# Projet de coopération au développement
## Collaboration ULB - ISIG

# Guide d'utilisation du dispositif conçu à Kinshasa

CODEPO - Cellule de coopération au développement
de l'École Polytechnique de Bruxelles

**Membres de l'équipe :**
Dietrich Colin
Esnault Teho
Giot Marie
Niedercorn Eliot
Trouillez Benoît

**Superviseurs :**
Bastin Orianne
Osée Michel
Verstraeten Maxime
Delchambre Lionel

Année académique 2021 - 2022

# Table des matières

**1 Introduction** **3**

**2 Objectifs du dispositif** **4**

2.1 Alertes . . . . . 4

2.2 Bilans journaliers . . . . . 4

**3 Différents modes d'utilisation du dispositif** **5**

3.1 Utilisation normale . . . . . 5

3.2 Test HPPC . . . . . 9

3.3 Test de décharge à faible courant . . . . . 11

**4 Câblage** **15**

4.1 Note importante - Alimentation ! . . . . . 16

4.2 Note importante - Composants non utilisés sur le PCB . . . . . 16

**5 Code arduino** **18**

5.0.1 Installation de l'IDE et ouverture des codes . . . . . 18

5.0.2 Installation des librairies . . . . . 19

5.0.3 Connexion de la carte . . . . . 20

5.0.4 Téléversement d'un code sur l'Arduino . . . . . 22

**6 Codes Python** **24**

6.1 Exécutable . . . . . 24

6.2 Code . . . . . 24

6.3 Erreurs courantes . . . . . 24

6.3.1 CSV ouvert . . . . . 25

6.3.2 Pas de réseau . . . . . 25

6.3.3 2 personnes exécutent l'exécutable en même temps . . . . . 25

6.4 Comment lire les CSV ? . . . . . 25

**7 Thingstream** **27**

7.1 Comment se connecter à la plate-forme Thingstream ? . . . . . 27

7.2 Comment définir l'adresse mail à laquelle Thingstream envoie les mails d'alerte ? . . 27

7.3 Facultatif : Envoi de sms . . . . . 31

**A Notes indicatives pour l'interprétation des données de solar_data_total.csv** **35**

# 1 Introduction

Ce document reprend les informations nécessaires à l'utilisation du dispositif de monitoring de batteries conçu à Kinshasa par l'ULB en collaboration avec la CAMESKIN dans le cadre du projet CODEPO 2021-2022.

FIGURE 1 – Dispositif final implémenté à la Cameskin à Kinshasa

Le dispositif a été développé de manière à être relativement simple d'utilisation à l'aide de l'écran LCD et des 4 différents boutons présents juste en dessous.

# 2 Objectifs du dispositif

Les objectifs du dispositif sont de mesurer les SOC (State Of Charge) , tensions, courants, températures des batteries afin de les monitorer ainsi que de mesurer la propreté des panneaux solaires par comparaison de l'ensoleillement avec la puissance fournie. Toutes ces informations de monitoring sont communiquées à l'utilisateur par le biais de 2 types de messages : des alertes et des bilans journaliers.

## 2.1 Alertes

Les alertes ne se déclenchent qu'en cas de dysfonctionnement des batteries ou des panneaux solaires. Lorsque le dispositif détecte le problème, il envoie un message MQTT à la plateforme Thingstream. Cette dernière envoie ensuite un mail ou un SMS aux personnes de référence (voir la section "Thingstream" pour savoir en quoi consiste la plateforme Thingstream et comment définir les personnes de référence). Chaque mail coûte 0,001$, chaque SMS coûte 0,07$ et cela doit être payé via la plateforme Thingstream. La facture finale est arrondi à 1$ si le montant est inférieur à cela.

## 2.2 Bilans journaliers

Les bilans journaliers sont les données récoltées par le dispositif au cours de la journée. Ils sont envoyés chaque jour au Broker Thingstream qui les stocke en attendant que quelqu'un les récupère via l'application (voir la section "Codes Python").

# 3 Différents modes d'utilisation du dispositif

Le dispositif possède 3 modes de fonctionnements différents sélectionnables à l'aide des boutons présents sur l'écran LCD :

- **Utilisation normale** - Le mode principal qui est utilisé de manière continue pour le monitoring des batteries.
- **Test HPPC (Hybrid Pulse Power Characterization)** - Ce mode correspond à la réalisation d'un test qui décharge une batterie isolée afin de déterminer les paramètres de cette dernière. La connaissance de ces paramètres est nécessaire au bon fonctionnement du mode d'utilisation normale.
- **Test de décharge à faible courant** - Ce dernier mode consiste aussi en un test de décharge sur une batterie isolée. Il permet également de déterminer des paramètres nécessaire pour le mode d'utilisation normale.

Les principes et fonctionnements des trois modes sont détaillés dans les sections suivantes.
Notons que les deux derniers modes d'utilisation (sections 3.2 et 3.3) ne doivent être réalisés qu'en cas de changement de système de batteries à monitorer. Si le cas ne se présente pas, uniquement le mode d'utilisation normale (section 3.1) est utilisé.

## 3.1 Utilisation normale

Le mode principal est le mode d'utilisation normale, celui qui a lieu de manière continue pour monitorer le pack de batteries. Il a été conçu pour monitorer un pack de 20 batteries disposée de la manière suivante : mise en parallèle de 5 branches de 4 batteries mises en séries.

Comme expliqué ci-dessus (voir section 2), le fonctionnement est le suivant. Des mesures de tension, courant et température sont prises sur le pack de batteries et sur la sortie des panneaux solaires à des noeuds intéressants. Des rapports journaliers sont fournis concernant la puissance des panneaux solaires et des batteries ainsi que l'état de charge des batteries. Des alertes sont également communiquées par le dispositif en cas de sous-tension, courant trop élevé, etc.

Pour réaliser ce mode de fonctionnement il faut :
- Le pack de batteries à monitorer (5x4 batteries de 12V)
- Être à proximité de la sortie des panneaux solaires (à l'entrée du charge controller)
- Le dispositif de mesure avec le mode 0.

Le capteur de courant 20A mesure le courant sortant de la batterie tandis que celui de 100A mesure le courant à l'entrée du charge controller (MPPT). Les prises de masse et d'alimentation sont respectivement connectées aux bornes négative et positive du pack entier ($\leq 50V$). Les mesures de tension *BAT1/BAT2/BAT3/BAT4/BAT5/BAT6* sont connectées au sein du pack à des noeuds importants. Les thermistances sont également placées à des endroits déterminés par l'utilisateur.

Le câblage pour le fonctionnement normal est représenté sur la figue suivante.

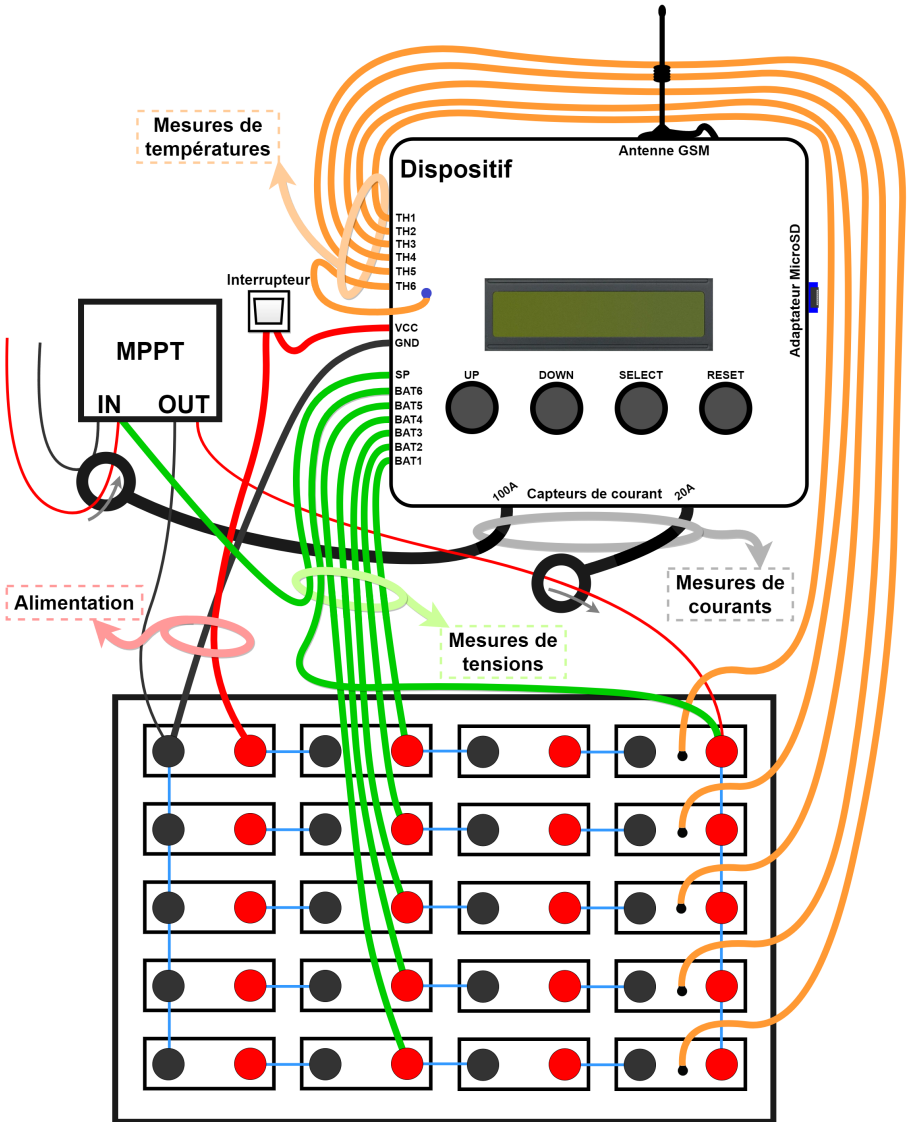



FIGURE 2 – Câblage du dispositif avec le pack de batteries et le MPPT - Utilisation normale (Les couleurs réelles des câbles sont visibles sur la figure 15)

Les valeurs des potentiomètres doivent respectivement être environ égales à $R = \frac{82.5}{V-2.5} 10^3 \Omega$ selon la valeur de la tension à son entrée correspondante [1]. Comme mentionné plus tôt, ces valeurs doivent être mises à jour dans le code du microcontrôleur qui doit ensuite être téléversé.

1. *RxDP* correspond à l'entrée *BATx*.

```
double RV = 32300;
double pot[nbBatteries] = {20400, 20400, 20400, 10000, 10000, 10000};
```

FIGURE 3 – Ligne de code à modifier dans *main.ino* (Valeurs des potentiomètres à modifier entourées en rouge et à entrer en Ω)

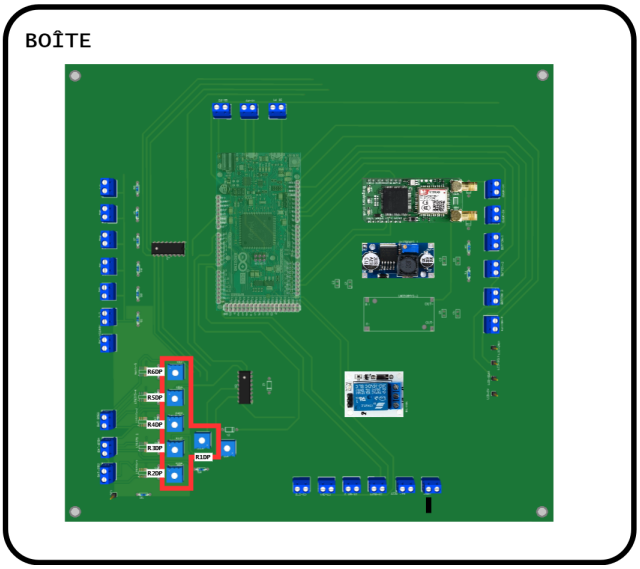



FIGURE 4 – Position des potentiomètres à calibrer sur le PCB

Une fois les calibrations et le câblage réalisés (sauf l'alimentation !), le dispositif peut être alimenté par le pack. Des explications sur l'utilisation du dispositif sont détaillées dans les paragraphes ci-dessous.

