New Energy Grid

⚡ HARDWARE INTEGRATION & DIY CABLE MANUAL

GEMS P1 Port Connection & DIY Cable Guide

Connecting GEMS to Digital Electricity Meters via P1 Port, DIY USB-to-RJ11/RJ12 Cable Assembly, Inverter Schematics & Fluvius e-MUCS / DSMR Telemetry

SYSTEM FIRMWARE
v9.25.3-release (Official Tag)

SUPPORTED STANDARDS
DSMR 5.0.2 / Belgian e-MUCS v1.4 / SMR 5.0

PHYSICAL INTERFACE
RJ12 (6P6C) or RJ11 (6P4C) to USB-UART

TARGET AUDIENCE
Electricians, Installers & DIY Homeowners

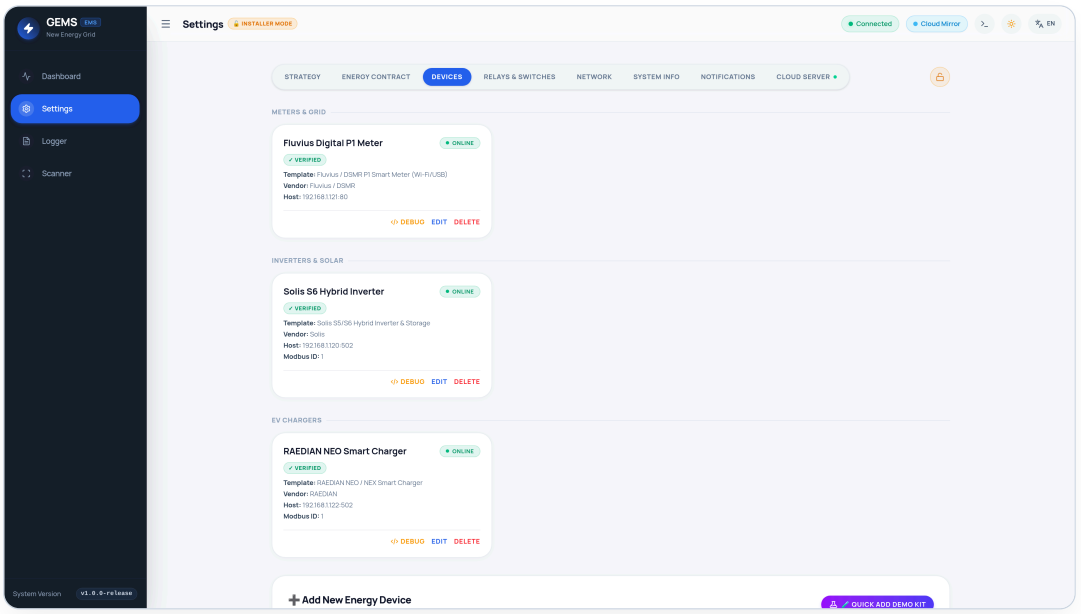


TABLE OF CONTENTS

1. Understanding the P1 User Port
2. Supported Smart Meters & Grid Operators
3. RJ12 vs RJ11 Pinout & Electrical Specs
4. Inverted Logic & Open-Collector Explained
5. Bill of Materials (BOM) & Tools Required
6. Method A: Hardware Transistor Inverter
7. Method B: FTDI Software Inversion (Clean)
8. Step-by-Step Cable Assembly & Testing
9. Activating the P1 Port (Mijn Fluvius / DNO)
10. Linux Recognition & Raw Serial Verification
11. Configuring the P1 Meter in GEMS UI
12. OBIS Telegram Decoding & Flanders Peaks
13. Comprehensive Troubleshooting & FAQ

1. Understanding the P1 User Port

Every modern digital utility electricity meter installed by distribution grid operators in Belgium (Fluvius, Sibelga, ORES, RESA), the Netherlands (Liander, Enexis, Stedin), and Luxembourg (Creos Smarty) features a physical customer port labeled **P1**.

The P1 port is a standardized serial communication interface designed specifically for homeowners and energy management systems. Unlike closed utility optical ports, the P1 port provides a unidirectional, real-time data stream directly from the utility revenue meter into GEMS.

What Data Does GEMS Read from the P1 Port?

