

Systeme de monitoring batteries pour bateau

Prévu pour surveiller la tension et le
courant de charge/décharge de 3
batteries 12V

Jérémy LANIEL et Yann HELLO

www.kp3d.re

Produit final



Contrôleur batteries V1

Ce boîtier permet de suivre l'état de vos batteries, ainsi que la température et le taux d'humidité, directement sur votre tablette ou Smartphone.

Une évolution existe avec 3 répéteurs LCD intégrés au boîtier.

A suivre, l'intégration d'un monitoring des entrées d'énergie (panneaux solaire, etc)

Fonctionnement

Vidéo : <https://youtu.be/l73krX4ICE8>

Branchement

Images à venir

Réalisation : matériel

Tous les composants électroniques s'achètent pour une poignée d'euros sur des sites comme Banggood, Amazon, Wish ou autres.

Le boîtier a été dessiné en CAO et construit dans « matériau ? » avec une imprimante 3D. Nous vous fournissons les fichiers STL si vous désirez les imprimer vous même.

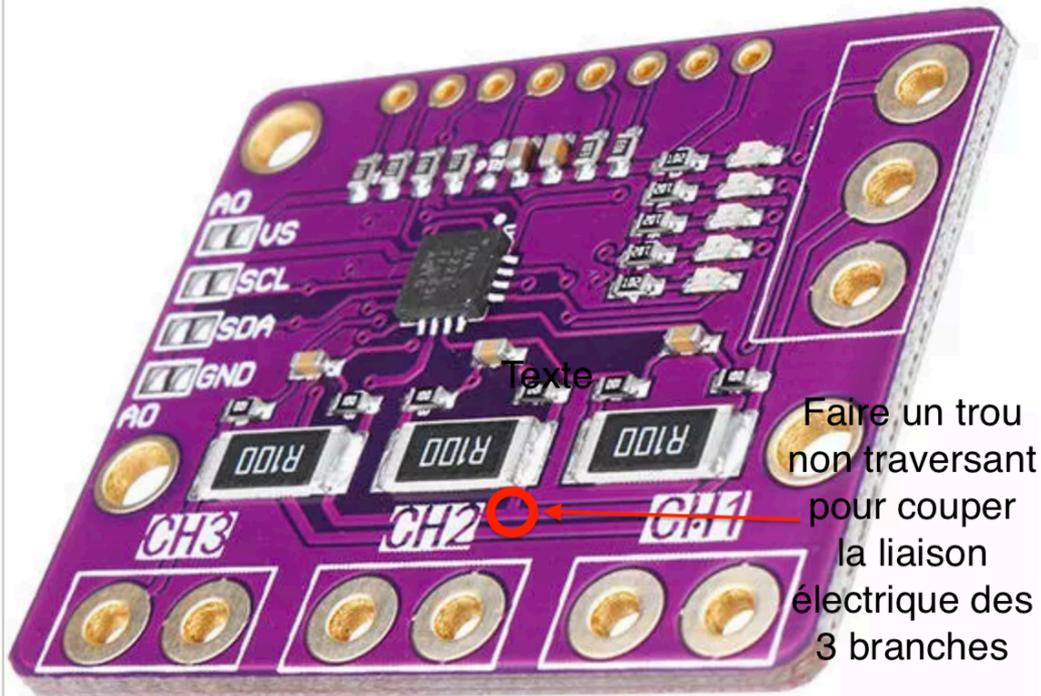
La liste des composants à acheter est ici :

- Carte WEMOS D1 (le cerveau) et son câble USB ;
- Module CJMCU 3221 (le capteur de tensions/courants à base du chip INA3221). On y fera quelques modifications mineures pour l'adapter à notre besoin ;
- Les 3 shunts 0,75 mOhms pour mesurer l'image du courant des batteries (https://www.banggood.com/External-Shunt-FL-2-100A75mV-50A75mV-Current-Meter-Shunt-Current-Shunt-Resistor-For-Digital-Amp-Meter-Analog-Meter-p-1418071.html?rmmds=search&ID=517855&cur_warehouse=CN) ;
- DHT11 (capteur de T° et humidité) ;
- Un interrupteur (avec voyant de préférence pour indiquer le ON) (https://www.banggood.com/12V-10A-22mm-LED-Autolock-Power-Push-Button-Switch-ONOff-3-Colors-p-1036447.html?rmmds=search&ID=229&cur_warehouse=CN) ;
- Des passes-fils pour maintenir les câbles en sortie du boîtier (https://www.banggood.com/20pcs-Rubber-Wiring-Grommets-Ring-Cable-Protector-5678mm-Inner-Dia-p-1070122.html?rmmds=search&ID=2243636&cur_warehouse=CN) ;
- Des connecteurs étanches pour le branchement du boîtier aux batteries (https://www.amazon.fr/Superseal-%C3%A9tanche-%C3%A9lectrique-terminal-connecteur/dp/B074MY5ZH6/ref=sr_1_75?mk_fr_FR=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crd=32SBURYJQCO5&keywords=connecteur+electrique&qid=1558280357&refinements=p_76%3A437878031&rnid=437877031&ps=1&s=gateway&sprefix=connecteur%2Caps%2C277&sr=8-75) ;
- Des cosses pour se brancher sur les batteries ;
- Du fil électrique.

Réalisation : CJMCU 3221 (base INA3221)

Modifications afin de rendre les 3 voies de mesures indépendantes

Etape 1

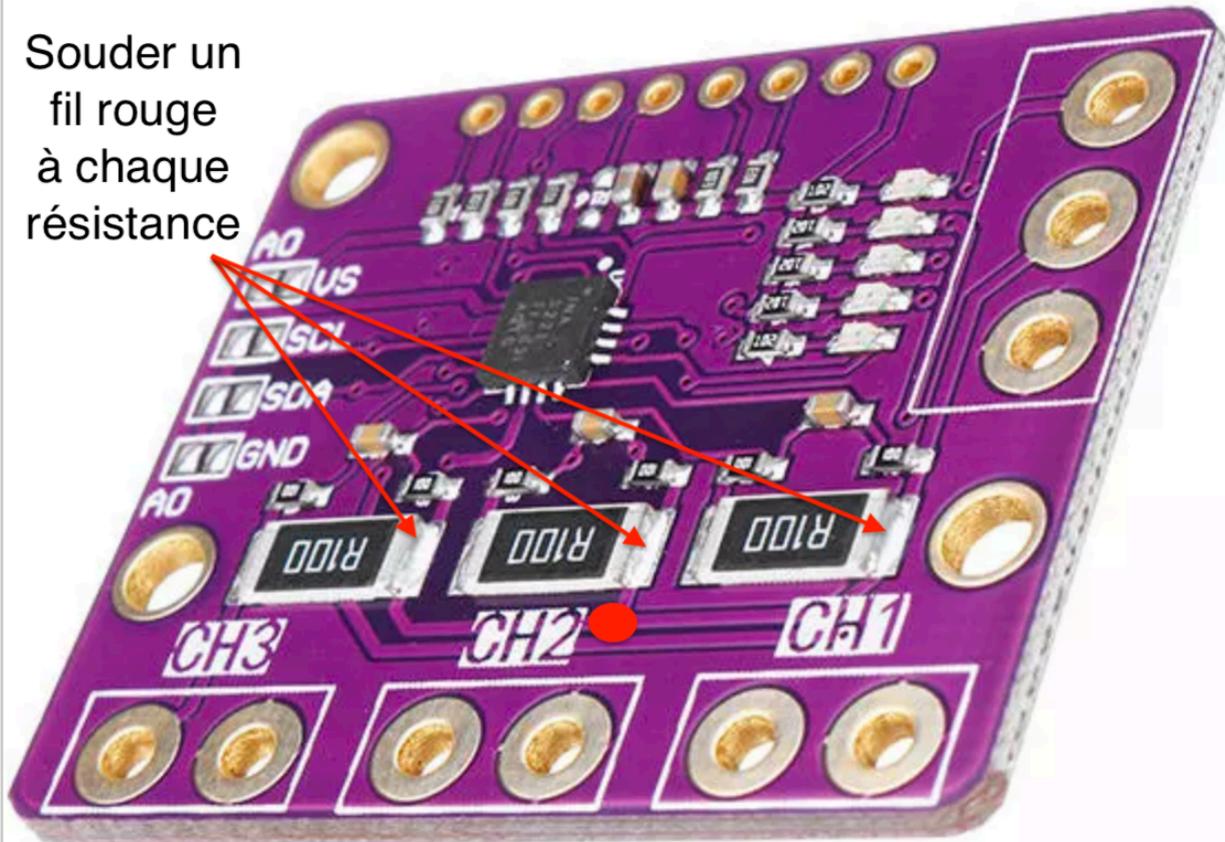


Réalisation : CJMCU 3221 (base INA3221)

