

[Accueil](#)[Boutique en ligne](#)[Neuf](#)

# Routeur solaire PV monophasé

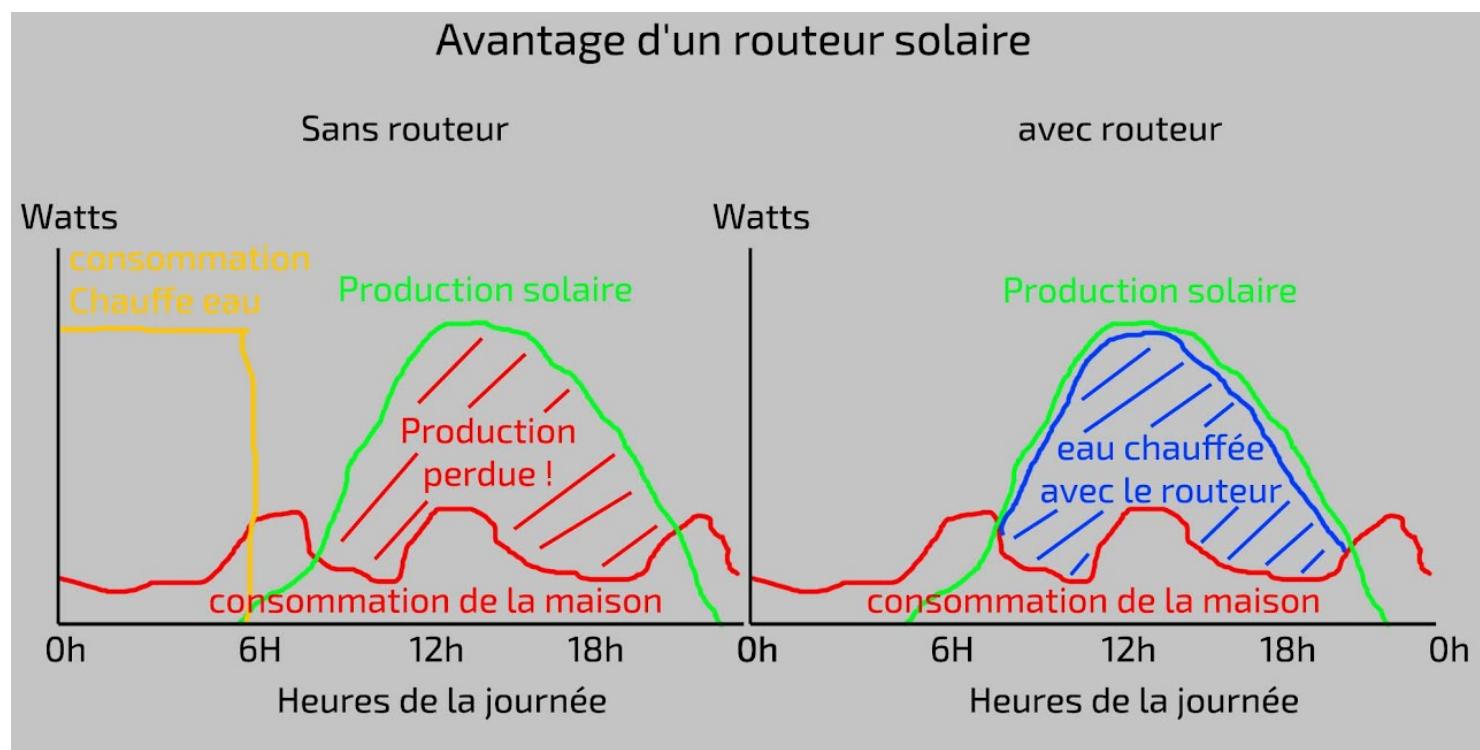
## Le Profes'Solaire

Utilisez des pièces du commerce pour fabriquer votre propre routeur solaire 2 sorties 16A pour 60-70 euros !

Pas de composants à souder, que des cartes à assembler.

Nécessite des connaissances dans le monde de l'Arduino (gestion des bibliothèques, cartes esp,...)

**ATTENTION : CE MONTAGE NECESSITE DES CONNAISSANCES EN ELECTRICITE. VOUS UTILISEZ DES TENSIONS DE 230V MORTELLES EN CAS DE MAUVAISES MANIPULATIONS.**  
**N'ENVISAGEZ PAS CE MONTAGE SI VOUS AVEZ UN DOUTE OU NE SAVEZ PAS FAIRE.**

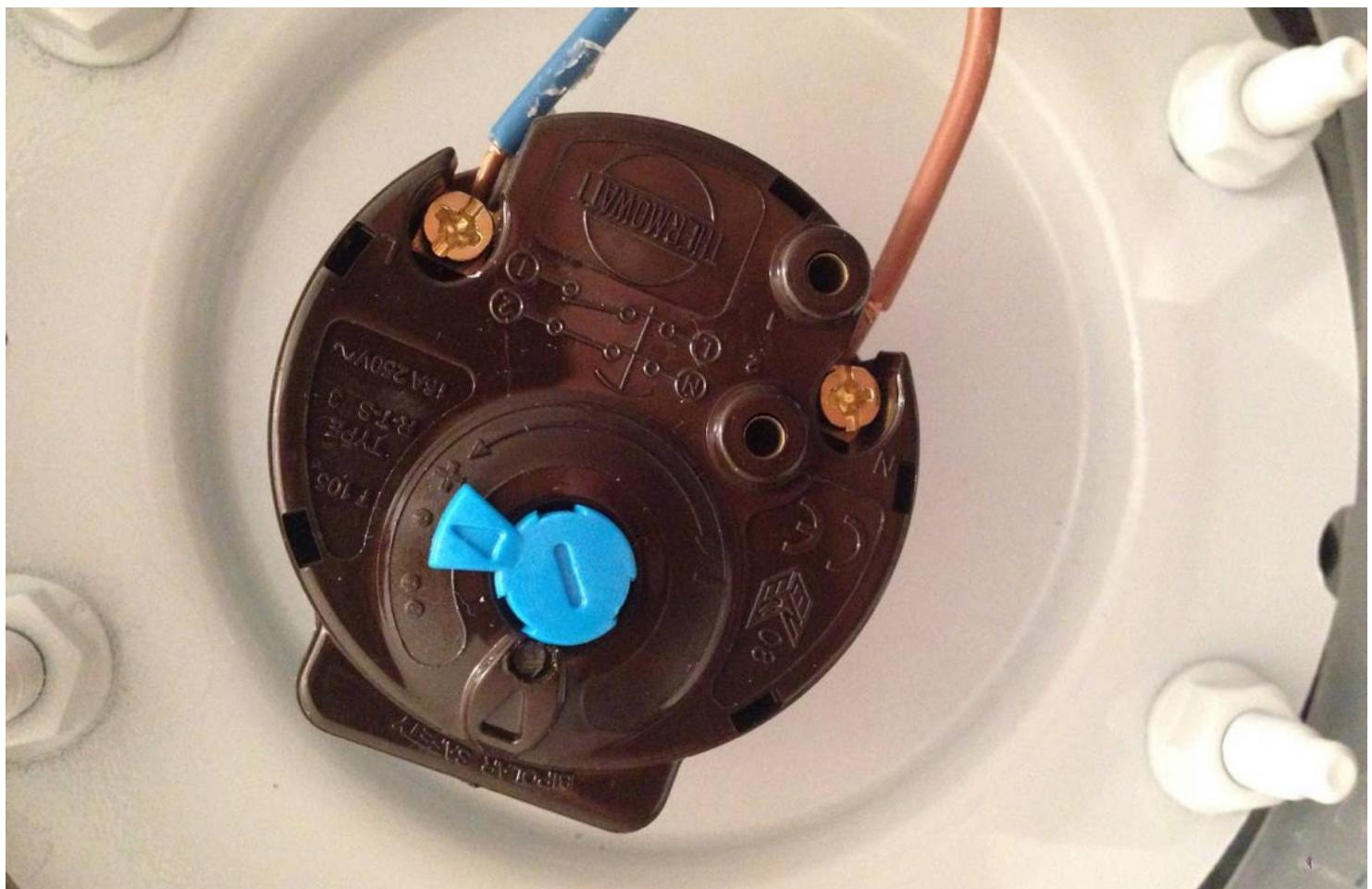


Selon une étude, en France **seulement 30% de la production photovoltaïque est utilisée**, soit 70% de la production retourne gratuitement sur le réseau.

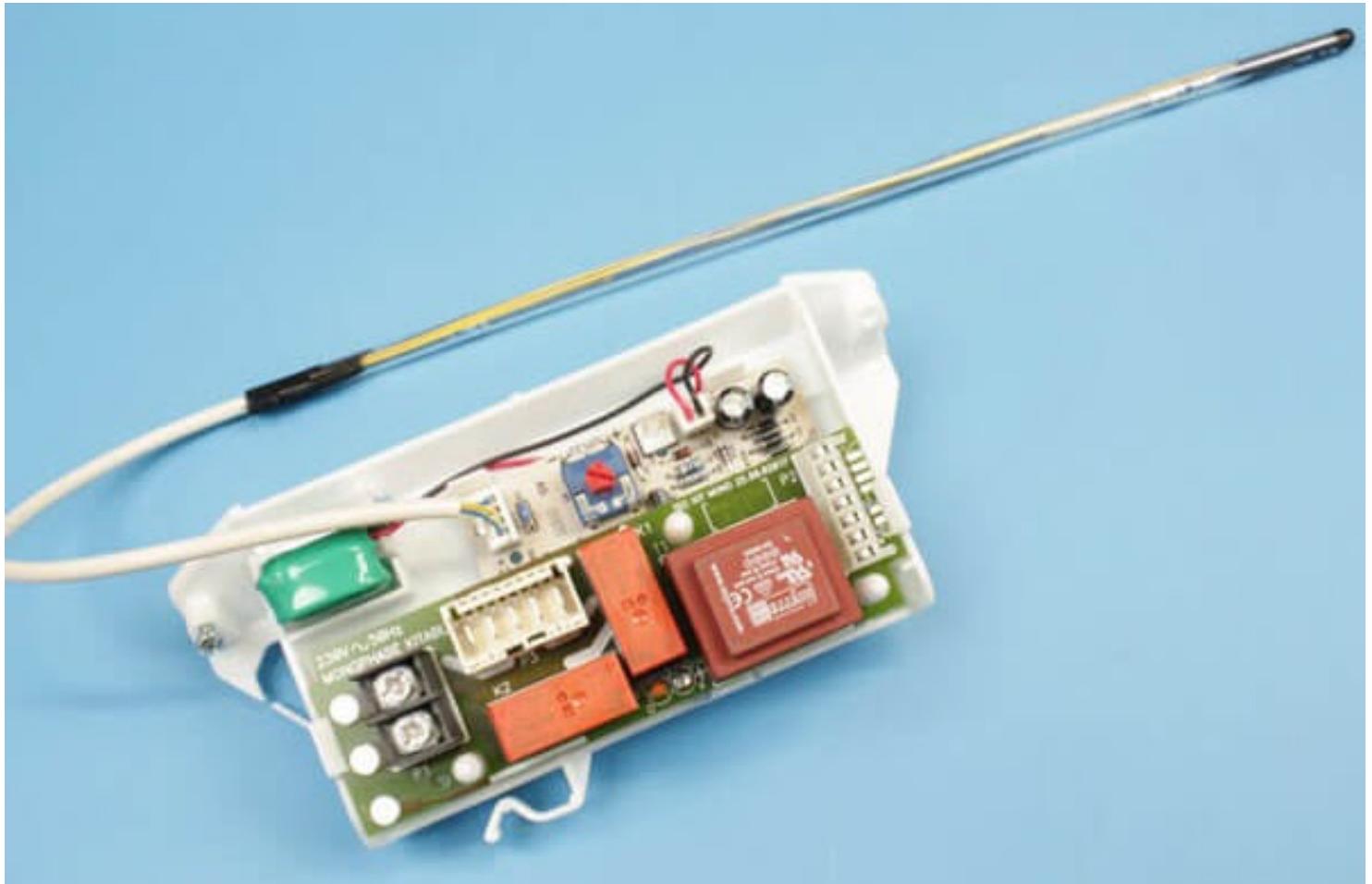
75 - premier routeur solaire Open Source avec capteurs du commerce ! 1 / 3



Le routeur fonctionne avec des charges résistives : donc en gros des résistances électriques avec thermostat mécanique. Si vous possédez un thermostat électronique voici une astuce pour pouvoir utiliser un routeur solaire.



## Thermostat mécanique



Thermostat électronique

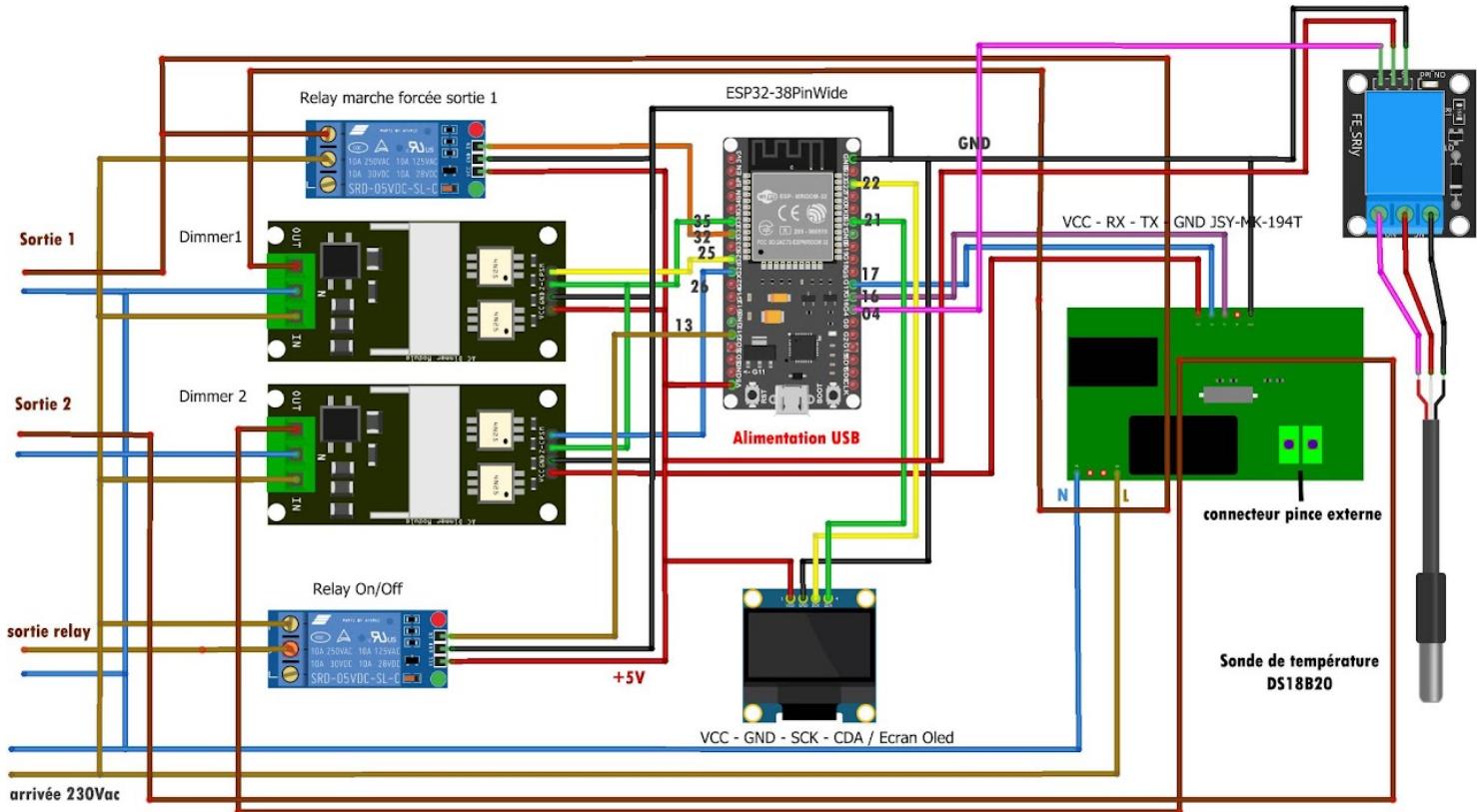
Voici la liste des pièces nécessaires :

- carte ESP32 + Shield
- carte Dimmer 24A -600V
- capteur de courant JSY - MK -194T
- écran OLED 1.3 pouces (4 pins )
- relay 30A (5Vdc-230Vac) pour marche forcée 3000w (déclenchement inversé : avec inversion de l'état de la commande avec jumper)
- radiateur externe

Options :

- relay 30A (5Vdc-230Vac) pour 3ème sortie 3000w (on/off) (déclenchement inversé : avec inversion de l'état de la commande avec jumper)
- gros relay avec radiateur pour grosse puissance (120A crête, 60A nominal), pour plus de 3000w
- Sonde de température DS18B20

# Schéma de câblage routeur 2 sorties Dimmer + 1 relay On/Off + 1 relay Marche forcée 30A et sonde de température dans le ballon



Il faut alimenter les capteurs avec la broche +5v car avec le +3.3v le capteur de courant manque parfois d'énergie et présente des bugs.

L'ESP32 est alimenté par son port USB et tous les capteurs sont alimentés par l'ESP32.

