

Read this before touching anything

**The inverter holds its last command forever.** It has no meter-timeout. If the controller stops talking, the battery keeps charging or discharging at whatever it was last told — indefinitely.

**Measured:** a controller went silent mid-command and the inverter held **5 kW of discharge for 113 seconds** until a person intervened.

**If anything looks wrong at any point: STOP THE ADD-ON.** That commands 0 W and the hardware watchdog holds it there. Nothing is at risk while you think.

## Order of operations

Do not reorder. Each step assumes the previous one passed.

| Step | Action                                            | Gate to pass          |
|------|---------------------------------------------------|-----------------------|
| 0    | Pre-visit checks (client, warranty, HA type)      | All boxes ticked      |
| 1    | Survey & photograph existing wiring               | Photos taken          |
| 2    | Bring system to 0 W, disconnect vendor controller | Battery idle          |
| 3    | Wire and power the T-CAN485                       | Board online          |
| 4    | Flash the ESP32 firmware                          | Entities appear in HA |
| 5    | Install & configure the add-on                    | All checks green      |
| 6    | Commissioning gates 1–6                           | Every gate passes     |
| 7    | E-stop: fit & test, or get signature              | Tested or signed      |
| 8    | Prove a maintenance cycle                         | Phases observed       |
| 9    | Handover & sign-off                               | Sheet complete        |

## The safety model

| Layer       | Covers                                                             | Where it lives      |
|-------------|--------------------------------------------------------------------|---------------------|
| 1 Watchdog  | Controller alive but silent → 0 W, and keeps writing it            | ESP32 firmware      |
| 2 Wind-down | Planned firmware update → 0 W before it starts                     | ESP32 firmware      |
| 3 E-stop    | <b>The HA machine itself dying</b> → 0 W after 30 s of bus silence | Optional Pi + RS485 |

**Only layer 3 covers the host dying.** Nothing running on the HA machine can cover its own death. On a site without the e-stop, a failed host leaves the battery latched until someone intervenes manually. See Step 7.

Step 0 — Before you travel

**Confirm with the client**

- ☐ Inverter is **GoodWe ES / BP family**
- ☐ Vendor controller may be **disconnected**
- ☐ Warranty / installer agreement acknowledged **in writing**
- ☐ Grid rules checked (BE: Synergrid C10/11)
- ☐ Vendor subscription — cancel only **after** sign-off

**Confirm on their system**

- ☐ HA is **OS or Supervised** (Settings → System → Repairs → : → System information)
- ☐ Grid power sensor exists, updates  $\leq 10$  s
- ☐ Note its entity id and update rate
- ☐ E-stop in the van, or signature form printed

**Stop conditions**

If HA is **Container** or **Core**, add-ons cannot be installed and this product cannot go in. If the grid sensor updates slower than ~10 s, the default tuning is wrong for the site — do not proceed without adjusting `gain` .

Never do these

| Never                                                              | Because                                                                                               |
|--------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| Flash third-party firmware on the inverter's WiFi dongle           | Common image targets different hardware — bricks it                                                   |
| Long-press (3–5 s) the inverter's WiFi Reset button                | Factory-resets the dongle, loses network access. Short press (~1 s) is safe                           |
| Connect our controller while the vendor box is still attached      | Two masters on one bus — contradictory commands                                                       |
| "Compensate" for an inverted meter sign in the tuning              | Fix <code>meter_invert</code> or the wiring. A wrong sign drives the grid away from zero at full gain |
| Raise the e-stop's 30 s threshold to stop it firing during updates | Firing during an update is correct behaviour                                                          |

## Steps 1–3 — Hardware

**Qualified persons only.** Mains AC and battery DC are present. The meter bus is low-voltage but sits inside an energised installation.

### 1 Survey and record

1. **Photograph** the meter-port wiring before touching it.
2. Identify the RS485 pair (A/B) running to the vendor controller.
3. Record inverter model + serial, battery capacity (Ah) and nominal voltage.

**VERIFY** — photos on file; A/B pair positively identified.

### 2 Disconnect the vendor controller

1. Bring the system to **0 W** — battery neither charging nor discharging (check the inverter display).
2. Disconnect the vendor controller from the meter port.
3. Leave it physically mounted but unplugged — that keeps the install reversible and gives you a rollback story.

**VERIFY** — battery at 0 W, vendor box disconnected.

**IF NOT** — do not continue while the vendor box can still write to the bus.

### 3 Fit and power the T-CAN485

1. Wire **A→A, B→B** to the meter port, plus its own supply.
2. If fitting the e-stop, connect its USB-RS485 adapter to the **same pair, in parallel**.
3. Power the T-CAN485 **last**.