Modifications afin de rendre les 3 voies de mesures indépendantes

Etape 2

Souder un
fil rouge
à chaque
résistance

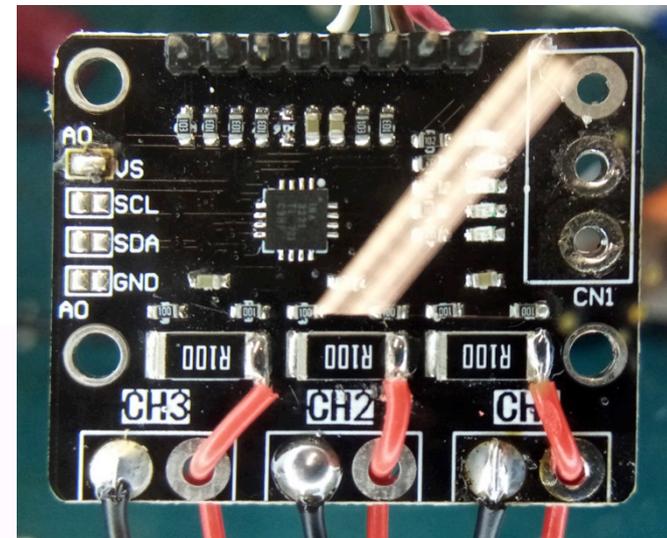
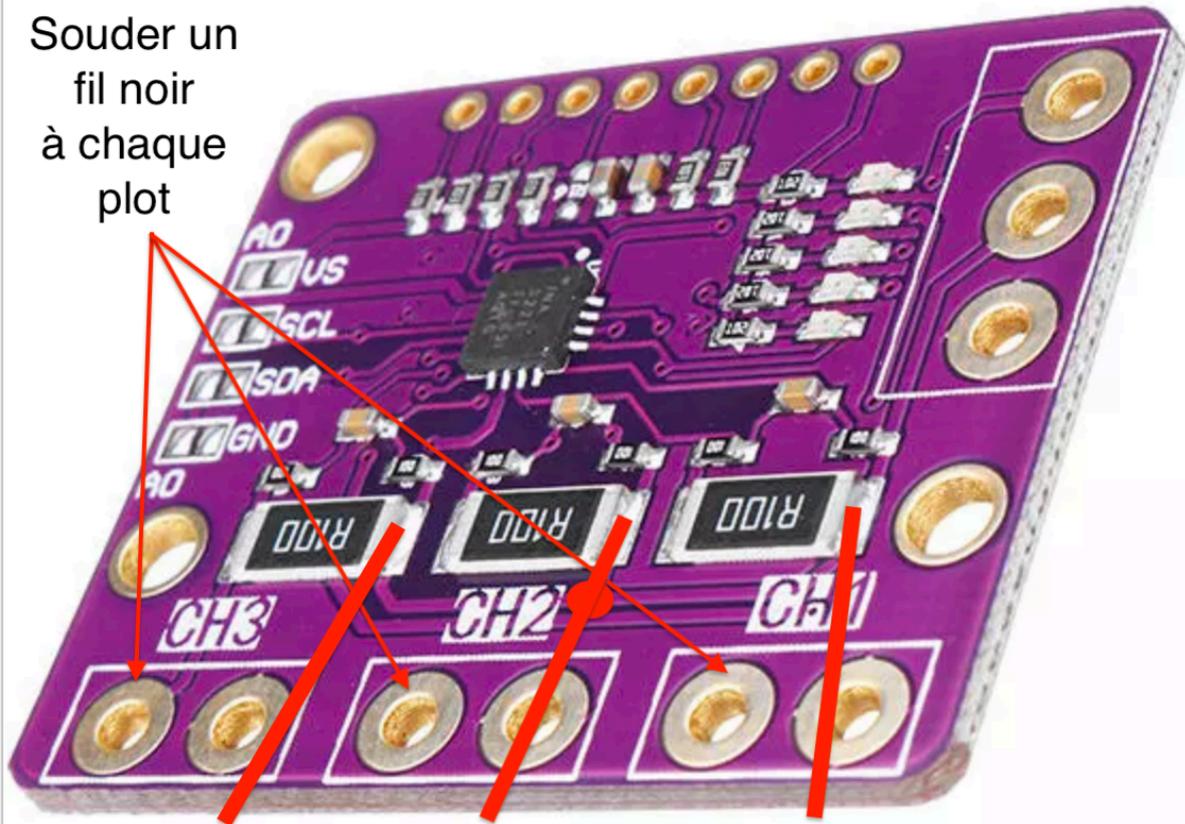


Réalisation : CJMCU 3221 (base INA3221)

Modifications afin de rendre les 3 voies de mesures indépendantes de ce module

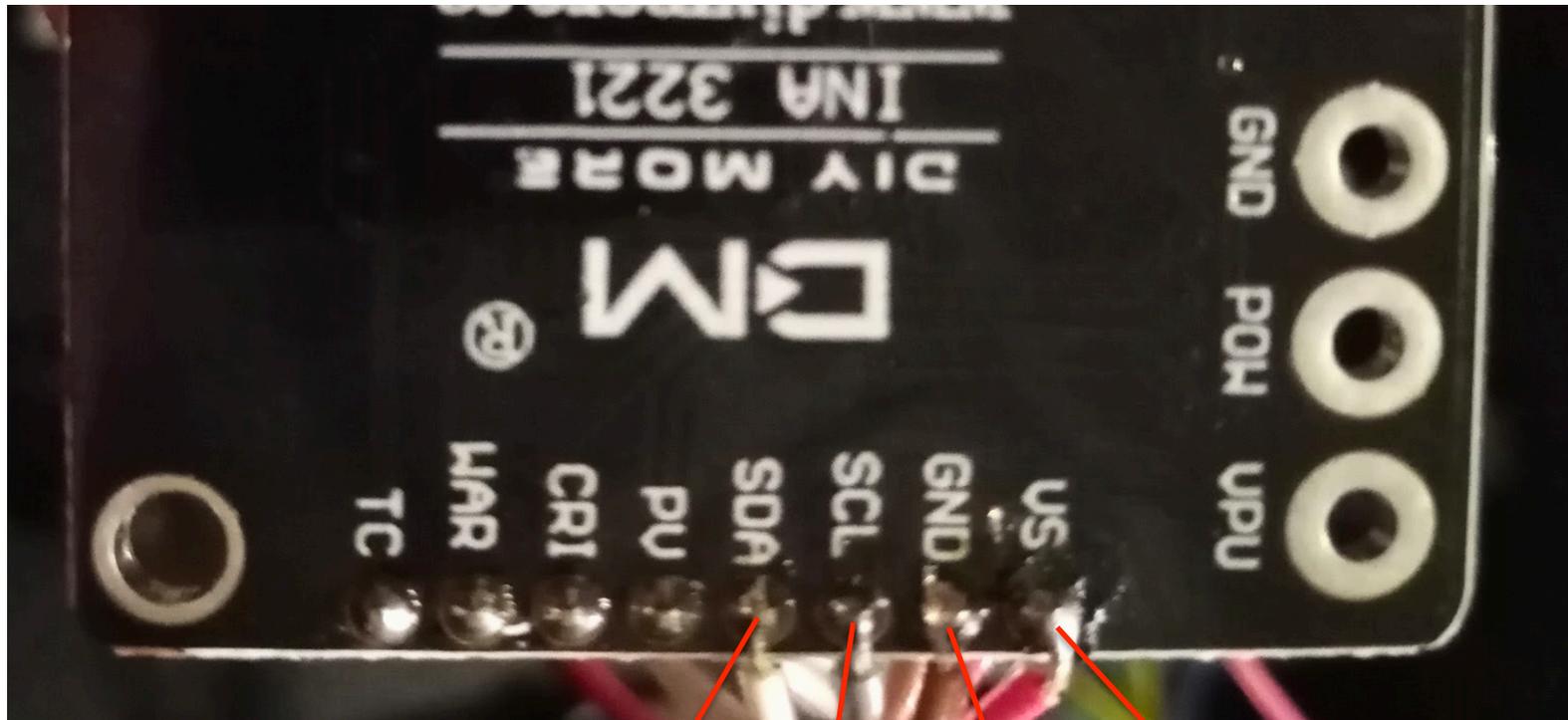
Etape 3

Souder un fil noir à chaque plot



Réalisation : CJMCU 3221 (base INA3221)

Branchement du module (vers carte numérique)