Le relay On/Off se déclenche par défaut à partir de 1000w de surplus et se coupe à 800w de surplus. Les paramètres peuvent être modifiés dans le programme en fonction de ce que vous branchez dessus :

## 79 - routeur solaire Open Source - programmation de l'esp32



Écran Oled en façade du routeur



Le Profes'S  laire v7.00



## Marche forcée

192.168.1.41

17:35:43 Fin à 19h49

Marche forcée automatique suivant le volume du ballon et du surplus sauvé dans la journée. Fonction activable ou désactivable et horaires modifiables. Si le surplus n'a pas été suffisant dans la journée suivant le volume de votre ballon, le routeur lancera une marche forcée à l'heure programmée pour une durée calculée suivant le surplus récupéré et le volume du ballon.

Le Profes'S  laire v8.01



## Marche forcée

192.168.1.92

10:52:55

26°C

Marche forcée automatique suivant la température du ballon. Par défaut la température mini est à 50 degrés (modifiable dans le programme). Si la fonction est activée dans le programme, le routeur fera un complément de chauffe sur les heures programmées pour arriver à 50 degrés. Si le surplus a été suffisant aucune marche forcée ne se mettra en route.

Sans modification du refroidissement du triac ne pas dépasser 1000w en sortie de carte Dimmer

(le radiateur d'origine est trop petit pour envoyer plus de puissance).

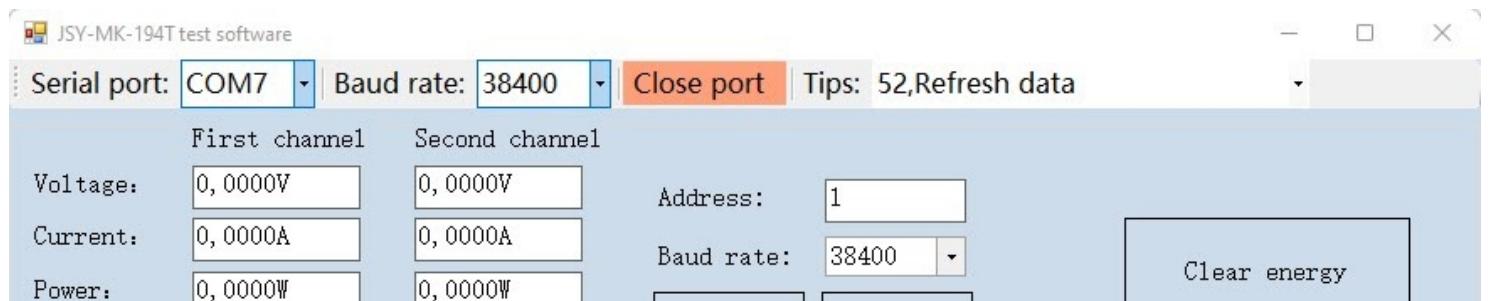
Si vous souhaitez router jusqu'à 3000w par sortie vous devez modifier le radiateur de refroidissement du triac.

- Fichier qui permet de modifier la vitesse du port TTL du capteur JSY : [\*\*vitesse port JSY\*\*](#) (Passage de 4800 à 38400bps)
- Si vous ne parvenez pas à modifier la vitesse du capteur JSY avec ce fichier vous pouvez le faire en utilisant l'application du capteur : [\*\*JSY.exe\*\*](#) (il faut connecter le capteur sur un convertisseur USB-TTL la vitesse par défaut est 4800bps)
- Lien firmware routeur avec 1 ou 2 carte(s) Dimmer et/ou 1 relay : [\*\*routeur solaire v8.11\*\*](#) avec web + marche forcée avec relay (non comptabilisé dans l'énergie sauvée) par page web, avec ou sans mqtt Home Assistant, marche forcée automatique suivant volume du ballon et surplus ou marche forcée automatique suivant la température de l'eau et mise à jour du firmware en wifi.
- [\*\*v9.11\*\*](#) : nouvelle version, merci par avance pour vos retours par rapport à la v8.11. Cette version est plus réactive et permet d'utiliser encore plus de surplus, elle prend en compte tous les relais, de type Arduino ou SSR.

### **A partir de la version 8.05, la troisième sortie Relay peut fonctionner sans Dimmer.**

La carte a utiliser dans Arduino IDE est : **ESP32 Dev MODULE** si cela ne fonctionne pas utilisez **DOIT ESP32 DEVKIT V1**

Partagez les photos de vos réalisations ! :)



Pour ceux qui veulent aller plus loin, voici les Datasheets du capteur : [JSY-MK-194T](#).

Vous aimez ce projet ?



Faire un don à  
**Le Profes'Solaire**

## Promouvoir l'autoconsommation photovoltaïque

### Ponctuelle

Mensuelle

0

EUR



### **Faire un don avec PayPal**

#### **Faire un don par carte bancaire**

Configuration à modifier dans le fichier du routeur :

```
const char* ssid = "votre réseau wifi"; // nom de votre réseau
wifi

const char* password = "votre mot de passe wifi"; // mot de passe de
votre réseau wifi

boolean mqtt = 0; // activer ou
désactiver MQTT Mosquitto pour Home Assistant : 0 ou 1

#define mqtt_server "192.168.1.10" // adresse de votre
serveur mqtt //

#define mqtt_user "mosquitto" // utilisateur mqtt //
#define mqtt_password "mosquitto" // mot de passe mqtt //
int relayOn = 1000; // puissance du surplus
pour déclencher le relay //

int relayOff = 800; // puissance du surplus
pour stopper le relay //

boolean marcheForceeVol = 0; // marche forcée
automatique suivant le volume du ballon : 0 ou 1

int volume = 200; // volume du ballon en
litres

boolean marcheForceeTemperature = 0; // marche forcée
automatique avec sonde de température DS18B20 (50 degrés) : 0 ou 1

byte HOn=01; // heure début marche
forcée

byte MnOn=30; // minute début marche
forcée

byte SecOn=00; // sec début marche
forcée

int temperatureEau = 50; // réglage de la
température minimale de l'eau en marche forcée, exemple 50 degrés

//////////
```

## 76 - Routeur solaire - modification carte Dimmer 2 /3



Configuration Home Assistant pour MQTT Mosquitto (à partir du firmware v8.01) : à insérer dans le fichier **configuration.yaml** :

**mqtt:**

**sensor:**

**- name: "Consommation"**

**state\_topic: "mosquitto/power1"**

**unit\_of\_measurement: 'w'**

**value\_template: "{{ value\_json.consommaion }}"**

**- name: "Puissance surplus"**

**state\_topic: "mosquitto/power1"**

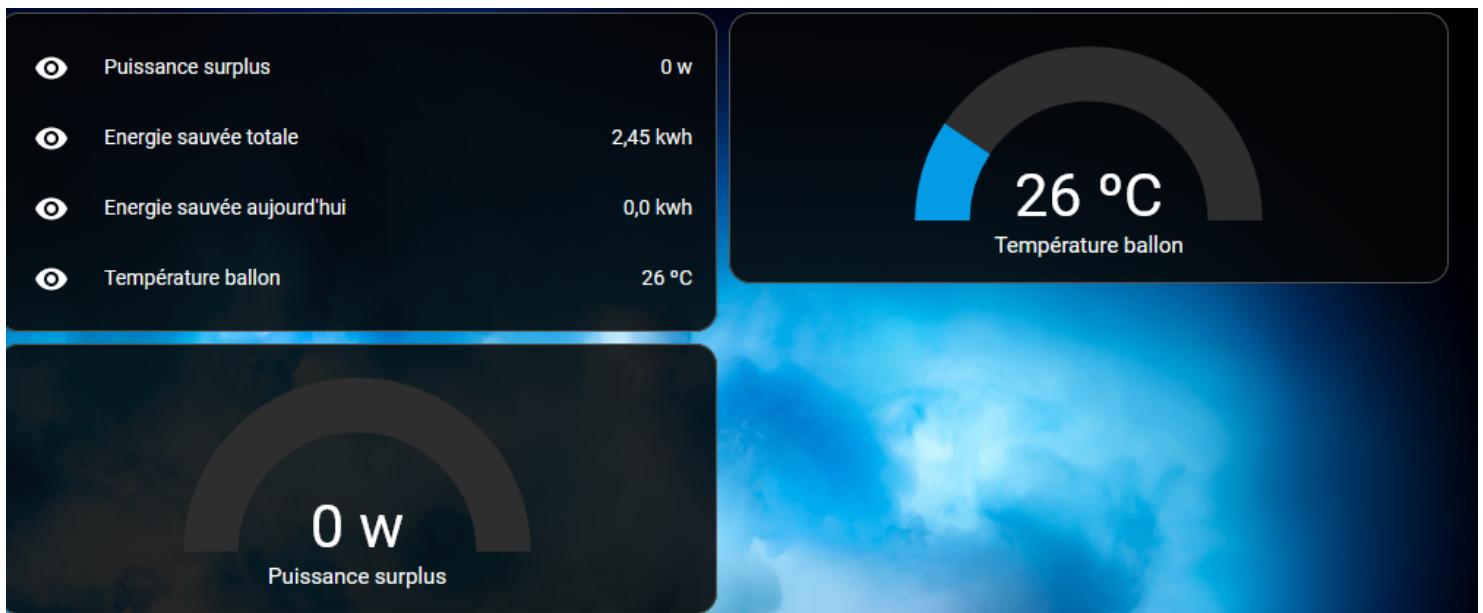
**unit\_of\_measurement: 'w'**

**value\_template: "{{ value\_json.surplus }}"**

```
- name: "Energie sauvée totale"
state_topic: "mosquitto/power1"
unit_of_measurement: 'kwh'
value_template: "{{ value_json.energie }}"

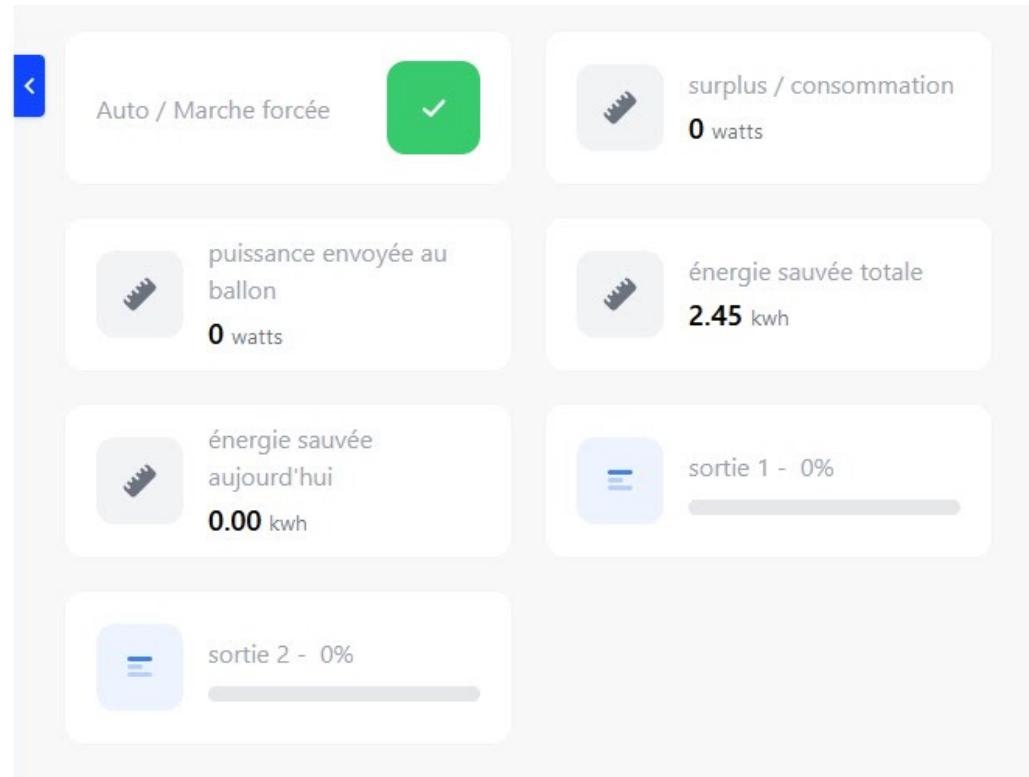
- name: "Energie sauvée aujourd'hui"
state_topic: "mosquitto/power1"
unit_of_measurement: 'kwh'
value_template: "{{ value_json.energoday }}"

- name: "Température ballon"
state_topic: "mosquitto/power1"
unit_of_measurement: '°C'
value_template: "{{ value_json.temperature }}"
```



Ajout de l'intégration dans Home Assistant (MQTT Mosquitto).

Puissance en temps réel envoyée au ballon, énergie sauvée au totale depuis le début et énergie sauvée par jour (remise à 0 tous les jour à minuit) et température du ballon avec la sonde.



The screenshot shows the Arduino IDE interface with the 'Tools' menu open. The sketch 'esp32\_oled\_ac\_v' is selected in the left pane. The 'Tools' menu contains various options for managing sketches and connecting to boards. A portion of the sketch code is visible on the right, specifically the configuration section at the top.

```
Formatage automatique Ctrl+T
Archiver le croquis
Gérer les bibliothèques... Ctrl+Maj+I
Moniteur série Ctrl+Maj+M
Traceur série
Mise à jour du Firmware WiFi101 / WiFiNINA
Téléverser les certificats racine SSL
Carte: "ESP32 Dev Module" ▶ Dimmer 2 24A-6
Port: "192.168.1.90" ▶
Obtenir les informations sur la carte
CPU Frequency: "240MHz (WiFi/BT)" ▶
Core Debug Level: "None" ▶
Erase All Flash Before Sketch Upload: "Disabled" ▶
Events Run On: "Core 0" ▶
Flash Frequency: "80MHz" ▶ e votre réseau
Flash Mode: "QIO" ▶ e passe de vot
boolean Flash Size: "4MB (32Mb)" ▶ er ou désactiv
#define JTAG Adapter: "Disabled" ▶ se de votre se
#define Arduino Runs On: "Core 1" ▶ sateur mqtt // e passe mqtt /
Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)" ▶ nce du surplus
PSRAM: "Disabled" ▶ ance du surplus
Upload Speed: "921600" ▶ e forcée autom
boolean Programmeur ▶ e du ballon en
byte Graver la séquence d'initialisation ▶ e forcée autom
début marche
```

Interface web Dash : si l'interface Dash plante régulièrement pensez à configurer Arduino IDE avant le téléchargement du programme :

- Events Run : "Core 0"
- Arduino Runs On : "Core 1"



ESP32 Dev Module



esp32\_oled\_ac\_v5.ino

```
1  /* Routeur solaire développé par le Profes'Sola
2  - 2 sorties 16A / 3000 watts
3  - 1 relais on/off
```

## Boitier imprimé en 3D :

Un grand merci à Anthony B. pour les fichiers ! Pour un routeur 1 Dimmer sans sonde de température

Les vis et insert

M3 x6 quantité 16

M3 x8 quantité 6

M3x10 quantité 2

M3x12 quantité 2

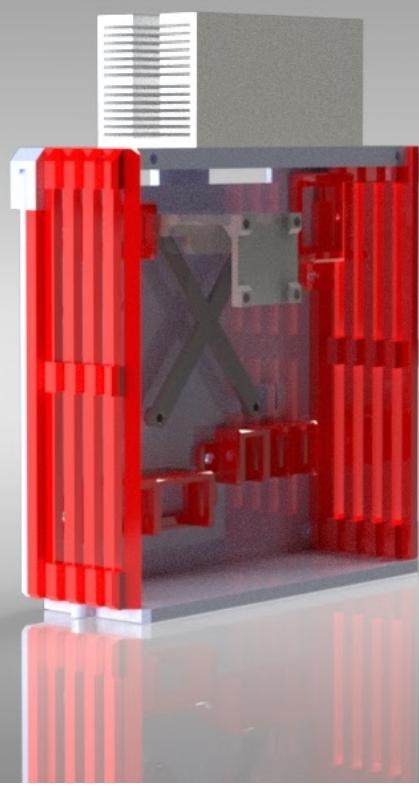
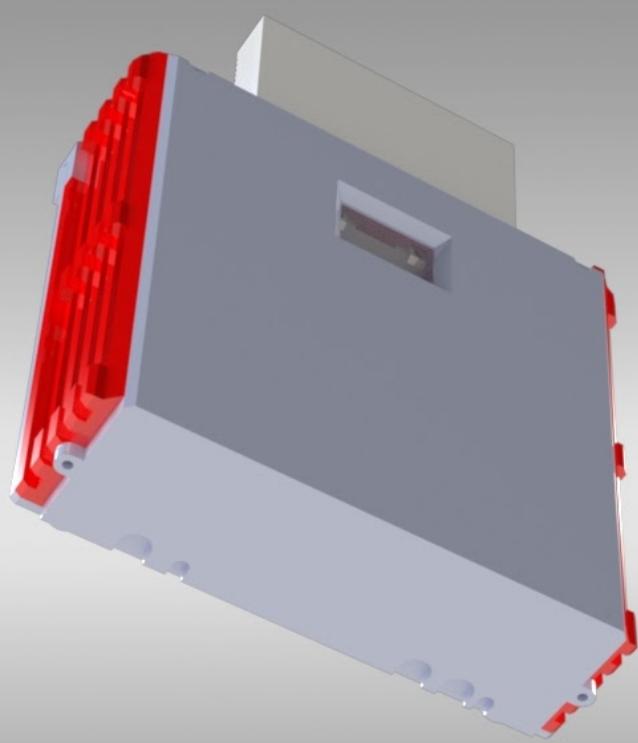
M3x16 quantité 2