Telemetry Metric	Frequency	Role in GEMS Optimization
Active Import Power (Watts)	Every 1 second	Instantaneous grid import. Used by the Flanders peak shaving controller to detect sudden load spikes (e.g. induction cooktop, tumble dryer) and trigger immediate battery discharge or EV throttling.
Active Export Power (Watts)	Every 1 second	Solar energy surplus exported to the grid. GEMS modulates EV charging currents (6A–32A) and turns on smart relays (SG-Ready heat pumps) to capture 100% of this surplus locally.
Per-Phase Current (A) & Voltage (V)	Every 1 second	Monitors Phase L1, L2, and L3 to prevent domestic main fuse trips and calculate unbalanced phase loads.
Fiscal Peak Active Demand (kW) `OBIS 1-0:1.6.0`	Every telegram	Authoritative maximum 15-minute average power demand recorded during the current calendar month by the utility. GEMS uses this for its **Smart Adaptive Ceiling**.
Energy Counters (kWh) Import T1, T2 & Export T1, T2	Cumulative	Official utility billing registers. Recorded in GEMS historical database to generate 100% accurate energy reports matching your electric utility bill.
Gas Meter Consumption (m^3)	Every 5 minutes	Sub-metered gas consumption transmitted wirelessly over M-Bus from the digital gas meter to the electricity meter.

2. Supported Smart Meters & Grid Operators

GEMS supports all smart meter models deployed under the Dutch Smart Meter Requirements (DSMR 4.0, 4.2, 5.0.2) and Belgian companion specifications (e-MUCS v1.4):

Grid Operator / Region	Meter Manufacturer & Models	Standard	Baud Rate
Fluvius (Flanders, Belgium)	Sagemcom T211 (3-phase), S211 (single-phase) Landis+Gyr E360	e-MUCS v1.4 / DSMR 5.0.2	**115,200 baud** (8N1)
Sibelga (Brussels, Belgium)	Sagemcom T211 / S211	e-MUCS v1.4 / DSMR 5.0.2	**115,200 baud** (8N1)
ORES & RESA (Wallonia, Belgium)	Sagemcom T211 / Landis+Gyr E360	e-MUCS v1.4 / DSMR 5.0.2	**115,200 baud** (8N1)
Liander, Enexis, Stedin (Netherlands)	Landis+Gyr (E350, E360), Kaifa (MA105, MA304), Iskra (ME382, MT382), Kamstrup	DSMR 5.0.2 DSMR 4.0 / 4.2	**115,200 baud** (DSMR 5) **9,600 baud** (DSMR 4 / older)
Creos (Luxembourg)	Smarty (Sagemcom T210-D)	e-MUCS / DSMR 5.0	**115,200 baud** (8N1)

3. RJ12 vs RJ11 Pinout & Electrical Specifications

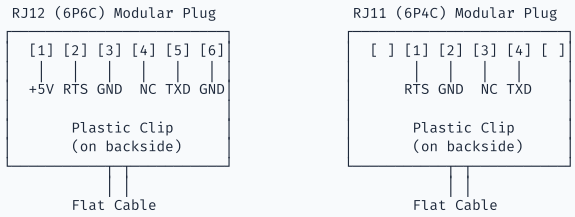
The physical P1 receptacle on the smart meter is a standard **6-position, 6-contact (6P6C) modular socket** (commonly referred to as an **RJ12** socket).

> 💡 **Can You Use an RJ11 (6P4C) Cable?**
>
> **Yes!** A standard telephone plug is physically an **RJ11 (6P4C)** connector. It has 6 plastic pin slots but only the center 4 slots contain metal contacts (corresponding to Pins 2, 3, 4, and 5 of the RJ12 socket).
>
> Because Pin 2 (RTS / Request) and Pin 5 (Data) and Pin 3 (GND) are all located on the center pins, **an RJ11 4-wire cable works perfectly** for reading telemetry into GEMS! The only pin omitted on RJ11 is Pin 1 (+5V power output). Because GEMS connects to the Raspberry Pi's USB port, the USB dongle draws its 5V power from the Pi anyway, making Pin 1 optional!