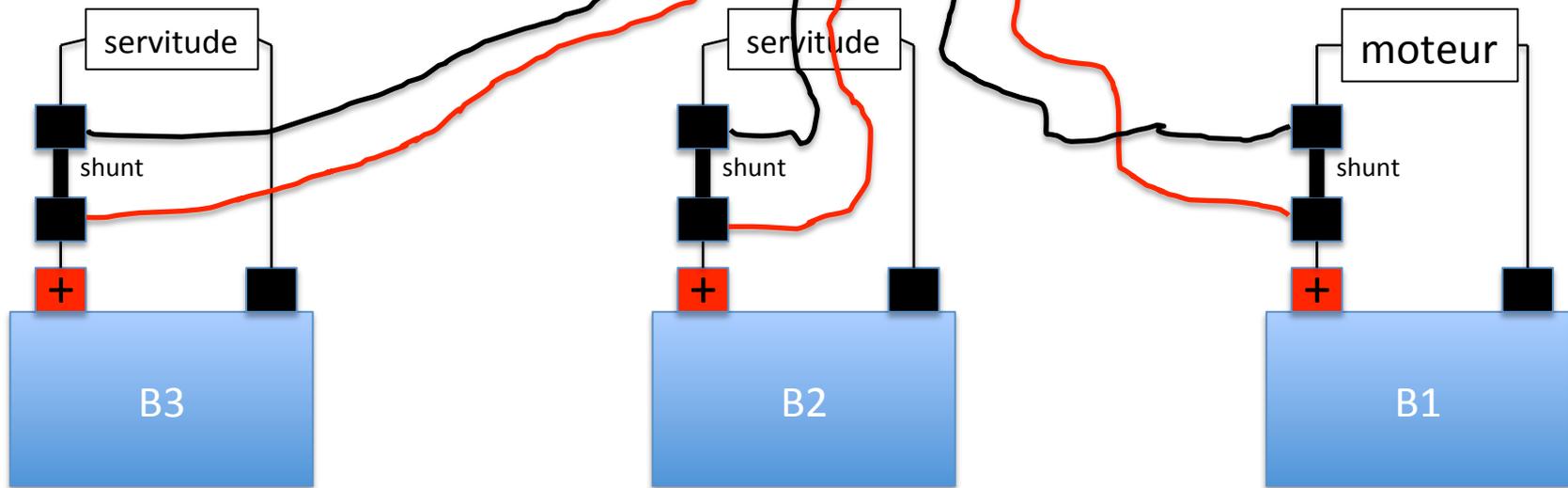
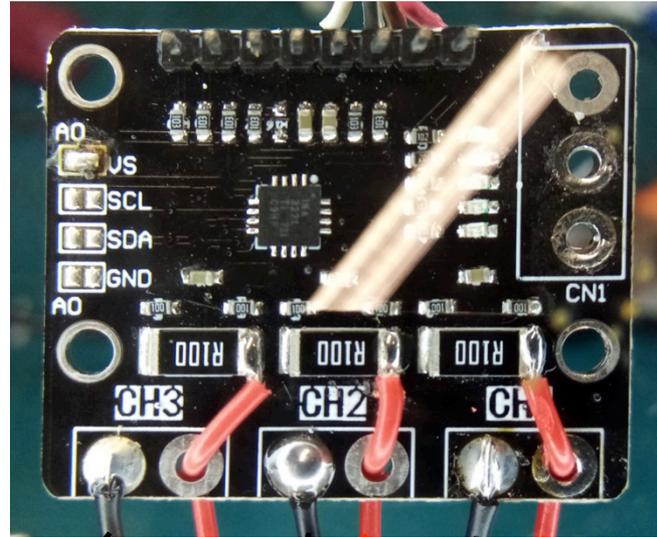
SDA SCL GND +3,3V

Vers carte WEMOS D1

Réalisation : CJMCU 3221 (base INA3221)

Branchement du module (vers batteries)

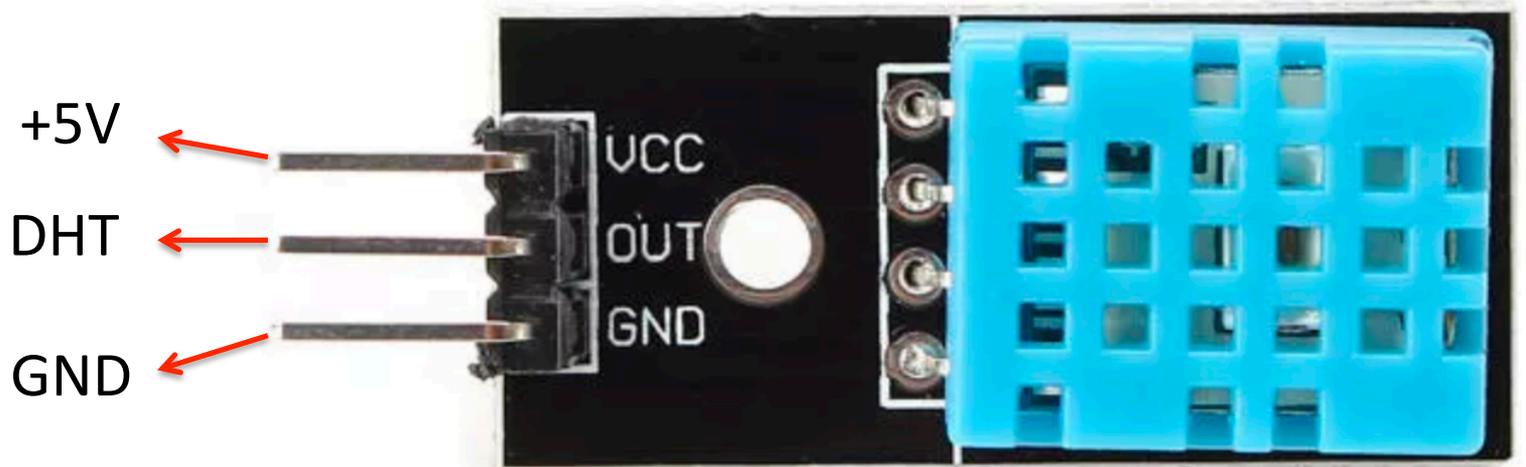
Section des fils sans grosse importance (pas de courant ou très peu)