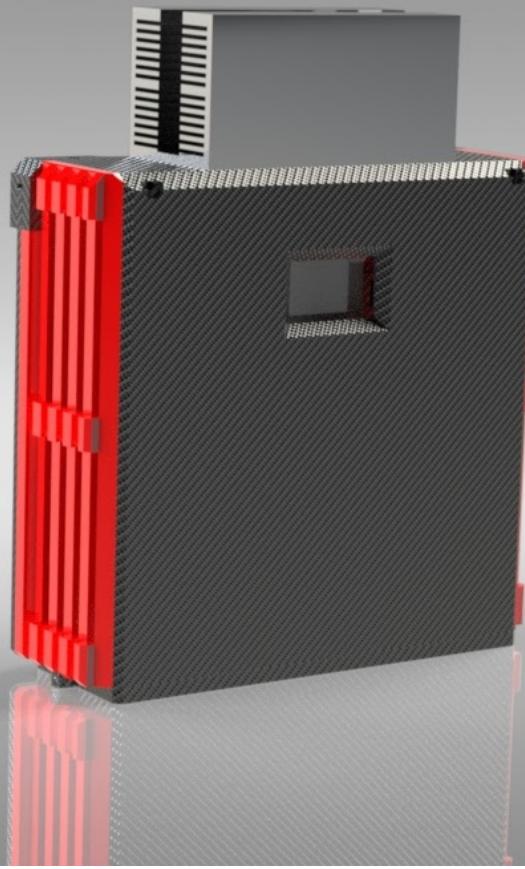
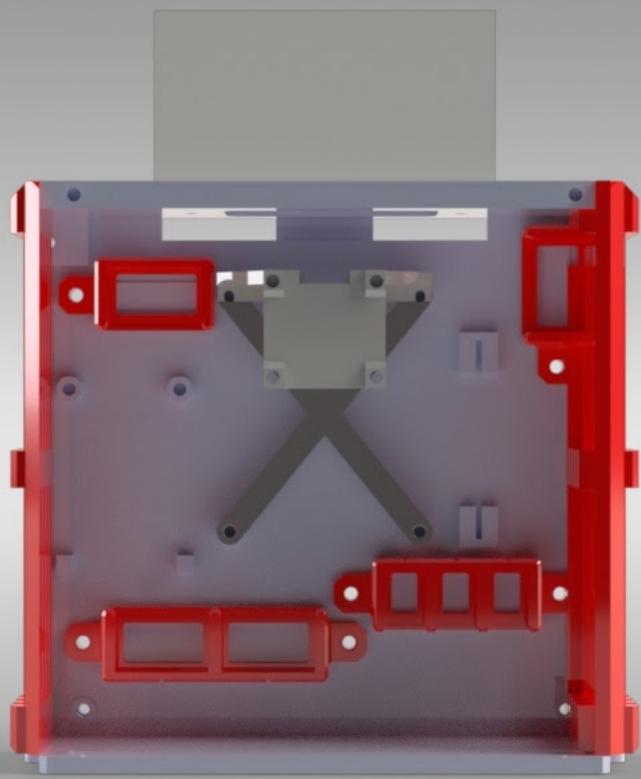
M3x22 quantité 1

Insert M3 4,2 x 3,6 quantité 32

Ajout de la mise à jour du routeur en wifi. Facilite la mise à jour du routeur pour les prochains Firmwares.

Mot de passe par défaut pour mise à jour : admin





voici les différents éléments à imprimer pour ce boitier : **Boitier 3D**

## Archives

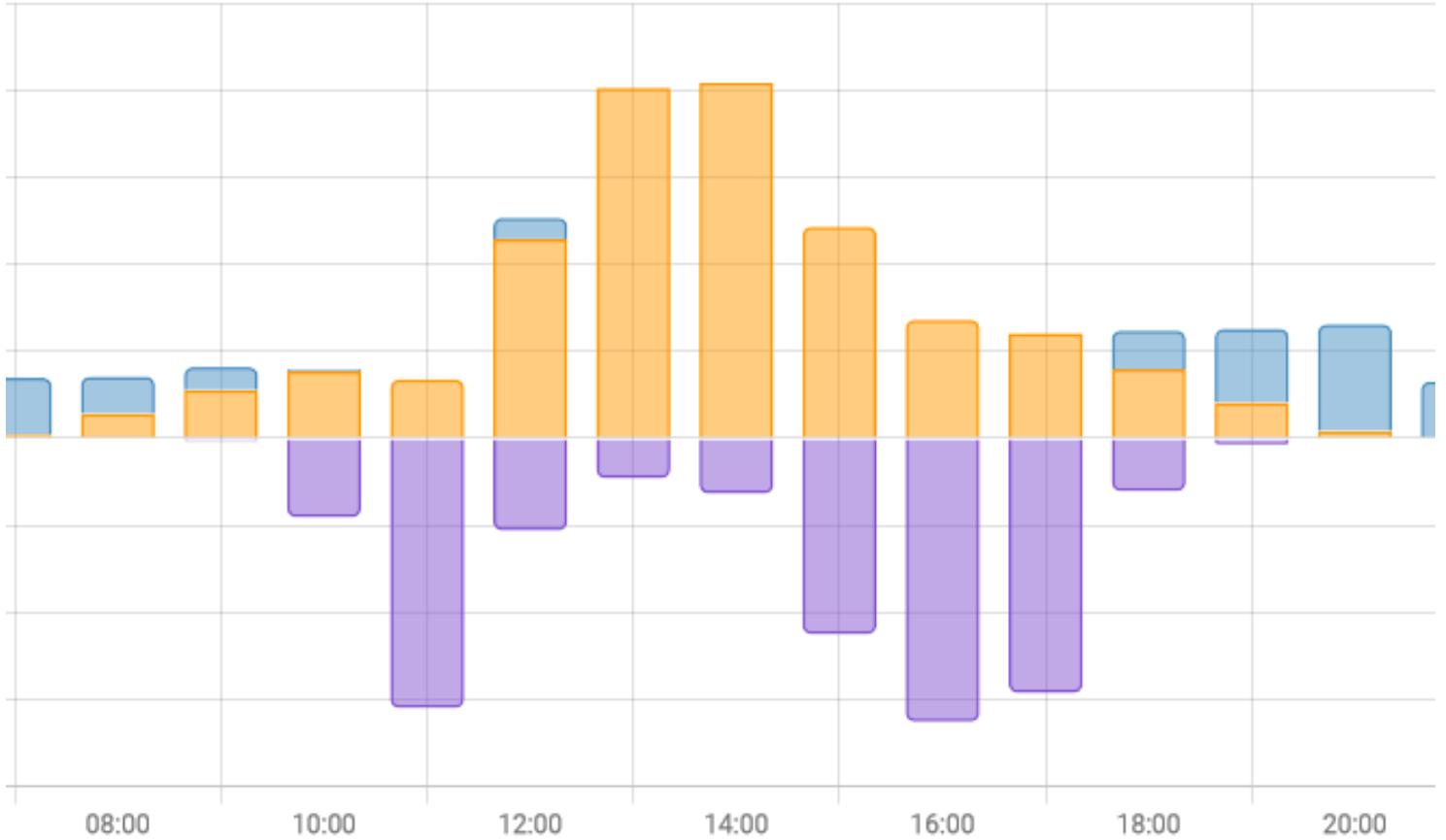
- [routeur v3](#)
- [routeur v4](#)
- [routeur v5](#)
- [routeur v6](#)
- [routeur v7](#)
- [routeur v8.01](#)

## Vos réalisations :

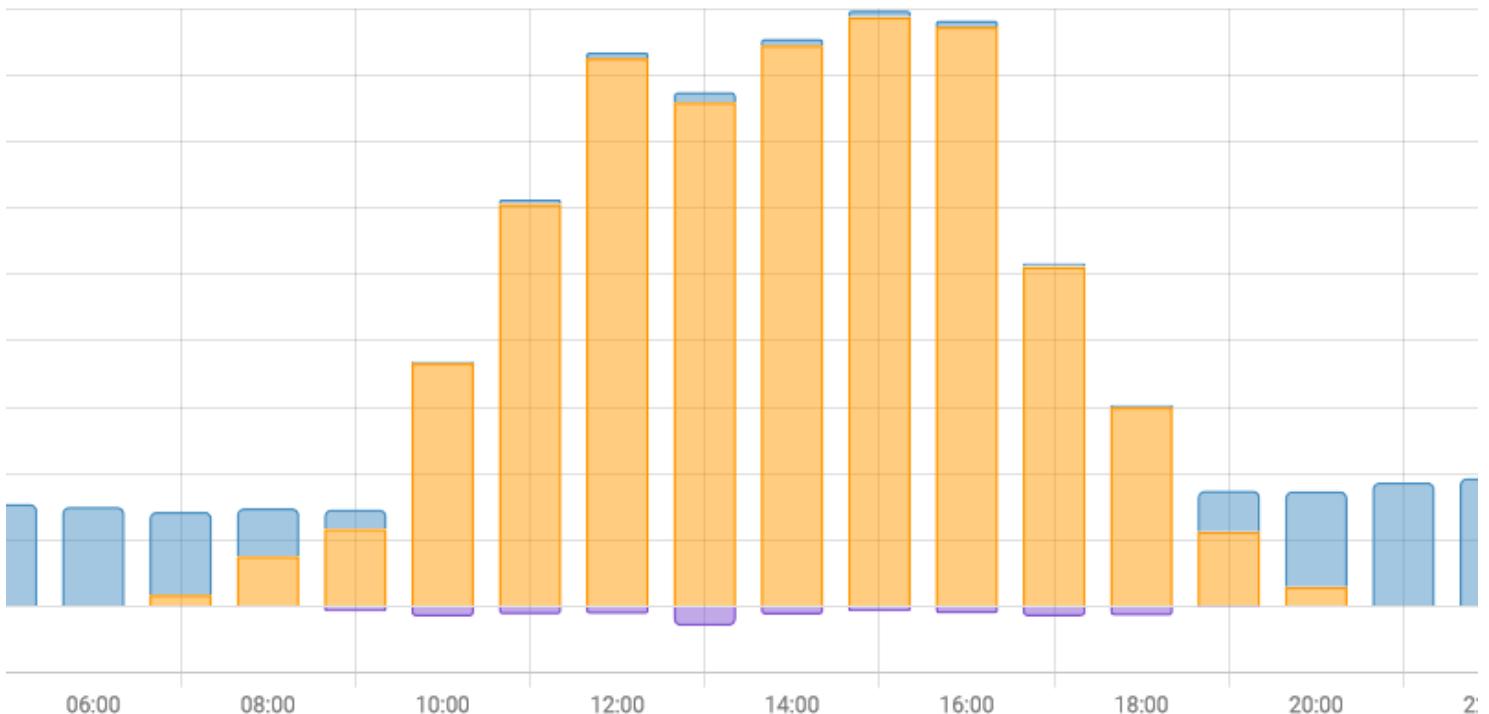
**le routeur de Mickaël :**

554 35  
910210

Sans le routeur : injection réseau en violet, consommation réseau en bleu, orange production pv