**VERIFY** — board joins WiFi and appears in ESPHome.

**IF NOT** — swapped A/B gives a **completely silent bus, not an error**. Swap and retry before suspecting anything else.

**It transmits within seconds of boot.** Never power it up while the vendor controller is connected.

Step 4 — Flash the firmware

4 ESPHome

- 1. Open the **ESPHome Device Builder** add-on on the client's HA.
- 2. Create a device from `firmware/goodwe-master.yaml`.
- 3. Set the **three substitutions** at the top — and nothing else.
- 4. Add the client's WiFi to ESPHome's `secrets.yaml`.
- 5. Install: first flash by USB, everything after that over the air.

**VERIFY** — device shows **Online**, and `number.<node>_goodwe_setpoint_w` exists in HA.

| Substitution | Set to                                             | Why it matters                                                                                            |
|--------------|----------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| name         | Node name, e.g. <code>goodwe-master</code>         | <b>Write it on the sheet.</b> Changing it later renames every entity in HA and silently breaks the add-on |
| max_w        | Inverter continuous rating, e.g. <code>5000</code> | Hard firmware limit, independent of HA. Last line of defence against a controller bug                     |
| wd_ms        | <code>30000</code>                                 | Watchdog timeout. Must stay well above the add-on heartbeat (10 s)                                        |

ESPHome Device Builder

Maa en bouw uw slimme apparaten en automatiseringen

Apparaten zoeken...

Filters

5 devices

Meerdere selecteren

bureau

bureau.yaml

Bewerken

Offline

goodwe-master

goodwe-master.yaml

Bewerken

Offline

goodwe-master-stage1

goodwe-master-stage1.yaml

Bewerken

Online

pressure sensor

pressure-sensor.yaml

Bewerken

Offline

warmtepomp zonne energie

esphome-web-7f9f98.yaml

Bewerken

Online

+ Apparaat aanmaken

Device Builder 1.12.0

ESPHome 2026.8.0

ESPHome Device Builder — the node must read **Online** before you continue. (Reference bench shown.)

**Testing firmware changes:** an update that lands new shutdown-path code still runs the **old** code on the way out. If you change wind-down or watchdog behaviour, **upload twice** before believing a test result.

Step 5 — Install and configure the add-on

5 Add-on

- 1. Settings → Add-ons → Add-on store → : → **Repositories** → add the repository URL.
- 2. Install **GoodWe RS485 Controller**. **Do not start it yet.**
- 3. Open **Configuration** and fill in the table below.
- 4. Start the add-on, open its **Web UI**.
- 5. Work down the **Commissioning** list until every line is green.

**VERIFY** — banner is green or amber, no red lines.

**IF NOT** — the banner names the failing item. Copy entity ids from **Developer Tools** → **States**. Do not type them from memory.

| Option                              | Set to                                                                                         |
|-------------------------------------|------------------------------------------------------------------------------------------------|
| meter_entity                        | Client's grid power sensor. <b>Positive must mean importing</b> — else set meter_invert        |
| soc_entity                          | sensor.<node>_goodwe_battery_soc — <b>the ESP32's own read</b> , never the cloud/dongle sensor |
| batt_entity                         | sensor.<node>_goodwe_inverter_ac_power                                                         |
| setpoint_entity                     | number.<node>_goodwe_setpoint_w                                                                |
| max_w                               | <b>1000</b> for commissioning. Raise at the end of Step 6                                      |
| estop_fitted                        | true only if you actually fitted one                                                           |
| peak_forecast_entity                | Capacity-tariff sites only, else leave empty                                                   |
| price_now_entity / price_avg_entity | Dynamic-tariff sites only, else leave empty                                                    |

Entity ids are the #1 install error

HA prefixes entity ids with the device's **area** at creation time, so the same firmware yields sensor.goodwe\_master\_... on one site and sensor.cellar\_goodwe\_master\_... on another. **A wrong entity id is not an error anywhere in HA** — it simply never produces a value. On the reference system two safety alarms pointed at non-existent entities and were dead for a day while reading "on".

# Reading the Web UI

Two states you must tell apart at a glance. The banner is the summary; the **Commissioning** list is the detail.

GoodWe RS485 Controller

NOT READY — grid power: sensor.p1\_meter\_active\_power is missing or not numeric

|                   |                |
|-------------------|----------------|
| Grid              | —              |
| Battery           | 884 W          |
| State of charge   | 78 %           |
| Commanded         | 0 W            |
| Why               | inputs-missing |
| Maintenance phase | idle           |
| Maintenance due   | now            |
| Cheap window      | no             |

Commissioning

- ✗ grid power: sensor.p1\_meter\_active\_power is missing or not numeric
- ✓ battery SoC: sensor.goodwe\_master\_goodwe\_battery\_soc = 78
- ✓ battery power: sensor.goodwe\_master\_goodwe\_inverter\_ac\_power = 884
- ✓ setpoint number: sensor.goodwe\_master\_goodwe\_setpoint\_w (range -5000...5000 W)
- ! no e-stop fitted: if this host dies the battery stays latched at its last command

Start / stop control

Force maintenance cycle

Abort maintenance

**HEALTHY.** Green banner. All checks ✓. *Why* reads `tracking`. **Grid near zero** and the commanded figure **resting** between updates — a command that moves every single cycle means the deadband is too small for this meter.