Réalisation : capteur T° et RH

Branchements

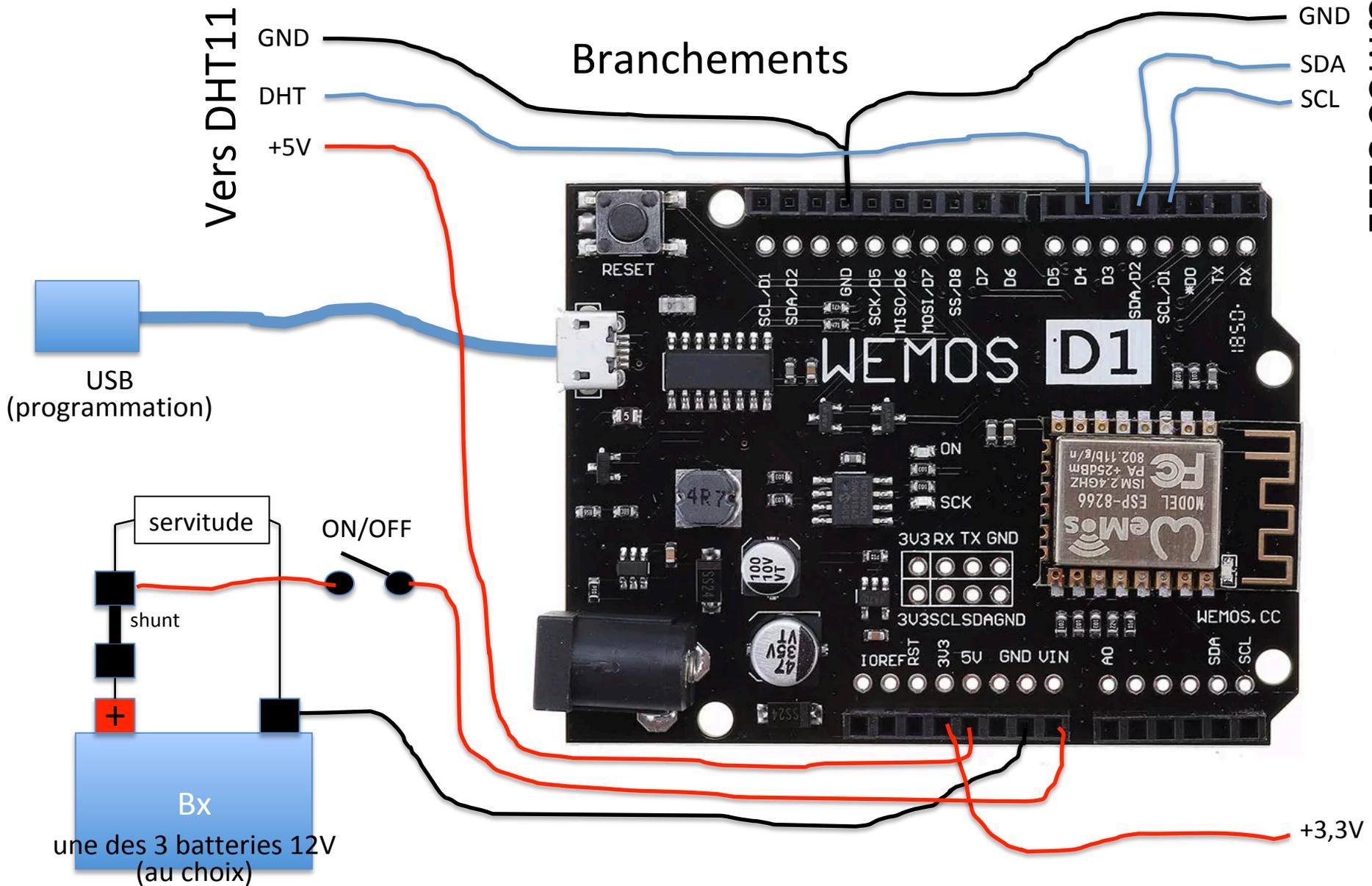
Vers carte WEMOS D1



Réalisation : carte WEMOS D1

option : alimentation sur une des 3 batteries à surveiller (moins sûr)

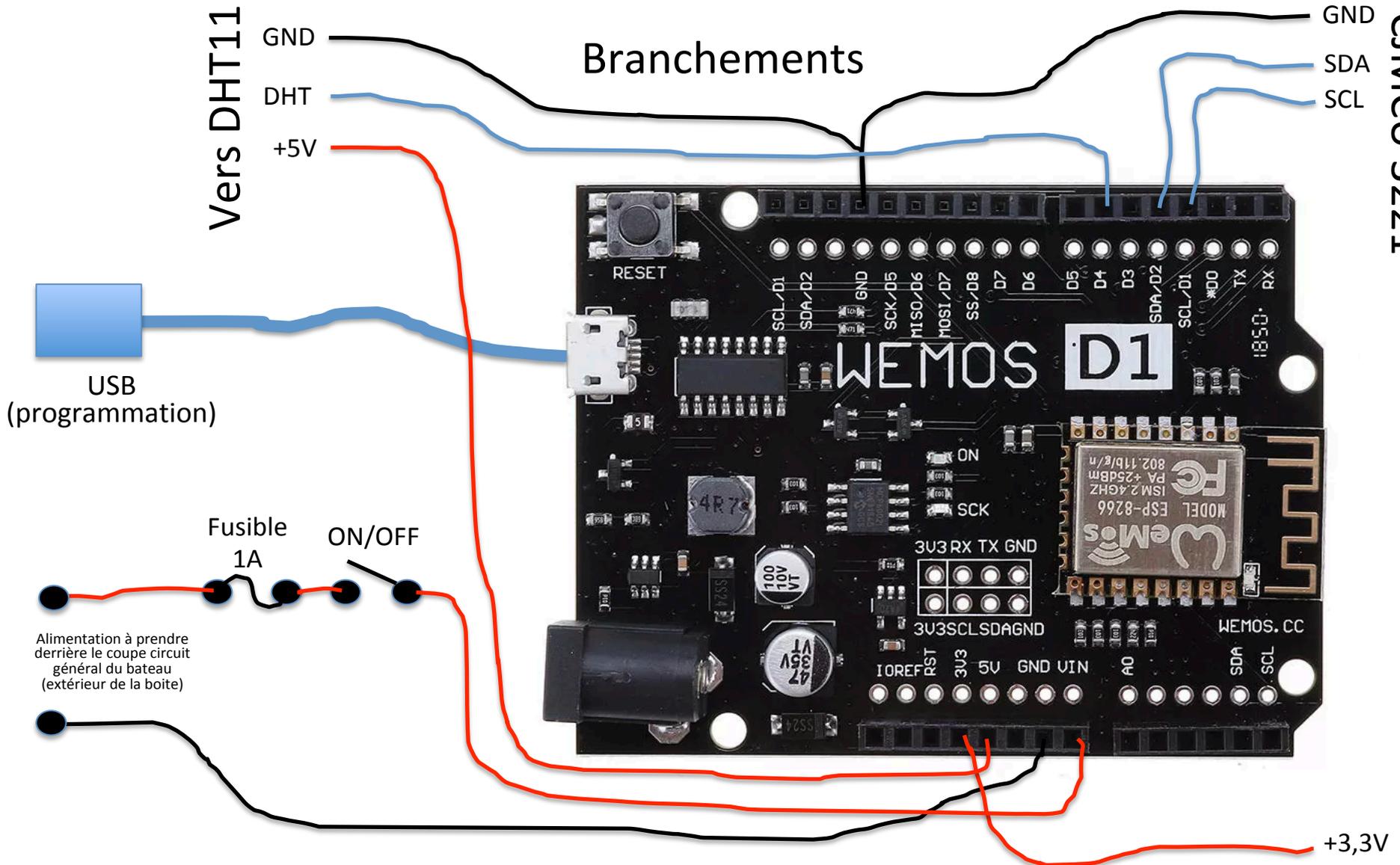
Vers CJMCU 3221



Réalisation : carte WEMOS D1

option : alimentation extérieure sécurisée

Vers CIMCU 3221



Réalisation : boitier

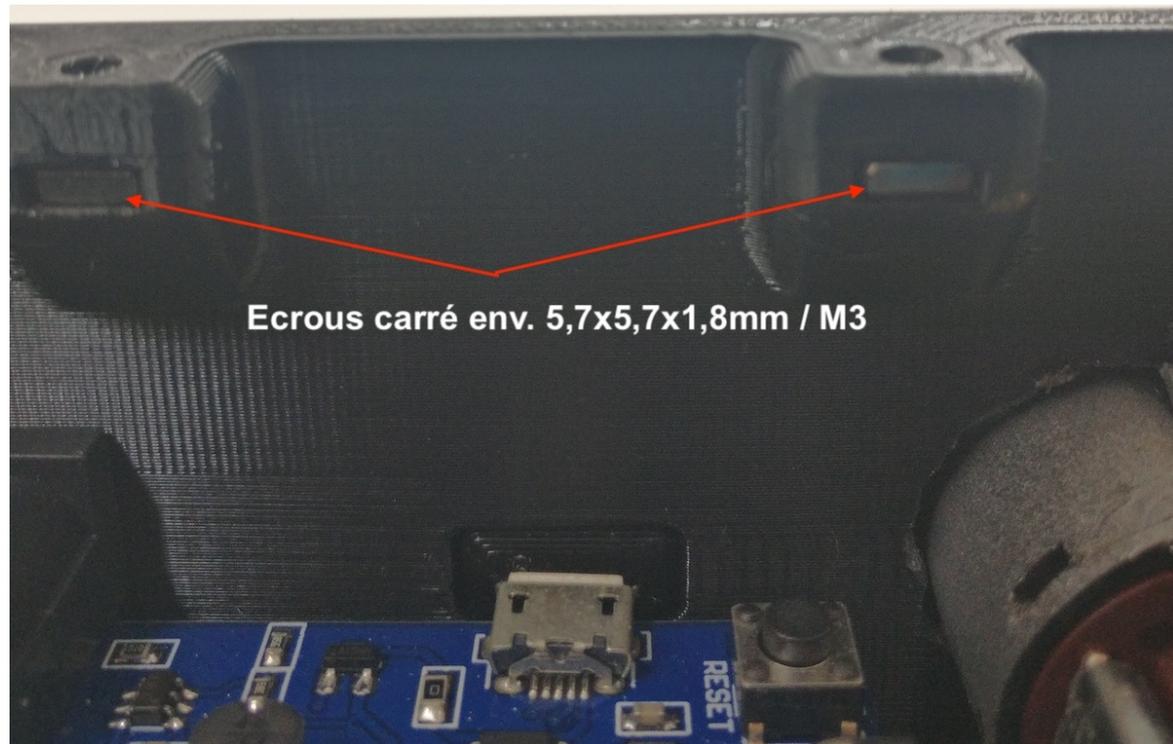
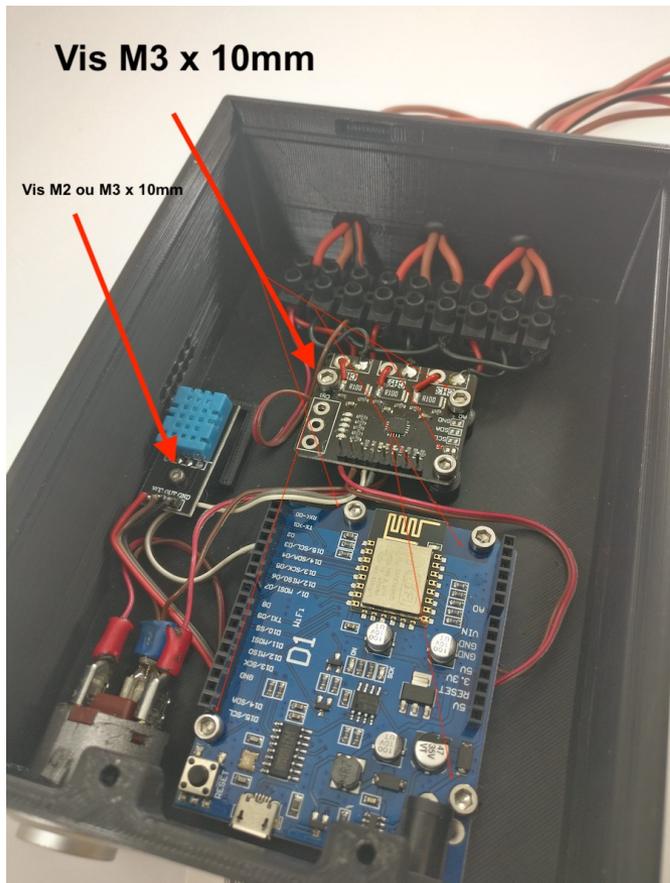
Le boitier a été modélisé sur FUSION 360. Il est ensuite fabriqué par impression 3D en PETG. Il y a 3 éléments (3 fichiers) à imprimer. La conception a été optimisée pour une impression sans support.