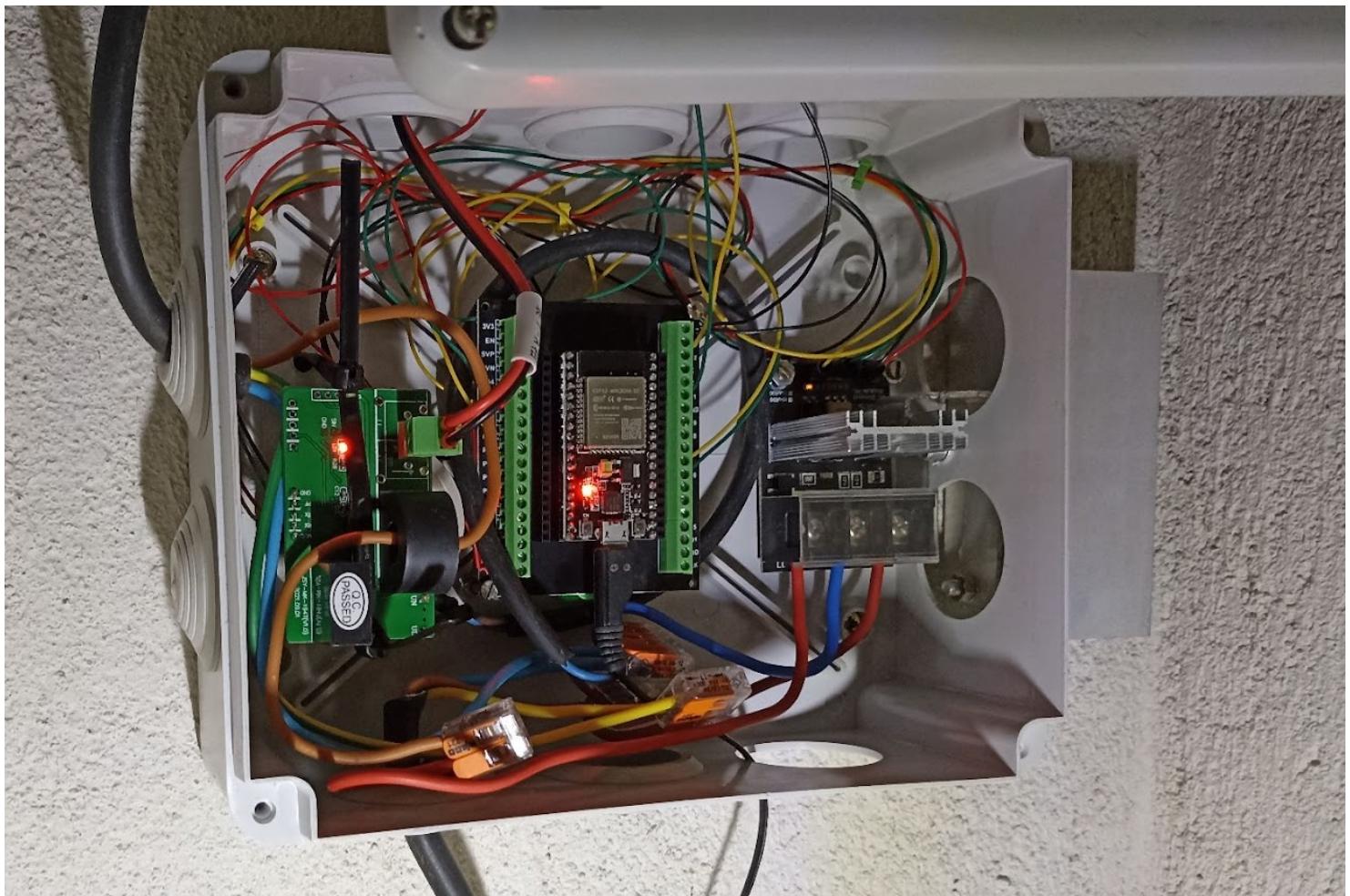


Avec le routeur :  
quasiment aucune injection

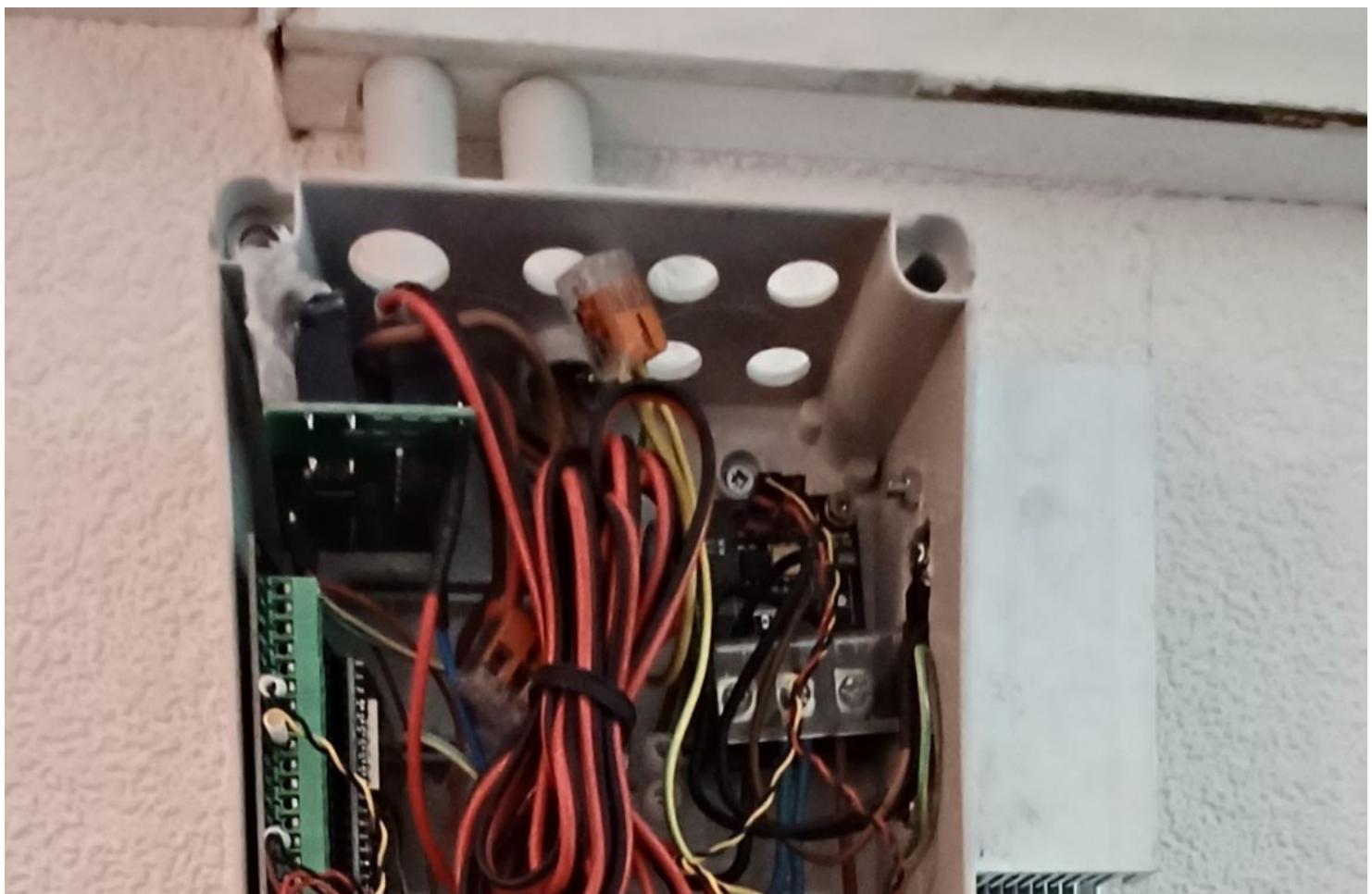


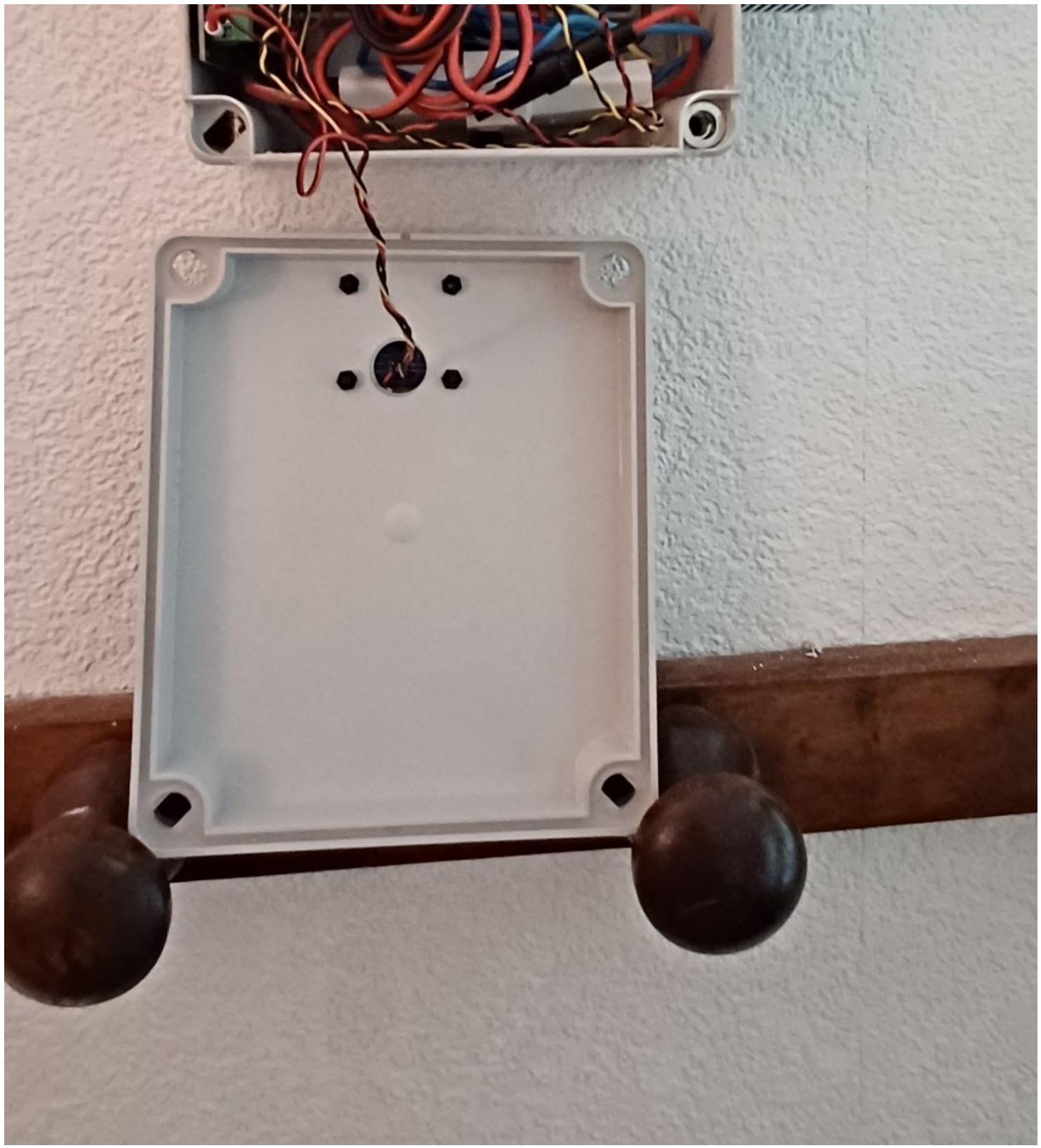
## Le routeur de François :





**Le routeur de Mathieu :**







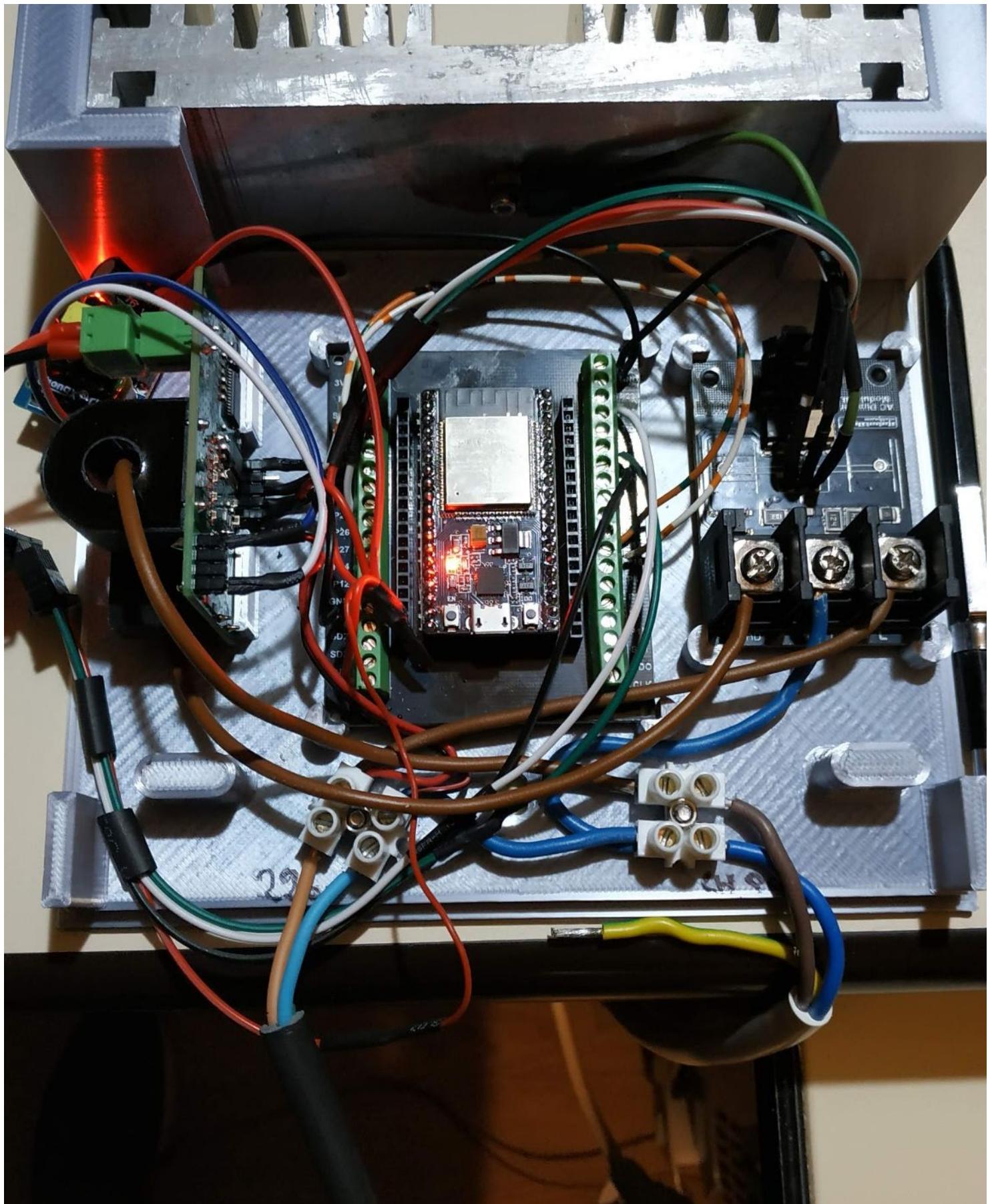
CE  
ATELIER



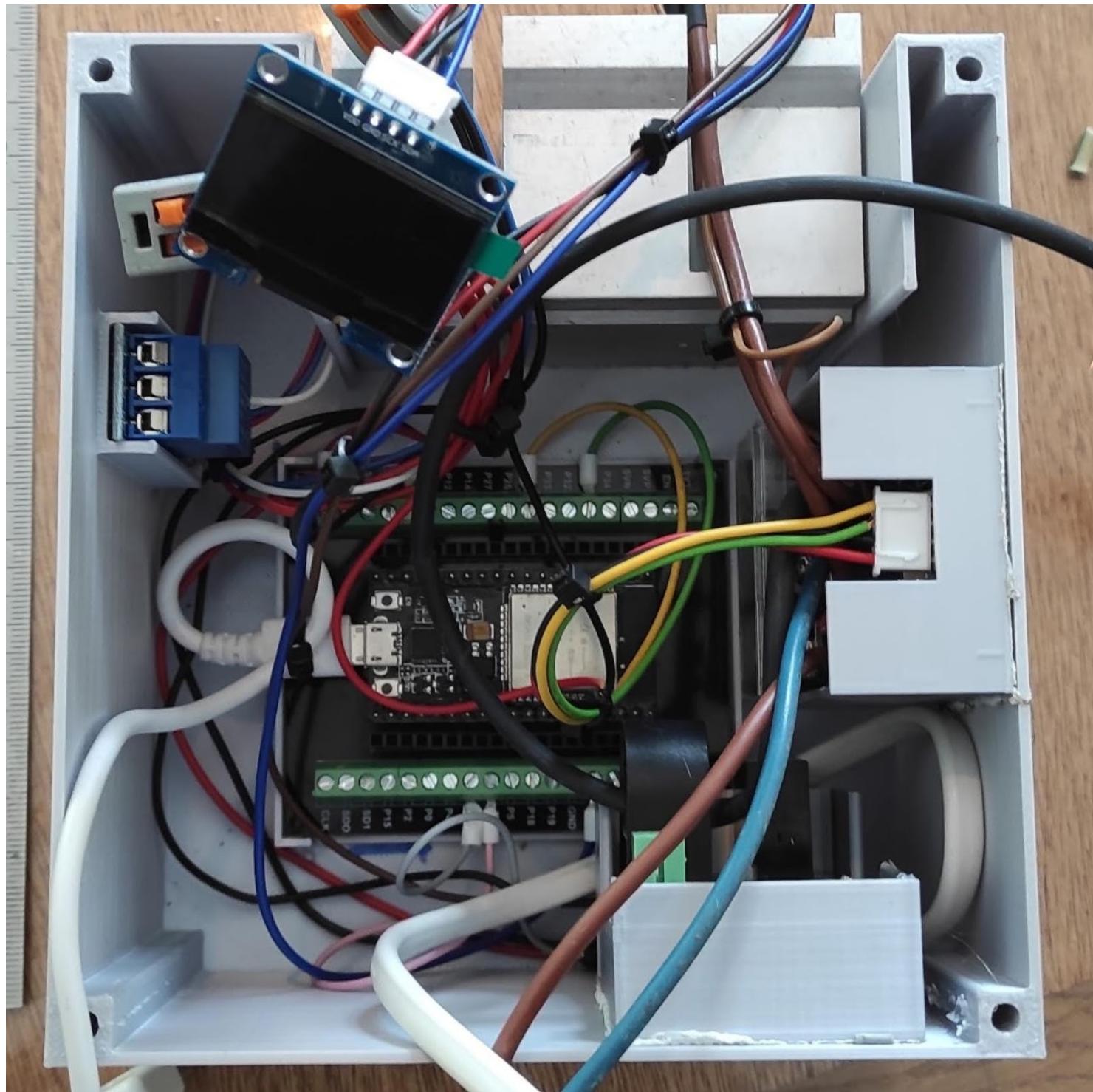
## le routeur de Daniel :

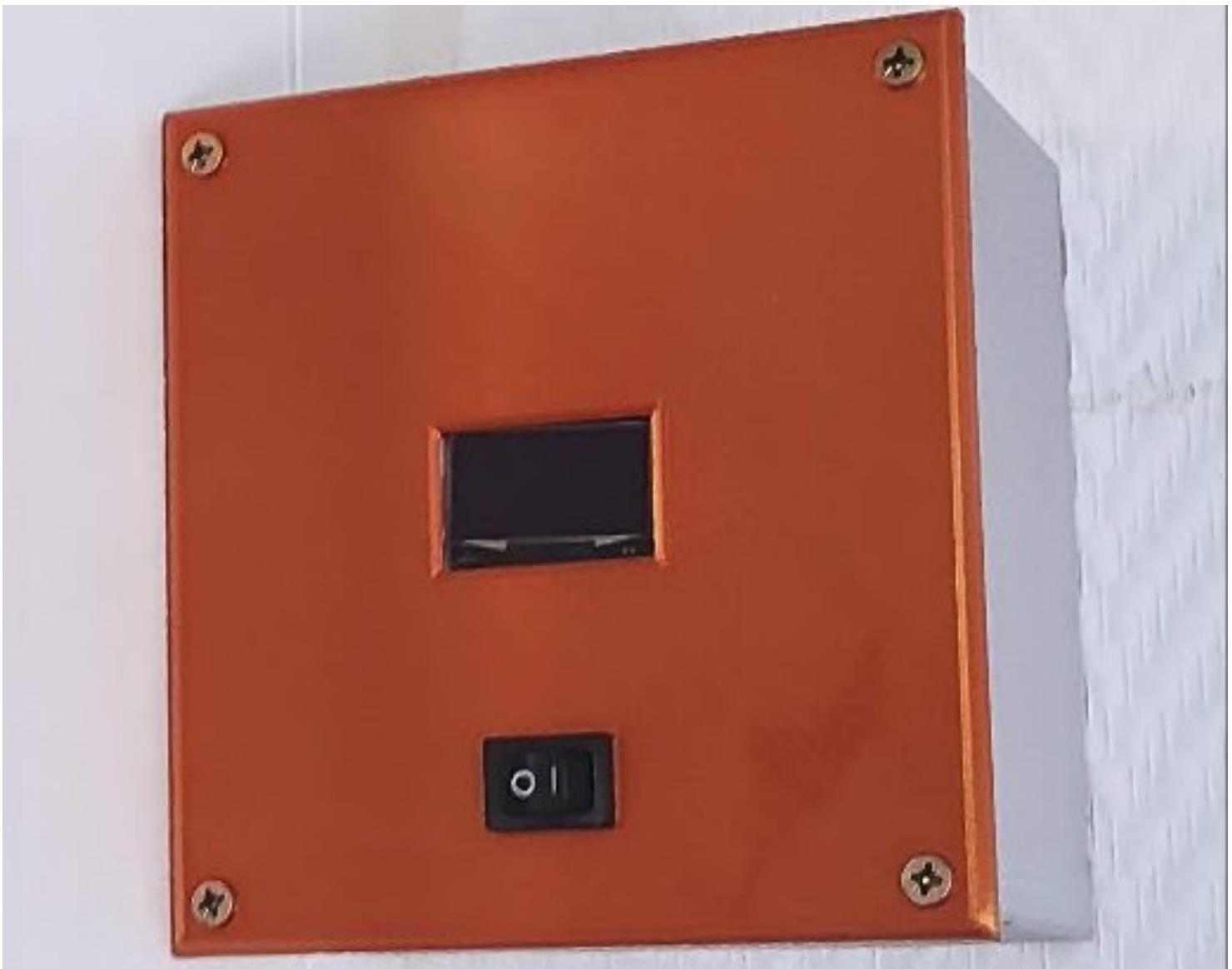
Bonjour et merci encore pour ce super projet qui a remplacer le PVmate qui a fait le boulot pendant 2 ans





