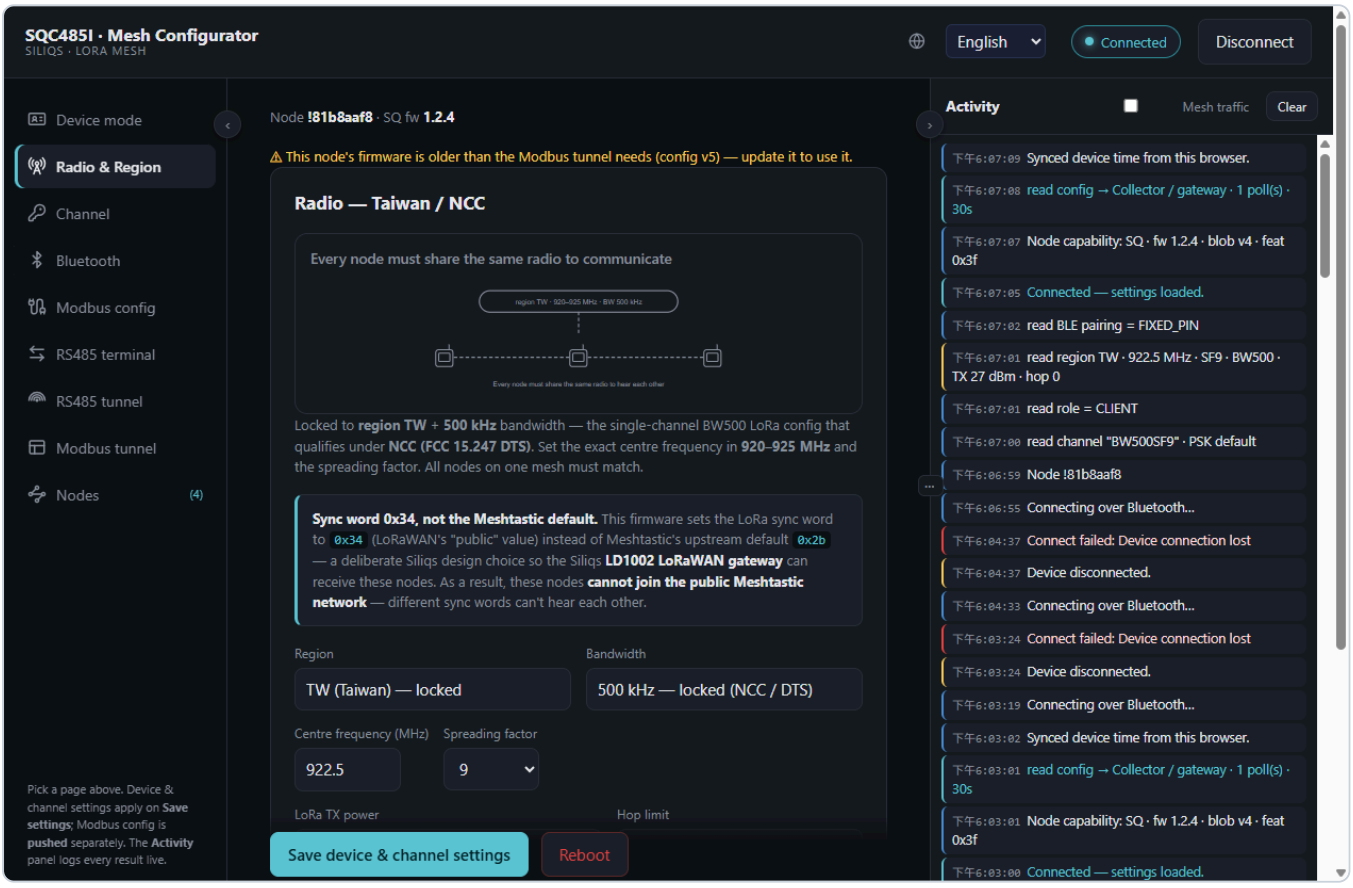


Choose what this node is for. "Current node status" at the top is the node's **current** state, read back from the device.

- **Sensor node** — reads its own RS485 / Modbus device and sends the readings onto the mesh. Pick a sensor behaviour:
  - Relay (default) — send readings and relay for the mesh
  - Mute — send readings, do not relay
  - Deep sleep · non-confirmed / Deep sleep · confirmed — wake, send, sleep (saves power)
- **Collector / gateway** — hands other nodes' data to the connected PC over USB or BLE (e.g. on to MQTT). Doesn't read RS485 and doesn't relay.
- **RS485 tunnel** — joins two nodes' RS485 buses over LoRa into one long wire (see 6.7).

Click **Apply device mode**.