Réalisation : montage des éléments

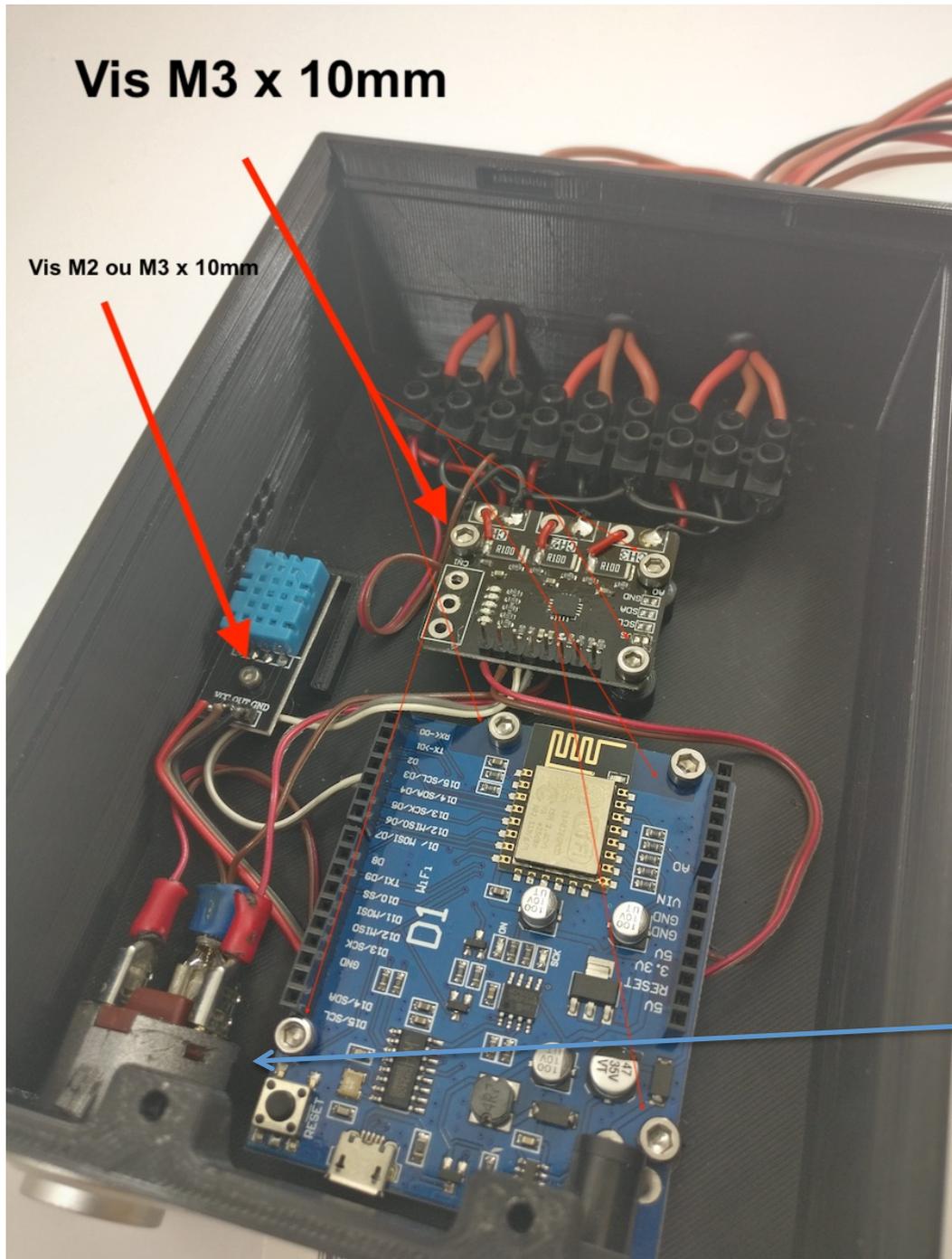
La visserie a été achetée sur <https://www.visseriefixations.fr>.

Toutefois, vous pourriez assembler les éléments électroniques avec un pistolet à colle chaude.



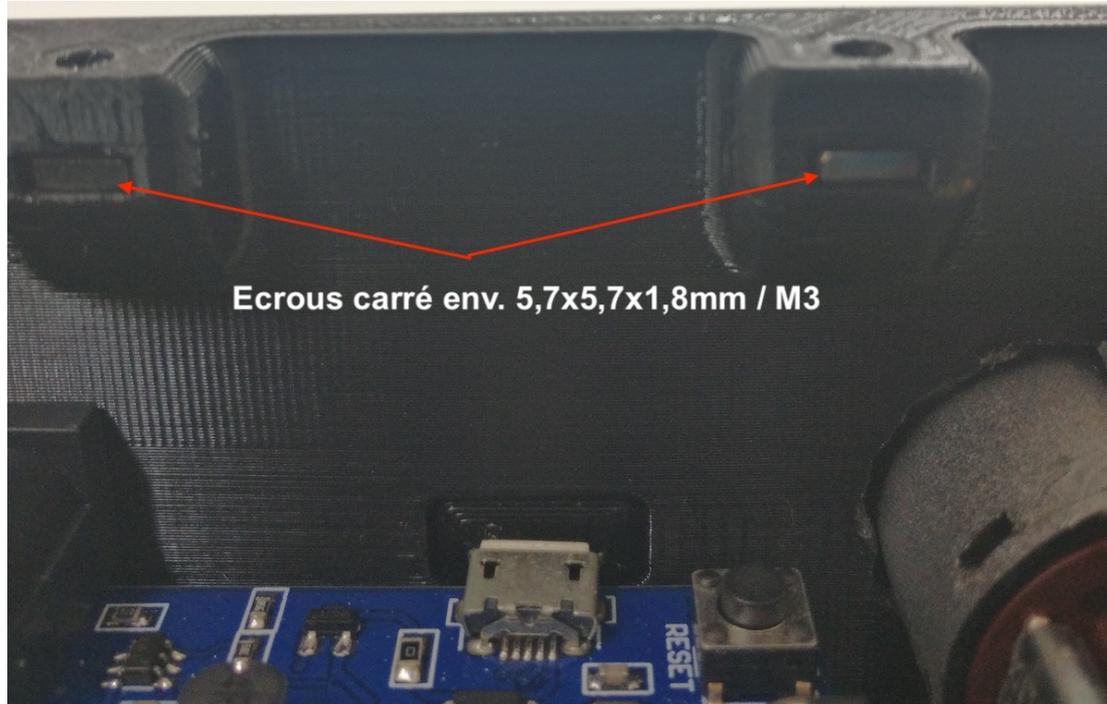
Vis M3 x 10mm

Vis M2 ou M3 x 10mm



Montage du boîtier avec des vis M3 10mm. Selon le capteur de température que vous trouverez, il est possible que ce soit du M2.