Official P1 Interface Pinout (DSMR 5.0 / e-MUCS)

Pin #	Signal Name	Electrical Description	RJ12 (6P6C)	RJ11 (6P4C)
Pin 1	**+5V Power**	+5 VDC Output (up to 250 mA on DSMR 5.0 meters). Powers external dongles.	Connected	*Not present* (Not needed)
Pin 2	**Data Request (RTS)**	Control input. **Must be pulled HIGH (+5V)** to start telegram streaming.	Connected	Connected (Pin 1 on 4-wire)
Pin 3	**Data Ground (GND)**	Signal ground reference.	Connected	Connected (Pin 2 on 4-wire)
Pin 4	**Reserved (NC)**	No connection / Reserved.	Connected / NC	Connected / NC
Pin 5	**Data Output (TxD)**	Open-collector data output (Inverted UART, 0V = Mark, 5V = Space).	Connected	Connected (Pin 4 on 4-wire)
Pin 6	**Power Ground (GND)**	Ground reference (internally tied to Pin 3 on the meter).	Connected	*Not present*

Physical Pin Orientation (Looking into the Plug with Gold Contacts Facing You, Cable Pointing Down):



4. Understanding Inverted Logic & Open-Collector

The most common reason why DIY P1 cables fail is **signal inversion**.

Standard TTL serial UART (e.g. from an Arduino or standard USB adapter) uses non-inverted logic:

- Idle line = High (+5V / +3.3V) → Logic 1
- Active start bit = Low (0V) → Logic 0

However, the DSMR / Fluvius P1 specification utilizes an **open-collector output with inverted logic**:

- Idle line = Pulled Low (0V) → Logic 1
- Active transmission = Pulled High (+5V) → Logic 0

If you connect Pin 5 directly to the RX pin of a standard USB serial converter, the operating system reads flipped 1s and 0s, resulting in completely garbled binary noise (framing errors, gibberish characters like `\xff\xfc\x00`).

To fix this, the signal must be inverted. GEMS gives you two clean options:

1. **Method A (Hardware Transistor Inverter):** Place a single inexpensive NPN transistor (e.g. BC547) and a pull-up resistor between Pin 5 and the USB RX pin. Works with *any* USB-serial adapter (CH340, CP2102, FTDI, PL2303).
2. **Method B (FTDI Software Inversion):** Use an authentic FTDI FT232RL or FT232R USB adapter. FTDI chips contain internal logic gates that can invert the RX polarity in hardware EEPROM without adding any external transistors!

5. Bill of Materials (BOM) & Tools Required

Building your own professional USB-to-P1 cable takes about 15 minutes and costs under €5 in parts.

Required Components

Component	Recommended Part	Approx. Cost	Purpose
USB-to-Serial UART Board	FTDI FT232RL USB UART Adapter (or CH340G / CP2102) with 5V logic jumper	€3.50 – €6.00	Translates serial bits to USB for the Raspberry Pi. FTDI is recommended for software inversion.
Modular Cable & Plug	RJ12 (6P6C) or RJ11 (6P4C) crimp plug with 1–2 meter 4-core or 6-core flat modular cable (or CAT5/CAT6 UTP cable)	€1.00	Physical plug that inserts into the smart meter P1 socket.
NPN Transistor *(Method A only)*	BC547, BC548, 2N3904, or 2N2222 general-purpose NPN bipolar transistor	€0.15	Inverts the open-collector signal from the meter before reaching the UART RX pin.
Resistor R1 (Pull-up) *(Method A only)*	1 kΩ (or 2.2 kΩ) 1/4W metal film resistor	€0.05	Pulls the transistor collector to +5V.
Resistor R2 (Base) *(Method A only)*	10 kΩ (or 4.7 kΩ) 1/4W metal film resistor	€0.05	Limits base current from P1 Data output into the transistor.
Heat-shrink Tubing	3mm and 6mm polyolefin heat-shrink sleeve	€0.20	Insulates soldered joints and prevents accidental short circuits.

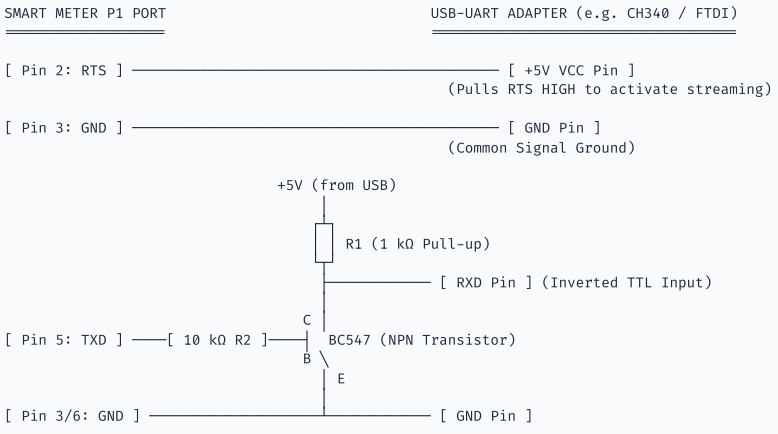
Required Tools

- **Modular Crimping Tool:** For crimping RJ11 / RJ12 connectors. (Alternatively, you can cut an existing telephone cable in half and strip the bare wires!).
- **Wire Stripper & Flush Cutters:** For cleanly exposing inner conductor strands.
- **Soldering Iron & Lead-Free Solder:** For soldering components to the USB adapter header pins.
- **Digital Multimeter:** For checking continuity and ensuring 0.00V shorts do not exist between 5V and GND.