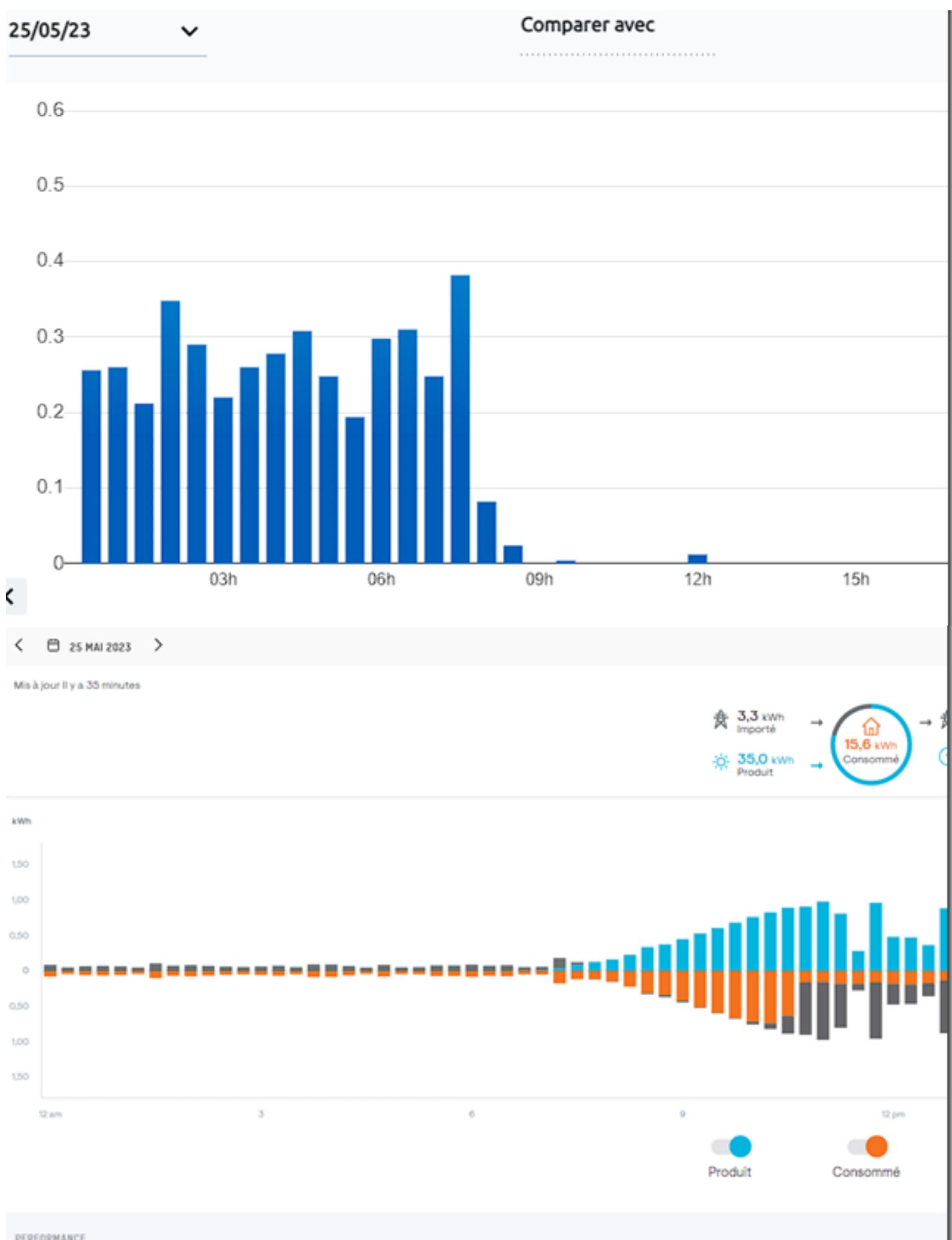
**Le routeur de Franck :**



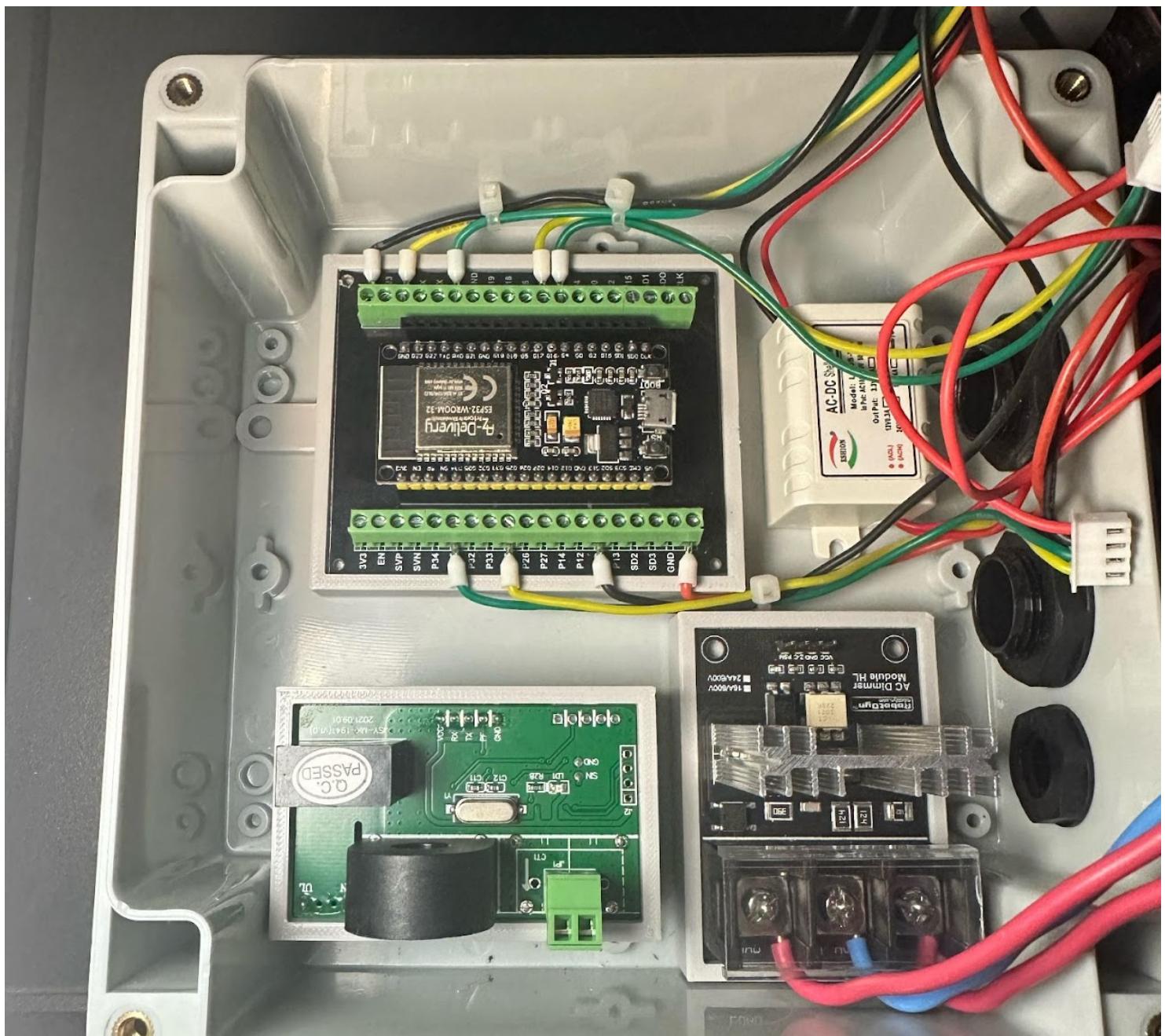




Avec le routeur :



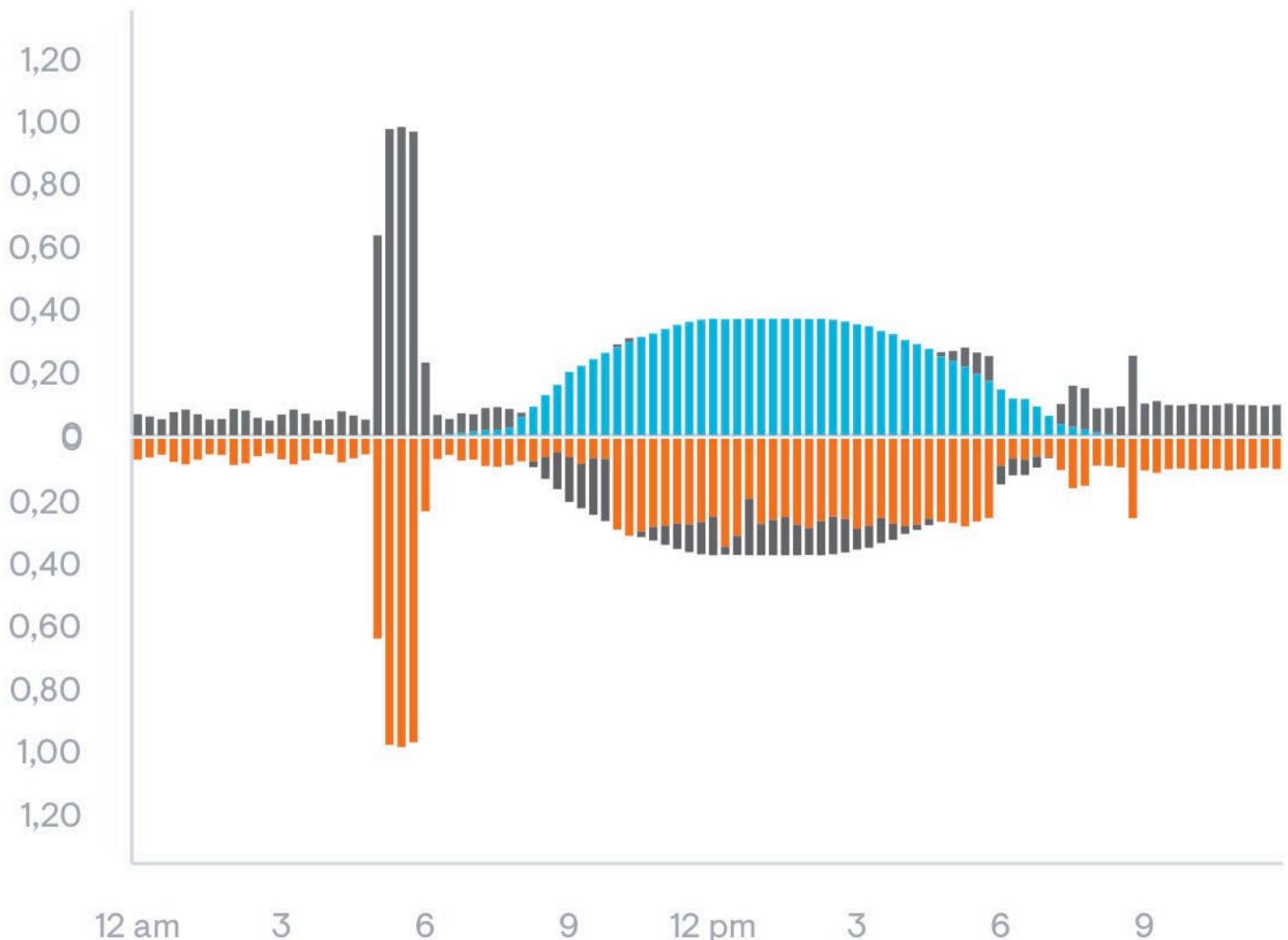
## Le routeur de Joris :



Avant le routeur :



12 am 3 6 9 12 pm 3 6 9

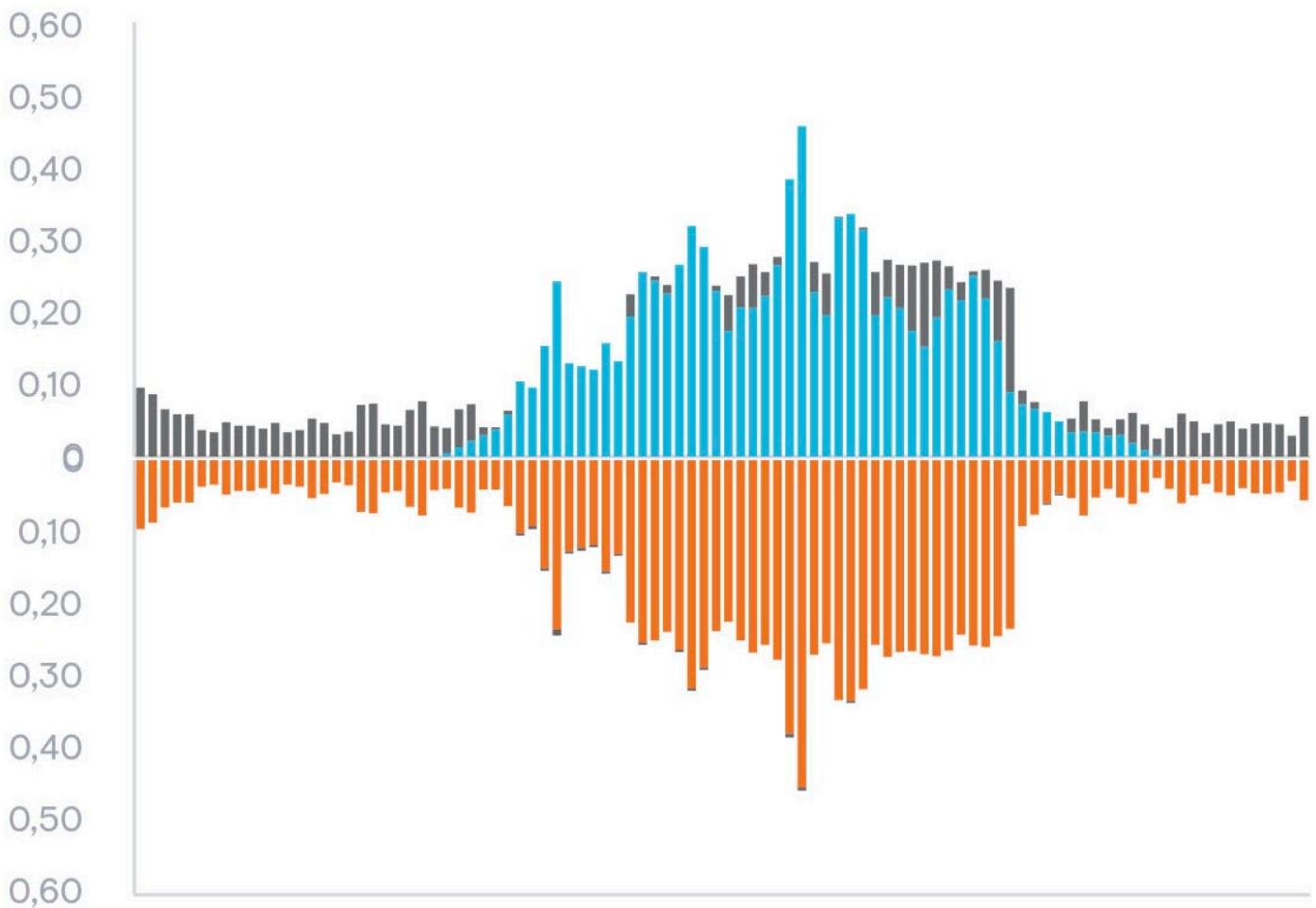




Avec le routeur :



