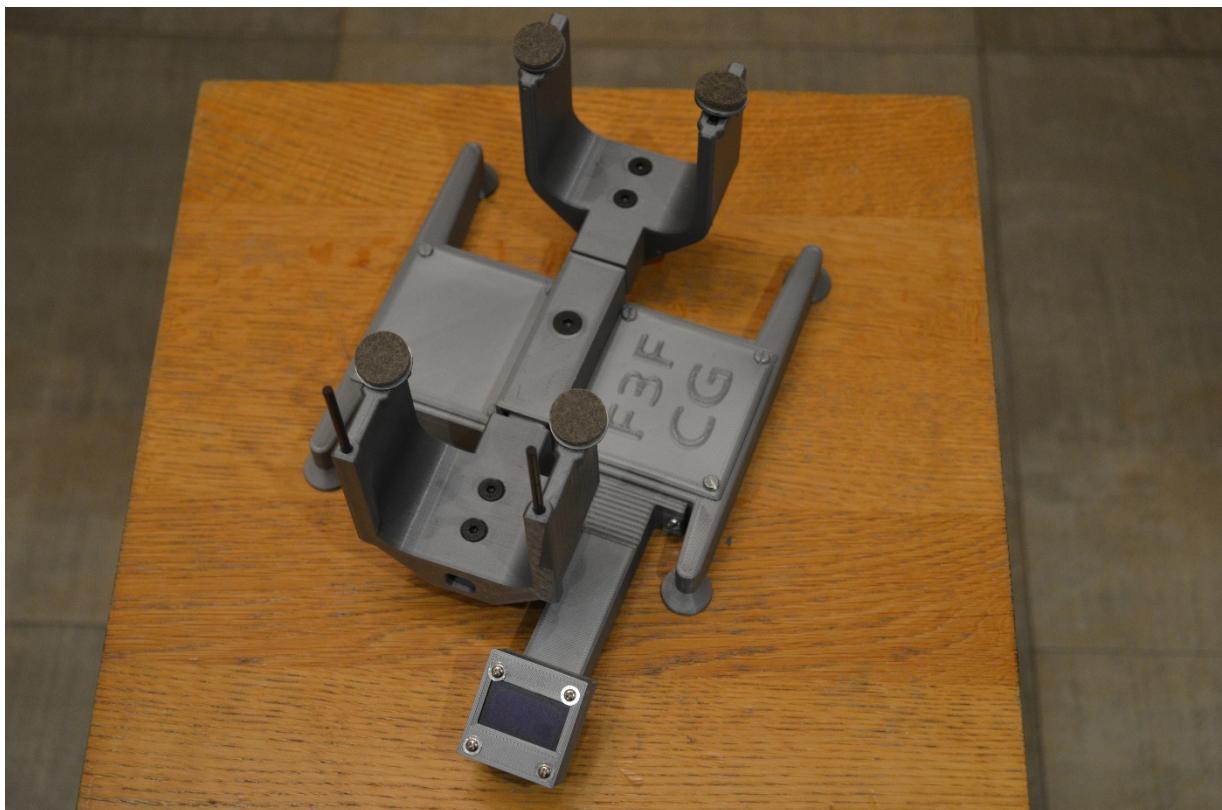


# Balance à CG basée sur Arduino pour les planeurs F3x (et autres).

Enjoy !!



Version	Date	Motif
1.0	26.02.2018	Création.
1.1	17.03.2018	Maj.

## 1 Introduction

Ce document présente les instructions de montage d'une balance à Cg utilisable pour des planeurs de FNx.

Fin 2017, de nombreux posts sont apparus sur les forums d'aéromodélisme, présentant une balance numérique dédiée à la détermination du Cg pour des planeurs de F3F et F3B.

En fouillant sur le net à la recherche d'informations sur cet équipement commercialisé un peu plus de 130 euros dans divers boutiques, je suis tombé sur le site **d'Olav Kallhovd**, pilote Norvégien de F3F (**merci à lui**), ([https://github.com/olkal/CG\\_scale](https://github.com/olkal/CG_scale)), qui a mis en open source toutes les informations nécessaires pour construire une balance très similaire aux modèles commerciaux en utilisant deux cartes Arduino et une imprimante 3D.

La photo ci-dessous présente la balance conçue par Olav. Tous les éléments nécessaires pour la construction sont disponibles sur son site, y compris les fichiers « .stl » pour les impressions 3D.

Cette balance peut être utilisée pour des planeurs de type F3F, F3B et F5J, les différentes largeurs étant disponibles pour l'impression 3D.



**Figure 1 :** Le modèle initialement conçu par Olav Kallhovd avec un écran déporté.

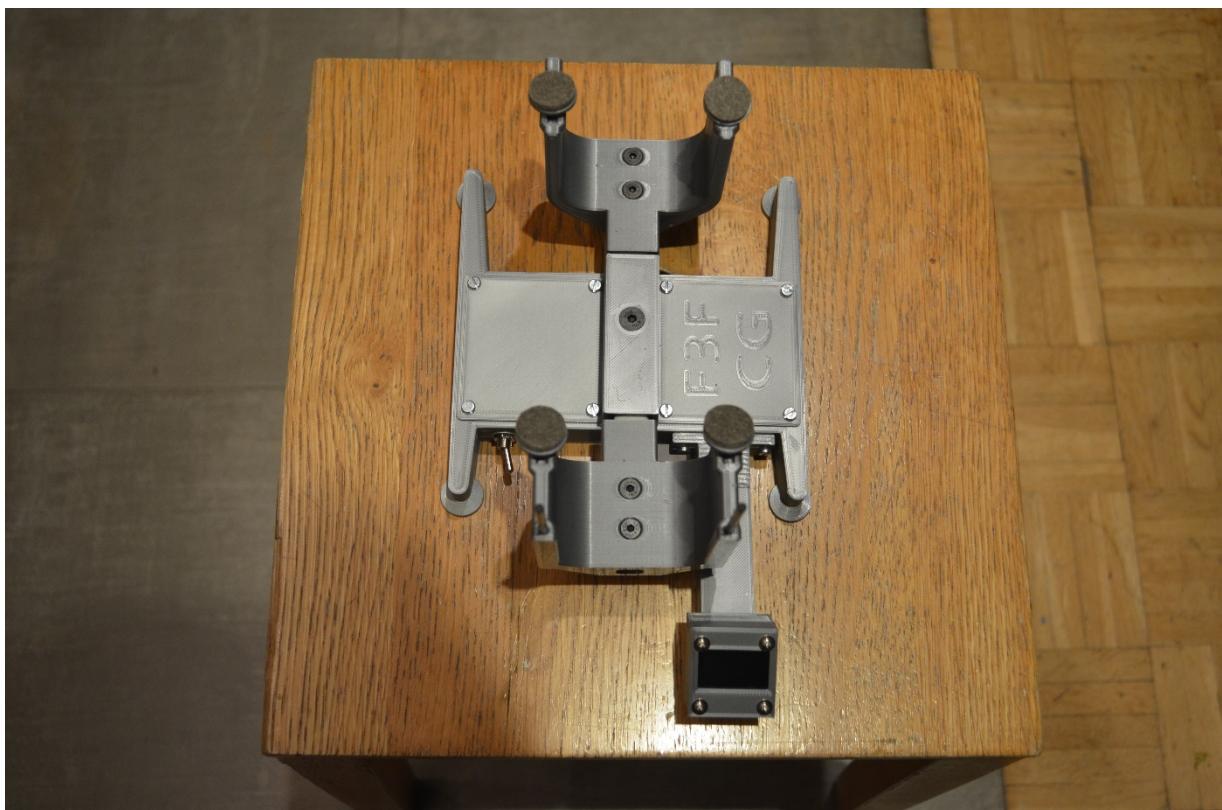
Après avoir construit un modèle identique à celui initialement proposé par Olav, avec l'aide de collègues de l'Aéroclub National des électriciens et Gazier (ANE) (Damien Donsez, Christophe Nocchi et Fred Hours), nous avons introduit quelques modifications pour faciliter et simplifier la construction de cette balance :