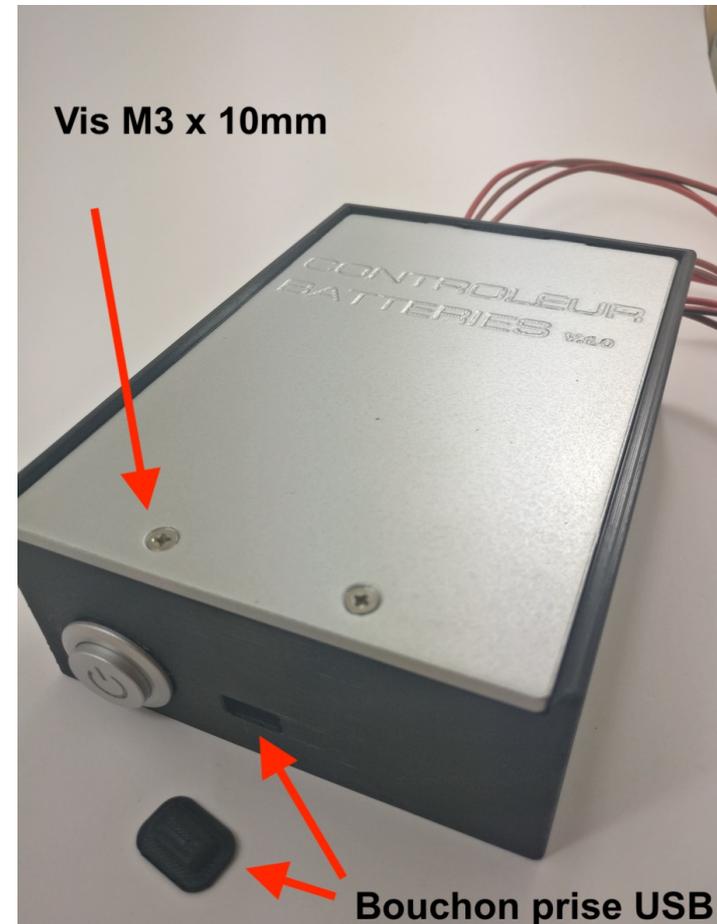
Le perçage pour l'interrupteur est de diamètre 6,80mm sur le fichier fourni. Ici, il a été agrandi après impression pour un inter beaucoup plus gros.



Ecrous carré env. 5,7x5,7x1,8mm / M3

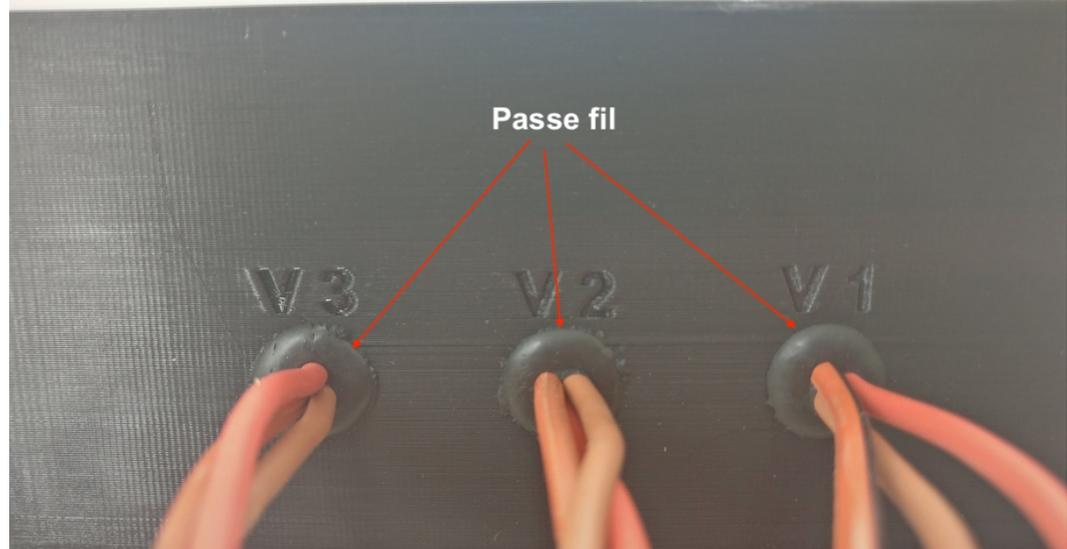
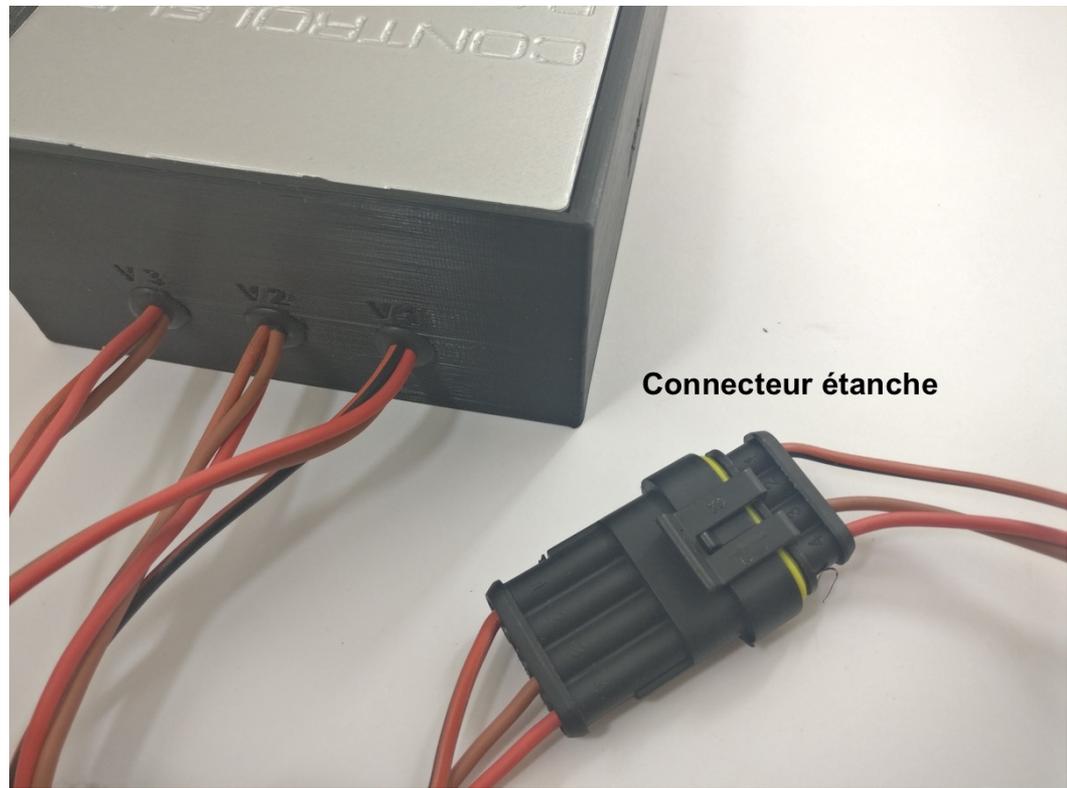
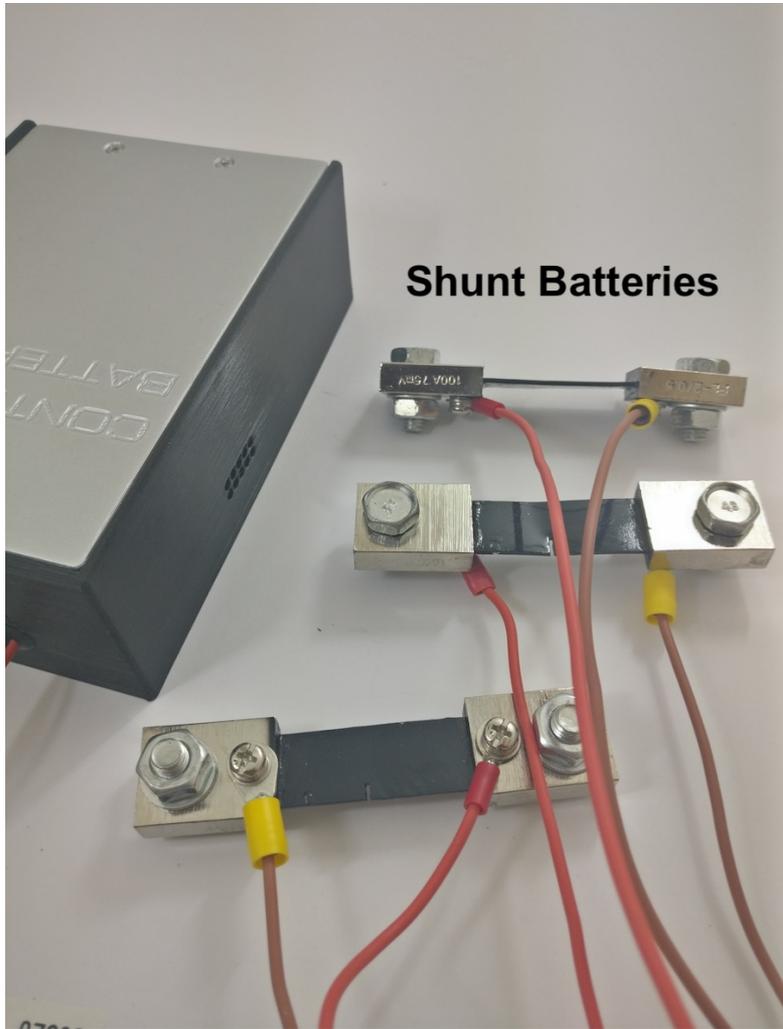
Fermeture de la porte par 2 x M3 10mm (tête fraisée) et 2 écrous insérés après impression du boîtier.