6. Method A: Hardware Transistor Inverter Build

This circuit works universally with **every USB-serial chip on the market** (CH340G, CP2102, PL2303, FTDI, Raspberry Pi GPIO UART).

Circuit Schematic



How the Transistor Circuit Operates

1. When P1 Pin 5 is **Low (0V / Idle)**, no current flows into the base of the BC547. The transistor is OFF (open). Resistor R1 pulls the USB adapter's RXD pin up to **+5V (High)**. This matches the standard UART idle state!
2. When P1 Pin 5 is **High (+5V / Bit 0)**, current flows through R2 into the base, turning the transistor ON (saturated). The collector pulls the USB adapter's RXD pin directly to **0V (Ground)**.
3. The logic is cleanly inverted at nanosecond speed, perfectly recreating the standard 115,200 baud UART waveform!

7. Method B: FTDI Software Inversion (Zero Extra Components!)

If you use an official **FTDI FT232R** USB-UART adapter, you do not need any transistors or resistors! The FT232R silicon has internal configurable inverter gates built into its I/O pads.

Clean 3-Wire Connection Diagram

```
SMART METER P1 PORT (RJ11 or RJ12)              FTDI FT232RL USB ADAPTER
═══════════════════════════════════             ═══════════════════════

[ Pin 2: Data Request (RTS) ] ──────────────────── [ +5V (VCC) ] (Pulls RTS High)

[ Pin 3: Data GND ] ────────────────────────────── [ GND ]

[ Pin 5: Data TxD ] ────────────────────────────── [ RXD ] (Direct connection!)
```

Configuring FTDI RX Inversion via Linux (ftdi_eeprom)

You can permanently configure the FTDI adapter's EEPROM on your Raspberry Pi in 60 seconds:

```
# 1. Install the open-source ftdi-eeprom utility
sudo apt-get update && sudo apt-get install -y ftdi-eeprom

# 2. Create a temporary configuration file
cat << 'EOF' > ftdi_p1.conf
vendor_id=0x0403
product_id=0x6001
invert_rxd=true
EOF

# 3. Flash the inversion flag into the FTDI EEPROM
sudo ftdi_eeprom --flash-eeprom ftdi_p1.conf

# 4. Unplug and replug the USB adapter
# The adapter now natively inverts incoming serial bits in hardware!
```

> 🌟 **Flashing via Windows (FT_Prog)**
> If you are on Windows, download the free **FT_Prog** utility from FTDI's official website. Plug in the adapter → Click **Scan and Parse** → Expand **Hardware Specific** → **Invert RS232 Signals** → Check the box for **Invert RXD** → Click **Program** (lightning bolt icon). Done!

8. Step-by-Step Cable Assembly & Verification

Step 1: Strip and Prepare the Cable

Remove 25 mm of the outer jacket from the telephone/UTP cable. Do not nick the inner conductor insulation. Strip 4 mm of insulation from each conductor strand.

Step 2: Crimp the RJ11 / RJ12 Connector

Hold the clear modular connector with the **plastic locking tab facing away from you (down)** and the gold contact blades facing up. Insert the wires from left to right:

- **For RJ12 (6P6C):** Pin 1 (Leftmost, +5V optional) • Pin 2 (RTS Request) • Pin 3 (GND) • Pin 4 (NC) • Pin 5 (TxD Data) • Pin 6 (Rightmost, GND).
- **For RJ11 (4-wire flat cable):** Wire 1 → Pin 2 (RTS) • Wire 2 → Pin 3 (GND) • Wire 3 → Pin 4 (NC) • Wire 4 → Pin 5 (TxD Data).

Push wires firmly until they bottom out against the front plastic end. Squeeze the crimper firmly until the ratchet releases.

Step 3: Solder the USB Adapter Connections

Connect the cable wires to your USB UART board according to Method A (with transistor) or Method B (direct FTDI). Ensure the voltage selection jumper on your USB-UART board is set to **5V** (not 3.3V).