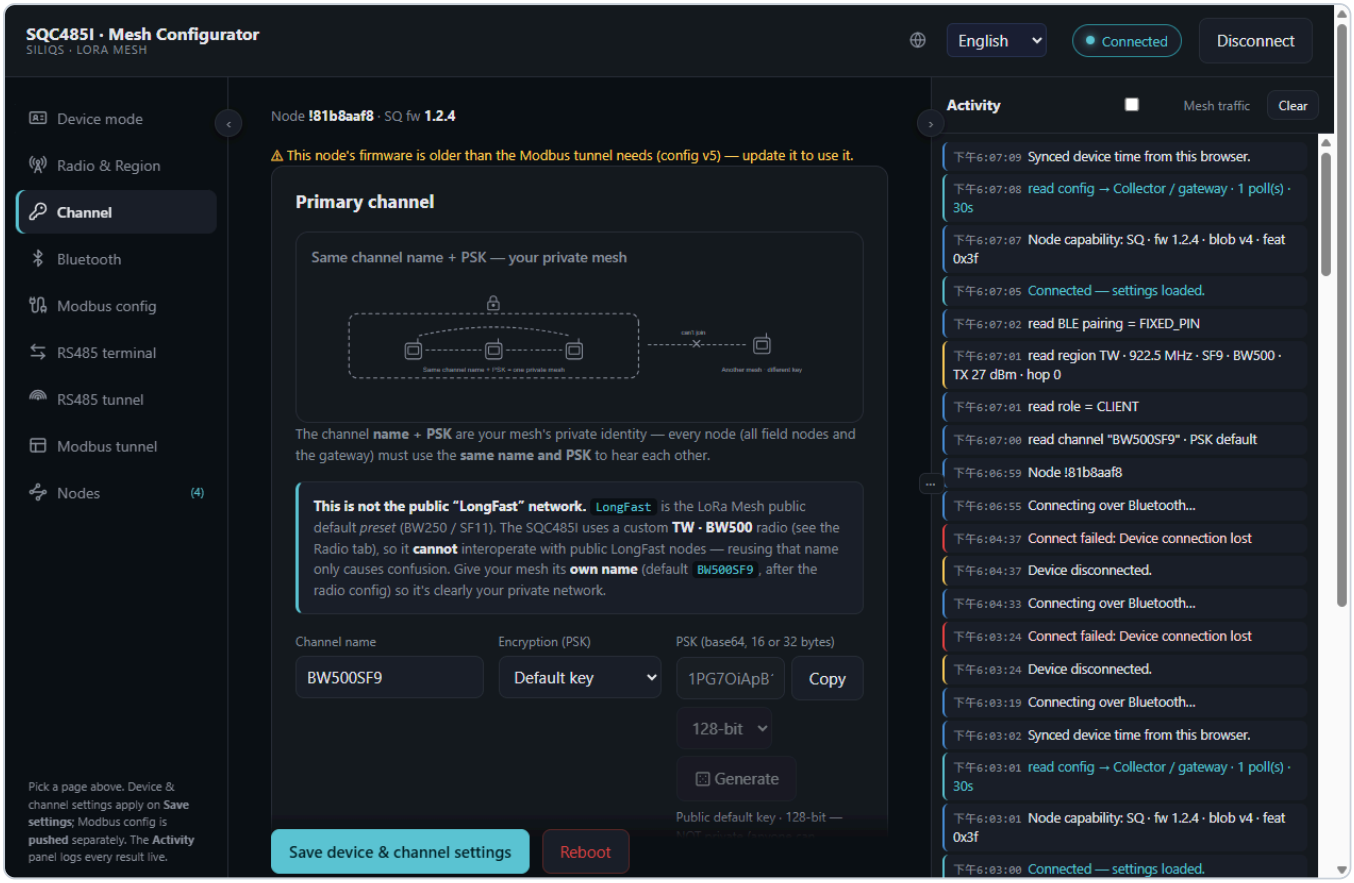
6.2 Radio & region



- Region is fixed to **TW** and bandwidth to **500 kHz** (NCC compliant).
- You can set: **centre frequency** (920–925 MHz), **spreading factor** (SF7–SF12), **LoRa TX power**, **hop limit**.
- **Every node in one mesh must use exactly the same radio settings**, or they can't hear each other.
- This firmware uses sync word `0x34`, not the public Meshtastic default, so these nodes **do not** join the public Meshtastic network (by design).

Click **Save device & channel settings**. Some radio changes need a reboot to take effect.