Un bouchon est également imprimé afin de fermer l'accès à la prise mini usb.



Vis M3 x 10mm

Bouchon prise USB



Réalisation :

Environnement de programmation

Le langage utilisé pour ce système est celui utilisé pour la programmation de la suite arduino (arduino IDE) avec comme cible le fameux micro-contrôleur wifi ESP8266.

Le programme est téléchargeable gratuitement sur ce site : <https://www.arduino.cc/>. On y trouve aussi les modes d'emploi, des tutoriels et un forum de discussion.

Un espace est réservé dans la mémoire de l'ESP8266, formaté comme un disque (format SPIFFS), pour y mettre les fichiers en html et javascript du site web que fournit ce système de monitoring.

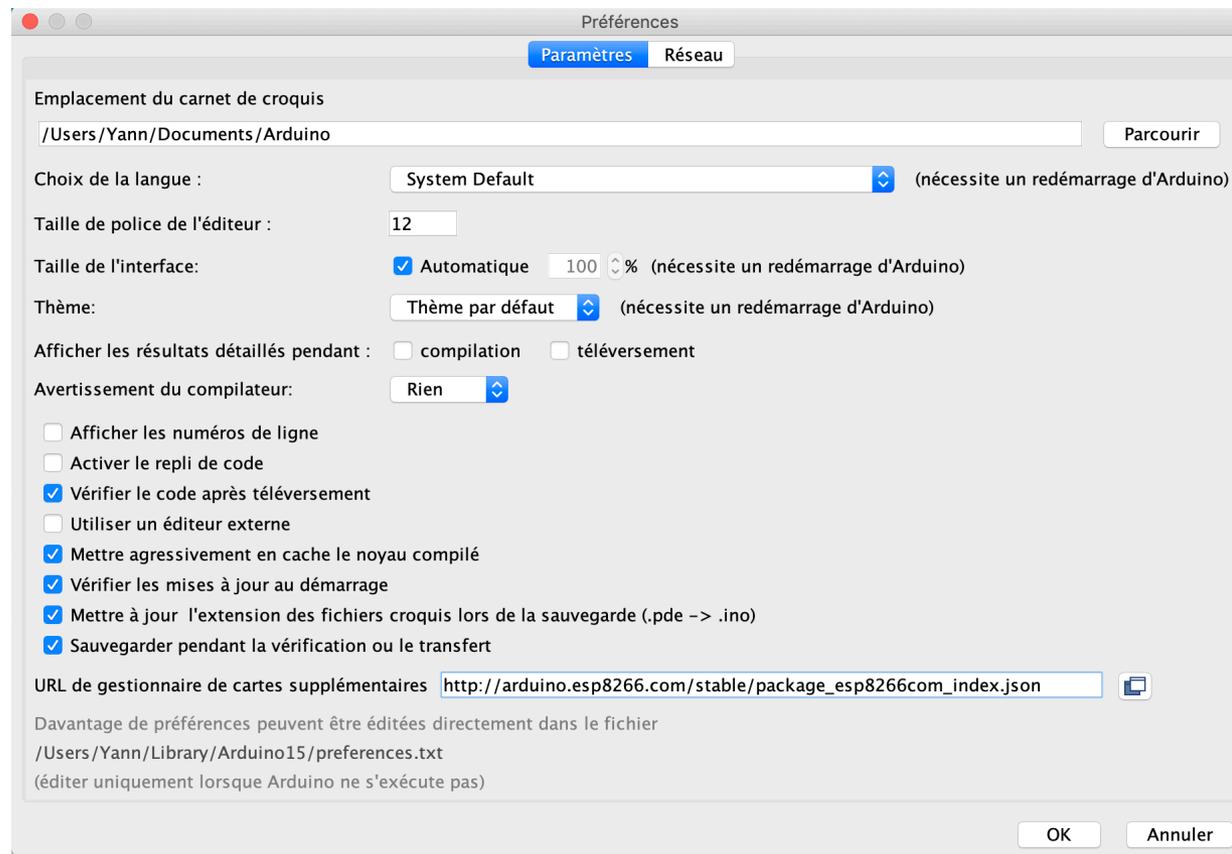
Une fois téléchargé et installé le programme arduino IDE, il faut le configurer afin qu'il puisse cibler l'ESP8266 et formater une partie de sa mémoire en SPIFFS (pour ceux que ça intéresse, nous avons mis un document d'explication de ce format dans le paquet de fichiers : Arduino_filesystem.pdf).

Réalisation :

Configuration de l'IDE arduino (1)

Une fois l'IDE arduino installé, il va falloir configurer l'IDE pour pouvoir cibler l'ESP8266 et y gérer son espace disque virtuel en mémoire flash.

Ouvrez l'IDE et rendez-vous dans les préférences dans le menu « arduino ». Dans la case «URL de gestionnaire de cartes supplémentaires», entrez l'adresse suivante : http://arduino.esp8266.com/stable/package_esp8266com_index.json

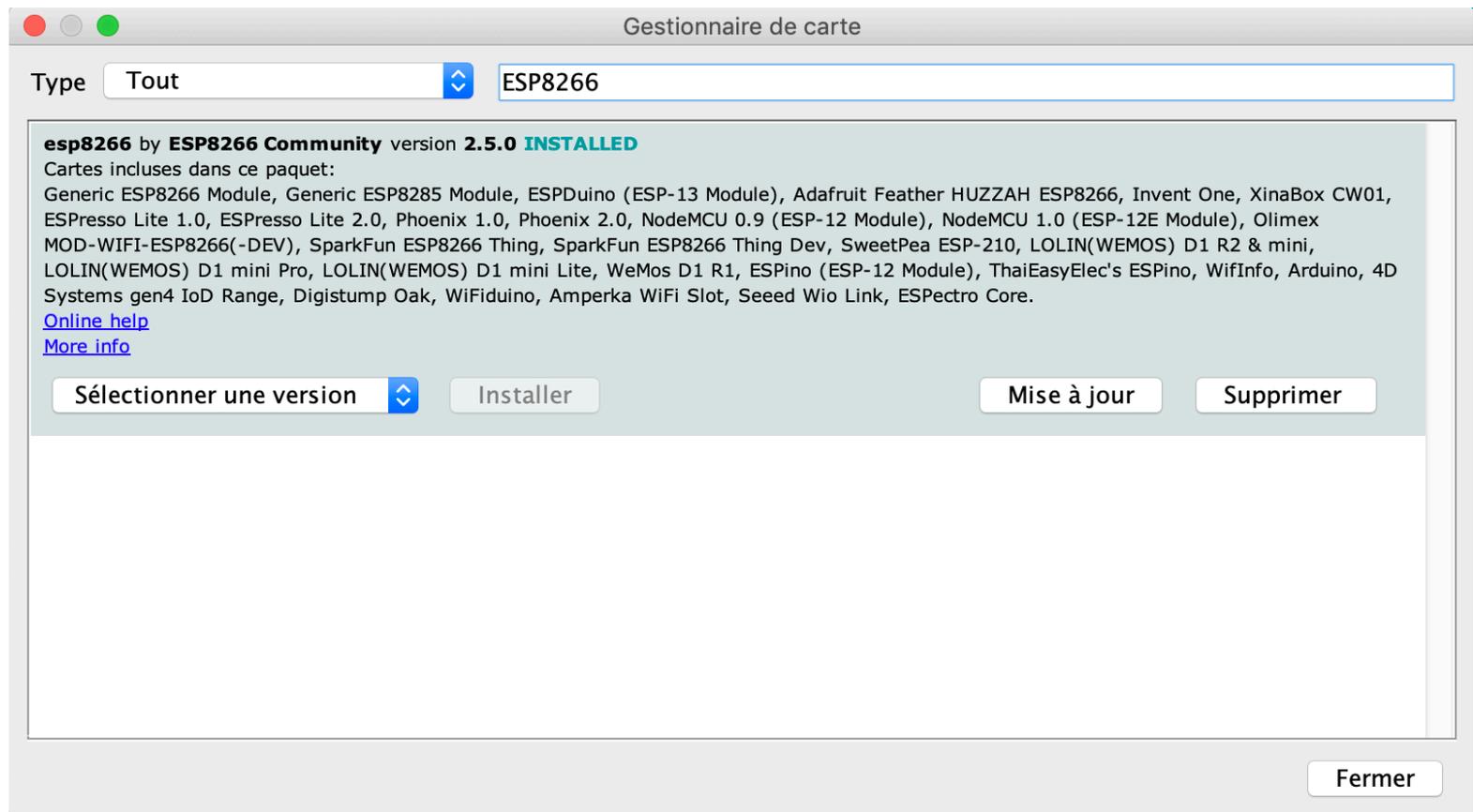


Réalisation :