Lorsque le dispositif est alimenté, l'écran affiche "*Activate click ?*" suivi de "*yes*" ou "*no*". L'utilisateur peut alors choisir s'il veut activer la communication à l'aide du click ou non ("*yes*" avec le bouton "*UP*", "*no*" avec le bouton "*DOWN*"). Si l'option n'est pas activée, aucune communication n'aura lieu avec le serveur et le dispositif n'est utilisé que de manière locale, à l'aide de l'écran. Une fois que le choix est fait, il suffit d'appuyer sur le bouton "*SELECT*".
Si le click est activé, une brève opération de connexion est réalisée par le prototype et peut prendre jusqu'à 30 secondes.

Après cela, le dispositif affiche sur l'écran les valeurs liées au monitoring ainsi que différentes options tels que le mode à utiliser, la date, l'heure, etc.

Le fonctionnement des boutons et de ce qui est affiché sur l'écran est illustré par le schéma suivant :

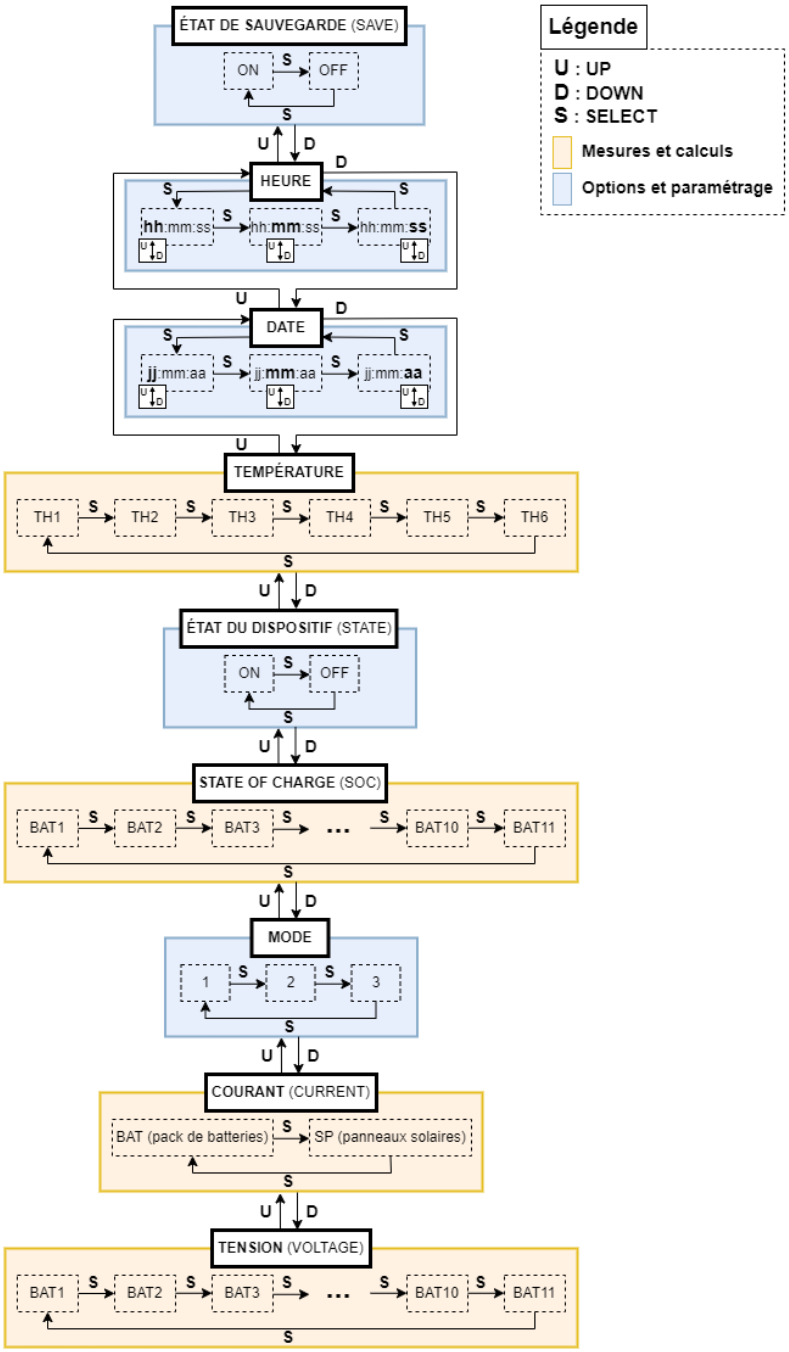



FIGURE 5 – Schéma explicatif de l'utilisation du dispositif par les boutons et écran - Utilisation normale

En plus de l'affichage des mesures et des paramètres, il est possible de réinitialiser le dispositif à l'aide du bouton "*RESET*".

### 3.2 Test HPPC

Si le dispositif doit monitorer un nouveau type de batterie, il est nécessaire de déterminer de nouveaux paramètres pour cette dernière afin d'assurer un bon fonctionnement de l'algorithme implémenté pour l'utilisation normale. Les batteries peuvent être caractérisée à l'aide de 3 paramètres déterminés par un test HPPC [2].

Le fonctionnement du test HPPC est le suivant : de brefs appels de courant élevés sont réalisés sur la batterie à intervalle régulier. Le reste du temps, aucun courant n'est débité afin de laisser la batterie atteindre un équilibre en tension. Dans notre cas, les appels de courant ont une durée de 20 secondes et le repos entre chacun de ces appels est de 5 minutes. En effectuant 5 répétitions du cycle mentionné plus tôt [3], le test est considéré comme suffisamment long pour assurer une bonne réalisation. Les mesures de la tension enregistrée durant l'entièreté du test sont analysés à posteriori afin de déterminer les paramètres de la batterie.

Pour réaliser ce test, il faut :

- Une seule batterie détachée du pack à monitorer (tension entre 5 et 50V).
- Une charge permettant de réaliser d'important appels de courants,de l'ordre de plusieurs dizaines d'ampères si possible, cela dépend de la capacité de la batterie (ex : un *Battery Tester* ).
- Le dispositif de mesure avec le mode 1.

Le capteur de courant 100A (câble noir du bas de la figure 6) mesure le courant sortant de la batterie tandis que la prise de masse et d'alimentation (câble rouge et câble noir allant à la batterie) sont respectivement connectées aux bornes négatives et positives de la batterie. La mesure de tension *BAT1* est également connectée à la borne positive de la batterie. On vient par la suite connecter a charge en parallèle sur la batterie. On retrouve sur la figure (6), le câblage du test HPPC.

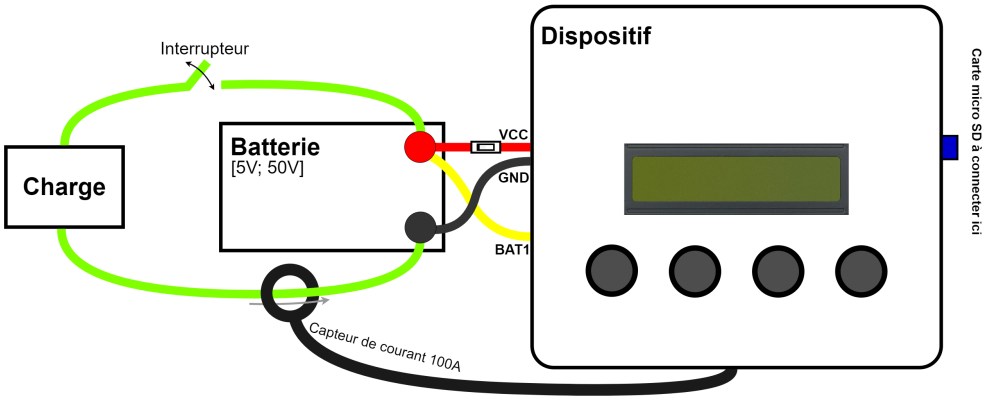



FIGURE 6 – Schéma de cablâge pour le test HPPC

2. HPPC : *Hybrid Pulse Power Characterization* (voir référence [1])
3. 1 répétition = 20 secondes d'appel de courant suivi de 5 minutes de repos.

Notez qu'il est important de vérifier que le potentiomètre *R1DP* soit calibré à $R \approx \frac{82.5}{V-2.5} 10^3 \Omega$ où $V$ est la tension de la batterie en volts [V] ; il faut dès lors changer la valeur du potentiomètre dans le code (figure 7) et téléverser sur le microcontrôleur (voir section 5.0.4).

```
double RV = 32300;
double pot[nbBatteries] = {20400, 20400, 20400, 10000, 10000, 10000};
double RVcc1 = 500000;
```

FIGURE 7 – Ligne de code à modifier dans *main.ino* (Valeur de potentiomètre à modifier entourée en rouge et à entrer en Ω)

Le mode 1 permet d'enregistrer les mesures de tension et de courant sur la batterie sur une carte micro SD connectée sur la droite du dispositif.
Une fois que la carte SD est insérée dans le dispositif, celui-ci peut être alimenté et le test peut être réalisé.

Une fois le test terminé les données sont stockés sur un fichier *.txt*. En analysant les données via un langage de programmation spécifique, on peut tirer les graphiques suivants de courant et tensions :

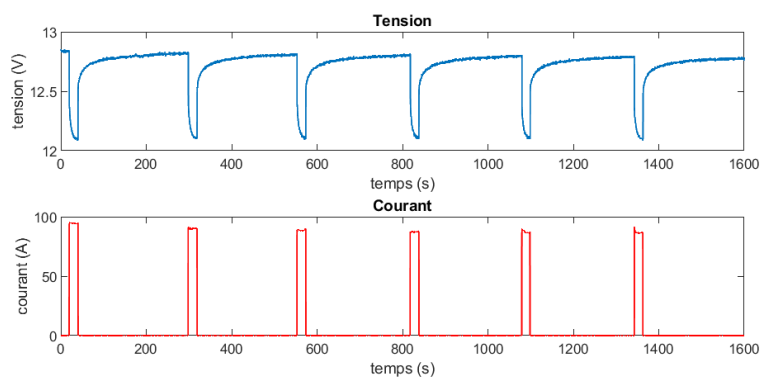



FIGURE 8 – Exemple de tension (bleu) et courant (rouge) sur la batterie durant un test HPPC

Les données peuvent être visualisées et analysées à l'aide du langage de programmation *MATLAB* et du code disponible sur le Drive et le GitHub (), *parameterFitting.m*. Pour cela, le fichier *DATAhppc.txt* stocké sur la carte micro SD doit être placé dans le même dossier que le code MATLAB. L'exécution du code peut être effectuée et les mesures de courant et tension ainsi que les paramètres de la batterie vont être donnés par le programme.

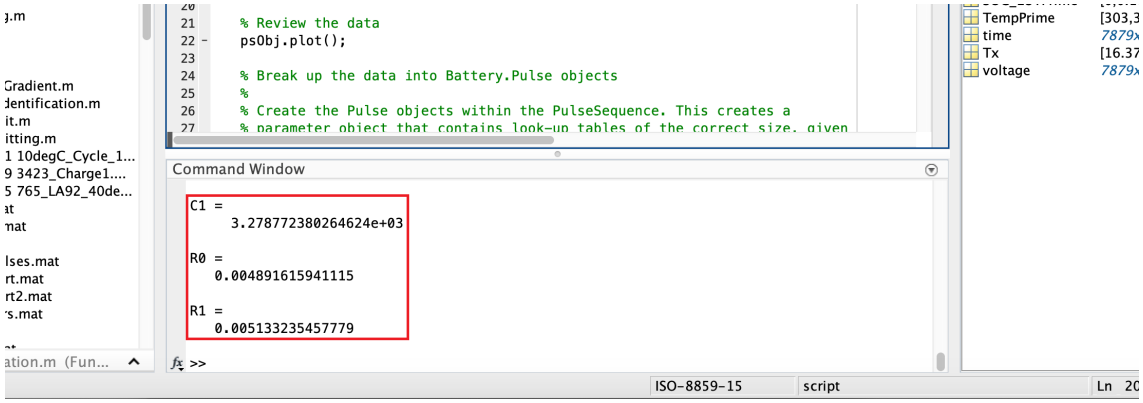



FIGURE 9 – Exemple de résultat du code *parameterFitting.m* (Valeurs des paramètres entourées en rouge)

Maintenant que les paramètres de la batterie sont déterminés, il est nécessaire de les mettre à jour sur le programme implémenté sur le dispositif. Le microcontroleur est un Arduino DUE et est donc implémenté à l'aide du logiciel *Arduino* (voir section 5). Dans le code *main.ino* trouvable sur le GitHub (), il faut modifier les paramètres à la ligne présentée ci-dessous située dans la partie "– *FILTRE KALMANN* –" en début de code.

```
/* -- FILTRE KALMANN-- */
char input[] = "{'C1': 1.39e3, 'R0': 4.2e-3, 'R1': 2.8e-3, 'SOCOCV': [-8.0
```

FIGURE 10 – Ligne de code à modifier dans *main.ino* (Paramètres à modifier entourés en rouge)

Une fois la modification effectuée, le programme peut être téléversé (voir section 5.0.4) sur la carte et la mise à jour des paramètres est réalisée.