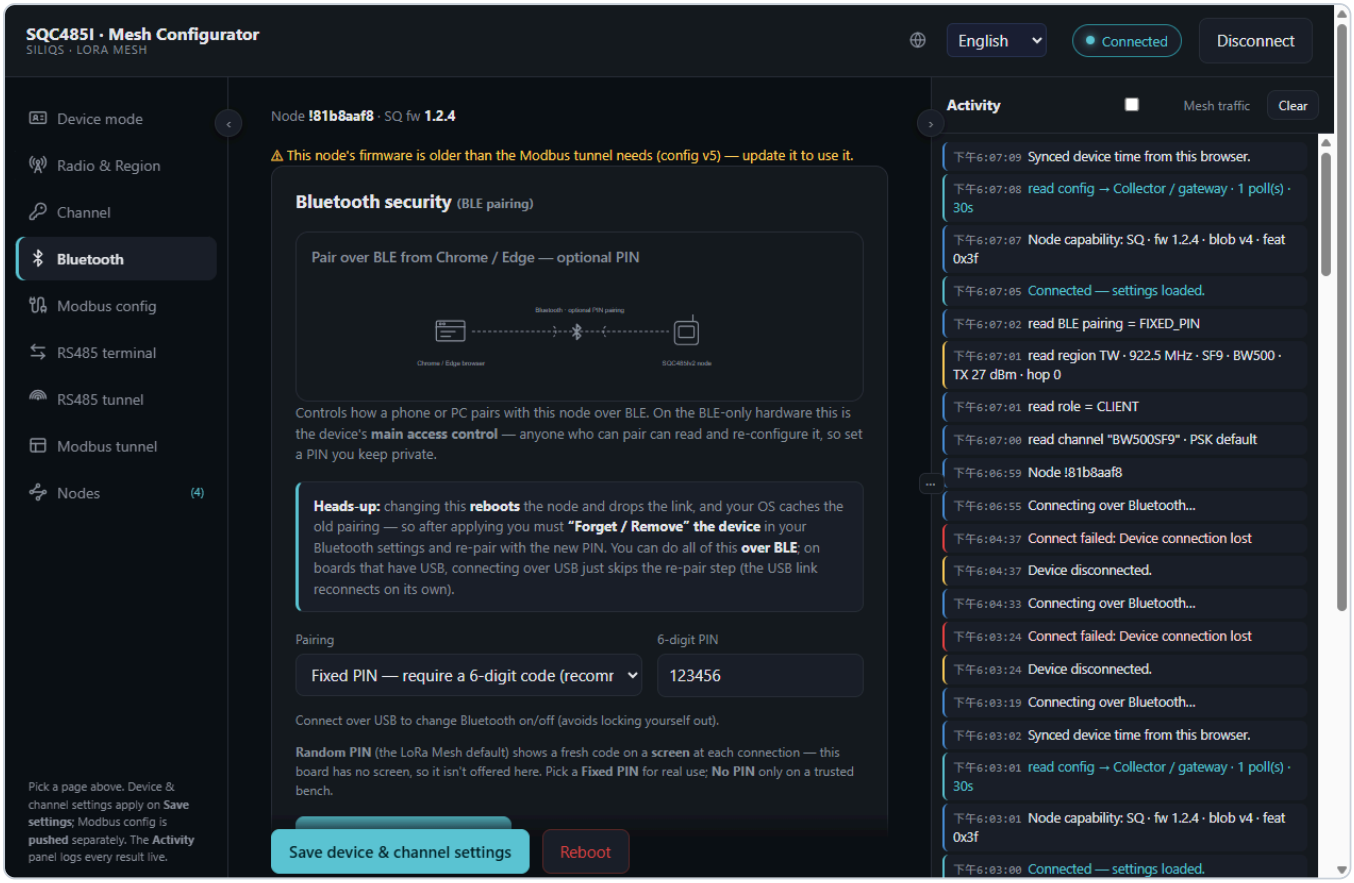
6.3 Channel



- The **channel name** and **encryption key (PSK)** are your mesh's identity — every node, including the gateway, must use the same values.
- Encryption:
  - **Default key** — a publicly known key; for bench testing only.
  - **Custom (base64)** — use this in real deployments. **Generate** creates a new key; **Copy** it and paste the same key into every other node.
  - **None (open)** — no encryption.
- "Name it BW500SF9" names the channel after the current radio settings.

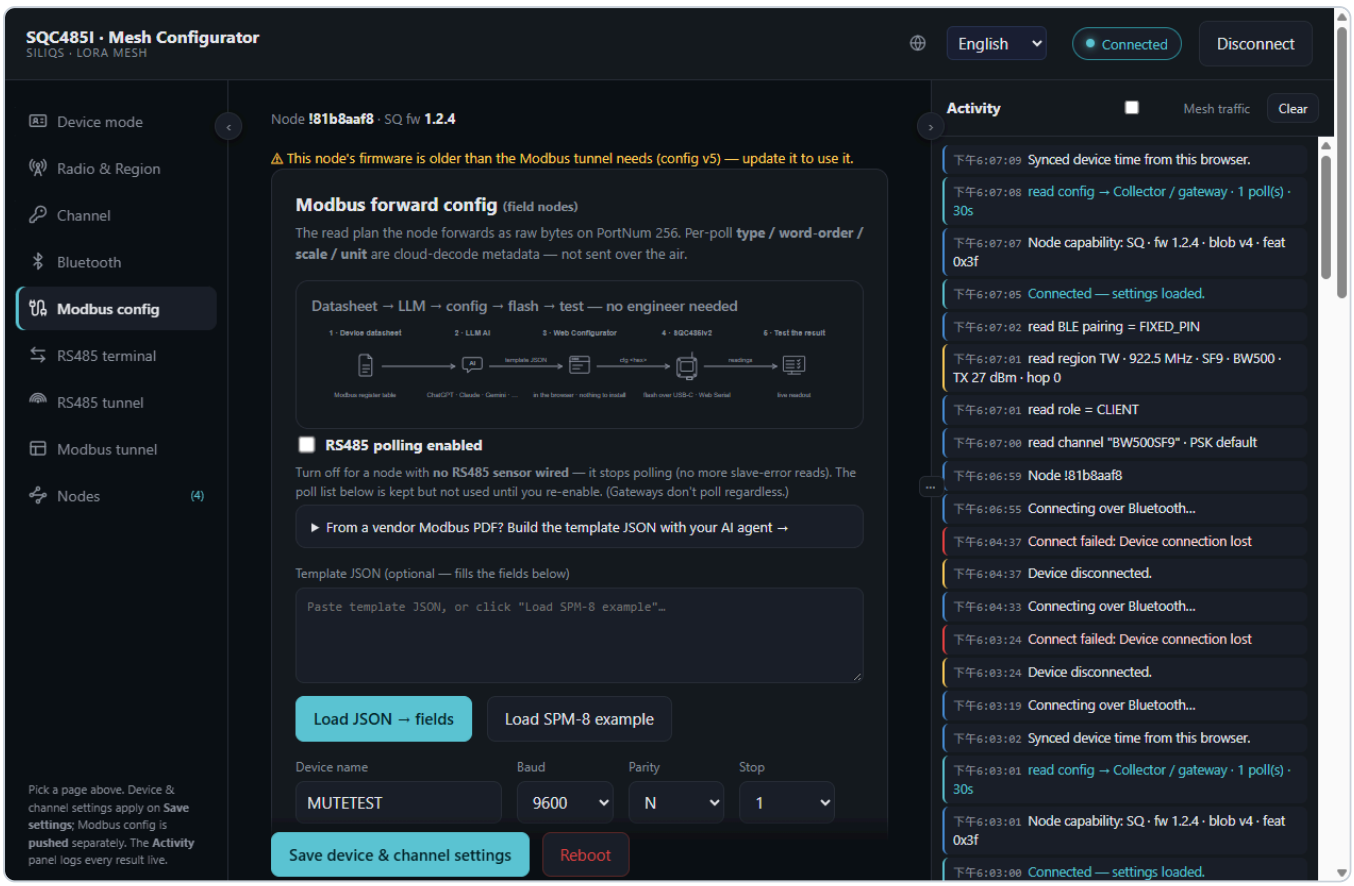
Click **Save device & channel settings**.

