

# Transformers – send Volt and AMP to MQTT

29-11-2025 – Jens Krogsgaard, Copenhagen - Denmark

---

## Table of contents

1. Summary .....	1
2. Dataflow .....	2
3. Wiring ESP32 and Energy meter INA226. ....	3

## 1. Summary

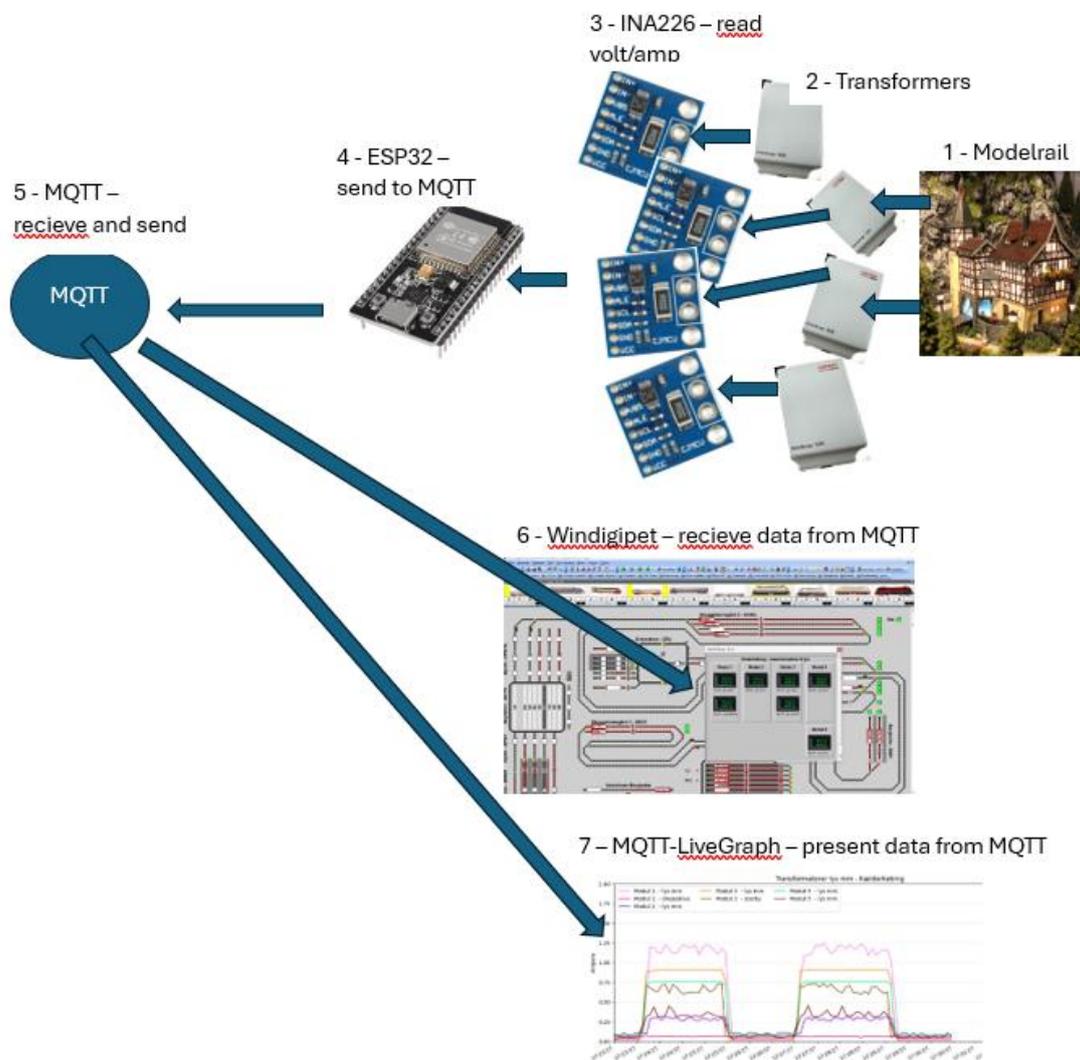
On my model railway layout, I have several secondary transformers that are used to supply power to building lights, various moving figures, and other accessories that do not require the digital power provided by the central station and its boosters. I currently have seven of these secondary transformers, all of which are DC. The transformers are distributed so that each one supplies a limited area. These are small transformers capable of delivering 1–2 amps, so it's important not to overload them.

Until now, I have used manually read multimeters to keep an eye on this. This has worked fine, but since the power consumption of my boosters is now being transferred to Windigipet, I would also like to display the consumption of the secondary transformers in Windigipet.