### 3.3 Test de décharge à faible courant

Un second test est nécessaire lorsqu'un nouveau type de batterie est à monitorer, il s'agit du test de décharge à faible courant.

Le fonctionnement est le suivant : la batterie doit être déchargée à faible courant sur une longue durée, en partant d'une charge complète à une décharge complète de la batterie. Un "faible courant" ($A$) est défini comme étant inférieur ou égal à un vingtième de la capacité ($Ah$) de la batterie ; par exemple, une batterie $400Ah$ doit débiter un courant inférieur à $\frac{400}{20} = 20A$ pour considérer le courant comme suffisamment faible.

Pour réaliser ce test, il faut :

- Une seule batterie détachée du pack à monitorer (tension entre 5 et 50V).
- Une charge permettant à la batterie de se décharger de manière continue avec un courant égal à un vingtième de la capacité de la batterie.
- Le dispositif de mesure avec le mode 2.

Le capteur de courant 20A mesure le courant sortant de la batterie tandis que la prise de masse et d'alimentation sont respectivement connectées aux bornes négative et positive de la batterie. La mesure de tension *BAT1* est également connectée à la borne positive de la batterie. La charge est connectée en parallèle sur la batterie. Le cablâge pour le test de décharge à faible courant est représenté ci-dessous.

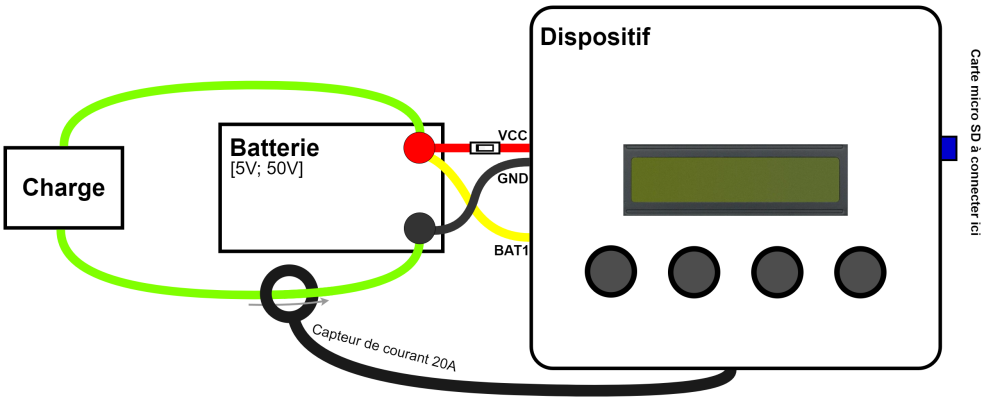



FIGURE 11 – Schéma de cablâge pour le test de décharge à faible courant

La valeur du potentiomètre *R1DP* doit être calibrée à $R \approx \frac{82.5}{V-2.5}10^3\Omega$ sa valeur doit être mise à jour au sein du microcontrôleur (voir figure 7).
Le mode 2 permet d'enregistrer les mesures de tension et de courant sur la batterie sur une carte micro SD conectée sur la droite du dispositif (voir figure 11).
En analysant de nouveau les données de la carte SD avec le logiciel *Mathlab*, on obtient des graphiques ayant une allure semblables à ceux de la figure 12.

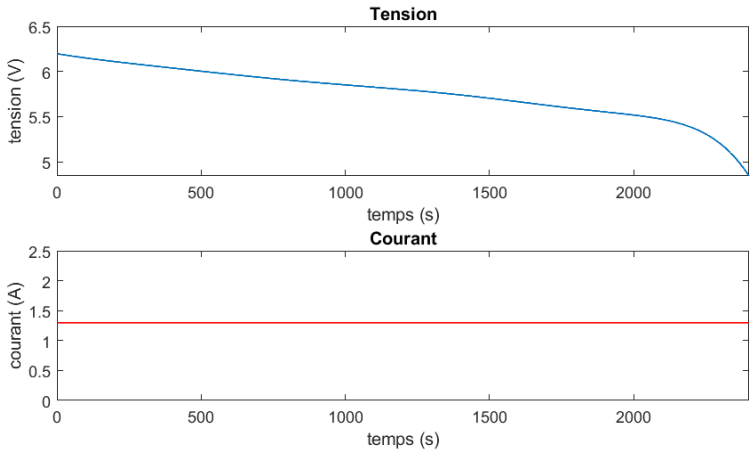



FIGURE 12 – Exemple de tension (bleu) et courant (rouge) sur la batterie durant un test de décharge à faible courant

Tout comme pour le test HPPC, les mesures peuvent être visualisées et analysées via MATLAB à l'aide du code *SOCOCVdetermination.m* disponible sur le Drive et sur le GitHub. Le fichier *DATAsococv.txt* provenant de la carte SD doit être placé dans le même dossier que le code et ce dernier peut être éxecuté. Les graphes de la tension et du courant mesurés apparaissent, ainsi que les paramètres expérimentaux nécessaire à la modélisation de la batterie.

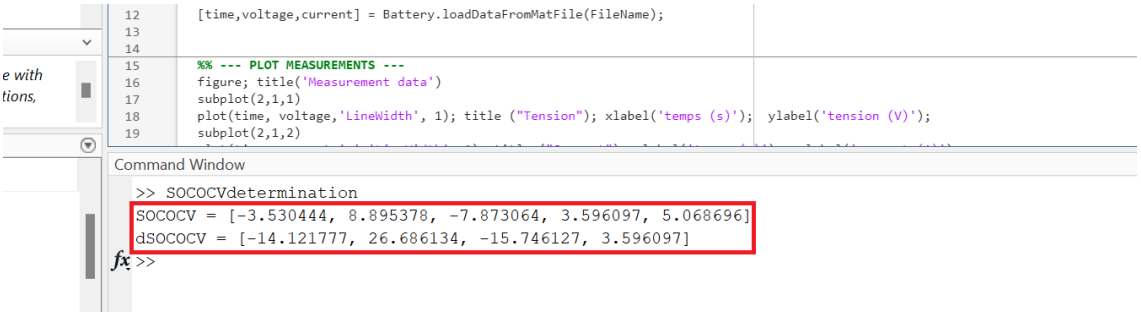



FIGURE 13 – Exemple de résultats du code *SOCOCVdetermination.m* (Valeurs des paramètres entourées en rouge)

De manière semblable au test HPPC, les paramètres doivent être mis à jour sur le microcontrôleur à l'aide d'Arduino dans le code *main.ino* trouvable sur le GitHub. La ligne à modifier est la même que celle présentée pour le test HPPC (voir figure 10). Cependant, les paramètres à modifier sont différents.

```
e-3, 'SOCOCV': [-8.05e-8, 1.64e-5, -1.03e-3, 0.034, 11.49], 'dSOCOCV': [-3.08e-7, 5.11e-5, 0.0023, 0.038]  'SOC_in
```

FIGURE 14 – Ligne de code à modifier dans *main.ino* (Paramètres à modifier entourés en rouge)

# 4 Câblage

Le schéma de câblage des connexions se trouvant à l'intérieur de la boîte est représenté ci-dessous.

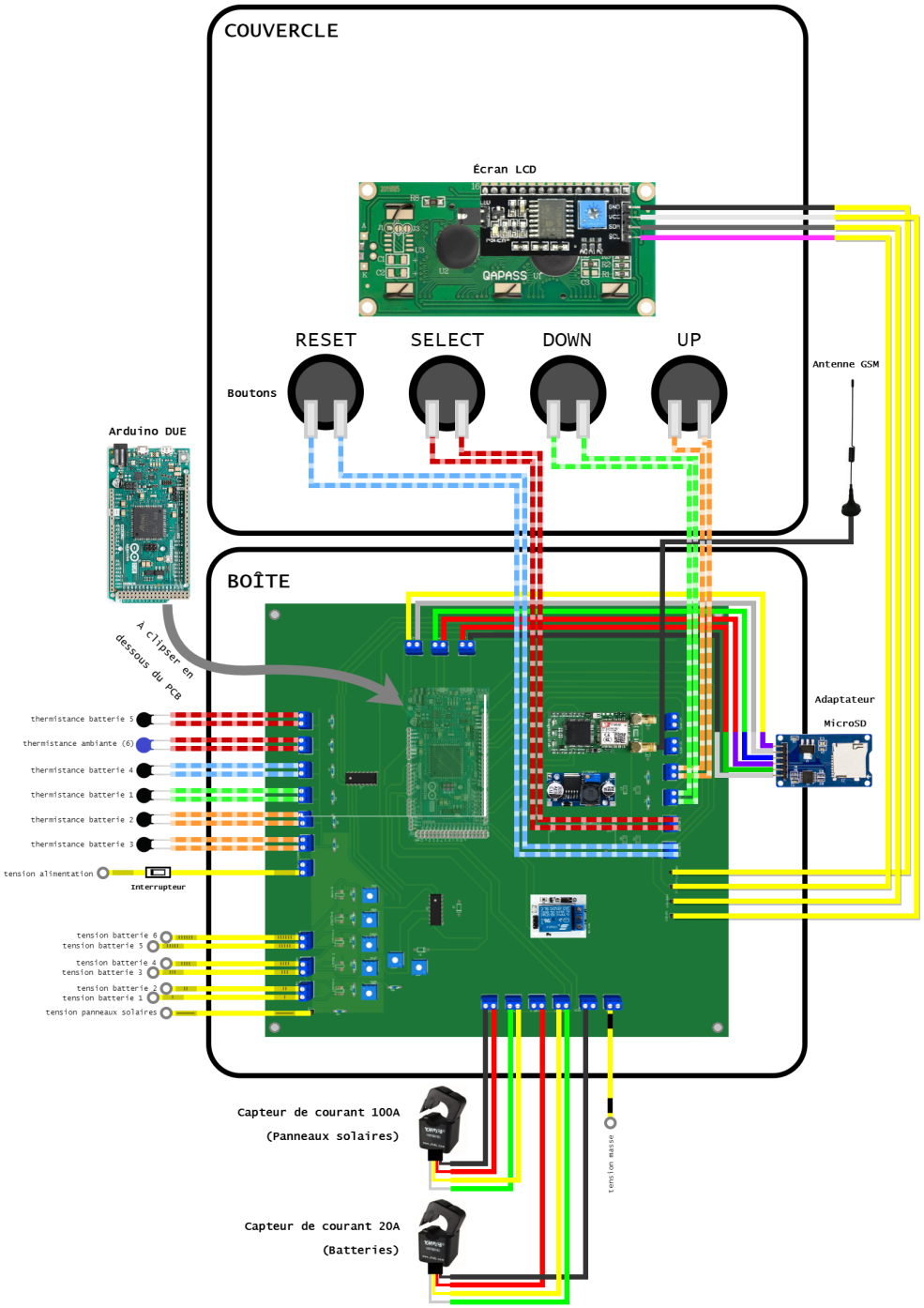



FIGURE 15 – Schéma de cablâge à l'intérieur du dispositif

On y retrouve le PCB (*Printed Circuit Board*), le microcontrôleur Arduino DUE, les prises de mesures de tensions, de courants et de température, l'écran, les boutons, l'adapteur microSD ainsi que l'antenne GSM.
L'ensemble des câbles reliant les différents modules du dispositif sont représentés à l'aide de couleurs qui respectent celles utilisées sur le dispositif initial.
Ce schéma est important car il permet de pouvoir vérifier qu'aucun fil ou composant n'a été déconnecté lorsqu'un dysfonctionnement du prototype apparaît.