Configuration de l'IDE arduino (2)

Validez, puis allez dans "Outils", "Type de carte", "Gestionnaire de carte". Dans le champ de recherche, entrez "esp8266".

Vous devriez voir apparaître une bibliothèque ESP8266 :



Réalisation :

Configuration de l'IDE arduino (3)

Cliquez sur la partie grisée puis cliquez sur "Install". Patientez quelques instants et une fois l'installation terminée, si vous vous rendez à nouveau dans "Outils", "Type de carte", vous devriez constater l'apparition d'une nouvelle option : "LOLIN (WEMOS) D1 R2 & mini". Sélectionnez cette dernière.

Dans le menu « outils », vous allez pouvoir configurer la mémoire flash (Flash Size) de la façon suivante : 4M (1M SPIFFS) et le « upload speed » à 115200 (le mode 921600 peut marcher mais pas tout le temps).

Ensuite, il faut pouvoir charger les fichiers html et javascript dans l'espace disque virtuel de la mémoire flash du WEMOS D1 à l'aide du protocole SPIFFS. Pour cela, il faut ajouter une ligne au menu "outils" de l'IDE Arduino de la façon suivante (on verra plus tard comment l'utiliser) :

- Si ce n'est pas déjà fait, créer une arborescence de dossiers « tools » puis « ESP8266FS » puis « tool » dans le répertoire « arduino » qui a été ajouté à l'installation de l'IDE et y mettre le fichier « esp8266fs.jar » fourni avec le paquet de fichiers
- Redémarrer l'IDE
- Vous devriez voir apparaître maintenant la nouvelle ligne «ESP8266 Sketch Data Upload» dans le menu «outils» de l'IDE.

Il faut maintenant installer les bibliothèques. Dans le dossier «arduino » puis « libraries » y mettre les dossiers du paquet (incluant tous leurs fichiers) :

- « arduinoWebSockets-master »
- « DHT_sensor_library_for_ESPx »
- « SDL_Arduino_INA3221 »
- « Timer-master »

Voilà, l'IDE arduino est prêt à être utilisé pour programmer notre ESP8266.

Réalisation : programme

Le paquet de fichiers comporte un dossier «MonitoringbatTRH» qu'il faut mettre avec tout ce qu'il contient dans le répertoire «arduino» de votre PC (garder l'arborescence). Ce dossier comporte le programme principal «MonitoringbatTRH.ino», ses deux sous-programmes «FonctionsACQ.h» et «FonctionsWifi.h» et un dossier «data» qui contient tous les fichiers nécessaires au serveur web.

Une facile personnalisation de ce programme est possible par le changement du nom du point d'accès wifi, son mot de passe, l'image de fond du serveur web et l'icône renvoyée sur la page client du site web.

Le nom (SSID) et le mot de passe du point d'accès se trouve dans le programme principal en ouvrant «MonitoringbatTRH.ino» par l'instruction : `WiFi.softAP("SSID", "password")`. Sauvegarder après modification (menu « fichier » puis « enregistrer »).

L'image de fond se trouve dans le dossier «data» : `Imagefond.jpg`. Elle peut être remplacée par n'importe quelle image JPG à condition qu'elle porte le même nom, que l'image fasse 900x426 pixels et que sa taille ne soit pas supérieure à 200Ko.

L'icône pour le navigateur du client s'appelle «apple-touch-icon.png», elle peut être aussi changée par n'importe quelle image au format PNG à condition qu'elle porte le même nom, que l'image fasse 72x72 pixels et que sa taille ne soit pas supérieure à 100Ko.

Réalisation :

Programmation de la carte WEMOS D1

Brancher la carte WEMOS D1 sur le PC via le câble USB.

Normalement, vous devriez voir apparaître votre port dans le menu "outils". Si ce n'est pas le cas il faut installer le bon driver sur le PC en cherchant sur internet la bonne version (là je ne vais pas m'étendre, tout dépend de votre environnement matériel).

Si vous voyez apparaître votre port, cliquez dessus, vous pourrez programmer la WEMOS D1 en deux étapes (attention, il sera peut-être obligatoire de baisser la vitesse de transmission «upload speed» dans le menu «outil» de l'IDE si une erreur survient) :

- Téléversement du programme en appuyant sur la petite flèche en haut à gauche de la fenêtre de l'IDE ouverte sur le programme principal («MonitoringbatTRH.ino»). Cela compilera le programme et l'enverra sur la carte ;
- Chargement des fichiers du serveur web qui sont dans le dossier «data» par l'option «ESP8266 Sketch Data Upload» du menu outils de l'IDE.

Une fois que ces deux opérations sont faites, vous pouvez débrancher le câble USB et utiliser le système alimenté par une des 3 batteries (via l'interrupteur ON/OFF).

Réalisation : tests

Si tout s'est passé correctement, à la mise ON du système, vous devriez voir apparaître un nouveau point d'accès wifi qui a le nom que vous avez rentré (ou «MonitoringBat» par défaut).

Il faut se connecter dessus en y rentrant le mot de passe (par défaut «ConnectBat»).

Ensuite, aller sur un navigateur (Firefox, Chrome, Safari, ...) et taper l'adresse suivante : <http://192.168.4.1>. Le site web se chargera et vous aurez en temps réel (après quelques secondes d'attente le temps que la liaison websocket s'installe) la tension et le courant de vos batteries ainsi que le relevé de température et humidité du capteur.

Réalisation : étalonnages (1)

Une fois que vous avez les valeurs de courants de vos batteries affichées sur votre téléphone ou tablette, il sera peut-être nécessaire d'étalonner votre système de mesure pour améliorer la précision.

En effet, le programme a été conçu pour lire le courant avec des shunts de 0,75 mOhms. Si la résistance du shunt que vous avez mis en place est légèrement différente, vous aurez des valeurs de courant erronées.

Pour étalonner l'appareil, il suffit soit :

- De mesurer précisément la valeur de résistance des shunts (avec un ohmmètre pour très faibles valeurs et changer les valeurs dans le programme :

```
#define lbat1_shunt_value (0.00075) // shunt resistor value of 0.00075 Ohm
#define lbat2_shunt_value (0.00075) // shunt resistor value of 0.00075 Ohm
#define lbat3_shunt_value (0.00075) // shunt resistor value of 0.00075 Ohm
```

- Soit de mettre une pince ampéremétrique précise qui mesure le courant de chaque batterie (I_{mes}) et que vous comparez à la valeur affichée (I_{aff}) sur votre tablette (à faire pour des courants supérieurs à 10A ça sera plus précis). La valeur de la résistance à changer dans le programme :

```
lbat1_shunt_value = 0,00075x(Iaff1/Imes1)
lbat2_shunt_value = 0,00075x(Iaff2/Imes2)
lbat3_shunt_value = 0,00075x(Iaff3/Imes3)
```

Réalisation : étalonnages (2)

Ensuite, il faudra faire en sorte que les batteries ne génèrent aucun courant. La valeur affichée étant un offset de mesure, il faudra l'enlever dans le programme (ici on a trouvé 213,333mA) :

```
Bat1I = lbat.getCurrent_mA(BAT1) - 213.333;
```

```
Bat2I = lbat.getCurrent_mA(BAT2) - 213.333;
```

```
Bat3I = lbat.getCurrent_mA(BAT3) - 213.333;
```

Rechargez le programme dans le WEMOS D1 et normalement votre système sera étalonné