6.4 Bluetooth



- **Pairing:**
  - **Fixed PIN** (recommended) — a 6-digit PIN is required to pair.
  - **No PIN** — anyone nearby can connect; use only on a trusted bench.
  - **Disable Bluetooth** — USB only.
- **Apply Bluetooth settings reboots the node**. If the PC was paired before, remove the old pairing in Windows *Settings → Bluetooth & devices*, then pair again with the new PIN.
- **BLE transmit power:** higher means more range and more battery use. **Set BLE power** applies immediately, no reboot.

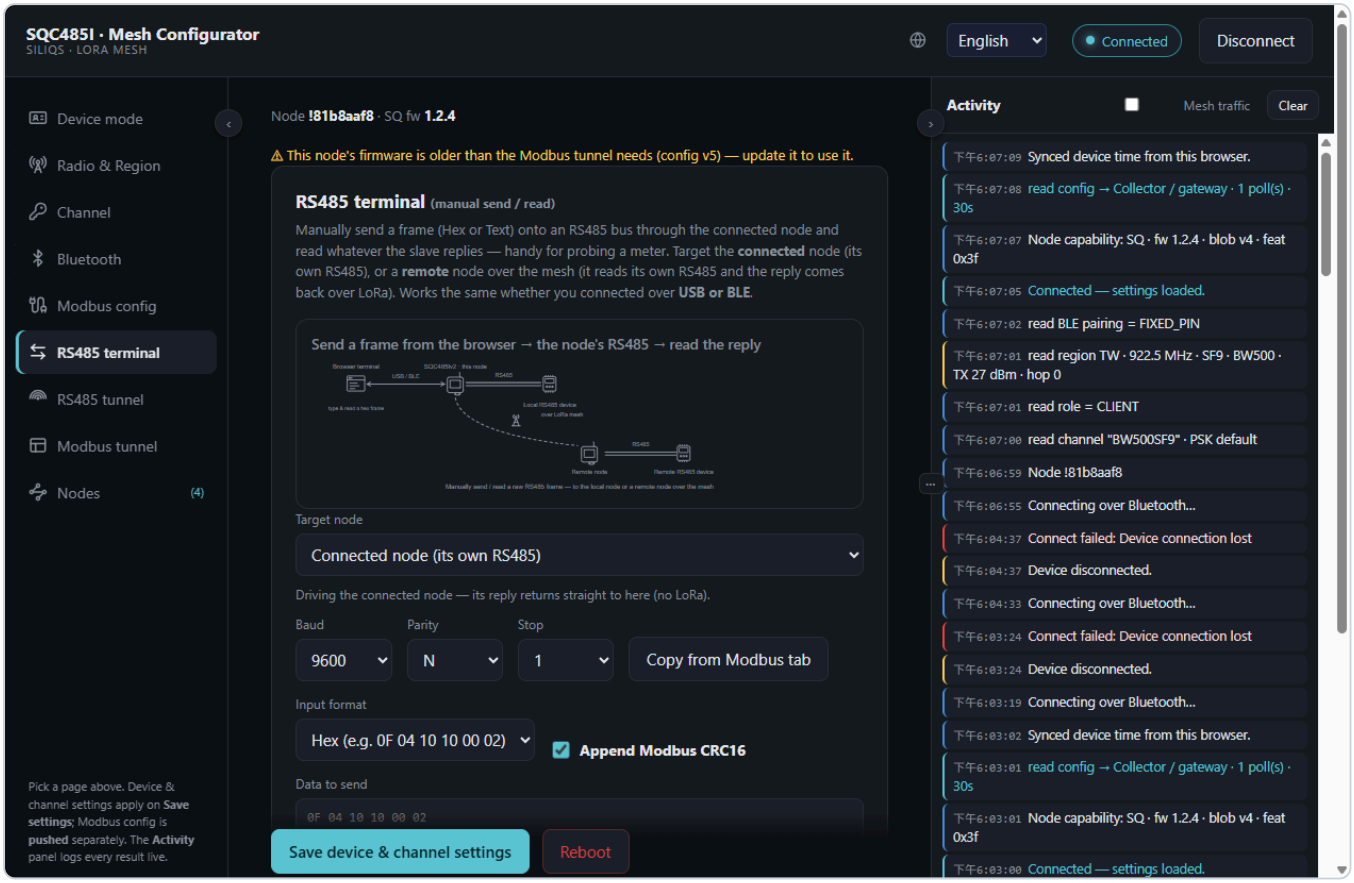
6.5 Modbus config (SQC485I)



Defines which Modbus registers the node reads over RS485, and how often.