GoodWe RS485 Controller

NOT READY — grid power: sensor.p1\_meter\_active\_power is missing or not numeric

|                   |                |
|-------------------|----------------|
| Grid              | —              |
| Battery           | 884 W          |
| State of charge   | 78 %           |
| Commanded         | 0 W            |
| Why               | inputs-missing |
| Maintenance phase | idle           |
| Maintenance due   | now            |
| Cheap window      | no             |

Commissioning

- ✗ grid power: sensor.p1\_meter\_active\_power is missing or not numeric
- ✓ battery SoC: sensor.goodwe\_master\_goodwe\_battery\_soc = 78
- ✓ battery power: sensor.goodwe\_master\_goodwe\_inverter\_ac\_power = 884
- ✓ setpoint number: sensor.goodwe\_master\_goodwe\_setpoint\_w (range -5000...5000 W)
- ! no e-stop fitted: if this host dies the battery stays latched at its last command

**NOT READY.** Red = must fix before going further; the banner names the first failing item. Amber = permitted, but the client must have signed for it — here, no e-stop fitted. Note *Why* reads `inputs-missing` and **Commanded** has gone to **0 W**: with a source missing, the controller zeroes the inverter rather than guessing.

## Step 6 — Commissioning gates

Judge every gate on the **wire or the meter** — never on how Home Assistant looks.

| # | Do                                                                                             | Pass                                                                                      | If it fails                                                                    |
|---|------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| 1 | Add-on running, control <b>stopped</b>                                                         | Steady 0 W written, inverter idle, no errors                                              | Check wiring and Step 5                                                        |
| 2 | By hand (Developer Tools → <code>number.set_value</code> ): <b>+300 W</b> , then <b>−300 W</b> | Battery discharges ~300 W, then charges ~300 W, within ~10 s                              | Sign inverted → fix <code>batt_invert</code> /wiring, <b>not</b> the tuning    |
| 3 | Compare ESP32 readings to inverter display                                                     | SoC identical; power within ~5 %                                                          | Wrong entity, or dongle sensor used by mistake                                 |
| 4 | <b>Set +300 W, then STOP the add-on</b>                                                        | <b>Inverter reaches 0 W within ~30 s</b> and stays                                        | <b>STOP THE INSTALL.</b> Nothing here is safe without this                     |
| 5 | <code>max_w</code> =1000, start control                                                        | Grid settles within tens of watts of zero and <b>rests</b>                                | Hunting → raise <code>deadband_w</code> ; slow meter → lower <code>gain</code> |
| 6 | Switch on a kettle/oven (~2 kW)                                                                | Grid back near zero within ~20 s; command <b>stops</b> rising once the battery catches up | Command keeps climbing = <b>runaway</b> . Stop the add-on immediately          |

**THEN** — raise `max_w` to the value the site is sold with and repeat gate 6 once.

## Step 7 — The e-stop

### 7a If fitted

1. Connect the Pi to the same RS485 pair; power it.
2. Run: `python3 rs485_log.py --out /home/pi/bus.log --panic`
3. Confirm it prints `PANIC ARMED`.
4. **Test it:** with the battery at +300 W, cut power to the T-CAN485.

**VERIFY** — within ~35 s the log shows a `TX` line and the inverter goes to **0 W**.

**IF NOT** — check the adapter is on the correct pair and that the tap was started with `--panic`.

**Arm the panic write only while our controller owns the bus.** Pointed at a bus somebody else drives, it is unrequested interference.

**It will fire during every firmware update** — the upload silence exceeds 30 s. That is correct. Do not raise the threshold.

### 7b — If not fitted: client acknowledgement

Read this to the client and have them sign. Note it on the sign-off sheet.

**Without the RS485 e-stop:** if the Home Assistant machine fails, loses power, or its storage fails, the battery inverter will continue charging or discharging at whatever level it was last commanded, **indefinitely**, until someone intervenes manually. The inverter has no automatic fallback of its own. The e-stop is the only component that prevents this.