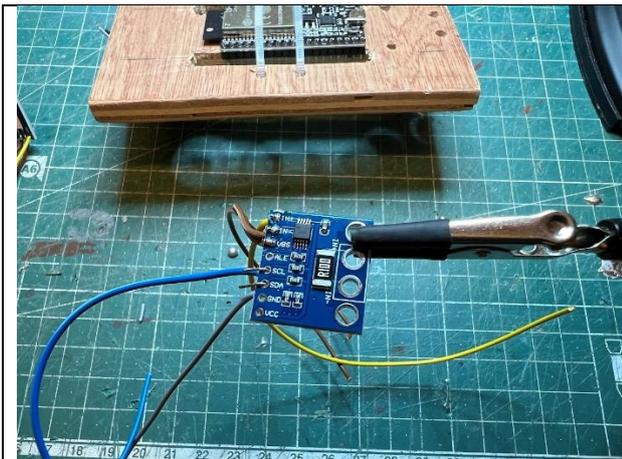


## 2. Dataflow

- 1 – Modelrail
  - On my model railway layout I have seven transformers for lights, etc. They are especially heavily loaded when night falls and all the lights turn on.
- 2 – Transformers
  - The seven transformers are from 1–2 amps and 12–18 volts DC.
- 3 – Energy meter INA226 – read – volt/amp
  - Each transformer is connected to an INA226 that measures voltage and current. An ESP32 can have up to four INA226 modules connected.
- 4 – ESP32 – send to MQTT
  - An ESP32 is a microcontroller that is programmed to receive and send data to MQTT.
- 5 – MQTT – receive and send
  - MQTT – a post office – receives and forwards data.
- 6 – Windigipet – receive data from MQTT
  - Show data in instruments
- 7 – MQTT -LineGrap
  - Show data as Line graphs



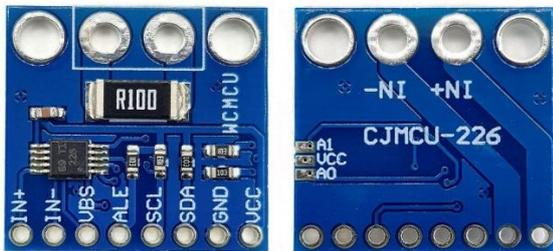
### 3. Wiring ESP32 and Energy meter INA226.



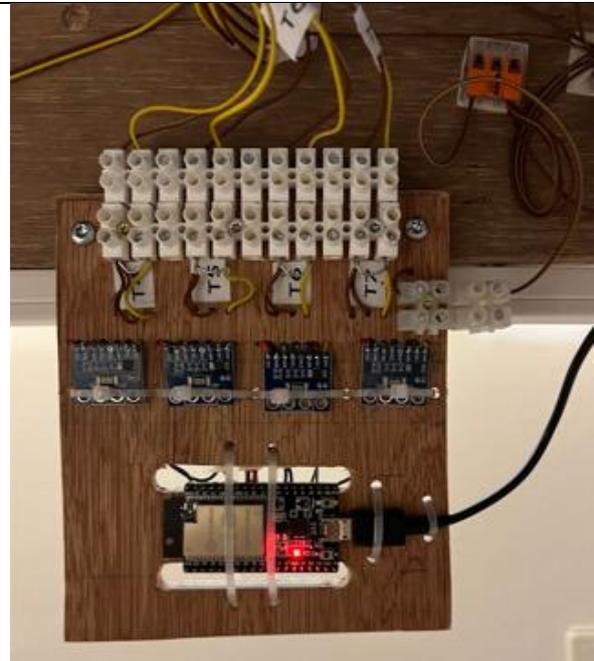
INA226 - Seven wires must be soldered onto each unit.



Up to 4 INA226 per ESP32 – each INA226 is assigned a unique address.