- L'écran LCD 1602 initial a été remplacé par un écran OLED 0,96" en I2C. Il est également possible d'utiliser un écran LCD 1602 en I2C et vous trouverez sur le site ci-dessous les fichiers Arduino pour les trois versions (version 1602 de base proposée par Olav, OLED 0,96" I2C et 1602 I2C). L'utilisation de l'I2C pour l'écran permet de simplifier le câblage et de **n'utiliser qu'une seule carte Arduino** au lieu de 2,
- La led externe visible sur la photo ci-dessus a été remplacée par la led présente de base sur les cartes Arduino. Si vous imprimez en PLA « clair » comme sur la photo, le clignotement de la led dans la base reste très visible,

Un petit PCB a été conçu pour faciliter l'intégration des quelques composants qui constituent l'électronique. Ce PCB s'intègre dans la base de la balance, et les fichiers Fritzing et Gerber associés sont également disponibles sur le github ci-dessous. Avec ces fichiers, il vous est possible de commander directement le PCB sur le site de Fritzing Fab (<https://aisler.net/fritzing>) ou Seed Fusion ([https://www.seeedstudio.com/fusion\\_pcb.html](https://www.seeedstudio.com/fusion_pcb.html)). Il est tout à fait possible de se passer de ce PCB pour monter la balance, en câblant les divers composants l'un à l'autre en fil à fil ... mais c'est moins simple et plus délicat !

- Quelques stl ont été repris pour agrandir quelques trous et nous avons créé un stl supplémentaire pour intégrer un bras de connexion permettant de rattacher l'écran OLED à la base.

Toutes ces modifications, y compris le fichier Fritzing et les gerber pour le PCB, sont disponibles en Open Sources sur le site [https://github.com/adesandr/CG\\_Scale\\_OlkalBreakout](https://github.com/adesandr/CG_Scale_OlkalBreakout). Une fois terminée, la balance en version écran OLED 0,96" ressemble à la photo ci-dessous.



**Figure 2 : Balance terminée en version OLED 0,96"**

Les étapes de montage de la version F3F avec écran OLED 0,96" en I2C sont présentées dans les chapitres suivants.

## 2 Impression et montage des éléments de la balance

### 2.1 Les composants nécessaires

La liste des composants nécessaires est disponible à l'adresse :

[https://github.com/adesandr/CG\\_Scale\\_OlkalBreakout/blob/master/Documentation/BOM.xls](https://github.com/adesandr/CG_Scale_OlkalBreakout/blob/master/Documentation/BOM.xls).

Tous les composants sont disponibles sur Hobbyking, Ebay, Amazon ou équivalent.

En prenant en compte le coût du PLA pour les impressions 3D, les diverses versions reviennent environ à 35 €

La balance se monte en une journée pour la version initiale et une demi-journée pour les versions I2c, calibrage compris, fonction de votre expérience avec un fer à souder et de vos connaissances de l'environnement Arduino. Je ne peux que vous encourager à vous lancer, vous verrez c'est extrêmement simple et le tuto est un peu long pour vous présenter pas à pas les différentes étapes du montage.

J'attire votre attention sur les deux warnings présents dans le fichier pour l'achat des cartes HX711 et de la carte Arduino Pro mini 5V. Certaines cartes HX711 génère un niveau de bruit trop élevé et impact les mesures et certaines cartes Arduino n'ont pas les broches A6/A7 de contrôle du bus I2c correctement placées par rapport au PCB.

### 2.2 Imprimer la balance

La photo suivante présente les divers éléments de la balance après impression avec les fichiers .stl associés. A noter que la partie basse du boîtier écran est à coller à l'époxy sur le « Bras balance ».

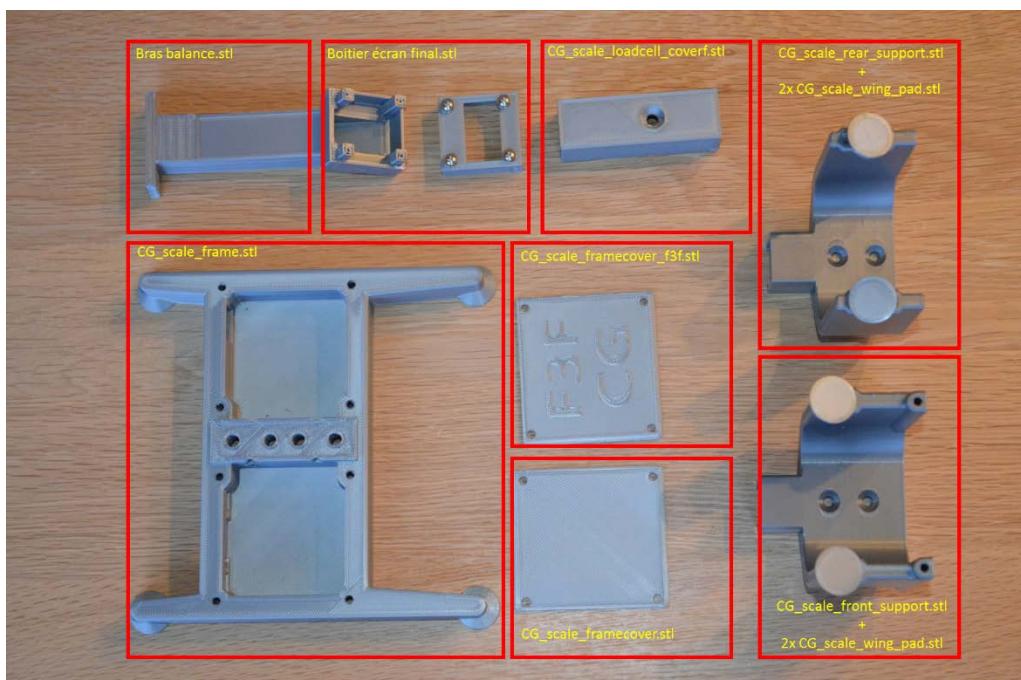


Figure 3 : Les divers éléments de la balance avec les fichiers .stl associés.

Une Imprimante 3D avec un plateau de 220x220 minimum est nécessaire. Le PLA Ice Filament disponible sur Amazon fonctionne bien, ainsi que le PLA Hobby King, avec 2 épaisseurs de paroi et un remplissage à 20%. L'impression de toutes les pièces de la balance prend **une vingtaine d'heures**.

Les pièces doivent être imprimées avec la même orientation que dans les fichiers STL.

Les trous pénétrants sont « aveuglés » pour une meilleure impression et doivent être repris par la suite à la Dremel.