\_\_\_\_\_

Client name

\_\_\_\_\_

Signature

\_\_\_\_\_

Date

## Step 8 — Prove a maintenance cycle

### 8 Do not wait a month to find out it never fires

1. Web UI → **Force maintenance cycle**.
2. Watch the phase go `drain` → `charge` → `hold` → `idle`.
3. Confirm the inverter **exports** during drain, **charges** during charge, sits at **0 W** during hold.
4. Set `maintenance_enabled: true`.

**SHORT VERSION (~10 min)** if the client cannot spare hours: set `maintenance_soc_floor` just below current SoC, `maintenance_soc_target` just below that, `maintenance_hold_min: 5`. **Restore real values afterwards** and record it as a partial test.

Why it matters: the full charge is when the BMS **balances cells**; the deep discharge **recalibrates the coulomb counter**. Skipping it breaks nothing visibly — it degrades the pack over months, and the first symptom is a state-of-charge reading nobody can trust.

Step 9 — Handover & sign-off

Tell the client, in these words

- “This is the Web UI. Green means it is working.”
- “If anything looks wrong, press stop. That sets the battery to zero and it idles safely.”
- “Once a month it will deliberately empty and then fill the battery. That is normal and it keeps the battery healthy.”

Commissioning record

Site / client

Inverter model / serial

ESPHome node name

Add-on version / max\_w

Meter entity + update rate

- ☐ Gates 1–6 all passed on the wire or the meter
- ☐ Gate 4 (watchdog) personally observed reaching 0 W
- ☐ Maintenance cycle proven — full / partial (circle one)
- ☐ E-stop fitted & tested OR §7b signed
- ☐ Client shown the Web UI and the stop button
- ☐ Vendor subscription cancelled (only after the above)

Technician

Signature

Date

## Troubleshooting

| Symptom                                                   | Likely cause                                                | Do this                                                                        |
|-----------------------------------------------------------|-------------------------------------------------------------|--------------------------------------------------------------------------------|
| Web UI says <b>NOT READY</b>                              | Wrong entity id, or sensor unavailable                      | Banner names it. Copy the id from Developer Tools → States                     |
| Configured, but nothing moves                             | Control is stopped                                          | Press start; banner turns green                                                |
| Inverter does nothing, no errors                          | A/B swapped, or vendor box still connected                  | Swapped RS485 gives a <b>silent</b> bus, not an error                          |
| HA shows one setpoint, inverter does another              | Firmware <code>max_w</code> below add-on <code>max_w</code> | Firmware wins by design. Align them                                            |
| Log shows writes rejected (HTTP 400)                      | Value outside the number entity's range                     | Lower <code>max_w</code> ; check the firmware substitution                     |
| <b>Command keeps climbing while the battery is pinned</b> | <b>Runaway</b> — sign inverted or saturation defeated       | <b>Stop the add-on now.</b> Verify <code>meter_invert</code> with a known load |
| Grid hunts, never rests                                   | Deadband too small, or meter too slow                       | Raise <code>deadband_w</code> ; if meter > 10 s, lower <code>gain</code>       |
| Battery flat overnight and stays flat                     | Maintenance drain never exited                              | Check the SoC entity is the ESP32's, not a cached cloud value                  |
| Maintenance never runs                                    | Disabled, or not yet due                                    | Check <code>maintenance_enabled</code> and the "Maintenance due" row           |
| E-stop fires on every update                              | Expected — upload silence > 30 s                            | Nothing to fix. Do not raise the threshold                                     |
| Add-on will not install                                   | HA is Container/Core                                        | Not supportable — see Step 0                                                   |

## Why the tuning is what it is

Every value came from hardware. Do not change them without measuring.

| Setting                                         | Value   | Reason                                                                                                                                                       |
|-------------------------------------------------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>gain</code>                               | 0.6     | Inverter needs 3–6 s to settle (1.4 s dead time, 94 % by 3.3 s) against a ~5 s cycle. <b>At the limit — do not raise</b>                                     |
| <code>slew_w</code>                             | 1000    | Most of a correction in the first cycle without running far ahead of the hardware                                                                            |
| <code>deadband_w</code>                         | 15      | Measured residual: mean 15.4 W, max 27 W. At 10 W the command never rests                                                                                    |
| <code>saturation_w</code> / <code>cycles</code> | 500 / 3 | Divergence = inverter at a limit → magnitude may fall, never rise. <b>The 3-cycle term is essential:</b> tested instantly it fires on every large correction |
| <code>step_w</code>                             | 10      | Register is 1 W but response lands on a ~17.6 W ladder. Finer is meaningless                                                                                 |
| <code>heartbeat_s</code>                        | 10      | Three chances to be heard before the 30 s watchdog acts                                                                                                      |

The failure this guards against is real: the vendor controller commanded **–14 547 W** against an inverter reporting –5250 W, and kept climbing for six minutes, because its integrator never stopped.