Step 4: Pre-Power Multimeter Verification (Crucial Safety Step!)

> ⚠️ **Mandatory Multimeter Check Before Plugging In**
>
> Before inserting the cable into your utility smart meter, take your digital multimeter and set it to **Continuity / Resistance (Ω) mode**:
>
> - Probe between **+5V (Pin 2 / USB 5V)** and **GND (Pin 3)**: Ensure there is **NO short circuit** (resistance must be infinite / OL).
> - Probe between **TxD (Pin 5)** and **GND**: Ensure no short circuit exists.
>
> A short between +5V and GND could trip the smart meter internal polyfuse or damage your Raspberry Pi USB bus!

9. Activating the P1 Port with Your Grid Operator

In Belgium (Fluvius), the P1 user port is **electronically locked by default** to protect consumer privacy. You must request activation via the web portal before the meter will transmit data.

Fluvius Activation Steps (Flanders, Belgium)

1. Go to **www.fluvius.be** (https://www.fluvius.be) and log in to your personal **"Mijn Fluvius"** portal.
2. Navigate to **Meters & Aansluitingen** → Select your digital electricity meter (EAN code).
3. Look for the section labeled **"Poorten digitaal beheer (P1 en S1)"**.
4. Click **"Activeer gebruikerspoorten"** and accept the privacy terms.
5. **Activation Time:** The cellular grid network transmits an over-the-air activation command to your smart meter. This typically completes within **15 minutes to 72 hours**.
6. **Visual Verification on Meter LCD:** Look at your smart meter's digital display.
   - When **LOCKED**: A solid yellow/black **padlock icon (🔒)** appears above "P1".
   - When **ACTIVE**: The padlock icon disappears, and a small **downward triangle (▼)** appears above "P1", indicating the port is open and streaming!

> 💡 **Netherlands Smart Meters (Liander, Enexis, Stedin)**
> In the Netherlands, smart meters under DSMR 4 and DSMR 5 have their P1 ports **open and active by default out-of-the-box**. No online activation request is required.

10. Linux Recognition & Raw Serial Verification

Plug your completed USB cable into any free USB port on your GEMS Raspberry Pi 5.

1. Verify USB Device Enumeration

Open Cockpit terminal on port 9090 or SSH into your Raspberry Pi and run:

```
dmesg | grep -E "ttyUSB|FTDI|ch341"
```

You should see output similar to:

```
[ 124.512] usb 1-1.2: FTDI USB Serial Device converter detected
[ 124.513] usb 1-1.2: Detected FT232RL
[ 124.514] usb 1-1.2: FTDI USB Serial Device converter now attached to ttyUSB0
```

2. Test Raw Telegram Stream in Terminal

Configure the serial terminal parameters (115,200 baud, 8 data bits, no parity, 1 stop bit) and read the live telegrams:

```
stty -F /dev/ttyUSB0 115200 cs8 -cstopb -parenb
cat /dev/ttyUSB0
```

Within 1 to 2 seconds, clean ASCII text telegrams will print on your screen:

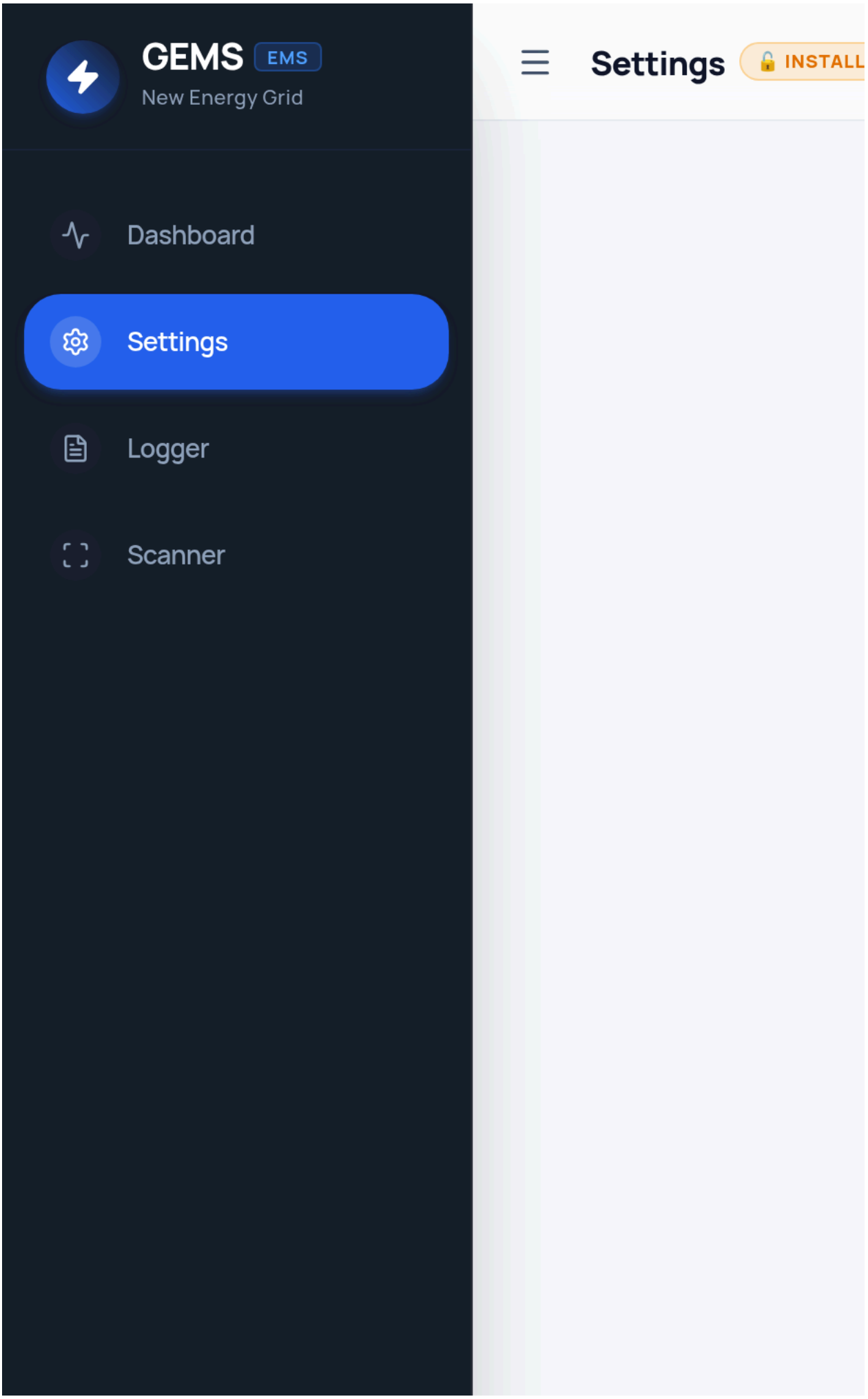
```
/FLU5\253769484_A

0-0:96.1.4(50217)
0-0:96.1.1(3153414733313030313233343536)
0-0:1.0.0(260912224500S)
1-0:1.8.1(004218.412*kWh)
1-0:1.8.2(003189.102*kWh)
1-0:2.8.1(001890.871*kWh)
1-0:2.8.2(000652.319*kWh)
1-0:1.7.0(01.850*kW)
1-0:2.7.0(00.000*kW)
1-0:1.6.0(260908183000S)(02.450*kW)
1-0:31.7.0(008.2*A)
1-0:32.7.0(231.4*V)
!5F2A
```

If you see readable text beginning with `/FLU5 ...` and ending with `!CRC`, **your DIY cable is 100% working!**

11. Configuring the P1 Meter in GEMS UI

With the cable connected and tested, adding the smart meter to GEMS requires zero coding:


GEMS
EMS
New Energy Grid
Settings
INSTALL
Dashboard
Settings
Logger
Scanner

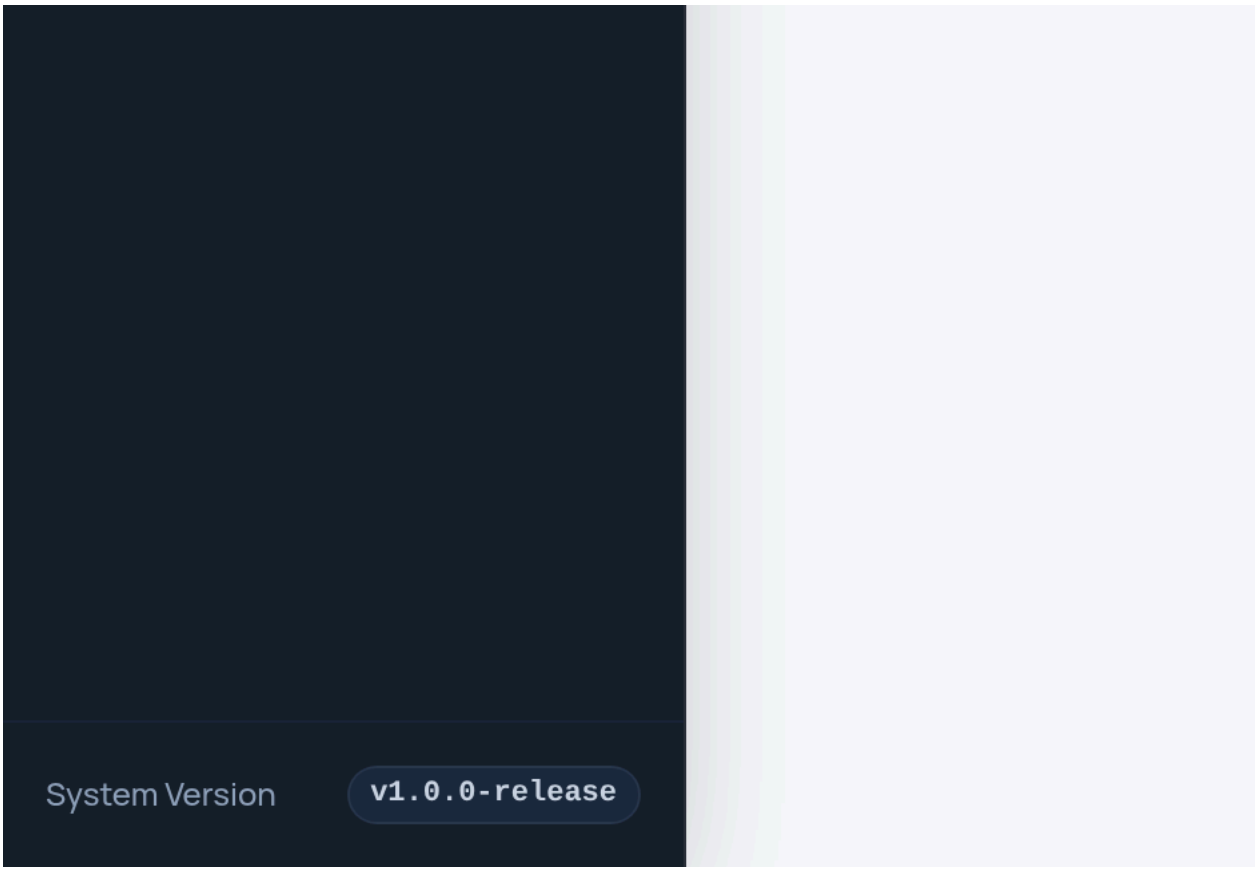


Figure 11.1: Configuring the DSMR P1 Smart Meter in GEMS Settings → Devices.

Step-by-Step Onboarding Workflow

1. Open your web browser and navigate to `http://ems.local`.
2. Click **Settings** in the navigation sidebar, then click the **Devices** tab.
3. Click the **+ Add Device** button.
4. Select the **Smart Meter** category card.
5. In the **Template** dropdown, select:
   - **DSMR P1 RJ12 Serial Dongle (/dev/ttyUSB0)** — for direct USB cable connection.
   - *(Or "Fluvius / DSMR P1 Smart Meter (Wi-Fi/USB)" if using a wireless network dongle like HomeWizard).*
6. Configure the connection fields:

Setting Name	Recommended Value	Installer Guidance & Explanation
Device Name	`Grid Main Smart Meter`	A clear descriptive label displayed on the dashboard and device list.
Serial Port Path	`/dev/serial/by-id/usb-FTDI_...` *(or `/dev/ttyUSB0`)*	GEMS automatically discovers connected USB serial adapters and displays clickable **persistent** `/dev/serial/by-id/*` **quick-select pills** so your P1 cable never swaps port numbers with USB-RS485 adapters after a reboot.
Baud Rate	`115200`	Choose **115200** for Fluvius (Belgium) and DSMR 5.0 (Netherlands). Only select **9600** if connecting to legacy Dutch DSMR 2/3/4 meters.
Poll Interval	`1` (or `2` seconds)	How often GEMS parses incoming telegrams. A 1-second interval ensures sub-second reaction for peak shaving.

7. Click **Add Device**. The device card will immediately appear with a green • **Online** badge, and live grid power will flow into your Dashboard!