## 2.3 Monter les éléments mécaniques de la balance

### 2.3.1 Etape 1 : Installer les jauge de contrainte.

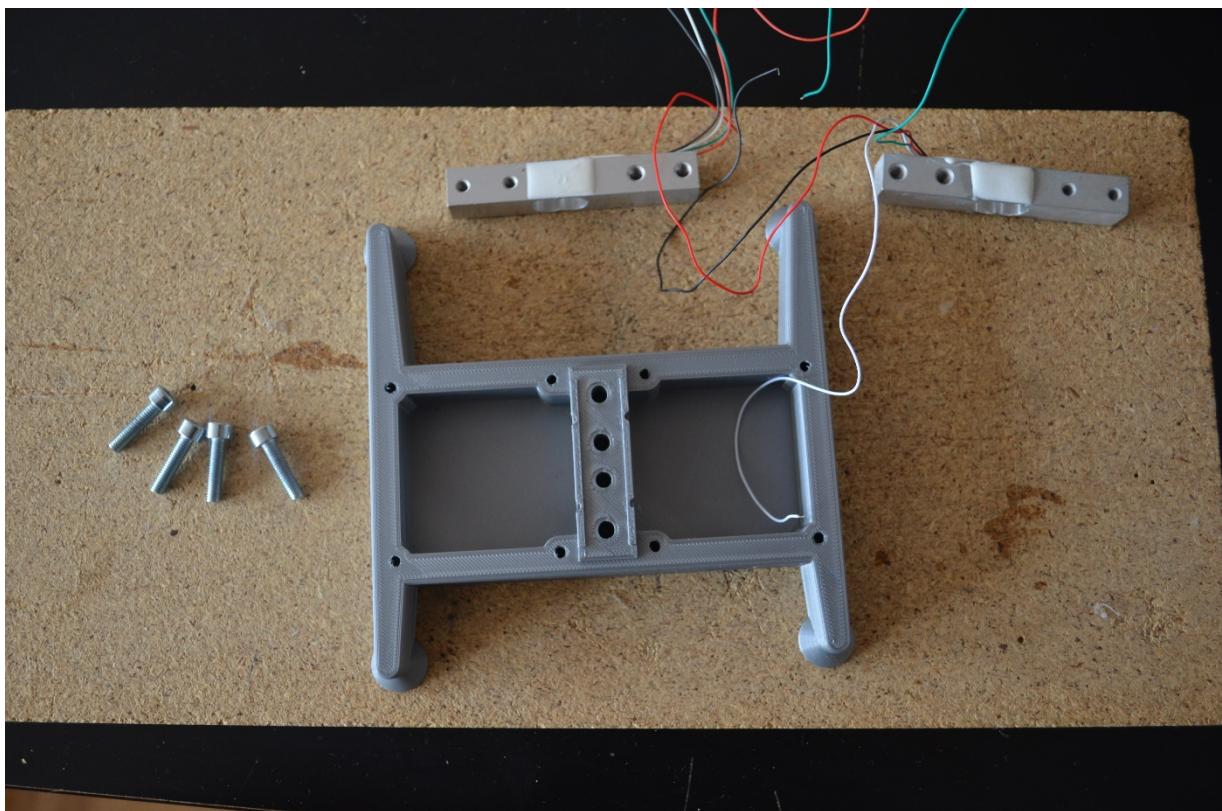
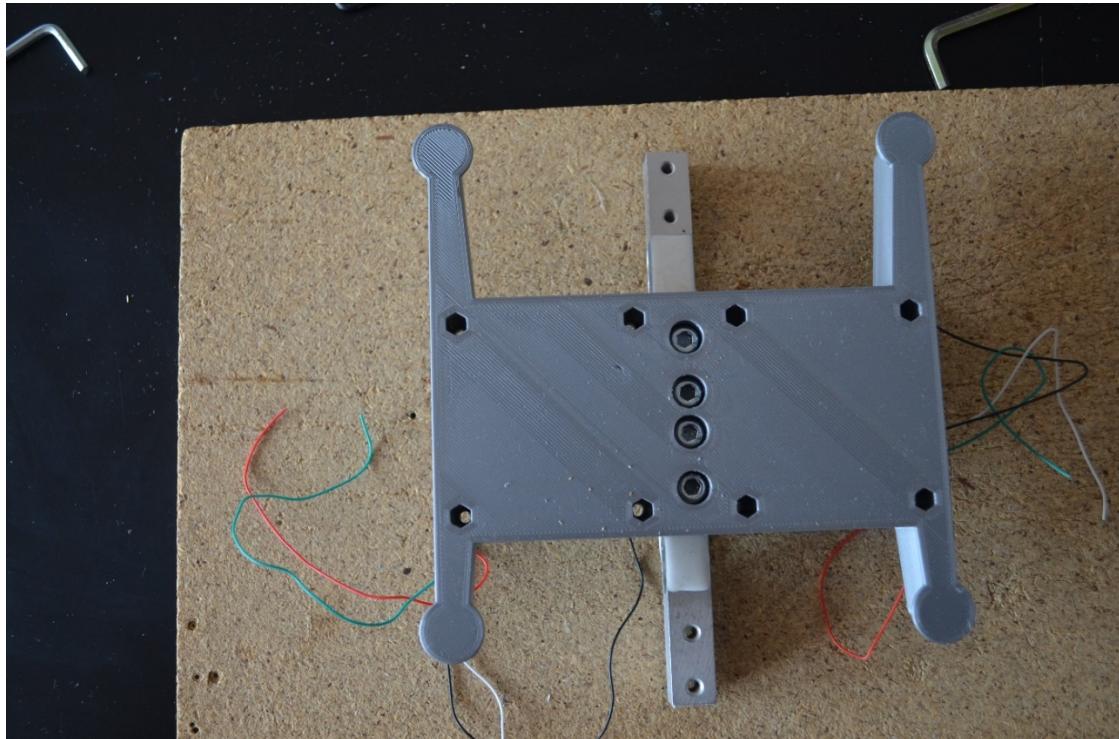


Figure 4 : Etape 1 – installation des jauge de contrainte.

Les jauge sont installées sur la base de la balance à l'aide de 4 vis BTR 5 mm x 20 mm. Les surfaces de contact des jauge de contrainte doivent être plates, et peuvent nécessiter un petit ponçage. Vérifier l'alignement global des deux jauge de contrainte pour que la balance soit précise.

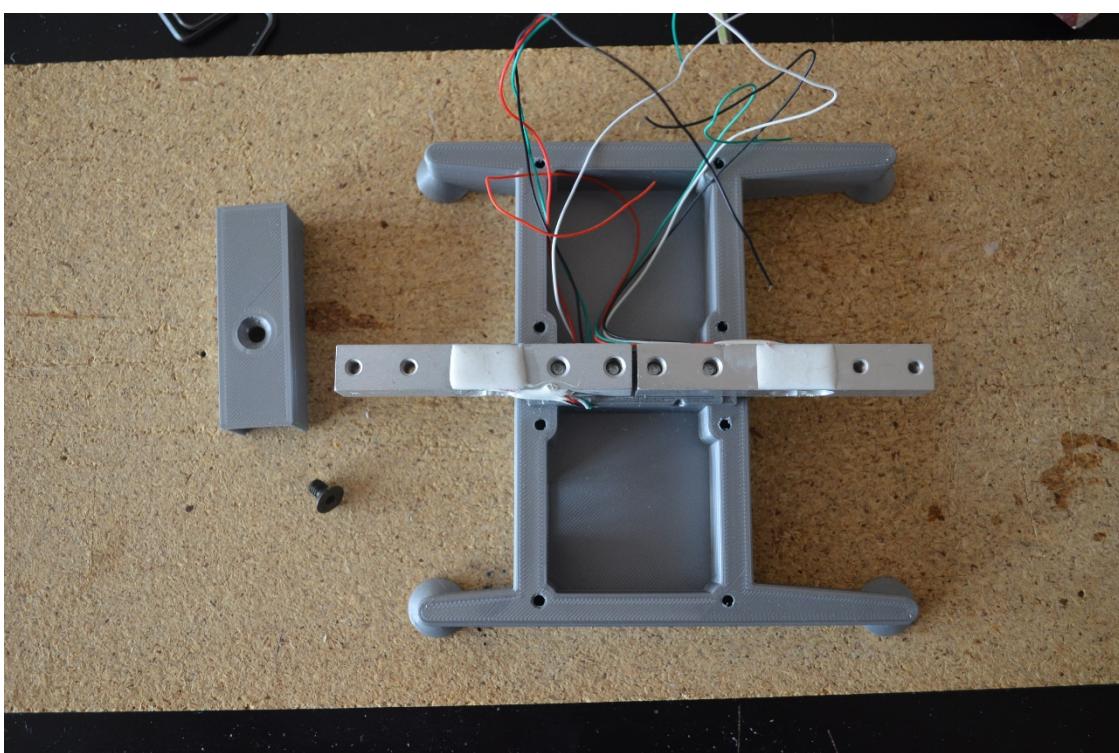
La jauge de 2 kg est installée à l'avant de la balance, celle de 3 kg est installée à l'arrière. L'avant de la balance est représenté vers le bas sur la photo de la figure 3.