1. Tick **RS485 polling enabled** (untick it on a node with no RS485 device wired).
2. Set **device name, baud, parity, stop bits, uplink interval** and the destination (broadcast by default).
3. Add rows to the poll table (slave, function code, register, count, type, word order, scale, unit).
    - Have the vendor's Modbus PDF? Click **Copy AI prompt**, give it to an AI assistant (ChatGPT, Claude…) together with the PDF, paste the returned template JSON and click **Load JSON → fields**.
    - **Load SPM-8 example** loads a power-meter example for reference.
4. Click **Push Modbus config to device**.
5. Use **Test now** to make the node read once immediately and check the values.

6.6 RS485 terminal

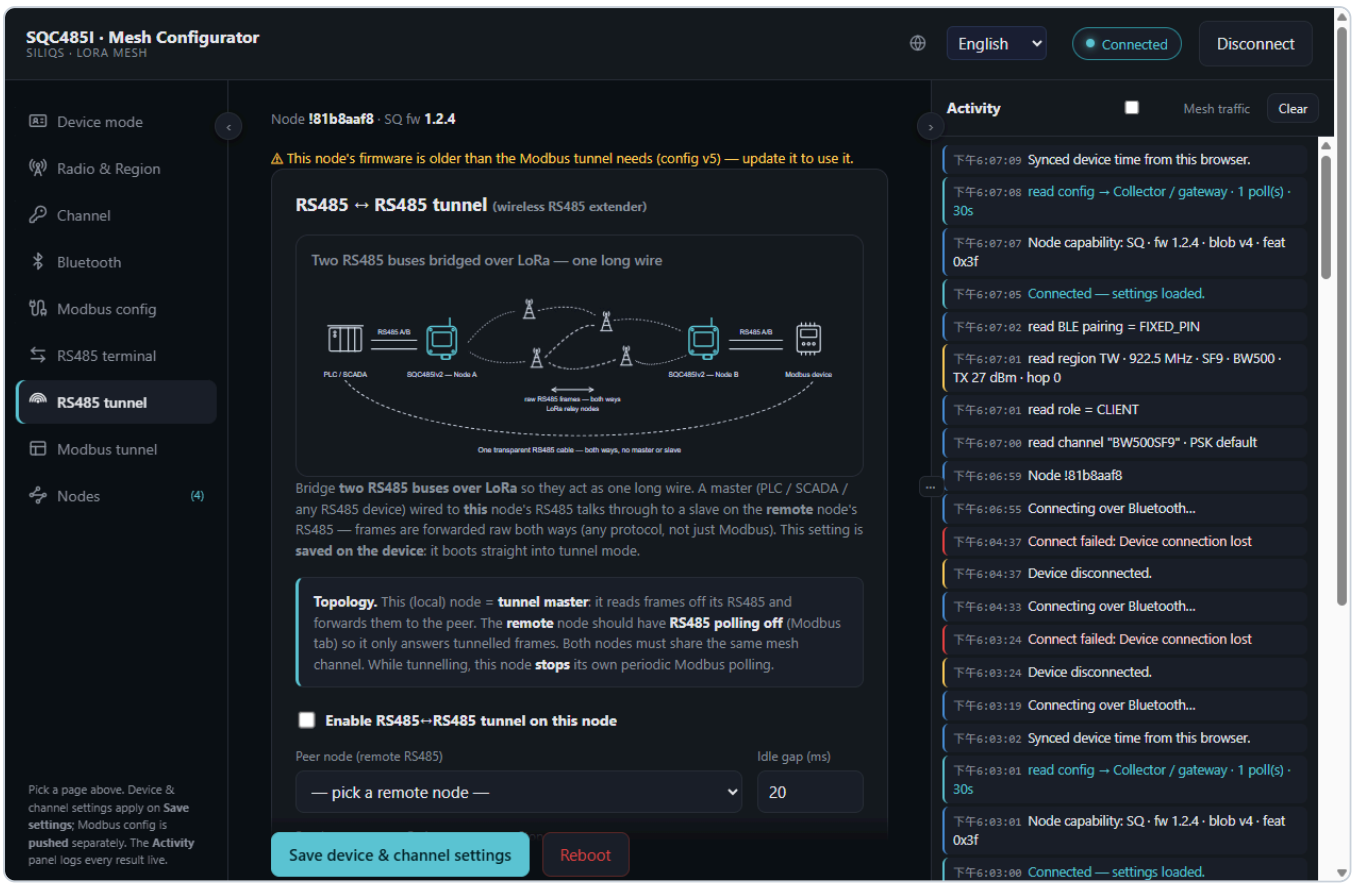


Send one frame onto RS485 by hand and see what the slave replies — useful for checking a meter's wiring and address.