Notons que chaque câble est annoté à l'aide d'un stickers afin de faciliter la compréhension du câblage.

## 4.1 Note importante - Alimentation !

Il est important de noter que l'alimentation du dispositif se fait sur une batterie 12V connectée à la masse (comme indiqué sur le schéma précédent). Une tension trop élevée au niveau de l'alimentation peut amener à la destruction du régulateur sur le PCB.

## 4.2 Note importante - Composants non utilisés sur le PCB

Quelques modules ont été ajoutés lors de la conception du PCB afin d'assurer un fonctionnement correct du dispositif. Après tests, quelques-uns se sont avérés facultatifs. On retrouve alors des composants non utilisés sur le PCB.
Il est donc normal de retrouver des pads non utilisés lorsque le PCB est contrôlé, ces derniers sont représentés sur la figure suivante.

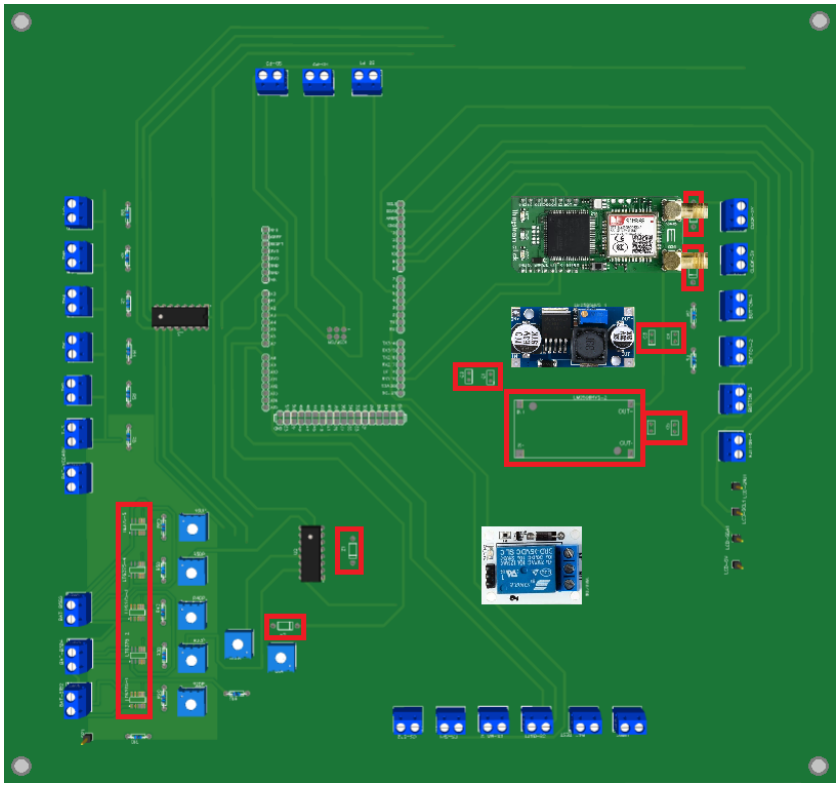


FIGURE 16 – Composants non utilisés sur le PCB (encadrés en rouge)

# 5 Code arduino

Le code arduino est le code implémenté sur la carte arduino présente sur le dispositif. Comme mentionné pour les tests HPPC et de décharge dans la section 3, il faudra modifier ce code lorsque l'utilisateur veut changer le type de batterie que le dispositif monitore. Ici sont donc reprises les différentes étapes nécessaires pour modifier avec succès le code arduino du dispositif avec un ordinateur tournant sur un système Windows.

### 5.0.1 Installation de l'IDE et ouverture des codes

Il faut premièrement installer l'IDE arduino.

Aller sur `https://www.arduino.cc/en/software` pour télécharger le logiciel *Arduino* (on parle ici d'un IDE).

Une fois l'IDE Arduino téléchargé, un dossier "Arduino" devrait apparaître sur l'ordinateur dans le fichier "Documents". C'est là qu'il faut placer le code arduino.

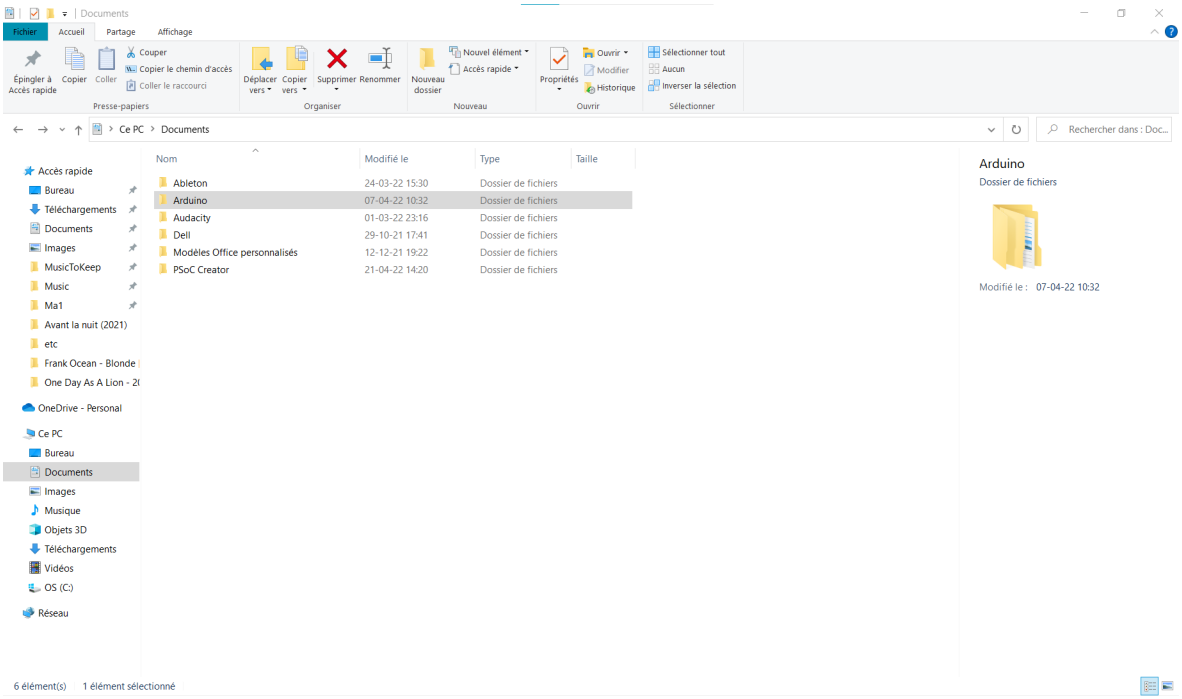



FIGURE 17 – Emplacement du dossier Arduino

Pour télécharger le code arduino, il faut se rendre sur `https://github.com/colinDietrich/Mobateli` pour télécharger : **arduino**. Pour cela, il faut clicker sur le bouton vert "code" et clicker

sur "Download ZIP".

Depuis ce fichier ZIP, il faut extraire le dossier **Mobateli/arduino/main** et le placer dans le dossier "Arduino".

### 5.0.2 Installation des librairies

Il y a au total 5 librairies à installer. La première doit s'installer manuellement. Pour cela, il faut ouvrir l'IDE et aller dans Croquis -> Inclure une bibliothèque -> Gérer les bibliothèques

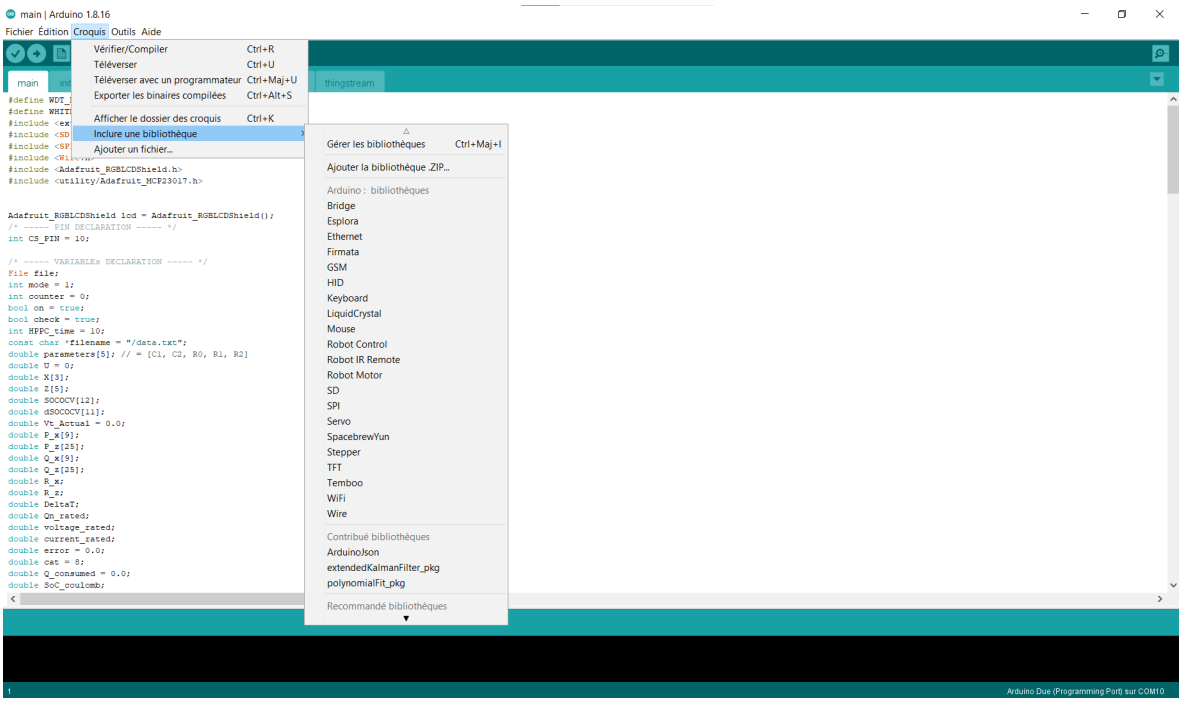



FIGURE 18 – Emplacement d'installation des librairies

La première librairie à installer manuellement est *Arduino_BuiltIn*

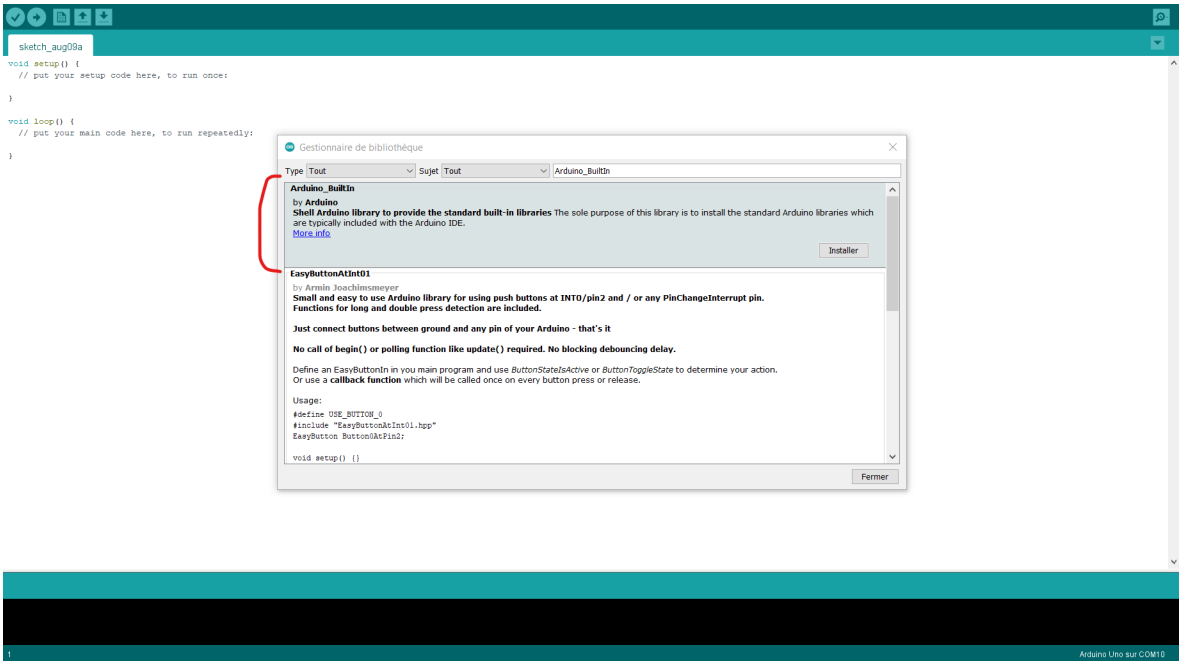



FIGURE 19 – Librairie *Arduino_BuiltIn*

Une fois celle-ci installée, un dossier "libraries" est automatiquement créée dans le dossier Arduino.