**Eviter de couper les fils qui se trouvent sur les jauge, sous peine de modifier les équilibres du pont de Wheatstone qui est intégré dans chaque jauge, ce qui modifie la précision de la balance.**



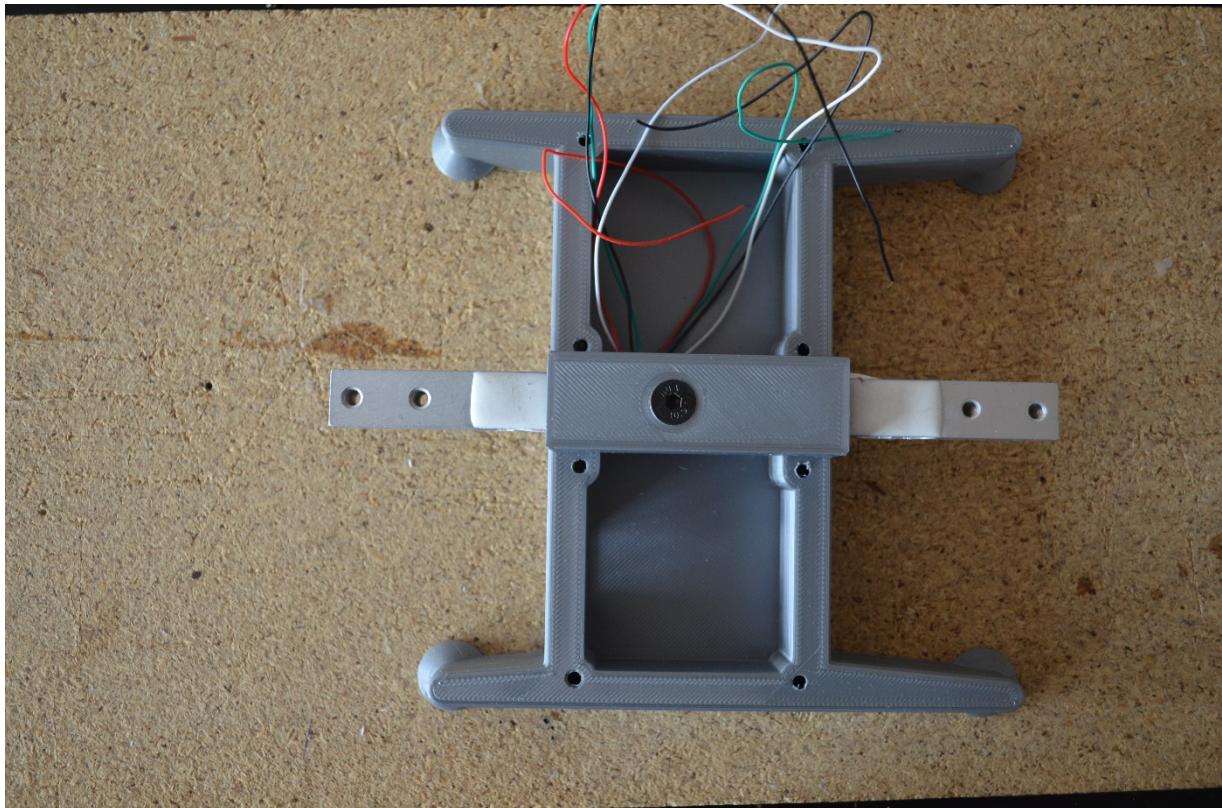
**Figure 5 : Etape1 – jauge de contrainte vissées sur la base de la balance.**

### 2.3.2      **Etape 2 : Installer le cache jauge.**



**Figure 6 : Etape2 – Installer le cache jauge.**

Le cache jauge est installé en utilisant une vis BTR M5 x 8 mm à tête plate.



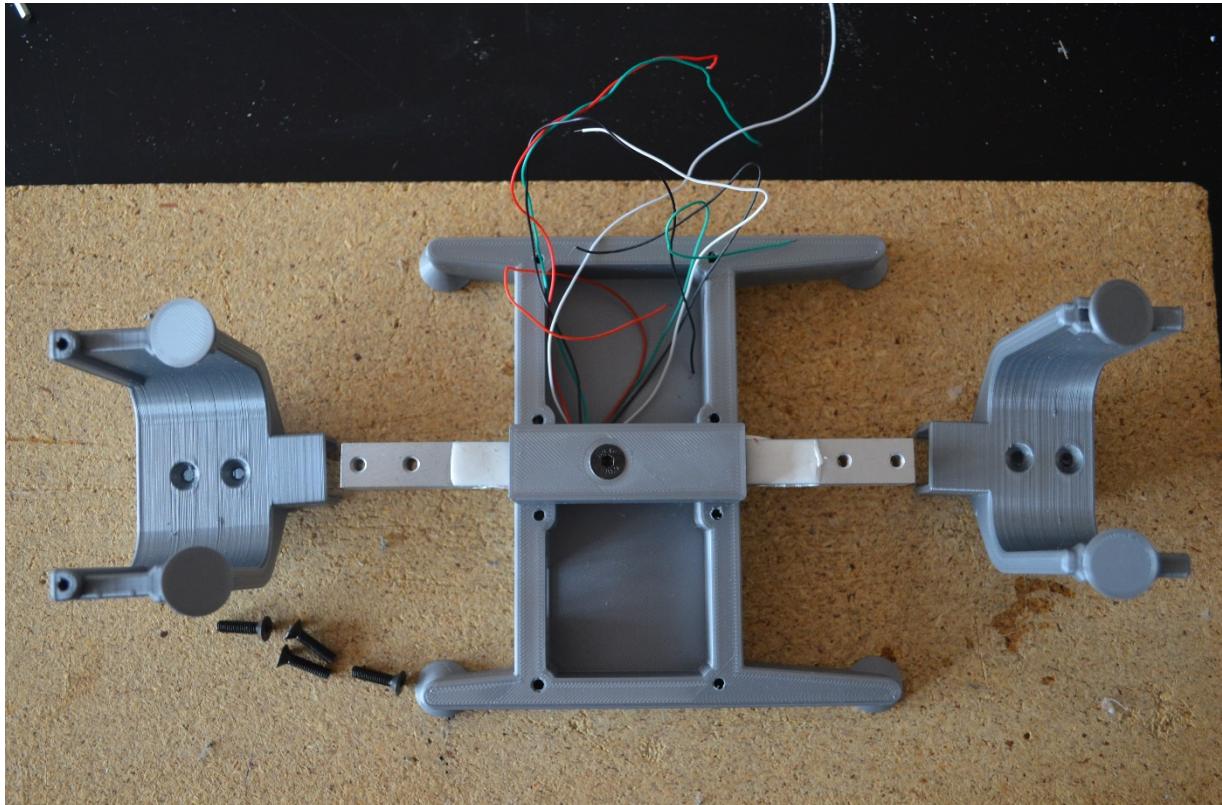
*Figure 7 : Etape2 – Installation du cache jauge avec une vis M5 x 8 mm.*