- **Target node:** the connected node (its own RS485) or a remote node on the mesh (the reply comes back over LoRa).
- Set baud etc. (**Copy from Modbus tab** fills them in).
- Choose Hex or Text input. Hex can append a Modbus CRC16 automatically.
- Click **Send to RS485**; the monitor below shows TX (sent) and RX (reply).

This page saves **nothing**.

6.7 RS485 tunnel

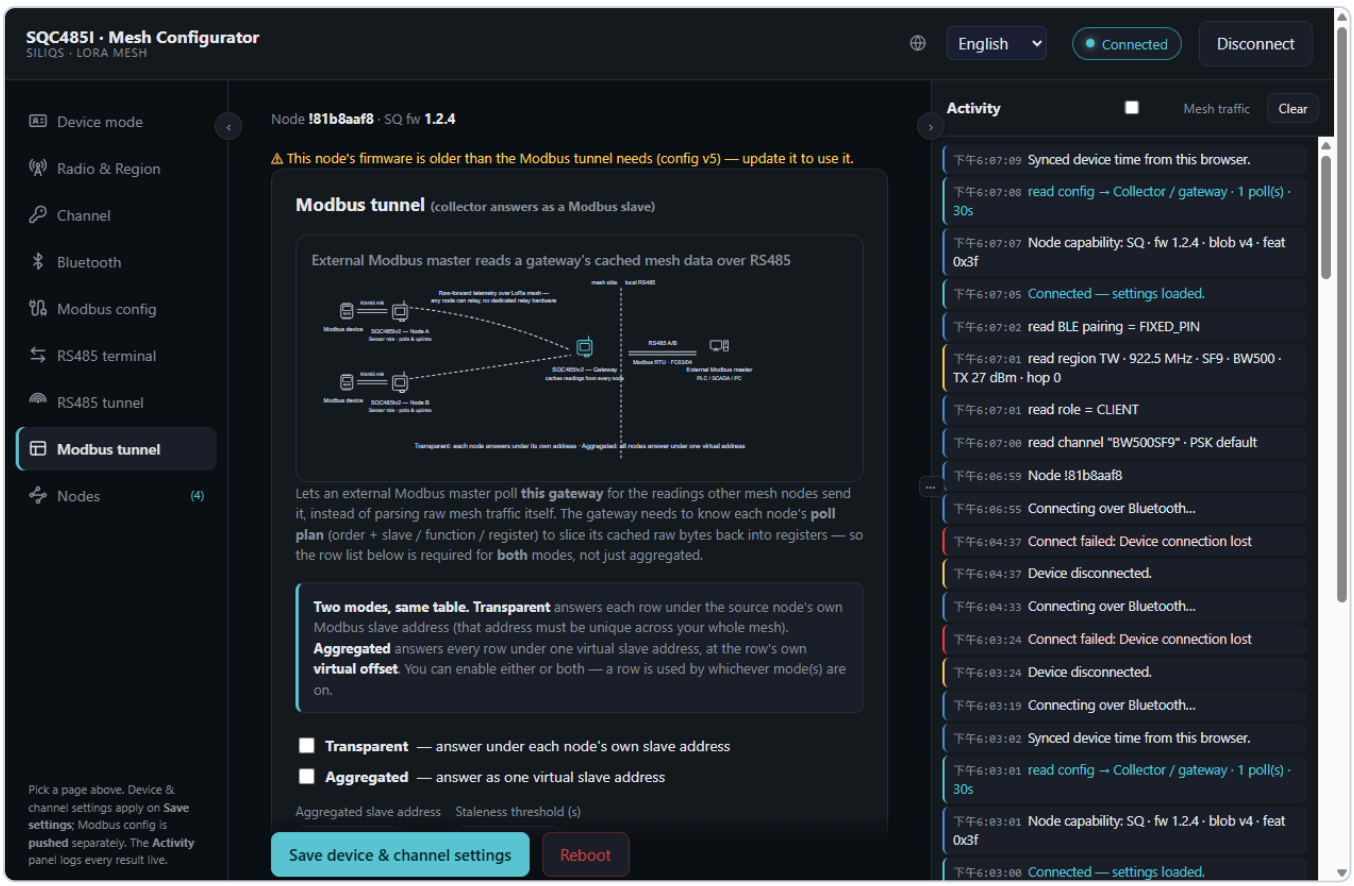


Bridges two nodes' RS485 buses over LoRa so they act as one long wire. A master (PLC, SCADA…) wired to this node can talk to a slave on the remote node's RS485, with any protocol.

1. Tick **Enable RS485↔RS485 tunnel on this node**.
2. Pick the **peer node**, set the idle gap and serial parameters.
3. Click **Apply & save to device** — it takes effect immediately.
4. On the remote node, **turn RS485 polling off** (Modbus config page).

To stop, untick and apply again.