kWh



12 am

3

6

9

12 pm

3

6

9

# Routeur Solaire



Puissance Surplus

1224 W



Energie Sauvée Aujoud'hui

2,75 kWh



Energie Sauvée Totale

161,9 kWh

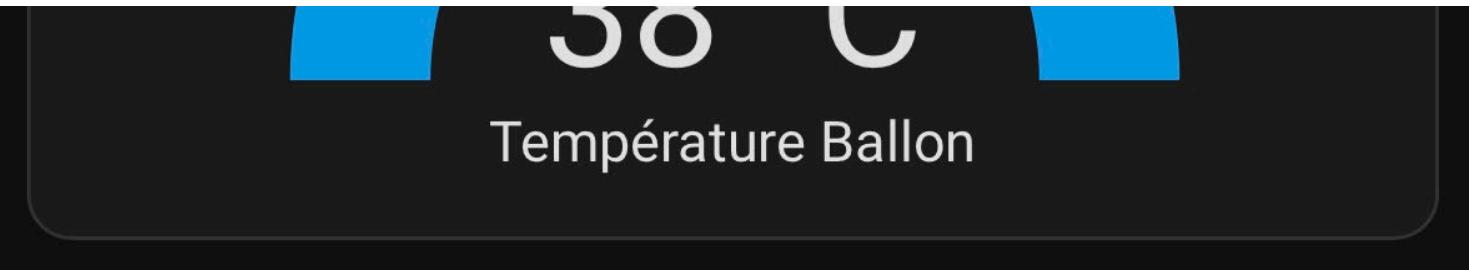


1224 W

Puissance Surplus

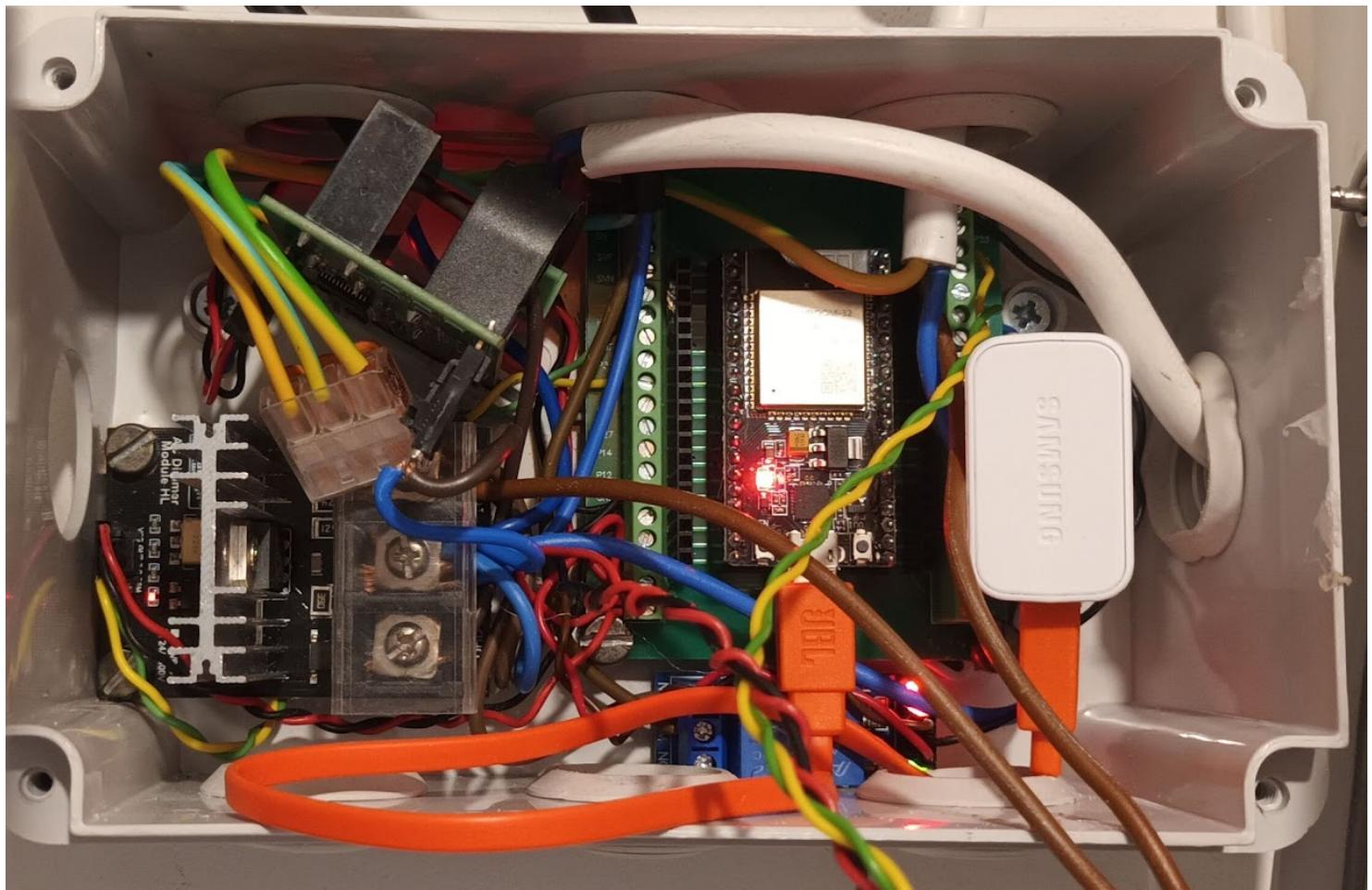


20 °C



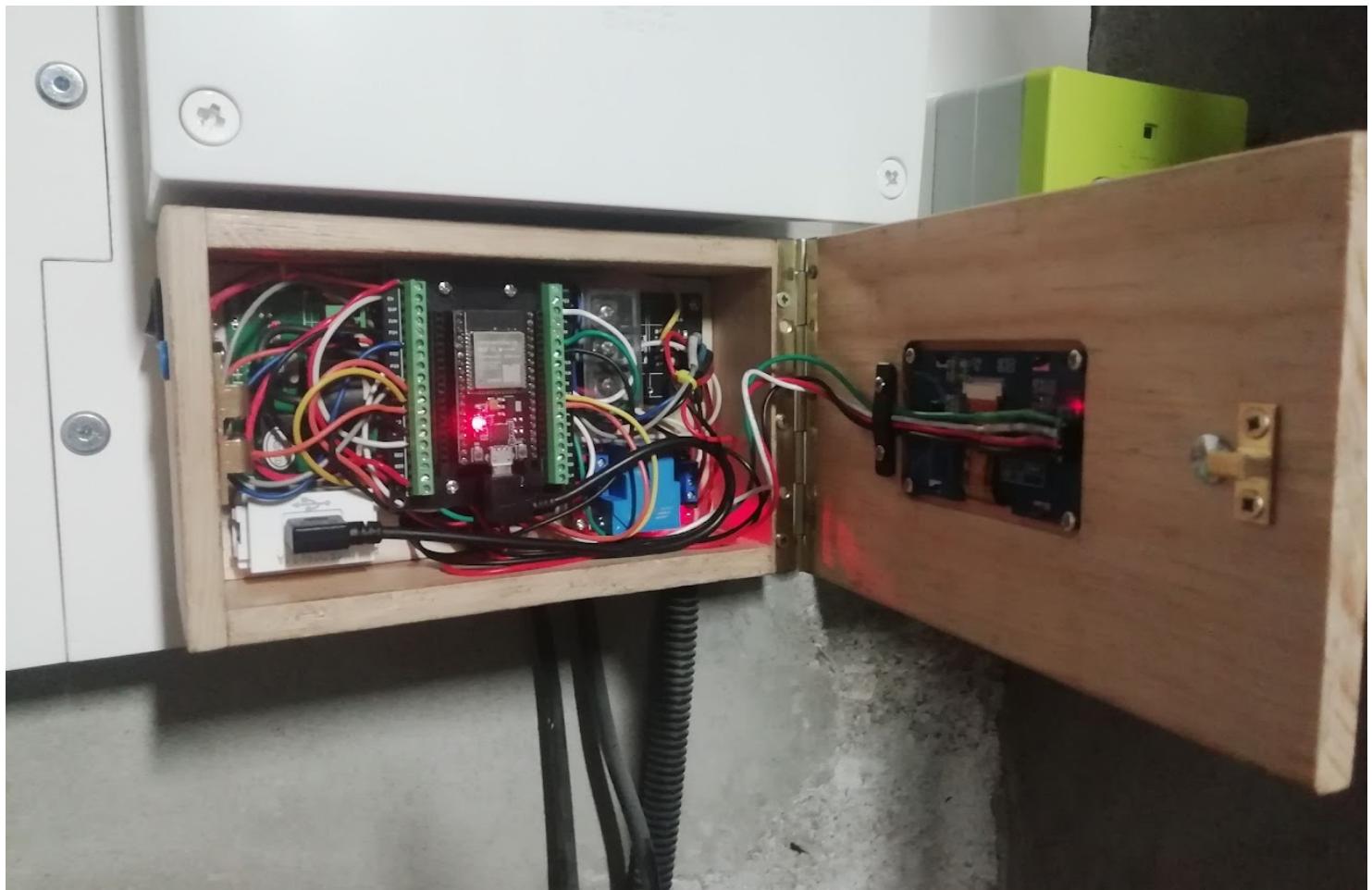
## Le routeur d'Arnaud :





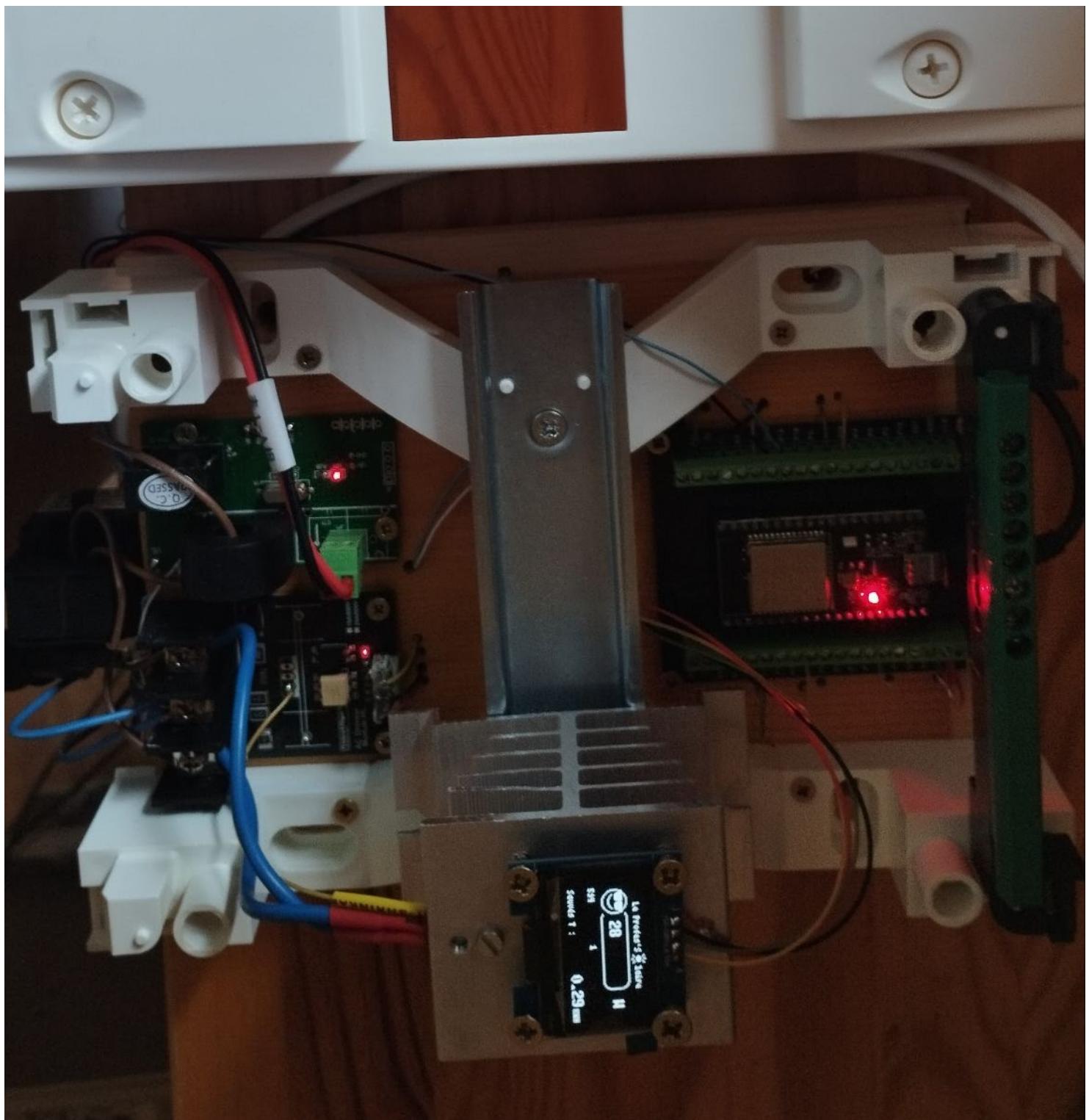
Le routeur de Grégory :

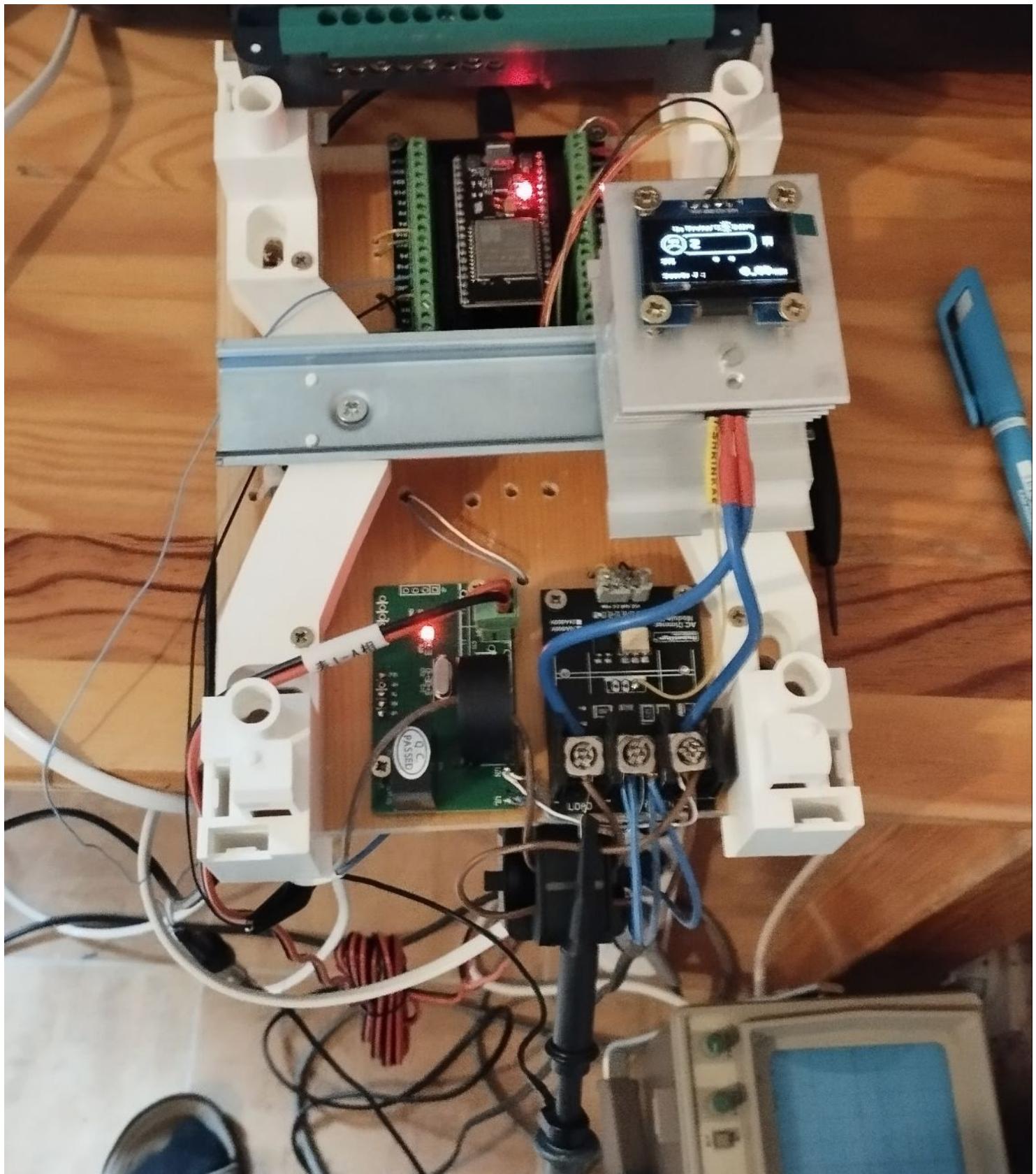




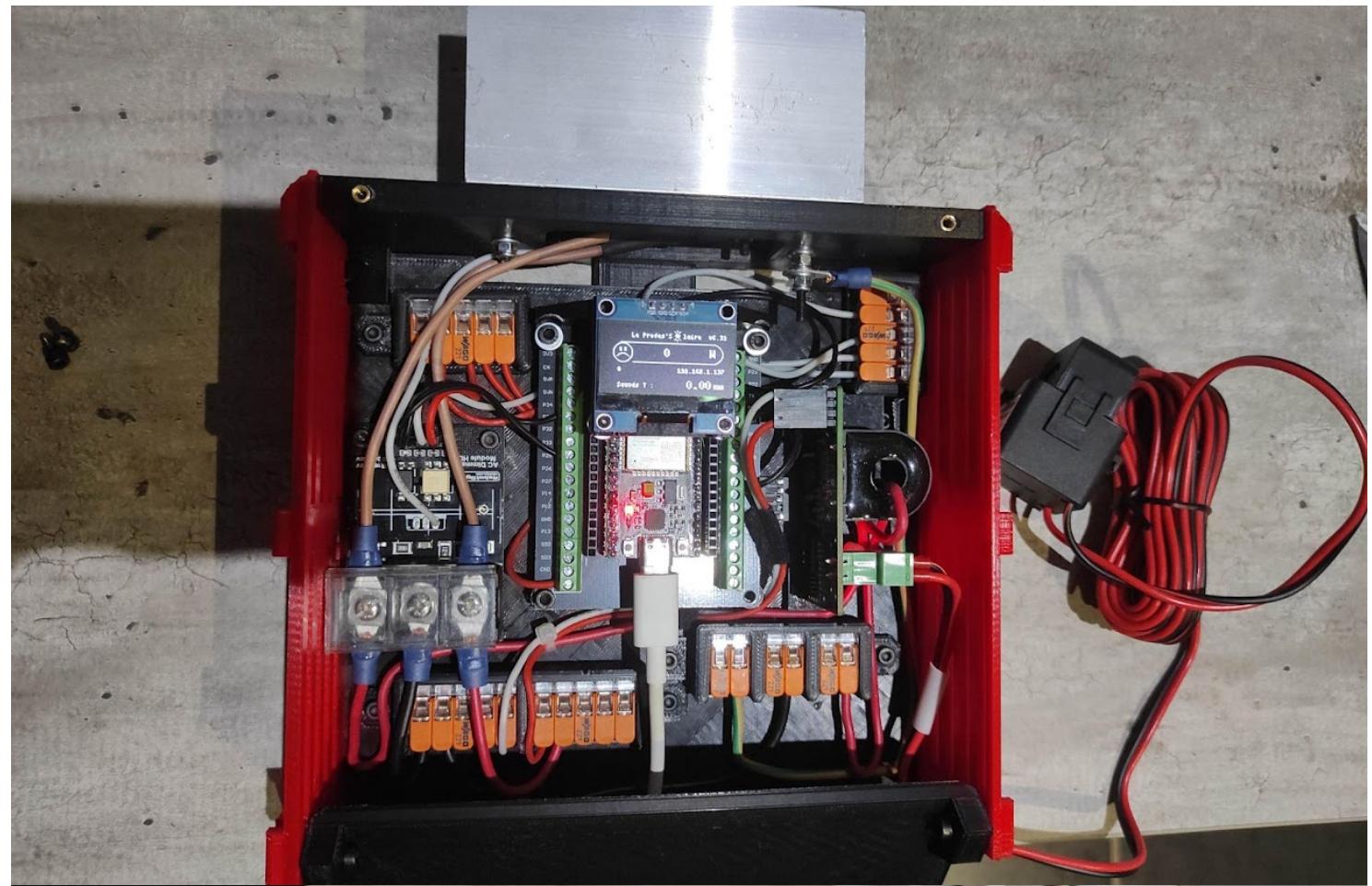
Le routeur de Jacques :