### 2.3.3      **Etape 3 : Installer les supports de planeur.**

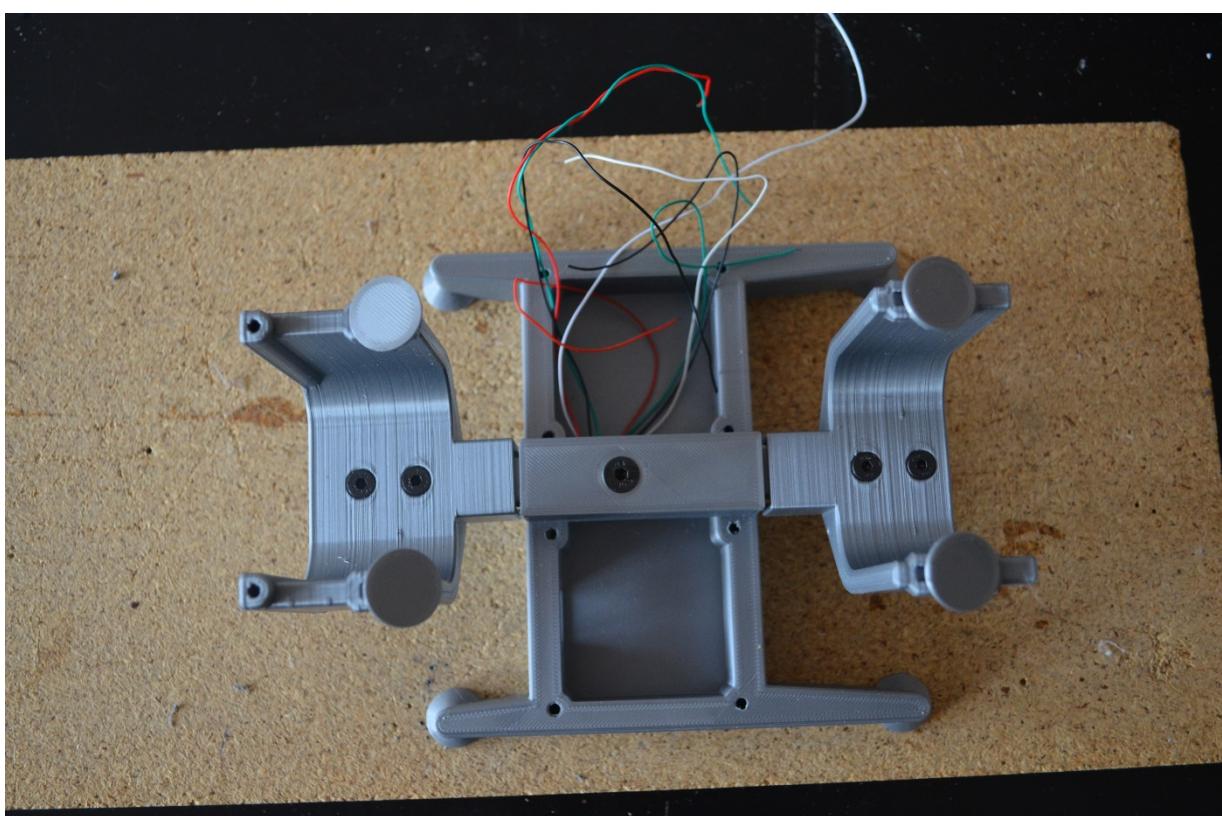
Les deux supports avant et arrière sont installés en utilisant 4 vis M5 x 16 mm.

**IMPORTANT:** Les 4 coussinets d'aile sont inclinés pour que la balance puisse être utilisée avec un planeur avec des ailes dièdres. L'articulation pour les coussinets doit être desserrée pour éviter toute liaison et frottements. Utiliser éventuellement un foret de 2 mm pour le trou permettant d'installer les coussinets s'ils sont trop serrés.

Fixez des tampons autocollants en feutrine (merci Leroy Merlin) sur les coussinets d'aile et sous le cadre si vous le souhaitez.