6.8 Modbus tunnel (collector answers Modbus queries)

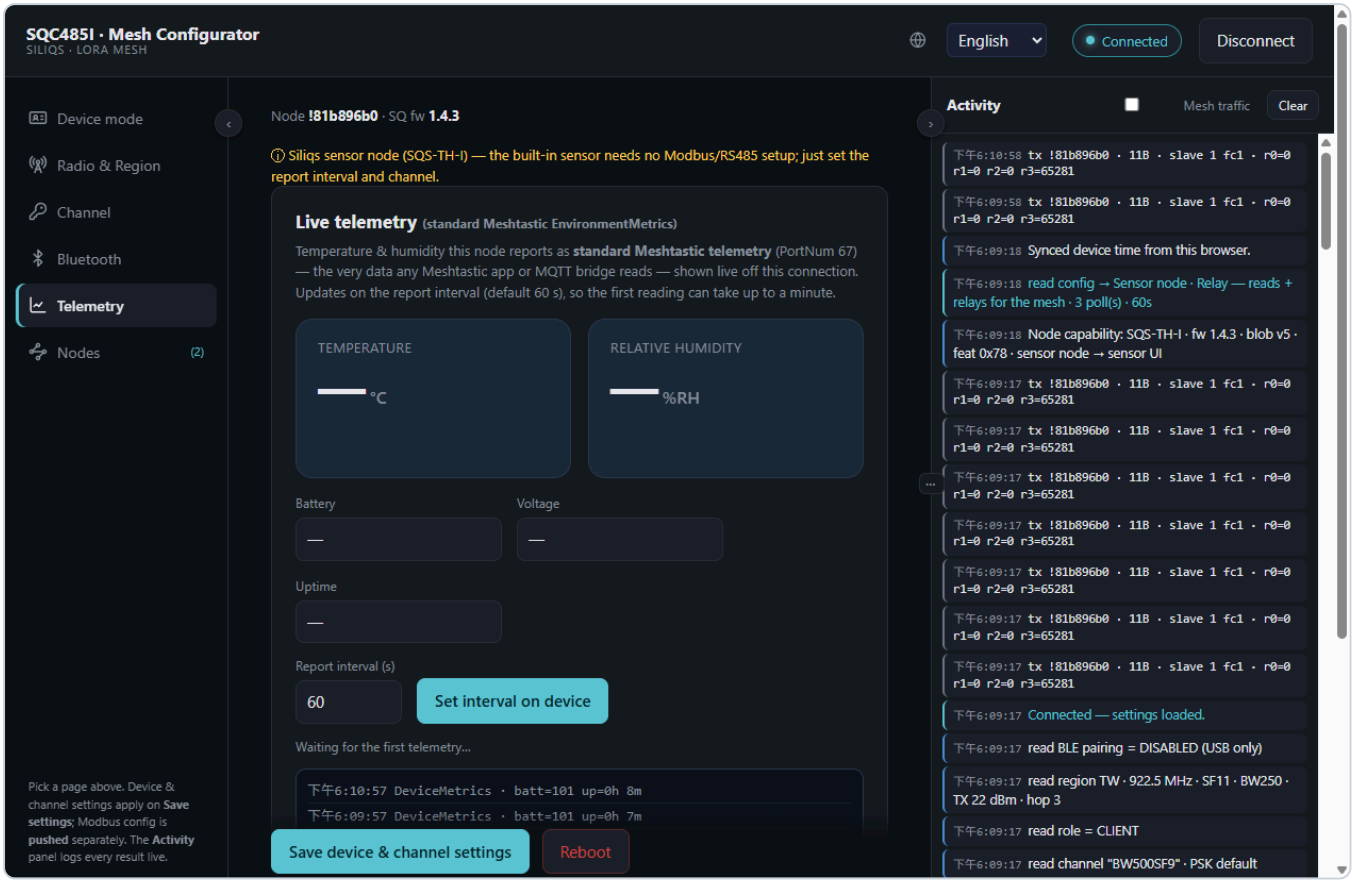


Lets an external Modbus master query the **gateway** over RS485 for the data other nodes report, without parsing mesh traffic itself.

- **Transparent** — answers under each source node's own slave address (must be unique across the mesh).
- **Aggregated** — answers everything under one virtual slave address at different offsets.
- List each source node's polls in the table and click **Save Modbus tunnel config to device**.