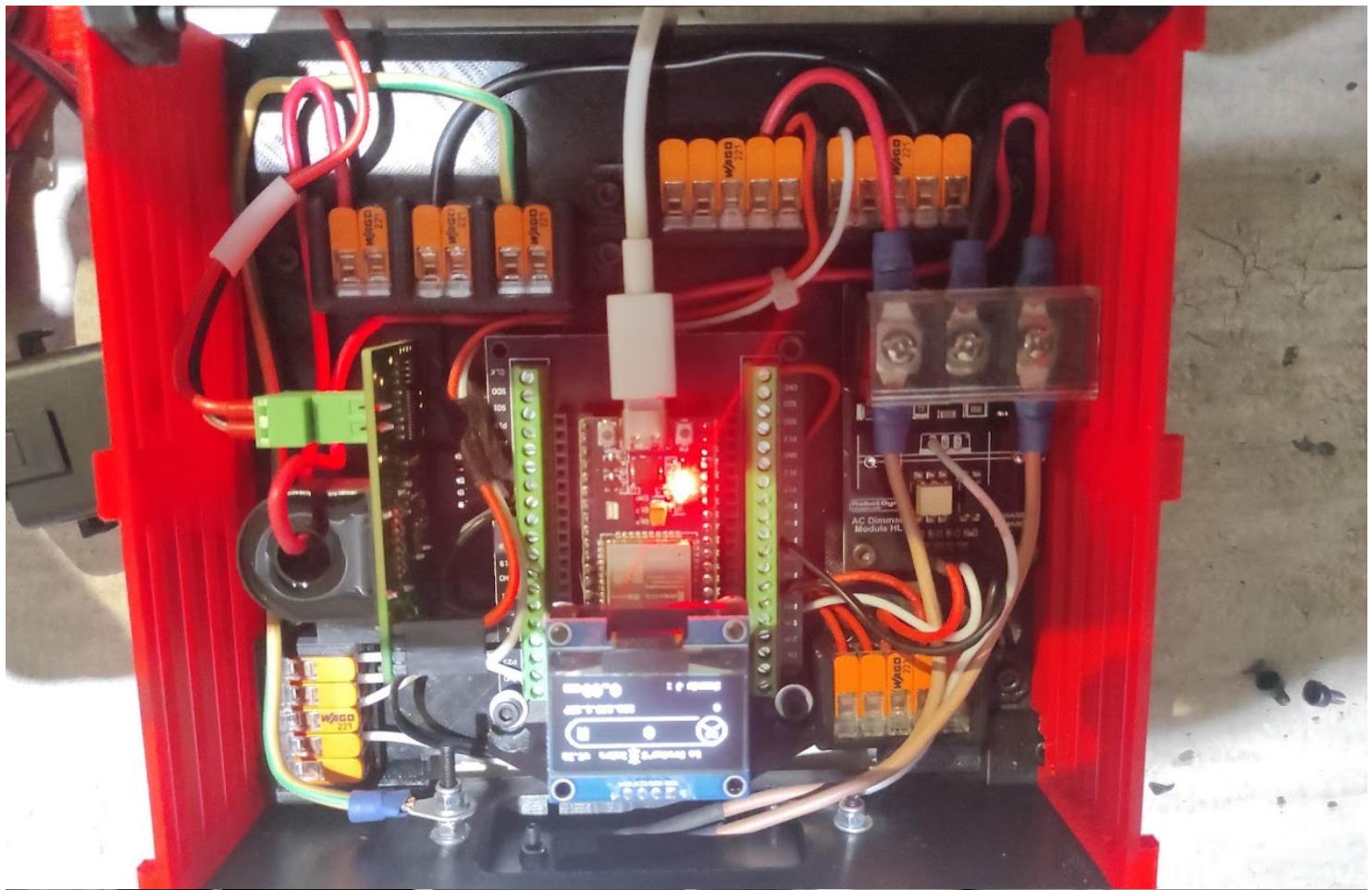






**Le routeur d'Anthony :**





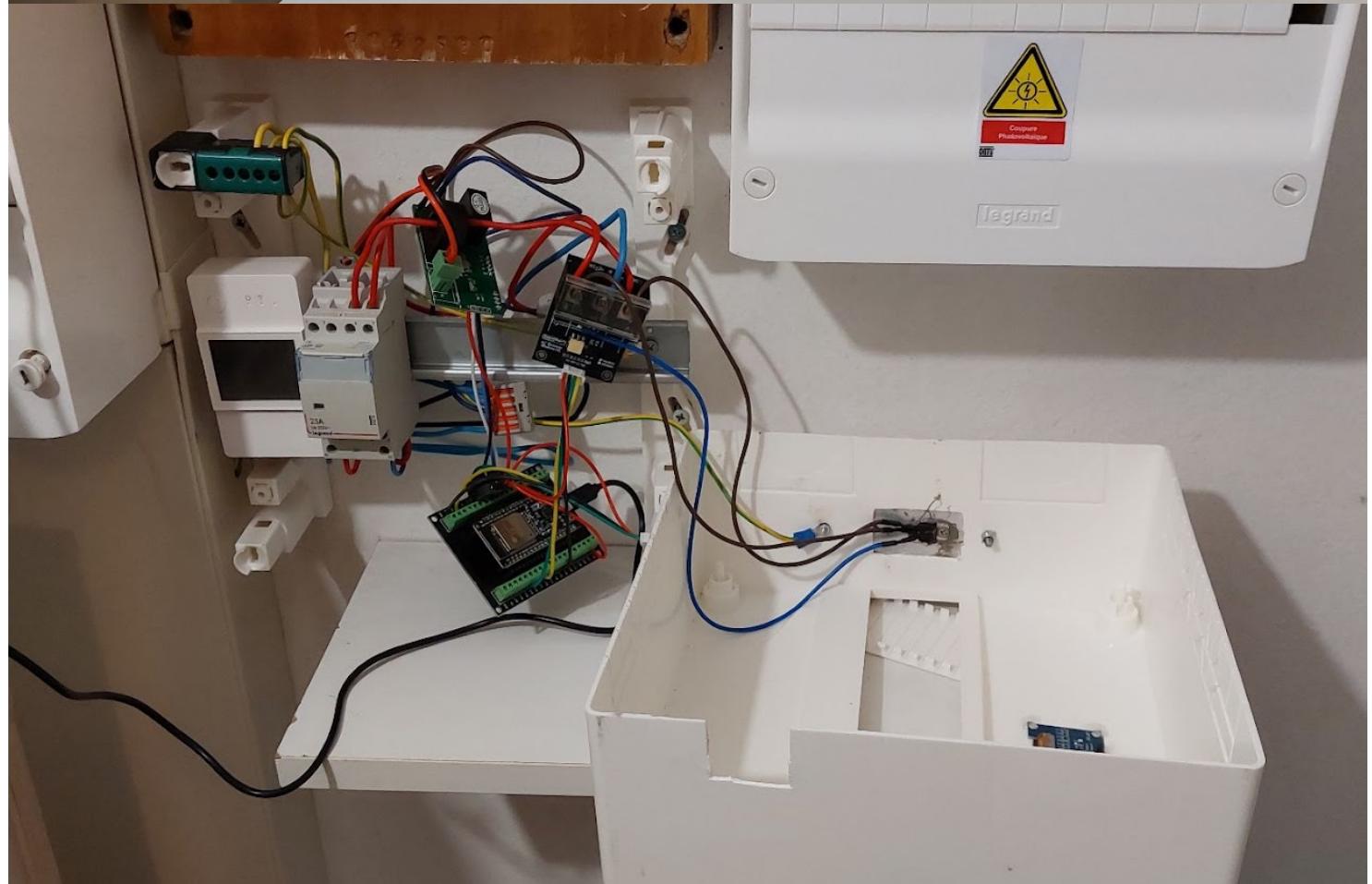
Routeur d'Adrien :



legrand



legrand



Sans routeur :

Mis à jour il y a 13 minutes

★ 8,9 kWh  
Importé

☀ 18,3 kWh  
Produit



★ 14,0 kWh  
Exporté

↑ 5,1 kWh  
Net  
exporté

kWh



1,00

0,80

0,60

0,40

0

-0,20

-0,40

-0,60

-0,80

-1,00

12 am

3

6

9

12 pm

3

6

9



Produit



Consommé



Importé/  
Exporté

Avec le routeur :

Mis à jour il y a 7 minutes

★ 1,5 kWh  
Importé

☀ 5,8 kWh  
Produit



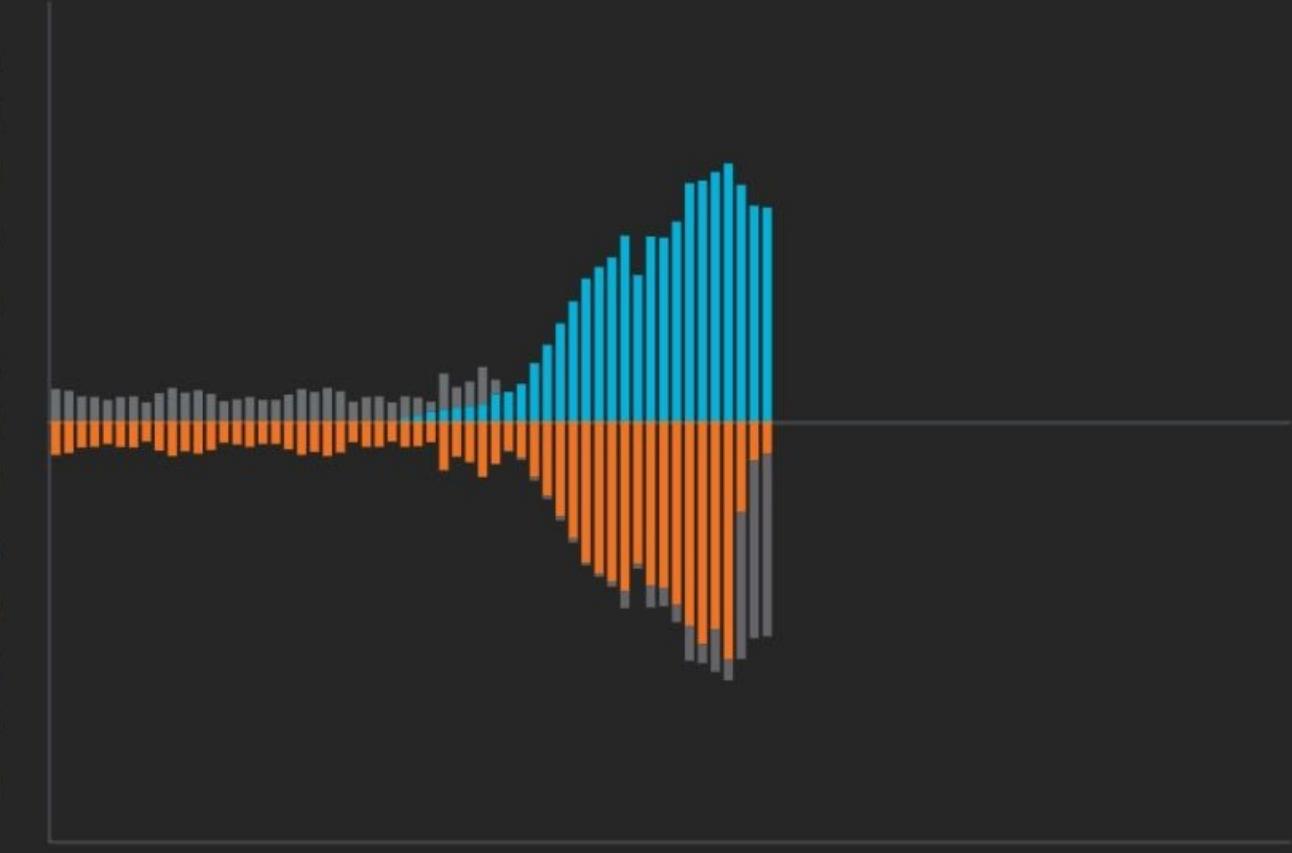
★ 1,2 kWh  
Exporté

⬇ 0,3 kWh  
Net  
importé

kWh



0,60  
0,50  
0,40  
0,30  
0,20  
0,10  
0  
-0,10  
-0,20  
-0,30  
-0,40  
-0,50  
-0,60



12 am 3 6 9 12 pm 3 6 9



Produit

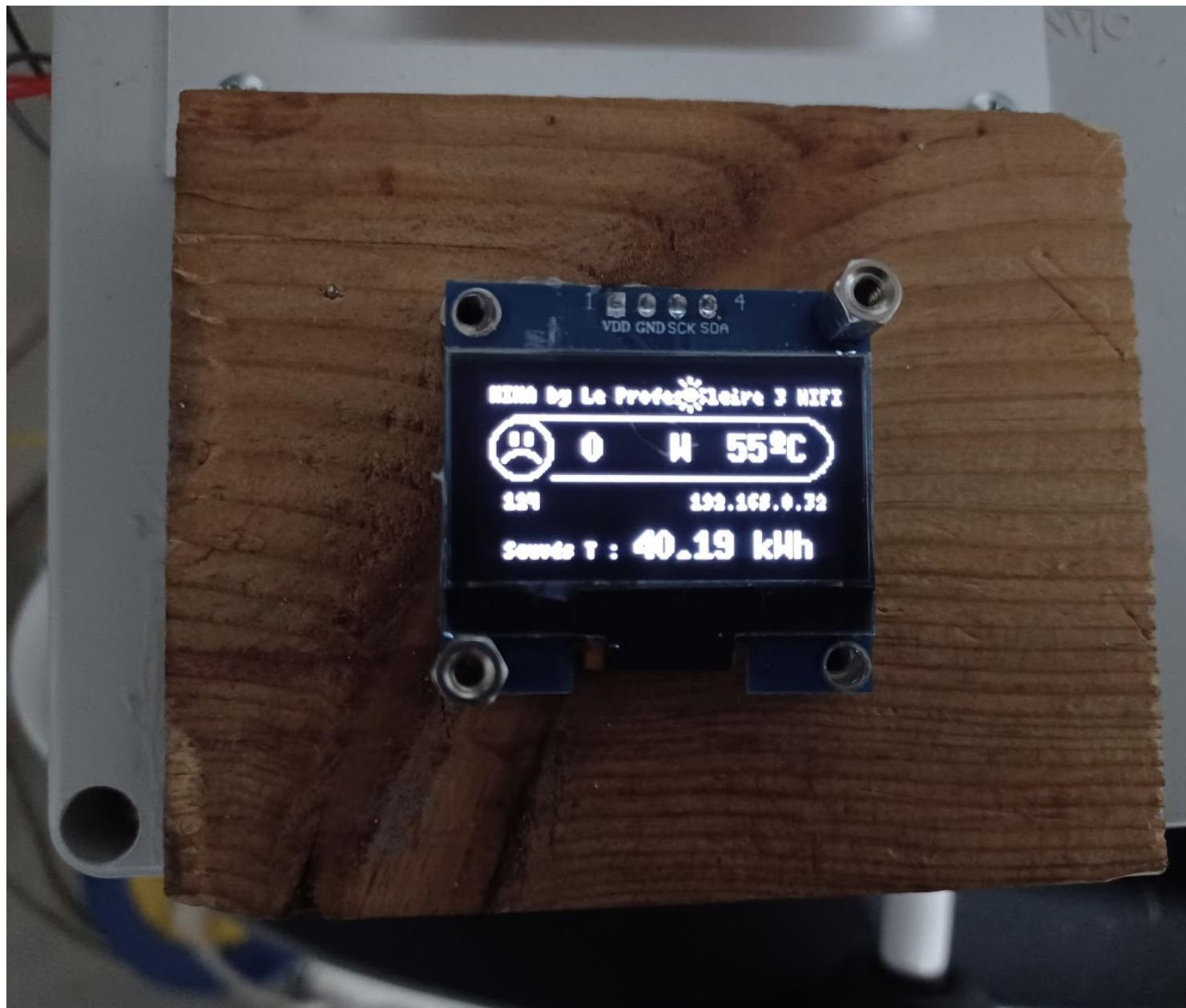


Consommé

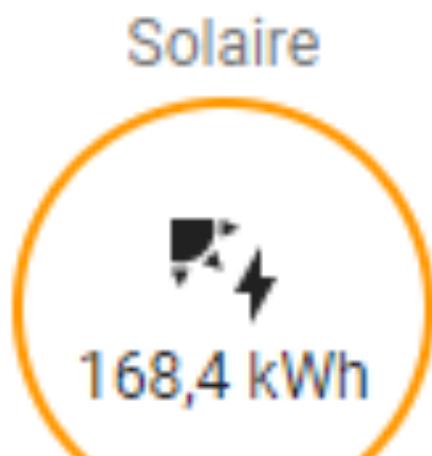


Importé/  
Exporté

Le routeur de Mickaël :