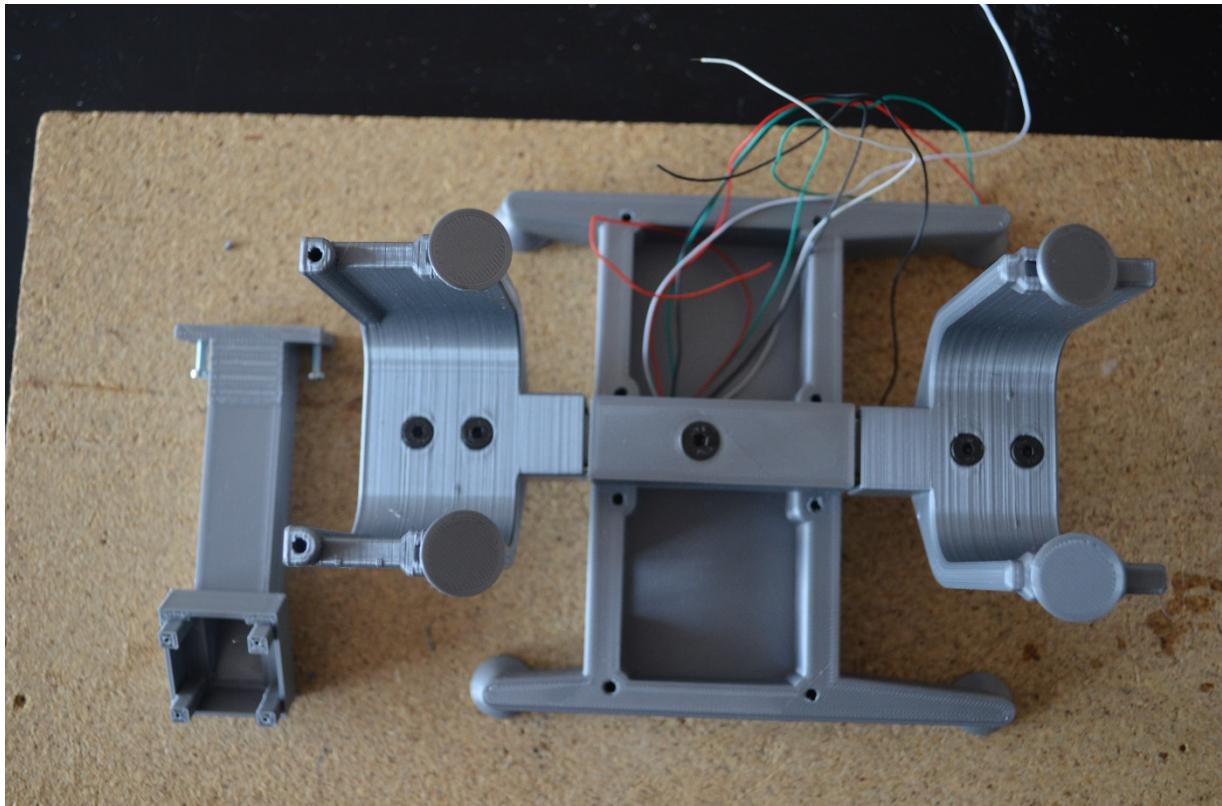


**Figure 8 :** Etape 3 – Installer les supports de planeur avant et arrière avec des vis M5 x 16 mm.



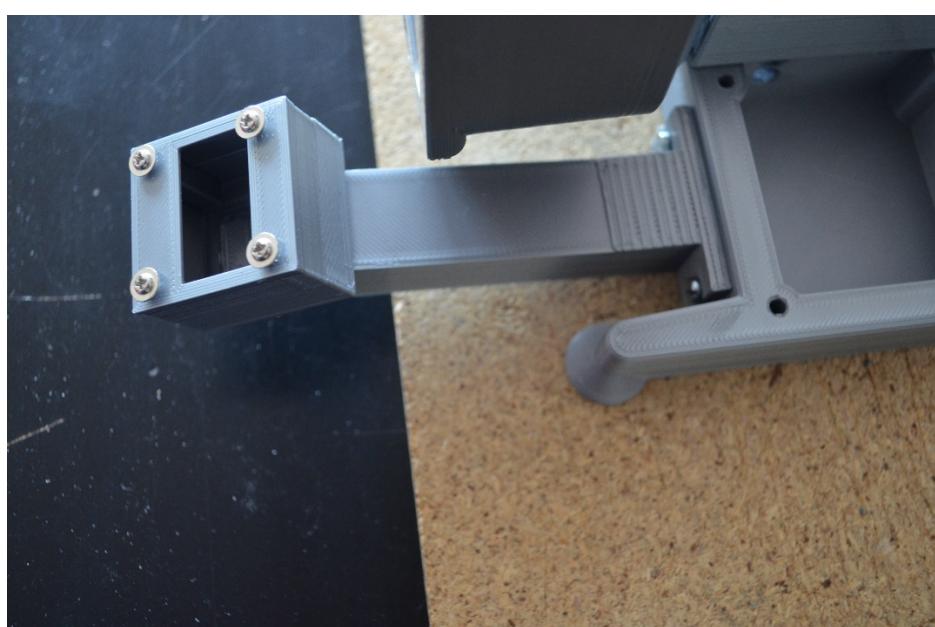
**Figure 9 :** Etape 3 – supports avant & arrière de planeur installés.

### 2.3.4     Etape 4 : Fixer le support pour l'écran



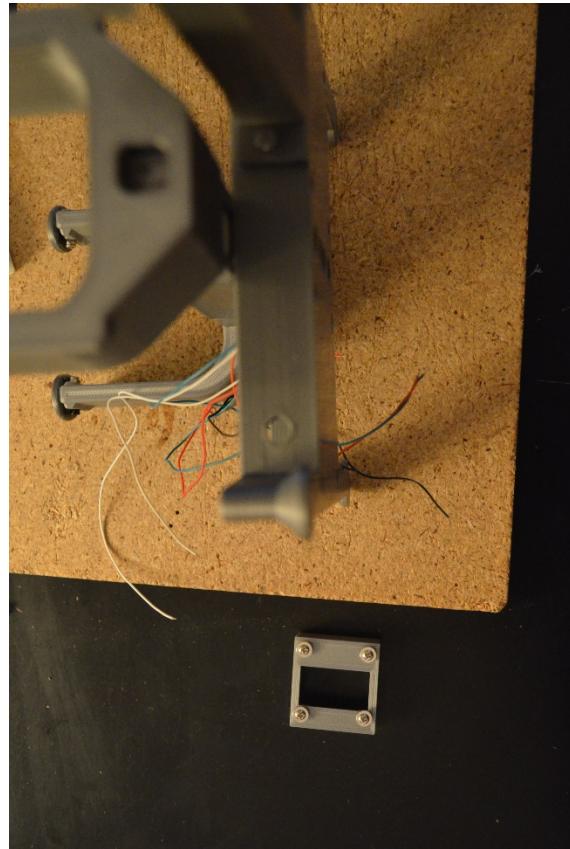
*Figure 10 : Etape 4 – Fixer le support pour l'écran*

Le support d'écran est fixé à l'aide de deux vis à tête plate M3 x 20 mm. A noter que la base du boîtier écran doit être collée au préalable à l'époxy, sur le bras support de l'écran.



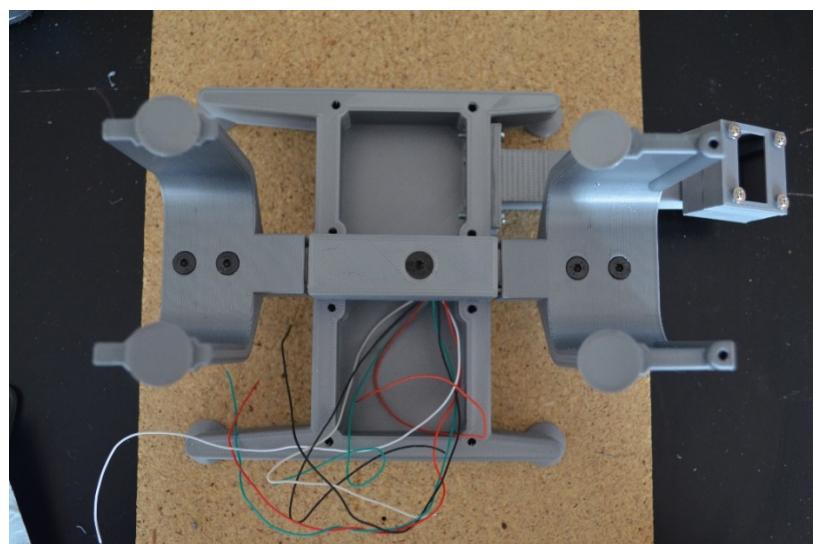
*Figure 11 : Etape 4 – Le support pour l'écran est fixé à l'aide de deux vis M3 x 20 mm.*

Pour terminer cette étape, libérer le trou de 6 mm occulter à l'impression prévu pour l'installation d'un interrupteur à bascule et coller deux tiges carbones de 3 mm x 45 m dans les deux trous du support avant de la balance.



**Figure 12 :** Etape 4 – Reprendre à la Dremel le trou pour le passage de l'interrupteur à bascule.

Le montage des éléments de la balance est terminé et nous pouvons passer à l'intégration des composants électroniques.