Pour installer les autres librairies arduino il suffit alors d'extraire du fichier ZIP et de glisser le dossier **Mobateli/arduino/libraries** dans le dossier "Arduino" et d'accepter la fusion avec le dossier existant.

### 5.0.3 Connexion de la carte

Pour connecter son ordinateur à la carte arduino et donc effectuer des modifications dans le code, il faut d'abord installer les drivers spécifiques à la carte arduino due qui est celle utilisée dans le dispositif.

Pour cela, il faut d'abord se rendre dans le gestionnaire de carte.

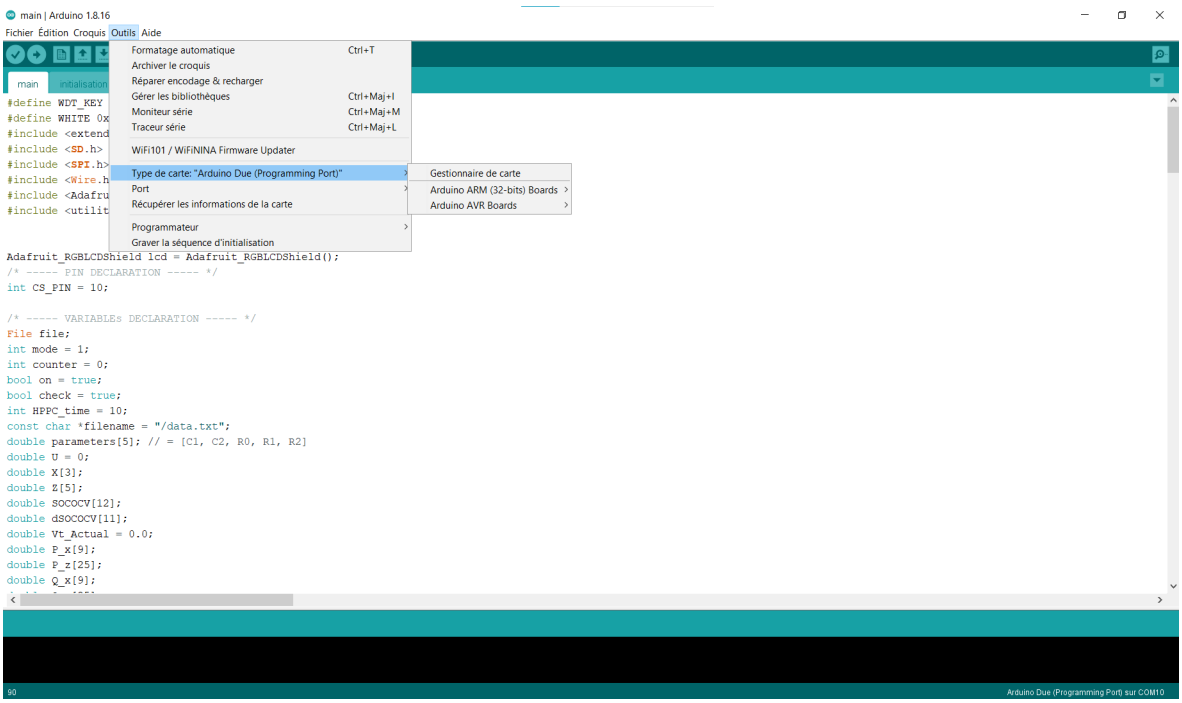



FIGURE 20 – Endroit où aller pour ouvrir le gestionnaire de carte.

Et effectuer la recherche *Arduino SAM Boards*.

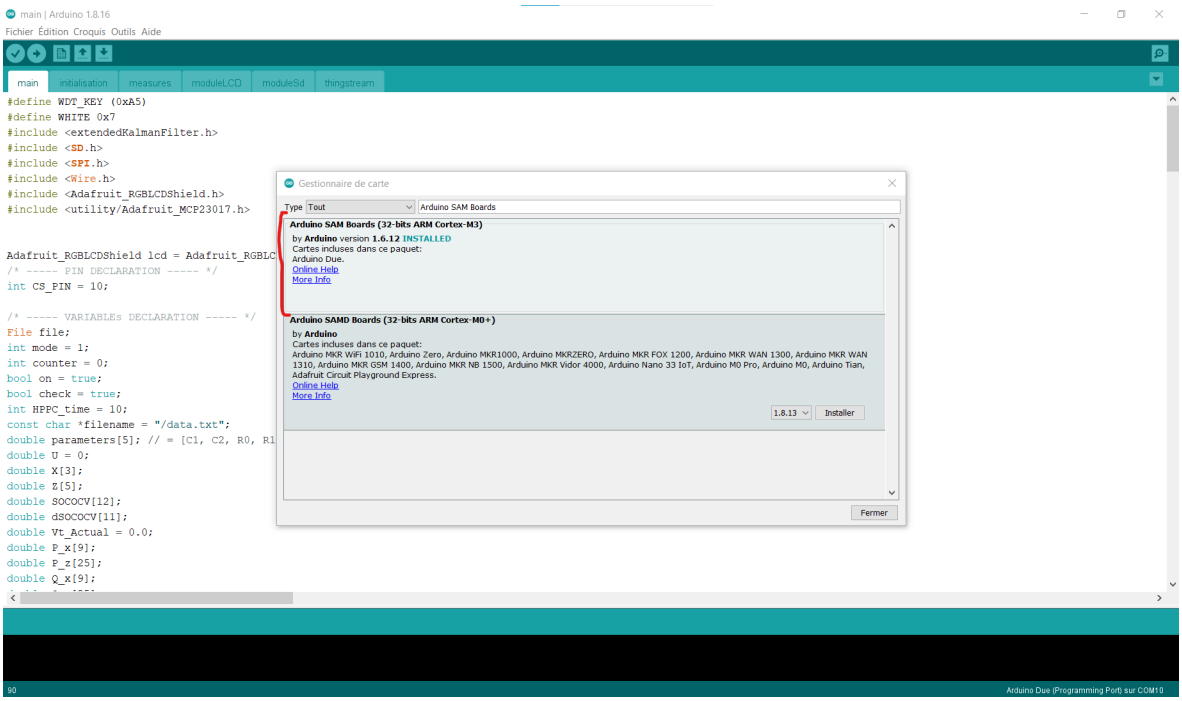



FIGURE 21 – Package à installer pour utiliser la carte arduino due.

Après avoir téléchargé le bon fichier, il faut se connecter par USB à la carte. Attention à bien se connecter au port gauche comme illustré sur la photo suivante.

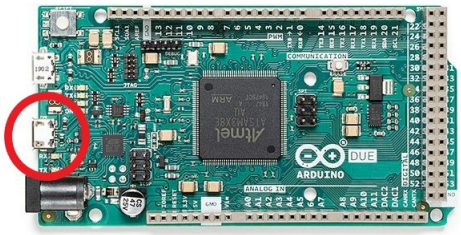



FIGURE 22 – Pour connecter la carte arduino au PC il faut faire attention de la connecter au bon port.

Une fois la carte connectée par USB, il faut clicker sur Outils -> Type de carte -> Arduino ARM (32-bits) Boards -> Arduino Due (Programming Port).

Et la dernière étape consiste à indiquer le port sur lequel la carte est connecté. Pour cela, il faut aller dans Outils -> Ports

### 5.0.4 Téléversement d'un code sur l'Arduino

Une fois la carte Arduino connectée, l'utilisateur peut téléverser un code sur celle-ci. Cette fonctionnalité n'est utilisé que si le code dans la carte doit être mise à jour (par exemple suite à un changement des paramètres à utiliser pour le monitoring, voir sections 3.2 et 3.3).

Pour faire cela, il faut appuyer sur la flèche indiquée sur la figure suivante une fois que le code a été modifié et que la carte Arduino est bien connectée.

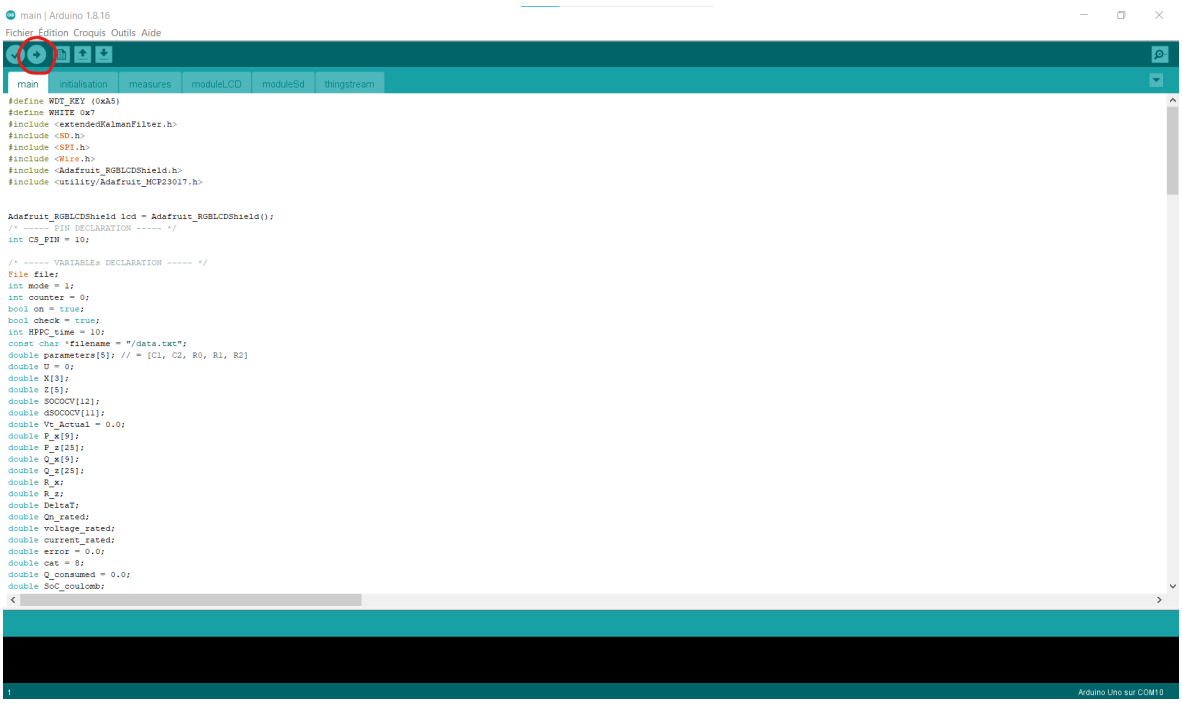



FIGURE 23 – Bouton où clicker pour donner les modifications de code à l'Arduino.

# 6 Codes Python

Afin de récupérer les bilans journaliers envoyés par messages MQTT par le dispositif, il faut être en possession des codes Python disponibles à l'adresse suivante :
https ://github.com/colinDietrich/Mobateli/tree/main/python_application.