12. OBIS Telegram Decoding & Flanders Peaks

The P1 data stream adheres to the international **OBIS (Object Identification System)** standard. GEMS parses every telegram and extracts the following key registers:

OBIS Code	Field Description	Example Value	How GEMS Uses This
`1-0:1.7.0`	Actual Active Power Import (+P)	`01.850*kW` (1,850 W)	Real-time grid import power. Plotted on dashboard and monitored for peak shaving.
`1-0:2.7.0`	Actual Active Power Export (-P)	`00.000*kW` (0 W)	Real-time solar surplus feed-in. Used for EV dynamic charging and relay boosting.
`1-0:1.6.0`	**Maximum Monthly Peak Active Demand**	`(2609081830005)(02.450*kW)`	**Authoritative Fluvius monthly peak demand** recorded on 2026-09-08 at 18:30. GEMS extracts both the exact timestamp and wattage to drive the Smart Adaptive Ceiling!
`1-0:1.4.0`	Current Quarter-Hour Average Demand	`01.920*kW` (1,920 W)	Running average demand for the active 15-minute billing quarter.
`1-0:1.8.1`	Energy Import Tariff 1 (Day / Peak)	`004218.412*kWh`	Fiscal daytime consumption counter for energy billing audits.
`1-0:1.8.2`	Energy Import Tariff 2 (Night / Off-Peak)	`003189.102*kWh`	Fiscal night/weekend consumption counter.
`1-0:2.8.1`	Energy Export Tariff 1 (Day / Peak)	`001890.871*kWh`	Solar energy injected into the grid during daytime hours.
`1-0:2.8.2`	Energy Export Tariff 2 (Night / Off-Peak)	`000652.319*kWh`	Solar energy injected during off-peak hours.
`1-0:31.7.0`	Instantaneous Current Phase L1	`008.2*A`	Current draw on Phase 1 for fuse overload protection.
`1-0:32.7.0`	Instantaneous Voltage Phase L1	`231.4*V`	Mains voltage monitoring (detects solar grid overvoltage > 253V).

13. Exhaustive Troubleshooting & Common Mistakes

Issue 1: Terminal shows completely garbled characters or question marks

Cause: Signal inversion is missing or incorrect baud rate.

- **Fix:** Ensure your hardware inverter transistor (Method A) is correctly wired with collector pulled up to +5V and base driven through a 10 kΩ resistor. If using Method B, verify that `invert_rxd=true` was successfully flashed to the FTDI EEPROM.
- **Baud Rate:** Verify baud rate is set to `115200` for Fluvius/DSMR 5. If your meter is an older DSMR 3 or 4, switch baud rate to `9600`.

Issue 2: `cat /dev/ttyUSB0` prints nothing at all (cursor just sits there)

Cause: Pin 2 (Data Request / RTS) is not pulled HIGH, or P1 port is electronically locked.

- **Check RTS Pin 2:** Measure voltage between Pin 2 and Pin 3 with your multimeter. It **must be +5.0V**. If it is 0V, the smart meter will remain silent! Connect Pin 2 directly to the USB adapter's +5V VCC line.
- **Check P1 Lock:** Look at your meter display. If there is a padlock icon (🔒) above "P1", the port is locked by the grid operator. Log into "Mijn Fluvius" and request port activation.

Issue 3: Linux reports "Permission denied" accessing `/dev/ttyUSB0`

Cause: The `gems` system user needs membership in the `dialout` group.

- **Fix:** Run: `sudo usermod -a -G dialout gems && sudo systemctl restart gems`.

Issue 4: USB device name changes between reboots (e.g. from `ttyUSB0` to `ttyUSB1`)

Cause: Multiple USB serial adapters plugged into the Raspberry Pi.

- **Fix:** Use the persistent hardware serial path instead of generic `/dev/ttyUSB0`. In terminal, run:

```
ls -l /dev/serial/by-id/
```

  Copy the permanent path (e.g. `/dev/serial/by-id/usb-FTDI_FT232R_USB_UART_A50285BI-if00-port0`) and paste it directly into the Serial Port field in GEMS!

> 🛠️ **Summary Checklist for a Flawless P1 Installation**
> - [✓] P1 port activated on Mijn Fluvius (padlock icon gone, triangle visible).
> - [✓] Pin 2 (RTS) pulled to +5V.
> - [✓] Signal inverted (via BC547 transistor or FTDI EEPROM).
> - [✓] Baud rate set to 115,200 (8N1).
> - [✓] Multimeter verified no 5V-to-GND shorts before plugging into meter.
> - [✓] GEMS device template added with poll interval of 1 second.