SolarSync P1 Meter

Technical Datasheet

For system integrators and developers | Firmware V012

Electrical Specifications

Parameter	Value
Supply voltage	5V from P1 port (DSMR 5.0)
Supply current	~50mA typical, ~70mA peak (WiFi TX)
Power consumption	<250mW (within DSMR 5.0 P1 port budget)
CPU frequency	240 MHz
Bluetooth	Disabled at boot (power savings)
Flash storage	Not used for serving (all HTML in PROGMEM)

Hardware Interface

Pin	GPIO	Function
P1_RX	GPIO16	Serial2 RX — receives P1 telegram data
P1_TX	GPIO17	Serial2 TX (unused, required by API)
P1_RST	GPIO12	Request-to-Send: LOW=request data, HIGH=idle

Serial2 runs at 115200 baud, 8N1.

DSMR 5.0 OBIS Codes Parsed

OBIS Code	Description	Unit
1-0:1.7.0	Instantaneous power import (grid → home)	kW
1-0:2.7.0	Instantaneous power export (home → grid)	kW
1-0:1.8.1	Cumulative import tariff 1 (low)	kWh
1-0:1.8.2	Cumulative import tariff 2 (high)	kWh
1-0:2.8.1	Cumulative export tariff 1 (low)	kWh
1-0:2.8.2	Cumulative export tariff 2 (high)	kWh
1-0:21/41/61.7.0	Per-phase power delivered L1/L2/L3*	kW
1-0:22/42/62.7.0	Per-phase power returned L1/L2/L3*	kW
1-0:31/51/71.7.0	Per-phase current L1/L2/L3*	A
1-0:32/52/72.7.0	Per-phase voltage L1/L2/L3*	V
0-0:96.14.0	Tariff indicator (0001=low, 0002=high)	—
0-0:96.7.21	Power failure count	—

0-0:96.7.9	Voltage sag/short interrupt count	—
0-0:1.0.0	Meter timestamp (YYMMDDhhmmssX)	—
*m3)	Gas meter reading (backwards scan)	m³

** Per-phase data only available on 3-phase meters that include these OBIS codes. Single-phase meters like the Ene5\T211 do not provide per-phase instantaneous data. The dashboard auto-detects and hides unavailable sections.*

HTTP API Reference

GET /power

Minimal JSON for machine-to-machine communication (used by SolarSync EV Charger):

```
{ "Solar2Grid": 0.780, "Grid2Home": 0.000 }
```

Content-Type: text/plain. Connection: keep-alive (essential for mDNS-based polling from ESP32 clients).

GET /api/data

Full JSON with all parsed meter values, per-phase data (if available), device info, WiFi diagnostics, and operational counters. Content-Type: application/json.

GET /info

Device identification for mDNS service discovery:

```
{ "type":"SolarSync", "model":"P1-Meter", "fw":"012", "ip":"192.168.68.106",
"mac":"88:57:21:E7:73:08" }
```

mDNS Service Discovery

The device registers itself via mDNS with the hostname SolarSync_XXXXXX (last 6 hex digits of MAC). It announces an HTTP service on port 80.

Important for ESP32-to-ESP32 communication: The /power endpoint uses HTTP keep-alive. Client devices should use HTTPClient with setReuse(true) and avoid calling http.end() between polls. This prevents repeated mDNS resolution which is unreliable on ESP32.

Network Architecture

The device operates in two modes:

- AP Mode: Open access point for first-time setup (192.168.4.1). Captive portal auto-redirects on Android, iOS, macOS, and Windows.
- STA Mode: Connected to home WiFi. All web pages served from PROGMEM (no flash I/O). Settings stored in ESP32 NVS (wear-leveled).

Critical design rule: WiFi.disconnect() is never called during normal STA operation. This preserves the mDNS responder stack. WiFi reconnection uses WiFi.enableSTA(true) + WiFi.begin(). Mode switching (AP → STA) is handled via ESP.restart() for a clean WiFi stack.