## 6.1 Exécutable

Le GitHub contient un exécutable, c'est-à-dire un fichier permettant de faire tourner le code sans devoir télécharger Python ou utiliser un environnement de programmation. Il peut être retrouvé à l'adresse suivante :
https ://drive.google.com/drive/folders/11-IiX5NF0pR_bZY7qZHjgFzzRWEFdXeU ?usp=sharing
Pour récupérer les données des bilans journaliers, il suffit de double-cliquer sur l'exécutable une fois qu'il est téléchargé. Un terminal va s'ouvrir et afficher les messages reçus ainsi que des messages liés au traitement de données. Une interface graphique va également s'ouvrir permettant de visualiser les données du jour sous forme de graphiques et un message d'alerte sur l'encrassement des panneaux va s'afficher si le logiciel à détecter que ceux-ci étaient sales. Une fois que le terminal n'affiche plus de nouvelles lignes pendant 10 secondes d'affilée, cela signifie que le code ne reçoit plus de nouveaux messages et que vous pouvez fermer le terminal en cliquant sur la croix en haut à droite. A présent, deux fichiers CSV se trouvent dans le dossier contenant l'exécutable. L'un reprend les informations des batteries ("batteries_data_total.csv") et l'autre, l'information des panneaux solaires ("solar_data_total.csv"). Lorsque l'utilisateur relancera le code plus tard pour récupérer de nouveaux messages, les nouvelles données seront directement ajoutées aux anciennes dans les même fichiers CSV.

## 6.2 Code

Pour utiliser le code dans un environnement Python, il faut récupérer les fichiers de codes sur le Github et lancer le fichier "Subscribe.py". Le fonctionnement est le même que pour l'exécutable.

## 6.3 Erreurs courantes

Le terminal peut renvoyer des erreurs et se fermer rapidement. Cette section reprend les erreurs courantes que l'utilisateur pourrait faire et donne leur résolution.

### 6.3.1 CSV ouvert

**PermissionError : [Errno 13] Permission denied : 'batteries_data_total.csv'**
Cette erreur signifie que l'utilisateur n'a pas fermé le fichier CSV que le code essaye de modifier. Il doit fermer les fenêtres affichant les CSV "solar_data_total.csv" et "batteries_data_total.csv".

### 6.3.2 Pas de réseau

**socket.gaierror : [Errno 11001] getaddrinfo failed**
Cette erreur signifie qu'il n'y a pas assez de réseau internet disponible pour que le code puisse se connecter au Broker Thingstream pour récupérer des messages. L'utilisateur doit trouver un connexion internet suffisante et relancer ensuite l'exécutable.

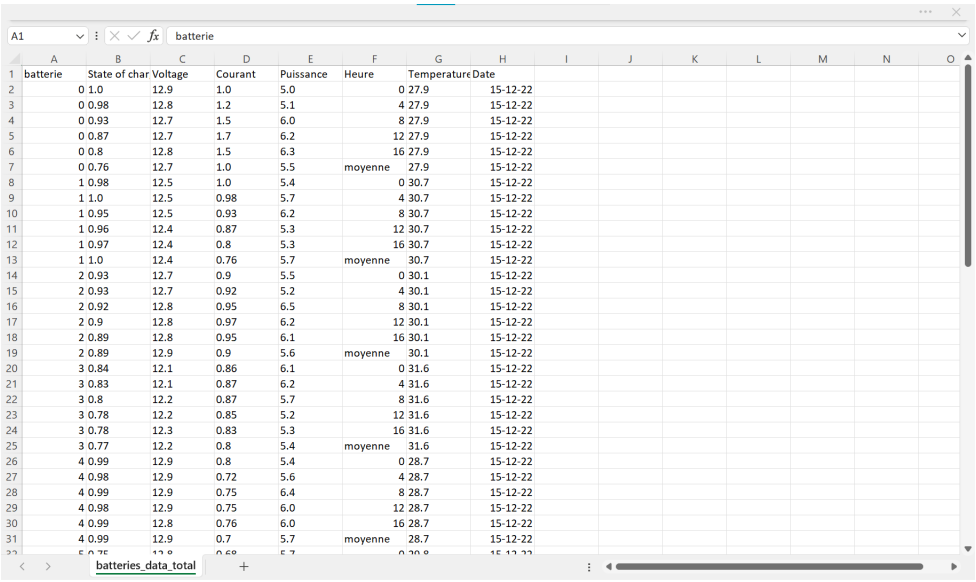
### 6.3.3 2 personnes exécutent l'exécutable en même temps

Il faut faire attention à ce que 2 personnes ne lancent pas l'exécutable en même temps car cela pourrait entraîner des bugs et potentiellement des pertes de données. Il est conseillé par défaut que ce soit toujours le même ordinateur qui lance l'exécutable.

## 6.4 Comment lire les CSV ?

Les fichiers CSV ( Comma-Separeted Values ) sont des fichiers de type tableur dont les valeurs sont séparés par des virgules.
En lançant l'exécutable deux fichiers CSV vont être créé.

— Le fichier "**batteries_data_total.csv**" qui reprend l'ensemble des informations sur les batteries. Comme les mesures de tensions et de courant prises sur 5h différentes de la journée : 4h, 8h, 12h, 16h, 22h. Les dates de mesures sont également indiqués comme pour les panneaux solaires. Une moyenne des mesures de la journée est également donnée.

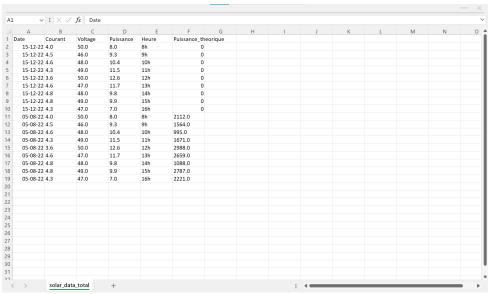


| batterie | State of char | Voltage | Courant | Puissance | Heure | Temperature | Date |
|---|---|---|---|---|---|---|---|
| 0 | 1.0 | 12.9 | 1.0 | 5.0 | 0 | 27.9 | 15-12-22 |
| 0 | 0.98 | 12.8 | 1.2 | 5.1 | 4 | 27.9 | 15-12-22 |
| 0 | 0.93 | 12.7 | 1.5 | 6.0 | 8 | 27.9 | 15-12-22 |
| 0 | 0.87 | 12.7 | 1.7 | 6.2 | 12 | 27.9 | 15-12-22 |
| 0 | 0.8 | 12.8 | 1.5 | 6.3 | 16 | 27.9 | 15-12-22 |
| 0 | 0.76 | 12.7 | 1.0 | 5.5 | moyenne | 27.9 | 15-12-22 |
| 1 | 0.98 | 12.5 | 1.0 | 5.4 | 0 | 30.7 | 15-12-22 |
| 1 | 1.0 | 12.5 | 0.98 | 5.7 | 4 | 30.7 | 15-12-22 |
| 1 | 0.95 | 12.5 | 0.93 | 6.2 | 8 | 30.7 | 15-12-22 |
| 1 | 0.96 | 12.4 | 0.87 | 5.3 | 12 | 30.7 | 15-12-22 |
| 1 | 0.97 | 12.4 | 0.8 | 5.3 | 16 | 30.7 | 15-12-22 |
| 1 | 1.0 | 12.4 | 0.76 | 5.7 | moyenne | 30.7 | 15-12-22 |
| 2 | 0.93 | 12.7 | 0.9 | 5.5 | 0 | 30.1 | 15-12-22 |
| 2 | 0.93 | 12.7 | 0.92 | 5.2 | 4 | 30.1 | 15-12-22 |
| 2 | 0.92 | 12.8 | 0.95 | 6.5 | 8 | 30.1 | 15-12-22 |
| 2 | 0.9 | 12.8 | 0.97 | 6.2 | 12 | 30.1 | 15-12-22 |
| 2 | 0.89 | 12.8 | 0.95 | 6.1 | 16 | 30.1 | 15-12-22 |
| 2 | 0.89 | 12.9 | 0.9 | 5.6 | moyenne | 30.1 | 15-12-22 |
| 3 | 0.84 | 12.1 | 0.86 | 6.1 | 0 | 31.6 | 15-12-22 |
| 3 | 0.83 | 12.1 | 0.87 | 6.2 | 4 | 31.6 | 15-12-22 |
| 3 | 0.8 | 12.2 | 0.87 | 5.7 | 8 | 31.6 | 15-12-22 |
| 3 | 0.78 | 12.2 | 0.85 | 5.2 | 12 | 31.6 | 15-12-22 |
| 3 | 0.78 | 12.3 | 0.83 | 5.3 | 16 | 31.6 | 15-12-22 |
| 3 | 0.77 | 12.2 | 0.8 | 5.4 | moyenne | 31.6 | 15-12-22 |
| 4 | 0.99 | 12.9 | 0.8 | 5.4 | 0 | 28.7 | 15-12-22 |
| 4 | 0.98 | 12.9 | 0.72 | 5.6 | 4 | 28.7 | 15-12-22 |
| 4 | 0.99 | 12.9 | 0.75 | 6.4 | 8 | 28.7 | 15-12-22 |
| 4 | 0.98 | 12.9 | 0.75 | 6.0 | 12 | 28.7 | 15-12-22 |
| 4 | 0.99 | 12.8 | 0.76 | 6.0 | 16 | 28.7 | 15-12-22 |
| 4 | 0.99 | 12.9 | 0.7 | 5.7 | moyenne | 28.7 | 15-12-22 |

batteries_data_total

FIGURE 24 – Tableau CSV des données sur les batteries.

— Le fichier "**solar_data_total.csv**" qui reprend l'ensemble des informations sur les panneaux solaire. Comme les mesures de tensions, de courant et de puissance des panneaux à chaque heures de la journée entre 8h et 16h. La date des mesures est également indiquée pour que l'utilisateur n'ait pas à se connecter chaque jours pour retrouver les données. On retrouve également dans ce fichier une évaluation de la puissance théorique que peut normalement produire les panneaux solaire, cette évaluation n'est pas la puissance réelle produite par les panneaux car elle ne prend pas en compte le fait que les batteries soient chargés et ne draine donc pas de courant. Cette puissance est présente à titre informatif pour rendre compte à l'utilisateur de la puissance pouvant être tiré de son montage de panneaux solaire. Une explication plus détaillé du modèle théorique des panneaux solaires et du MPPT est présentée dans le rapport du projet. Il est cependant important de noter que la puissance théorique est utilisé dans l'exécutable pour déterminer l'encrassement des panneaux solaires.
Lorsque l'exécutable python n'arrive pas à se connecter aux données météorologiques, la puissance théorique est mise à 0.



| Date | Courant | Voltage | Puissance | Heure | Puissance_theorique |
|---|---|---|---|---|---|
| 15-12-22 | 4.0 | 50.0 | 8.0 | 8h | 0 |
| 15-12-22 | 4.5 | 46.0 | 9.3 | 9h | 0 |
| 15-12-22 | 4.6 | 44.0 | 10.4 | 10h | 0 |
| 15-12-22 | 4.3 | 49.0 | 11.5 | 11h | 0 |
| 15-12-22 | 3.6 | 50.0 | 12.6 | 12h | 0 |
| 15-12-22 | 4.6 | 47.0 | 11.7 | 13h | 0 |
| 15-12-22 | 4.8 | 48.0 | 9.8 | 14h | 0 |
| 15-12-22 | 4.8 | 49.0 | 9.9 | 15h | 0 |
| 15-12-22 | 4.3 | 47.0 | 7.0 | 16h | 0 |
| 05-08-22 | 4.0 | 50.0 | 8.0 | 8h | 2112.0 |
| 05-08-22 | 4.5 | 46.0 | 9.3 | 9h | 1564.0 |
| 05-08-22 | 4.6 | 44.0 | 10.4 | 10h | 995.0 |
| 05-08-22 | 4.3 | 49.0 | 11.5 | 11h | 1671.0 |
| 05-08-22 | 3.6 | 50.0 | 12.6 | 12h | 2988.0 |
| 05-08-22 | 4.6 | 47.0 | 11.7 | 13h | 2699.0 |
| 05-08-22 | 4.8 | 48.0 | 9.8 | 14h | 3088.0 |
| 05-08-22 | 4.8 | 49.0 | 9.9 | 15h | 2787.0 |
| 05-08-22 | 4.3 | 47.0 | 7.0 | 16h | 2221.0 |

solar_data_total

FIGURE 25 – Tableau CSV des données sur les panneaux solaires.

# 7 Thingstream

Thingstream est la plateforme qui gère l'envoi et la réception des messages MQTT (plus d'informations sur la plateforme thingstream et sur le protocole MQTT sont accessibles dans le rapport du projet). Dans le cadre du premier prototype, le nom du domaine Thingstream utilisé est "**codepo2021**".