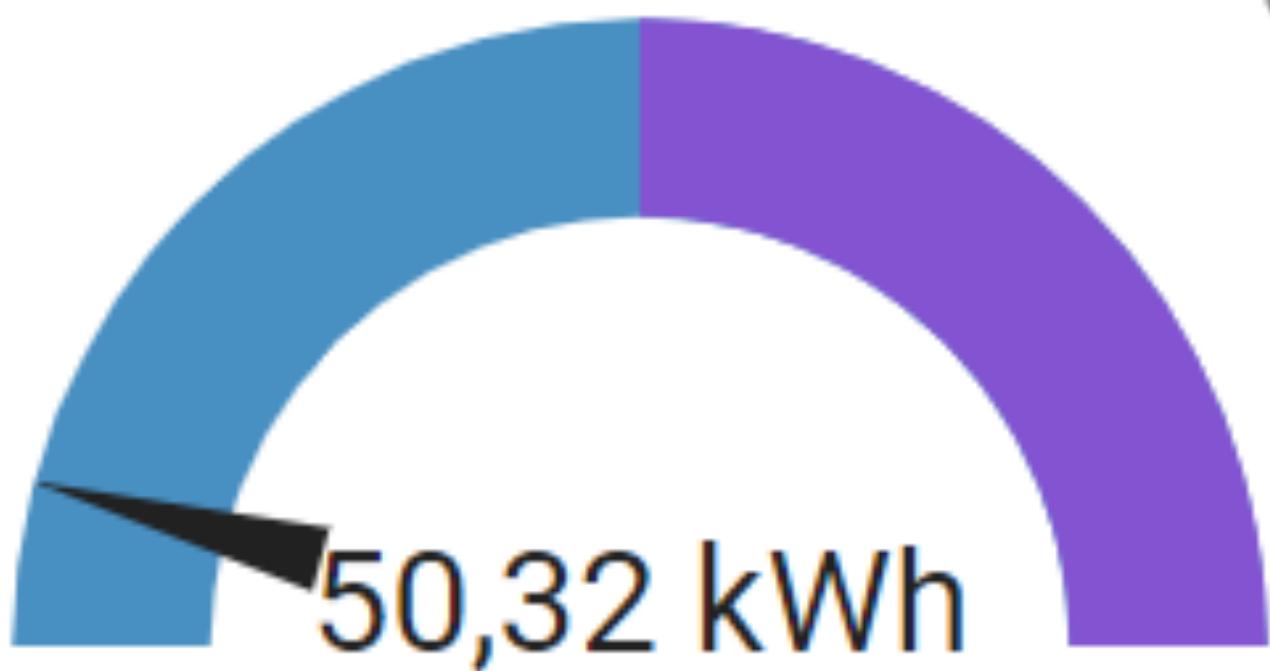


## Distribution d'énergie



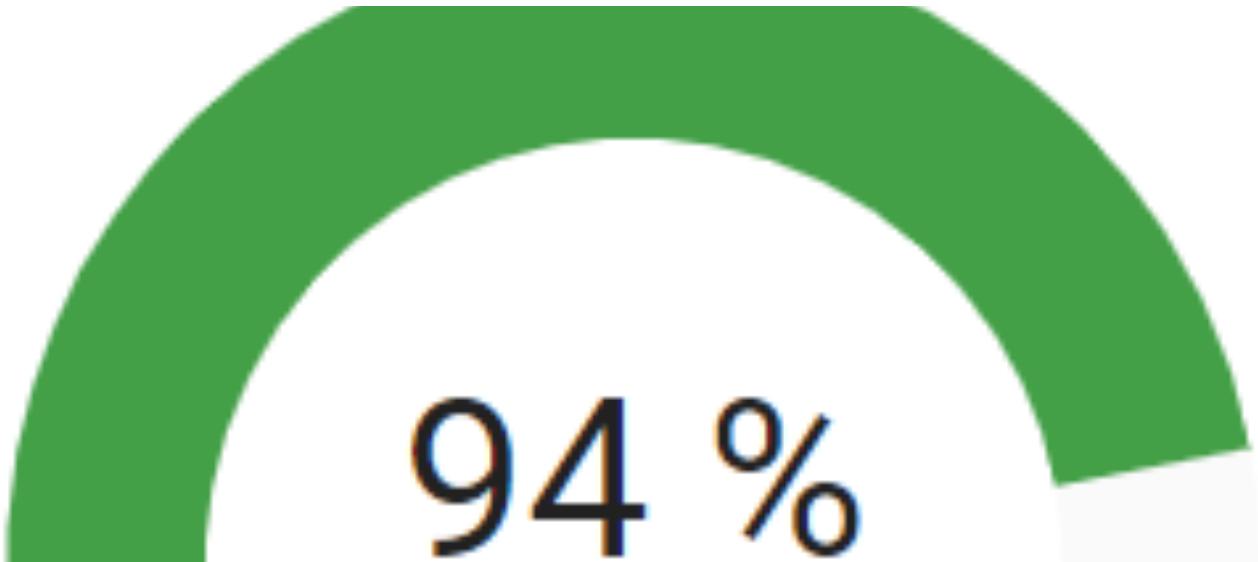


i



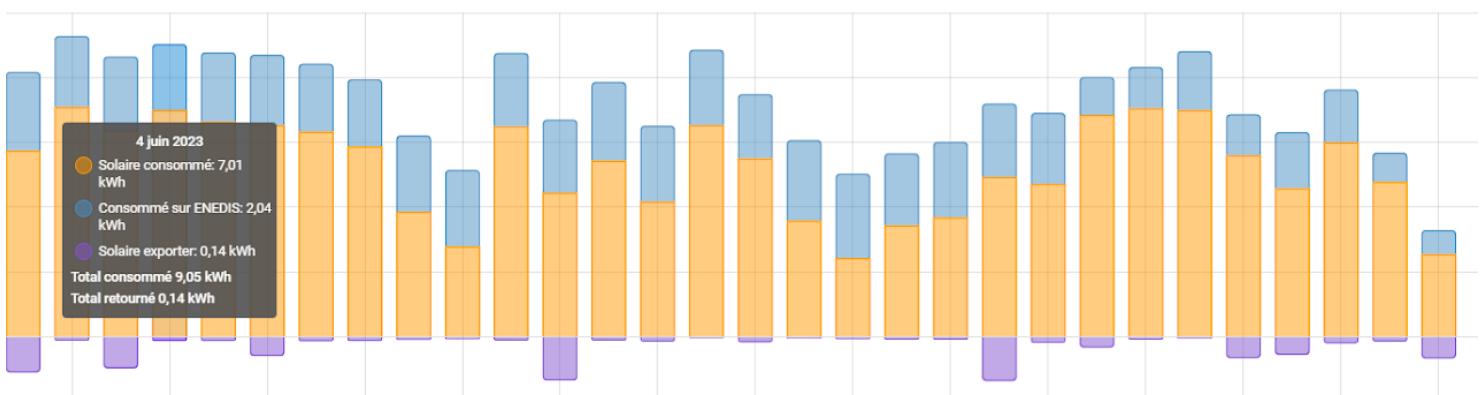
Consommation nette du réseau

i



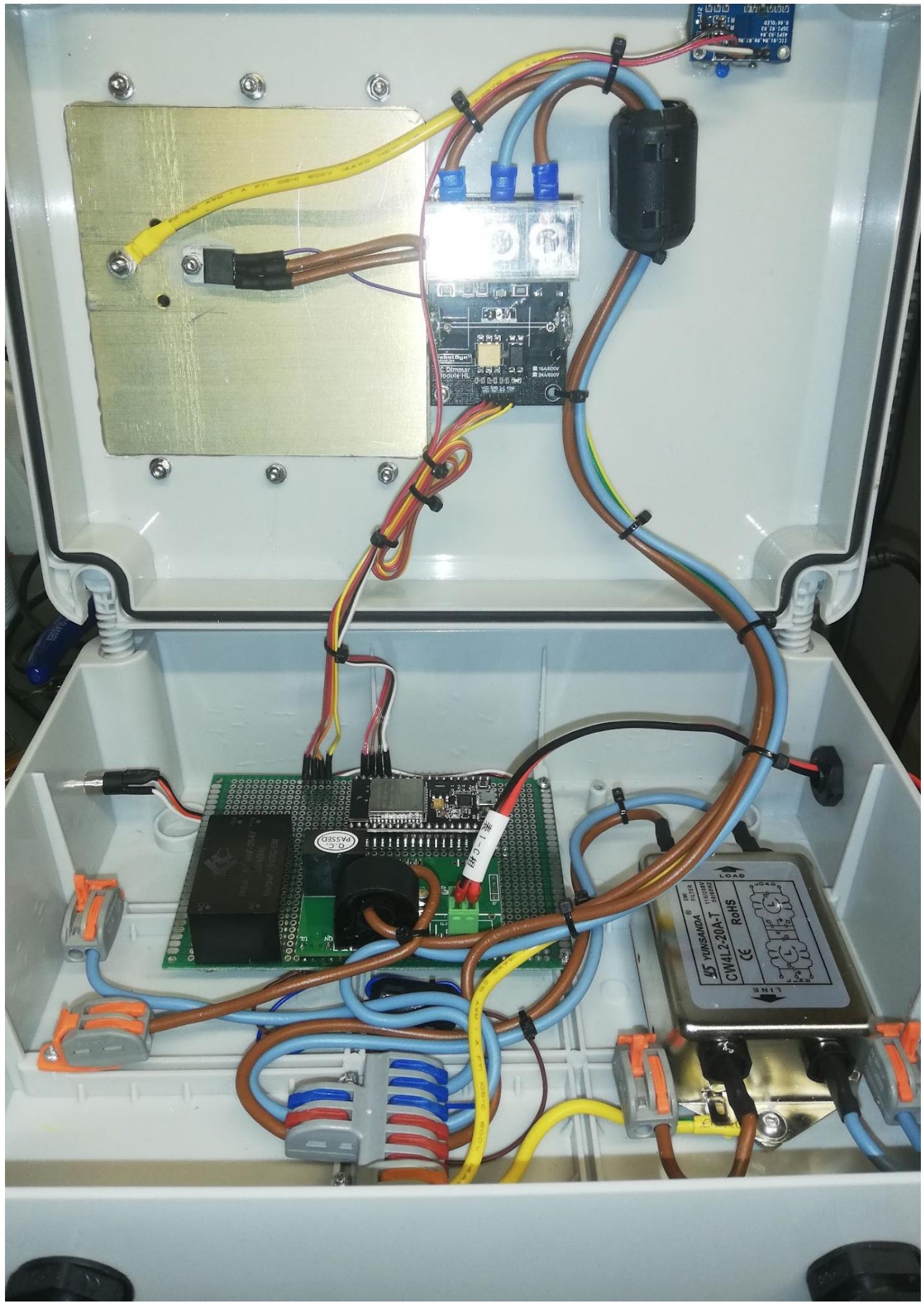
94 %

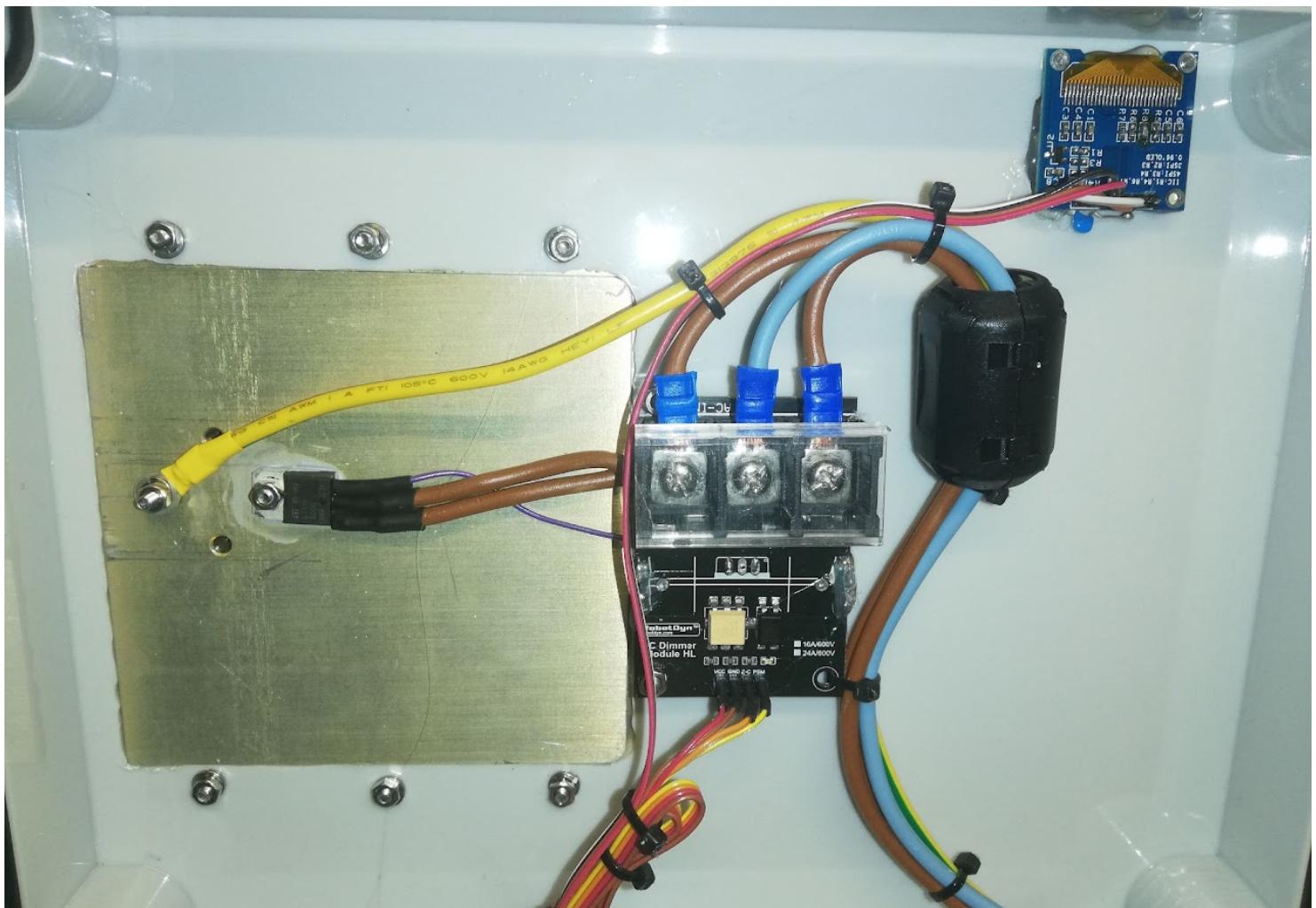
## Énergie solaire auto-consommée

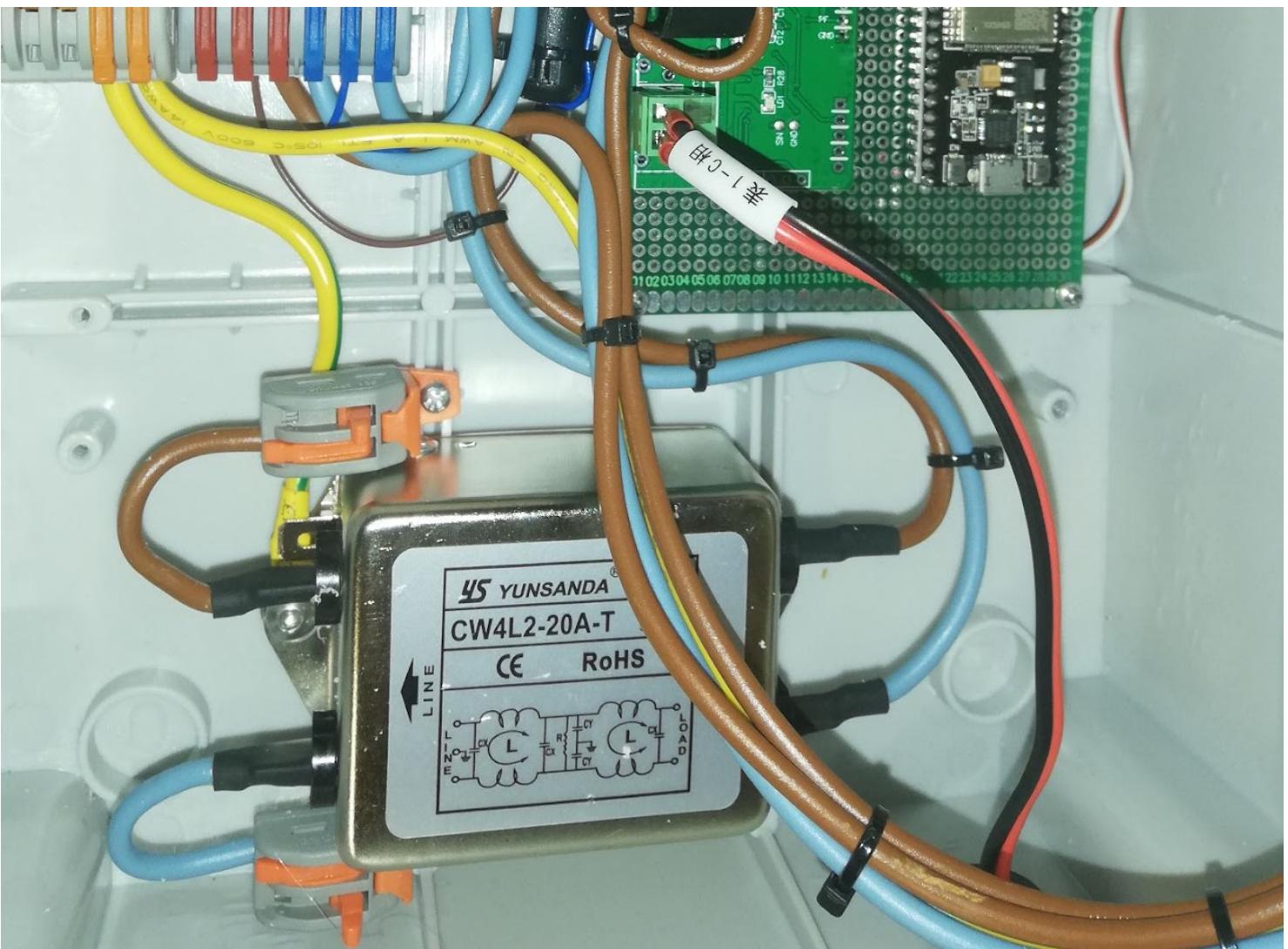


Le routeur de Sébastien :









professolaire@gmail.com - Siret : 52529320500015 © 2020-2023