Homemade mounting plates for reading seven transformers

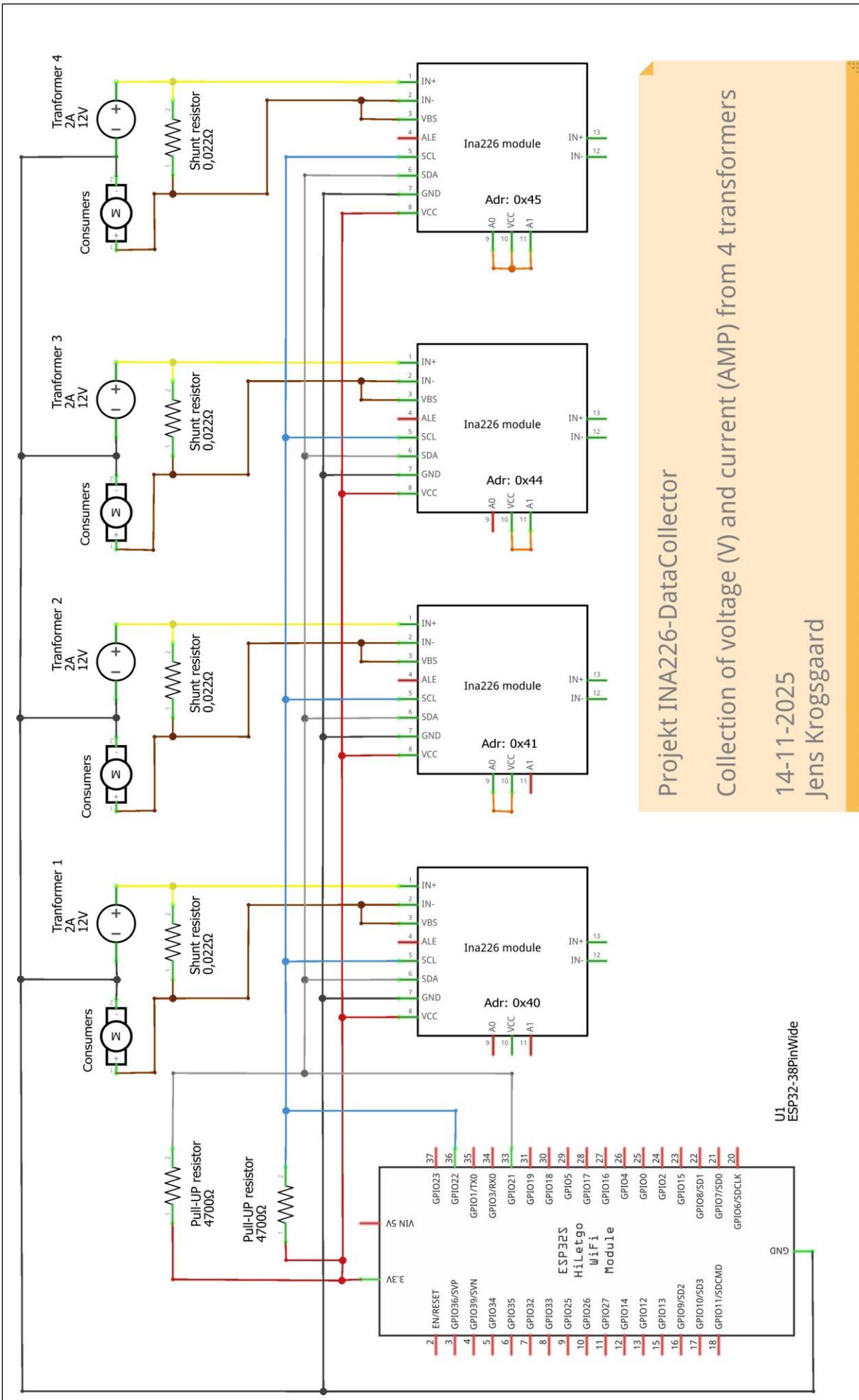


Mounting plate is installed and ready for use

Each INA226 is assigned an address by soldering combinations of A0, A1, and VCC together. There are a total of 4 combinations.



Trafo No	INA226 Address (Hex)	A1 State	A0 State	I2C Address-bit (A1, A0)	Lodningsinstruktion
1	0x40	Open	open	0, 0	No solder (Standard)
2	0x41	Open	to VCC	0, 1	Solder A0 to VCC.
3	0x44	to VCC	Open	1, 0	Solder A1 to VCC.
4	0x45	to VCC	to VCC	1, 1	Solder A0 and A1 to VCC.



Projekt INA226-DataCollector  
 Collection of voltage (V) and current (AMP) from 4 transformers  
 14-11-2025  
 Jens Krogsgaard

U1  
 ESP32-38PinWide

## Wiring Description

The system consists of an ESP32-S WiFi module connected to four INA226 sensors, each measuring voltage and current from a separate 12 V / 2 A transformer.

Each transformer output passes through a 0.022  $\Omega$  shunt resistor, and the INA226 measures the voltage drop across the shunt to calculate current, while also monitoring the supply voltage.

All INA226 modules share the same **I<sup>2</sup>C bus**, with the following wiring:

- **SDA** from all INA226 modules is connected to GPIO21 on the ESP32.
- **SCL** from all INA226 modules is connected to GPIO22 on the ESP32.
- Two **47 k $\Omega$  pull-up resistors** are used to pull SDA and SCL up to 3.3 V.

Power and addressing:

- All INA226 modules are powered from **3.3 V** and share a common **GND** with the ESP32.
- Each INA226 has a unique I<sup>2</sup>C address, configured by wiring the **A0 and A1** address pins.

Measurement wiring:

- For each transformer, the positive output passes through the **0.022  $\Omega$  shunt resistor** and into the INA226 **IN+** pin.
- The load/consumer is connected between the shunt output and the transformer's negative terminal, which connects to **IN-**.
- All transformer grounds are tied together and connected to the system ground.

This wiring configuration allows the ESP32 to simultaneously read voltage and current from all four transformers via the shared I<sup>2</sup>C bus.