## 7.1 Comment se connecter à la plate-forme Thingstream ?

Il faut se rendre sur le site : https ://portal.thingstream.io/app
Pour accéder au domaine sur lequel fonctionne le dispositif, il faut d'abord être invité par un utilisateur existant.

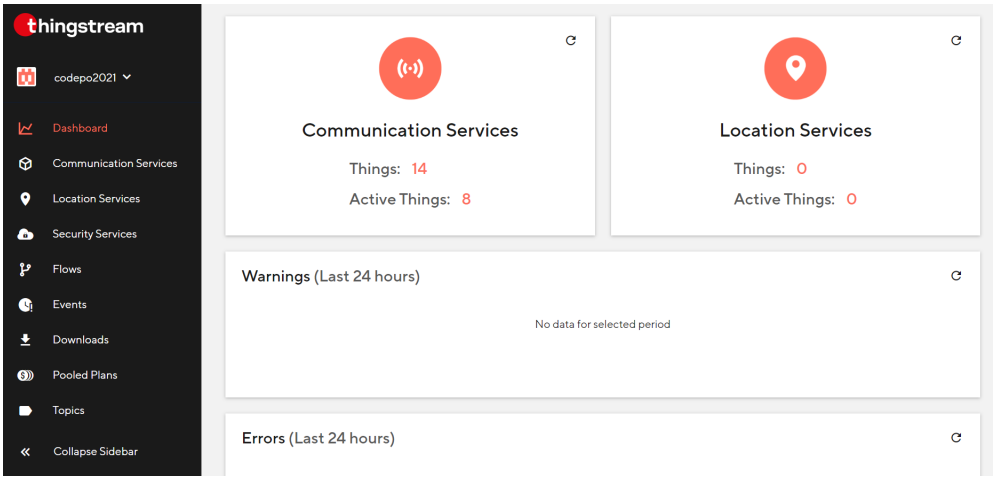



FIGURE 26 – Plateforme Thingstream "codepo2021"

## 7.2 Comment définir l'adresse mail à laquelle Thingstream envoie les mails d'alerte ?

1. Une fois connecter sur le bon domaine Thingstream, l'utilisateur se retrouve sur la page d'accueil, il faut se rendre dans l'onglet "Flows", ensuite clicker sur les 3 points verticaux à droite de "Alertes par Email" et clicker sur "Edit Flow".

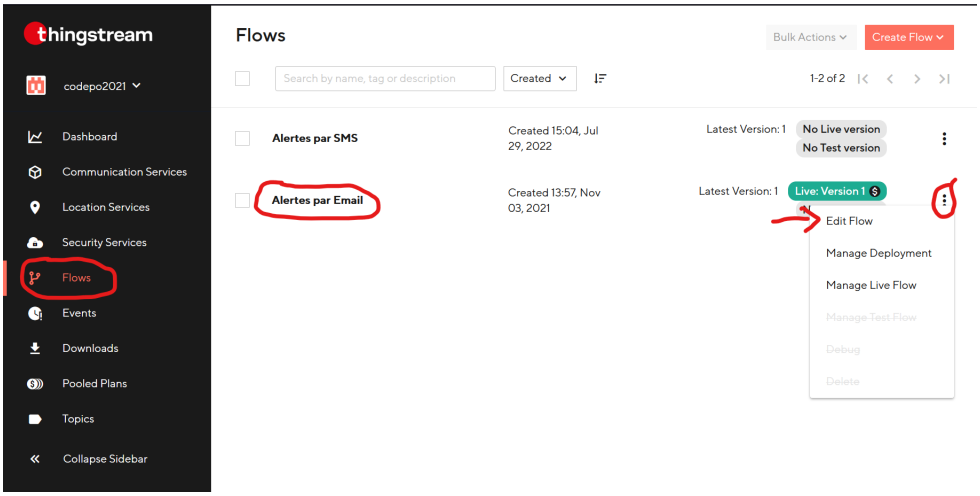



FIGURE 27 – Étape 1 : Accéder au Email flow.

2. Une fenêtre avec des blocs multicolores devrait apparaitre, c'est ici que l'utilisateur peut choisir à quel adresse email il veut envoyer les alertes. Avant de faire ça, il faut d'abord désactiver temporairement l'envoi d'email en clickant sur "Manage" et puis passer de "Live mode" au "Test mode".

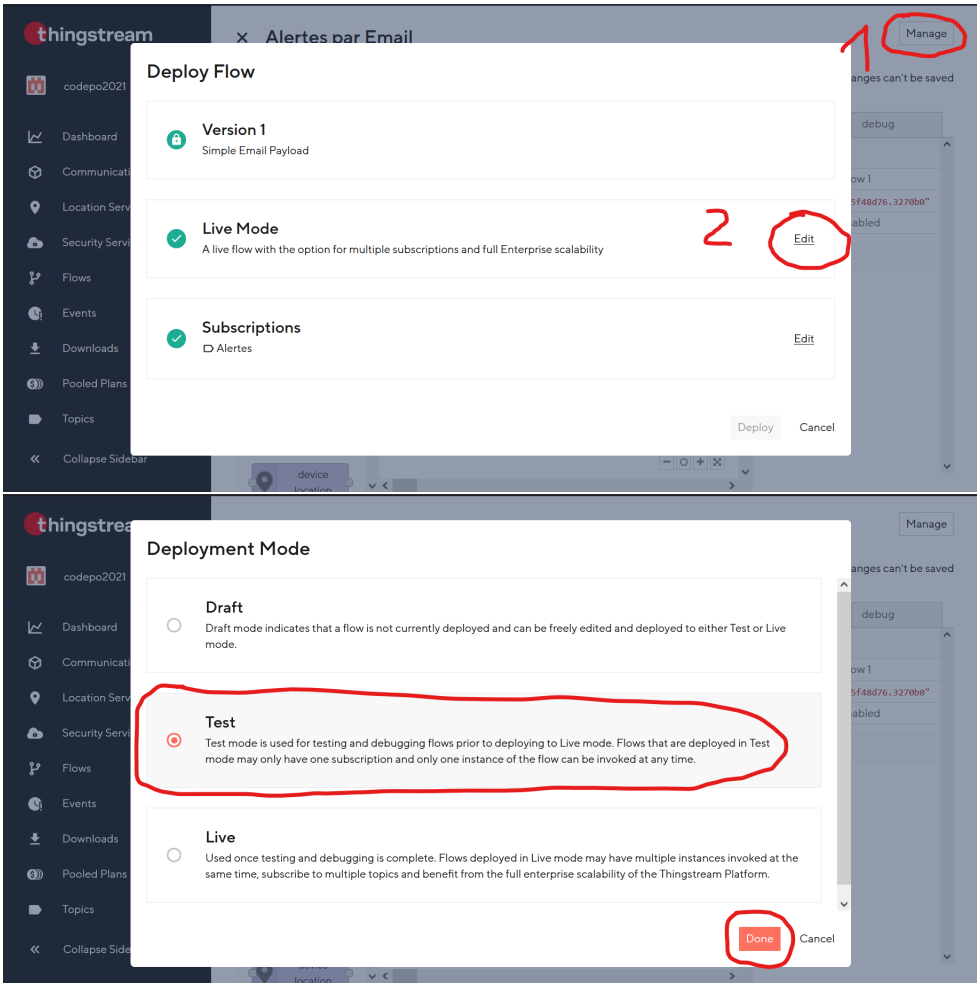



FIGURE 28 – Étape 2 :Manage flow

3. L'utilisateur peut enfin changer l'adresse email en double clickant sur l'élément "Email Payload" et changer l'adresse mail par la ou les adresses que l'utilisateur souhaite.

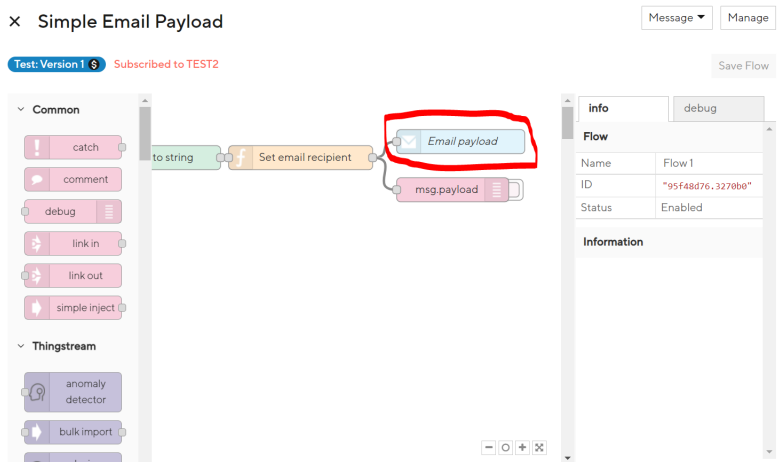



FIGURE 29 – Étape 3 :Changer l'adresse email

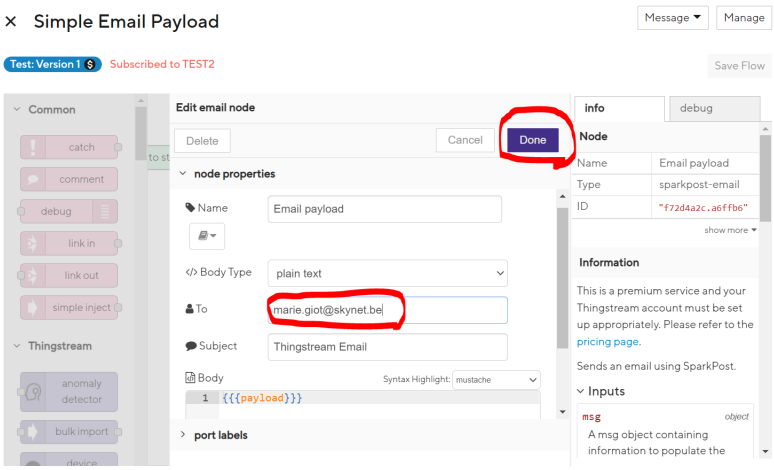



FIGURE 30 – Étape 3 :Changer l'adresse email

Le flow doit ensuite être sauvegardé. L'utilisateur doit ensuite repasser du "test mode" au "live mode" comme vu précédemment.

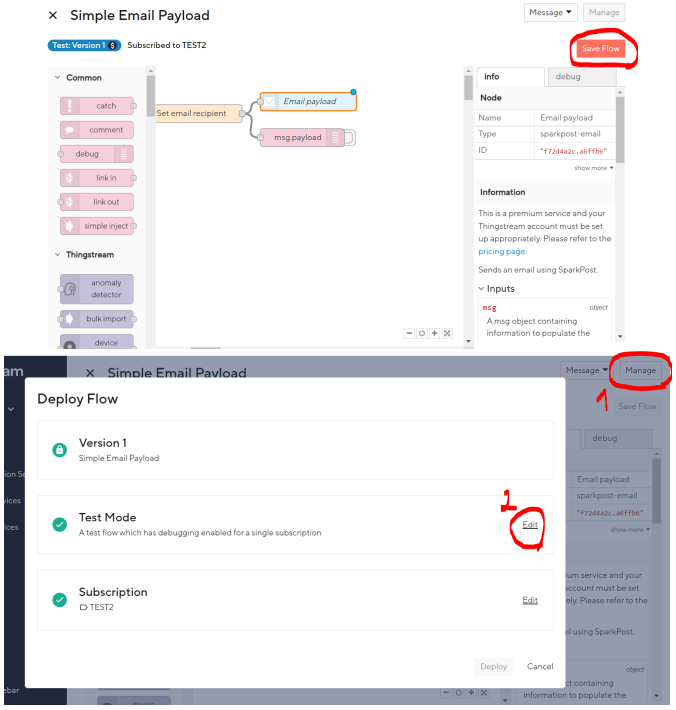



FIGURE 31 – Étape 4 :Retour au live mode

## 7.3 Facultatif : Envoi de sms

La plateforme thingstream permet, comme alternative à l'email, d'envoyer des messages d'alertes sous forme de sms. Cette option n'a pas été activé car elle est plus cher que les emails (0.07 dollars par sms contre 0.001 dollars par email). Néanmoins cette option a quand même été préparé si l'utilisateur veut l'activer. Pour l'activer il faut suivre les étapes suivantes :

1. Il faut commencer par se rendre et se connecter à la plateforme thingstream (Voir Section 7.1). Une fois sur la plateforme, il faut se rendre dans l'onglet "Flows", ensuite clicker sur les 3 points verticaux à droite de "Alertes par SMS" et clicker sur "Edit Flow".