**Figure 13 :** Etape 4 – La balance avec tous les éléments mécaniques montés.

## 2.4 Intégration des composants électroniques

### 2.4.1 Les composants nécessaires au montage.

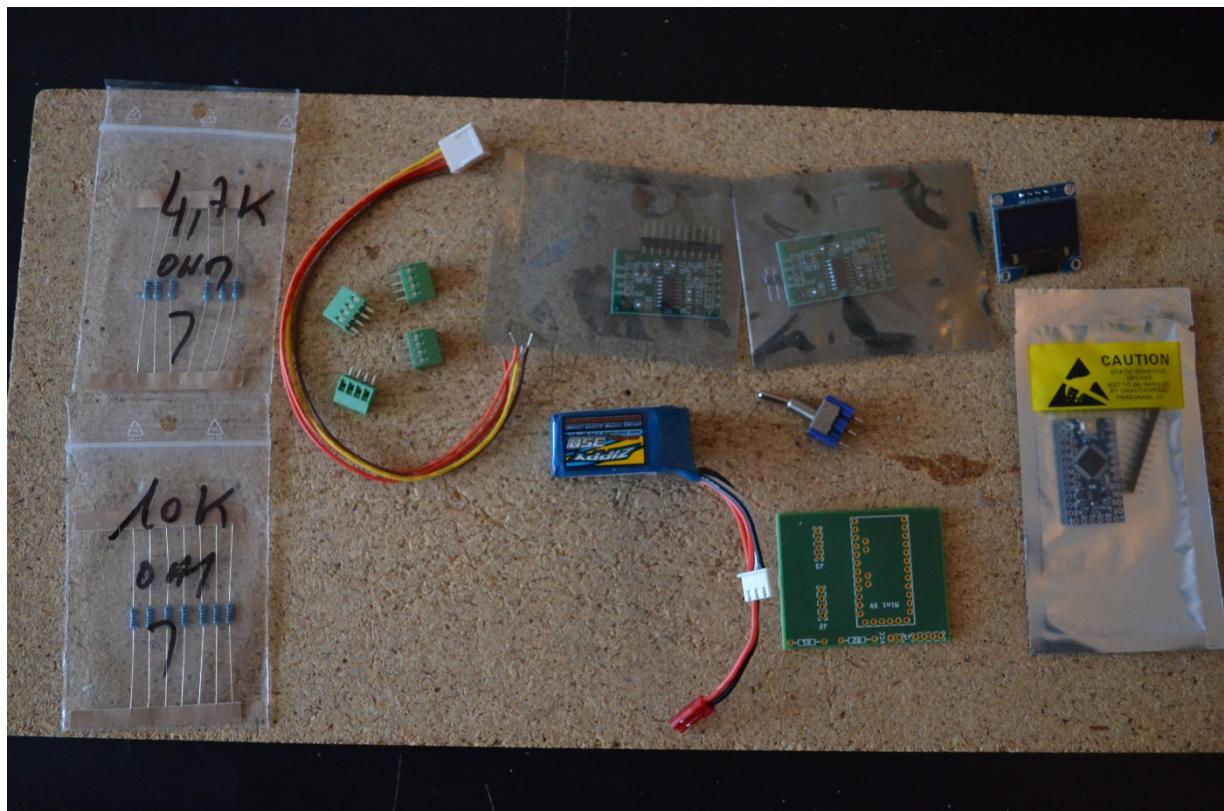
La photo suivante présente les composants nécessaires au montage de la balance. Une résistance 4,7 KΩ et une résistance 10 KΩ sont nécessaires pour le pont diviseur de tension qui permet de mesurer la tension de la batterie.

Une nappe 4 fils permet de connecter le PCB à l'écran OLED 0,96".

4 borniers à vis au pas de 2.54 sont utilisés pour souder sur les deux HX711 pour simplifier la connexion des jauge de contraintes. Il est possible de s'en passer, mais le montage est beaucoup plus simple en les utilisant.

Deux cartes HX711 et une carte Arduino Pro mini 5V sont également utilisées, ainsi qu'un interrupteur à bascule et une batterie LiPo de dimension maximum 41 mm x 21 mm x 16 mm.

Une petite plaque de prototypage (ou breadboard) vous sera également utile pour aider à réaliser quelques soudures (environ 3 €).



**Figure 14 :** Les composants électroniques nécessaires au montage de la balance.

La carte Arduino Pro mini n'ayant pas de port USB, il est également nécessaire de se procurer un programmeur TTL-USB. Vous pouvez acheter un programmeur de ce type sur Amazon,

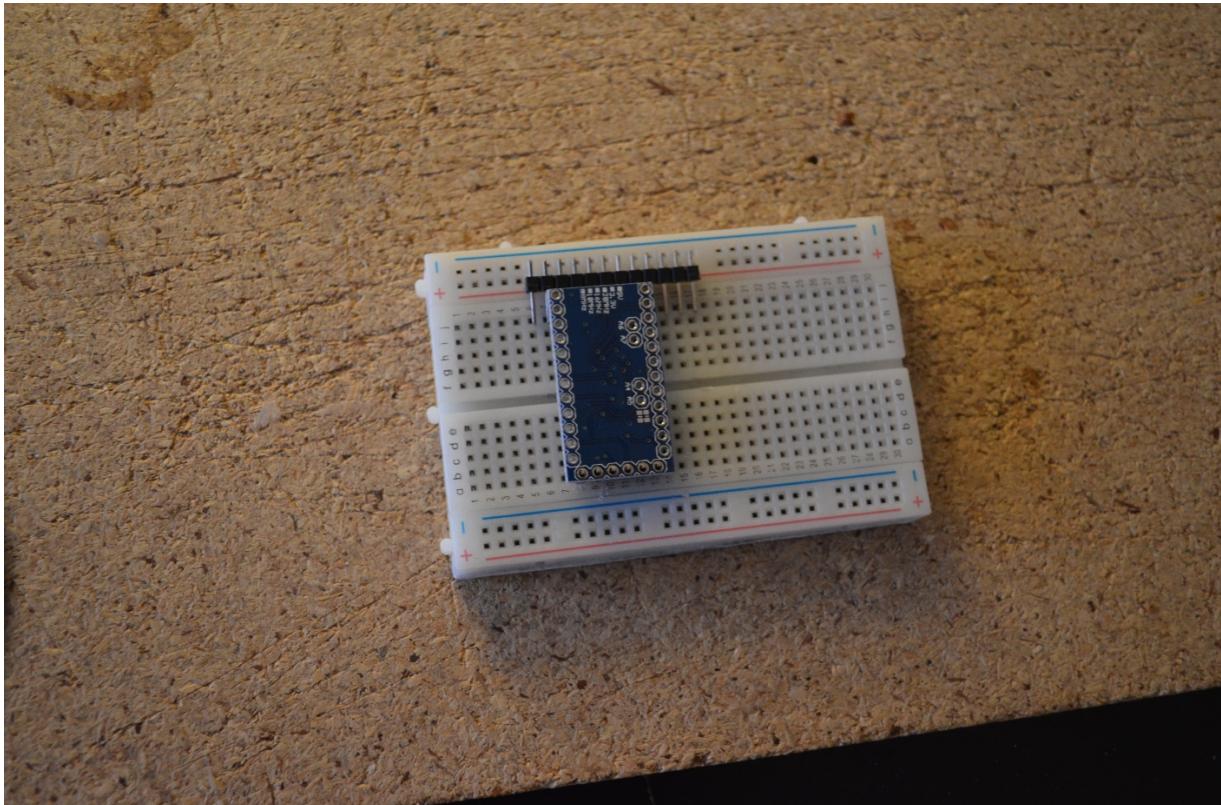
[https://www.amazon.fr/gp/product/B06Y38P7P2/ref=oh\\_aui\\_detailpage\\_o07\\_s00?ie=UTF8&psc=1](https://www.amazon.fr/gp/product/B06Y38P7P2/ref=oh_aui_detailpage_o07_s00?ie=UTF8&psc=1). Ce type de programmateur coûte entre 7€ et 8 €.