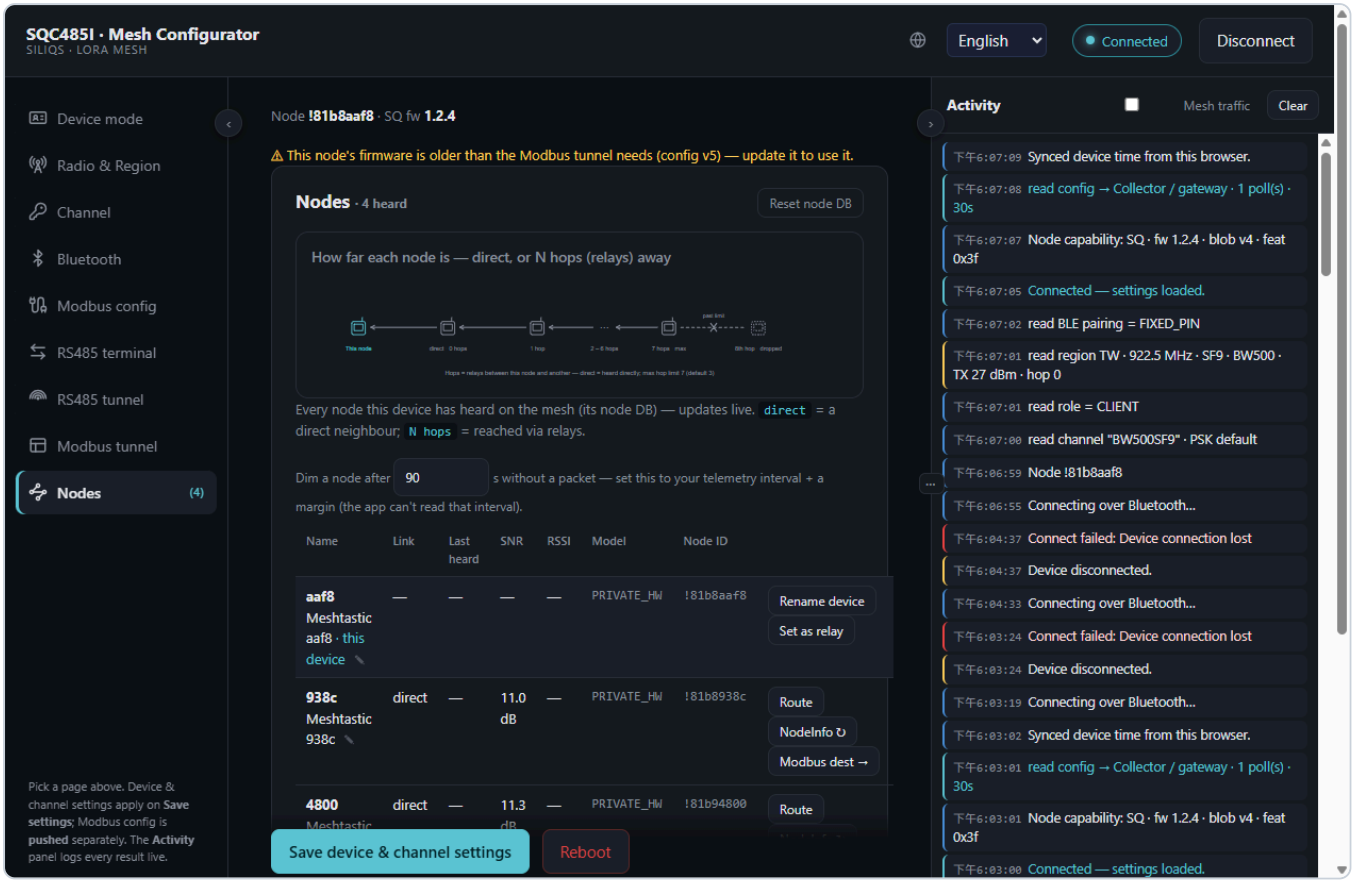
> Needs newer firmware (config v5). If the page shows "this node's firmware is older than the Modbus tunnel needs", update the firmware first.

6.9 Telemetry (SQS sensor nodes)



Shown only for SQS sensor nodes (e.g. SQS-TH-I temperature & humidity). Displays live temperature, humidity, battery, voltage and uptime, and sets the **report interval** (**Set interval on device**). The first reading can take up to one report interval.

6.10 Nodes



Every node this device has heard on the mesh: link (direct or N hops), last heard, SNR, RSSI, model, node ID.

- **Rename device** (on the "this device" row) — sets the node's real name (long name plus a short name of up to 4 characters), which is shared across the mesh. An input window opens:



- ✎ next to a name — a nickname kept on this PC only; the node is not changed.
- **Route** — trace the path to that node. **Get NodeInfo** — ask it to report its name and model now.
- **Reset node DB** — clears every node this device remembers so it learns them again.

6.11 Save and reboot

- **Save device & channel settings** — writes the Radio & region and Channel pages.
- **Reboot** — restarts the node. Some radio settings need it. The USB link drops; click **Connect (USB)** again.

---

7. Working offline

The Windows app contains the whole configurator, so **no internet connection is needed** — it works on sites without network access.

---

8. Troubleshooting

Problem	What to do
The node's COM port isn't in the list	Try a cable you know carries data; try another USB port; check the node has power. Windows 10/11 installs the driver automatically.
Disconnects a few seconds after connecting, again and again	Usually **not enough power**. Plug the node straight into a USB port on the PC, not through a hub (especially an unpowered one). ESP32-C3 models draw extra power when Bluetooth starts at boot and keep rebooting on a weak supply.
"Connection timed out — try again."	The node didn't answer. Wait a few seconds and connect again; if it still fails, unplug and re-plug the USB cable (restarts the node) and connect again.
"The port is already open"	Another program is using that COM port (another serial tool, another app window). Close it and retry.
Bluetooth connects, then drops within seconds (Device connection lost)	The node isn't paired in Windows yet. Pair it with the PIN in *Settings → Bluetooth & devices* (section 5), then connect from the app.
Bluetooth scan finds nothing	Check the node's Bluetooth is on (6.4), the PC's Bluetooth is on, and you are close enough.
Bluetooth pairing fails	Check the PIN; remove the old pairing in *Settings → Bluetooth & devices* and pair again.
A change doesn't seem to apply	Check the Activity panel for errors; make sure you used that page's button (table in section 6); radio changes may need **Reboot**.
SmartScreen blocks the installer	See section 2: **More info → Run anyway**.

---

9. Release notes

Version	Date	Notes
1.0.2	2026-11	Bluetooth chooser now reminds Windows users to pair the node in Settings first
1.0.1	2026-11	Fixed: input window for "Rename device" and node nicknames; reconnecting after Disconnect; COM port not released after a connection timeout
1.0.0	2026-11	First release

Siliqs · https://mesh.siliqs.net