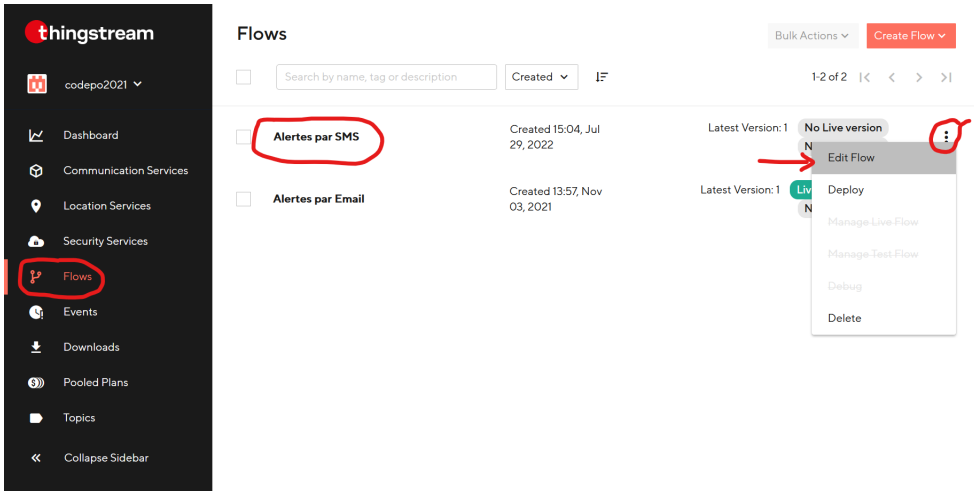



FIGURE 32 – Étape 1 : Accéder au SMS flow.

2. Une fenêtre avec des blocs multicolores devrait apparaitre, c'est ici que l'utilisateur peut choisir à quel(s) numéro(s) il veut envoyer les alertes. Pour cela, il faut double-clicker sur le bloc bleu "SMS sending".

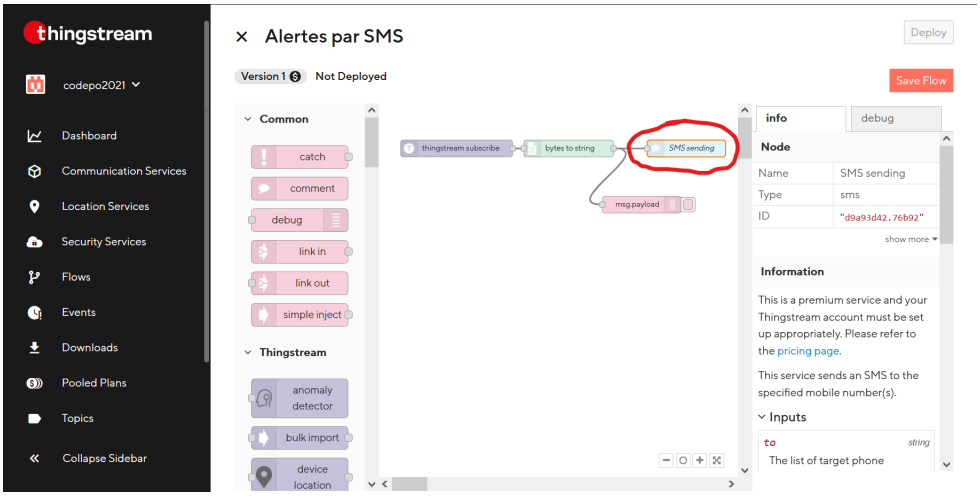



FIGURE 33 – Étape 2 : Choisir le numéro de réception.

L'utilisateur doit mettre le numéro de téléphone auquel il veut recevoir les alertes dans la case "To". Si il veut mettre plusieurs numéros de téléphones différents il doit les séparer par un "-". Par exemple, si l'utilisateur veut que les deux numéros de téléphones +243 123 456 789 et + 243 987 654 321 reçoivent les alertes alors il doit rentrer dans la case : **+243123456789-+243987654321**.

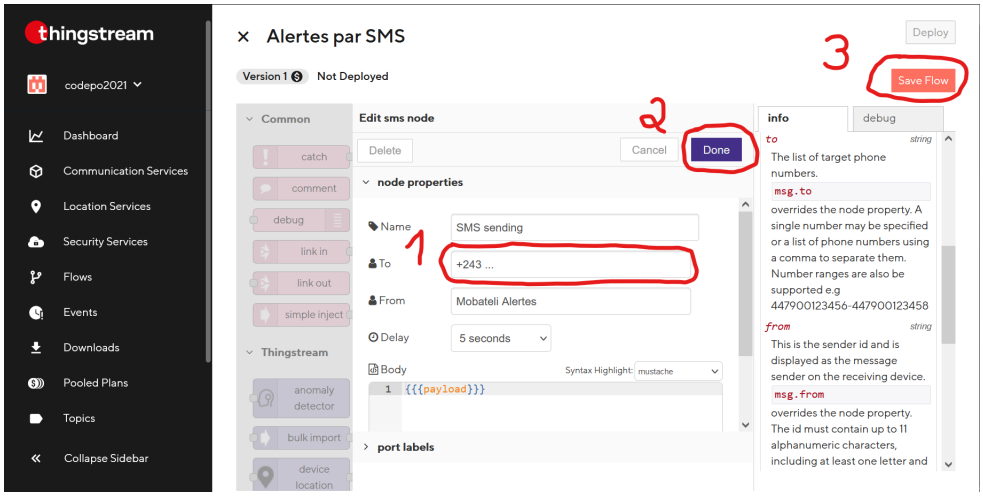



FIGURE 34 – Étape 2 : Choisir le numéro de réception.

Pour finir, il doit appuyer sur le bouton rouge "Save Flow".

3. La dernière étape consiste à activer l'envoi des alertes par sms. Pour cela, il faut clicker sur "deploy", puis sur "edit" et finalement sur "live mode" :

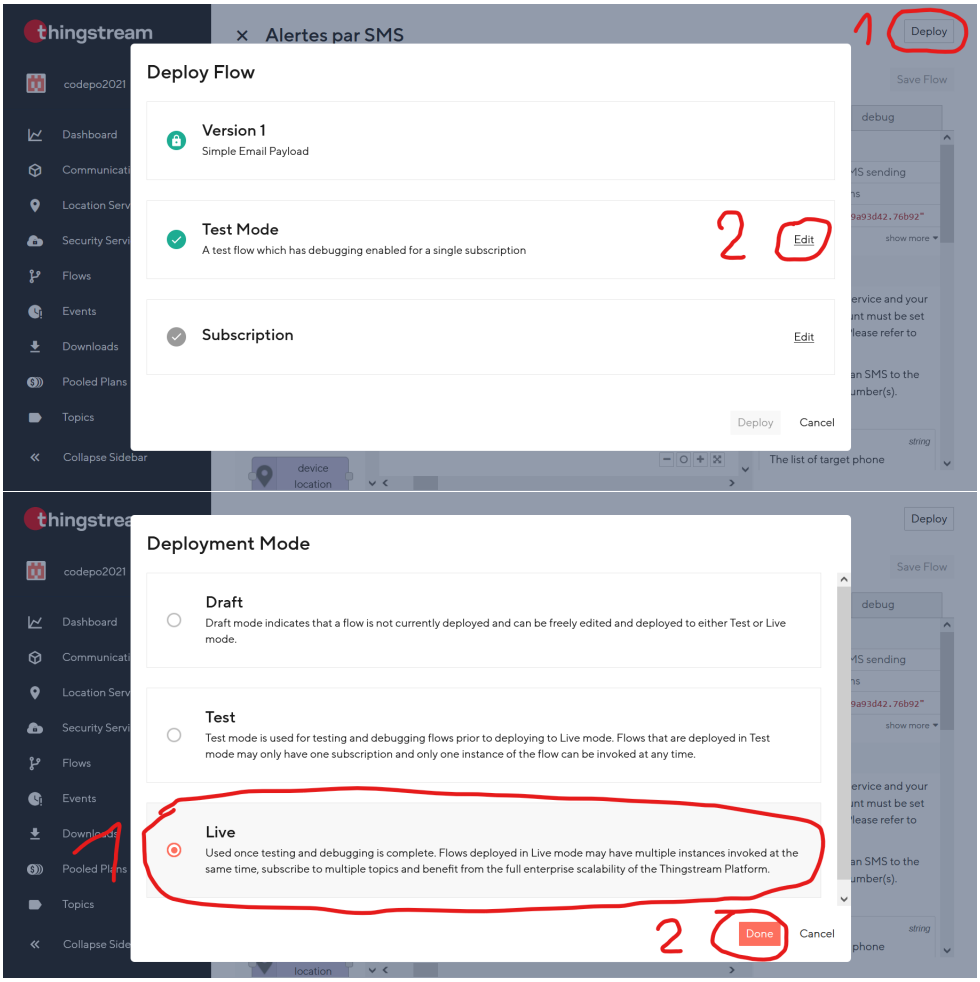



FIGURE 35 – Étape 2 : Choisir le numéro de réception.

## Références

[1] S. David. « Pulse power characterisation for lithium ion cells in automotive applications - Small and large signal cell impedance analysis ». Thèse de doct. CHALMERS UNIVERSITY OF TECHNOLOGY, 2016.

## A Notes indicatives pour l'interprétation des données de solar_data_total.cs

Ces valeurs sont valables uniquement pour les installations faites par la CODEPO à la Cameskin.

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | Date | Courant | Voltage | Puissance | Heure | Puissance_theorique | |
| 2 | 15-12-22 | 4.0 | 50.0 | 8.0 | 8h | 0 | |
| 3 | 15-12-22 | 4.5 | 46.0 | 9.3 | 9h | 0 | |
| 4 | 15-12-22 | 4.6 | 48.0 | 10.4 | 10h | 0 | |
| 5 | 15-12-22 | 4.3 | 49.0 | 11.5 | 11h | 0 | |
| 6 | 15-12-22 | 3.6 | 50.0 | 12.6 | 12h | 0 | |
| 7 | 15-12-22 | 4.6 | 47.0 | 11.7 | 13h | 0 | |
| 8 | 15-12-22 | 4.8 | 48.0 | 9.8 | 14h | 0 | |
| 9 | 15-12-22 | 4.8 | 49.0 | 9.9 | 15h | 0 | |
| 10 | 15-12-22 | 4.3 | 47.0 | 7.0 | 16h | 0 | |
| 11 | 05-08-22 | 4.0 | 50.0 | 8.0 | 8h | 2112.0 | |
| 12 | 05-08-22 | 4.5 | 46.0 | 9.3 | 9h | 1564.0 | |
| 13 | 05-08-22 | 4.6 | 48.0 | 10.4 | 10h | 995.0 | |
| 14 | 05-08-22 | 4.3 | 49.0 | 11.5 | 11h | 1671.0 | |
| 15 | 05-08-22 | 3.6 | 50.0 | 12.6 | 12h | 2988.0 | |
| 16 | 05-08-22 | 4.6 | 47.0 | 11.7 | 13h | 2659.0 | |
| 17 | 05-08-22 | 4.8 | 48.0 | 9.8 | 14h | 1088.0 | |
| 18 | 05-08-22 | 4.8 | 49.0 | 9.9 | 15h | 2787.0 | |
| 19 | 05-08-22 | 4.3 | 47.0 | 7.0 | 16h | 2221.0 | |

La puissance est égale à 0 soit parce que l'exécutable n'a pas été lancé le jour de la prise de ces données soit parce que l'utilisateur qui a lancé l'exécutable n'a pas une bonne connexion internet

La puissance théorique est une indication de la puissance maximale que les panneaux solaires pourraient fournir à cette heure-là. Elle est calculée à partir des données météorologiques

Les données sont fournies toutes les 2 heures au cours de la journée

Les puissances réelle devrait être aux alentours des 1300W si les panneaux sont propres et que les batteries ne sont pas chargées au maximum

Figure 36