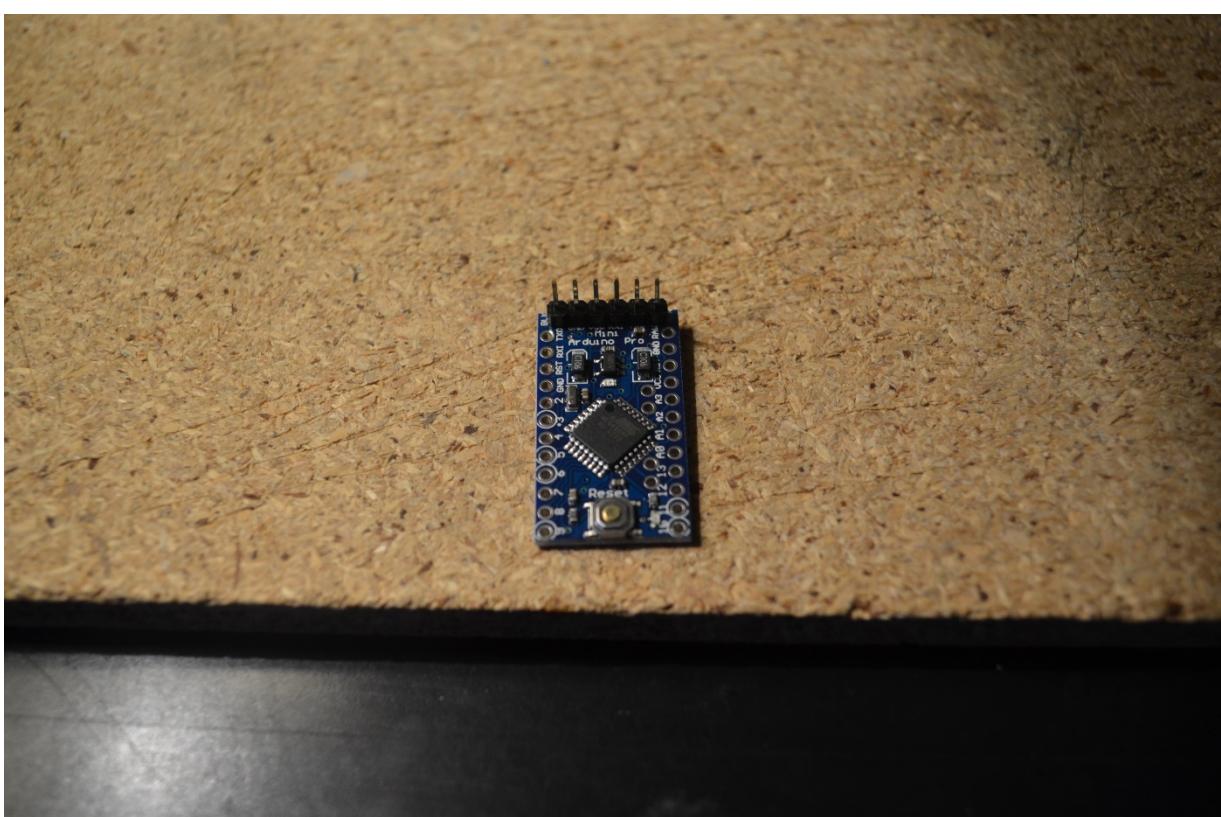
**Figure 15 :** Programmateur TTL-USB 3.3V 5 V (Goaxing Tech par exemple).

#### 2.4.2 Etape 5 : préparer la carte Arduino

Nous allons commencer par souder le connecteur de programmation sur la carte Arduino. Pour cela une petite breadboard va nous faciliter les soudures. Le connecteur de programmation doit être soudé « vers le haut » pour faciliter la programmation de la balance quand les divers composants auront été intégrés dans la base.



**Figure 16 :** Etape 5 – soudure du connecteur de programmation sur l'Arduino.

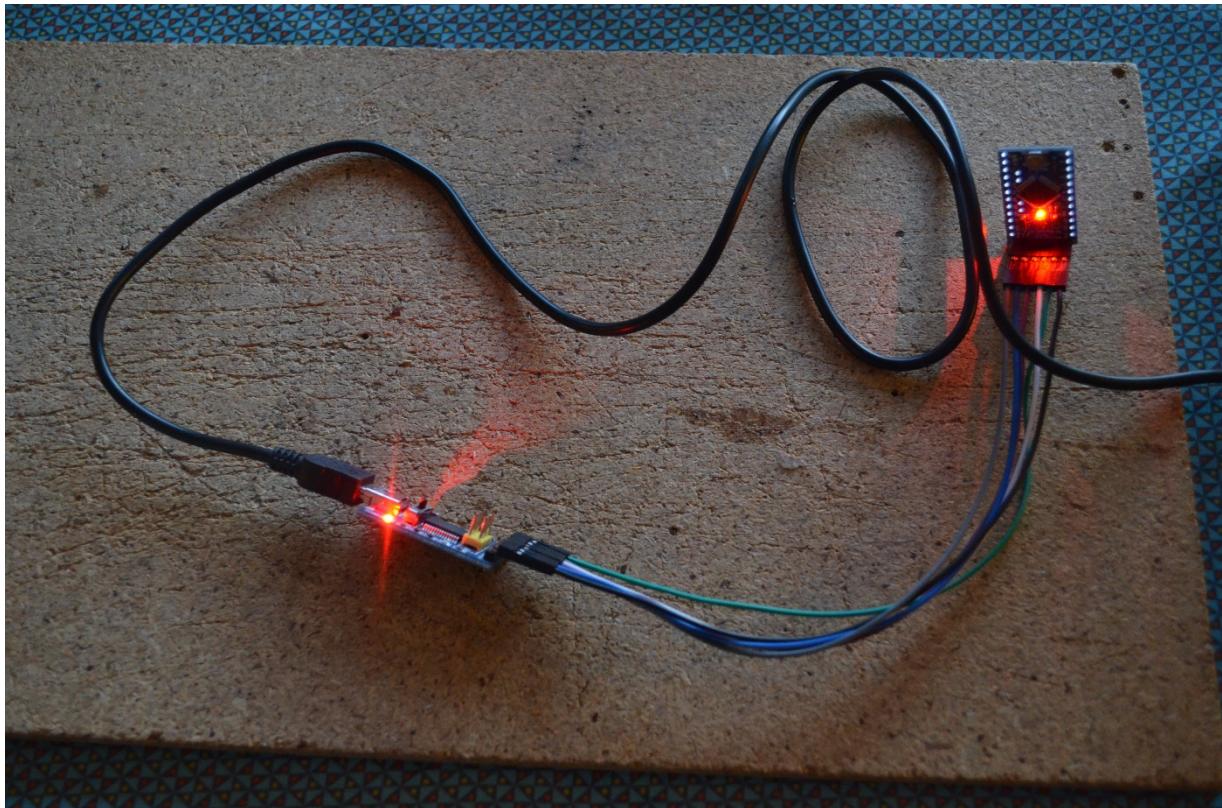


**Figure 17 :** Etape 5 – le connecteur de programmation doit être soudé « vers le haut ».

Une fois le connecteur soudé sur la carte, je vous recommande d'effectuer un test de bon fonctionnement de cette dernière.

Pour ce faire, relier la carte Arduino à votre PC par l'intermédiaire du programmeur TTL-USB en respectant bien les polarités (notamment GND et VCC).

Toutes les cartes Arduino étant livrées avec un programme réalisant le clignotement de la led installée sur la carte, si votre carte est fonctionnelle, après branchement au PC, la led doit clignoter avec une fréquence d'environ 1s.



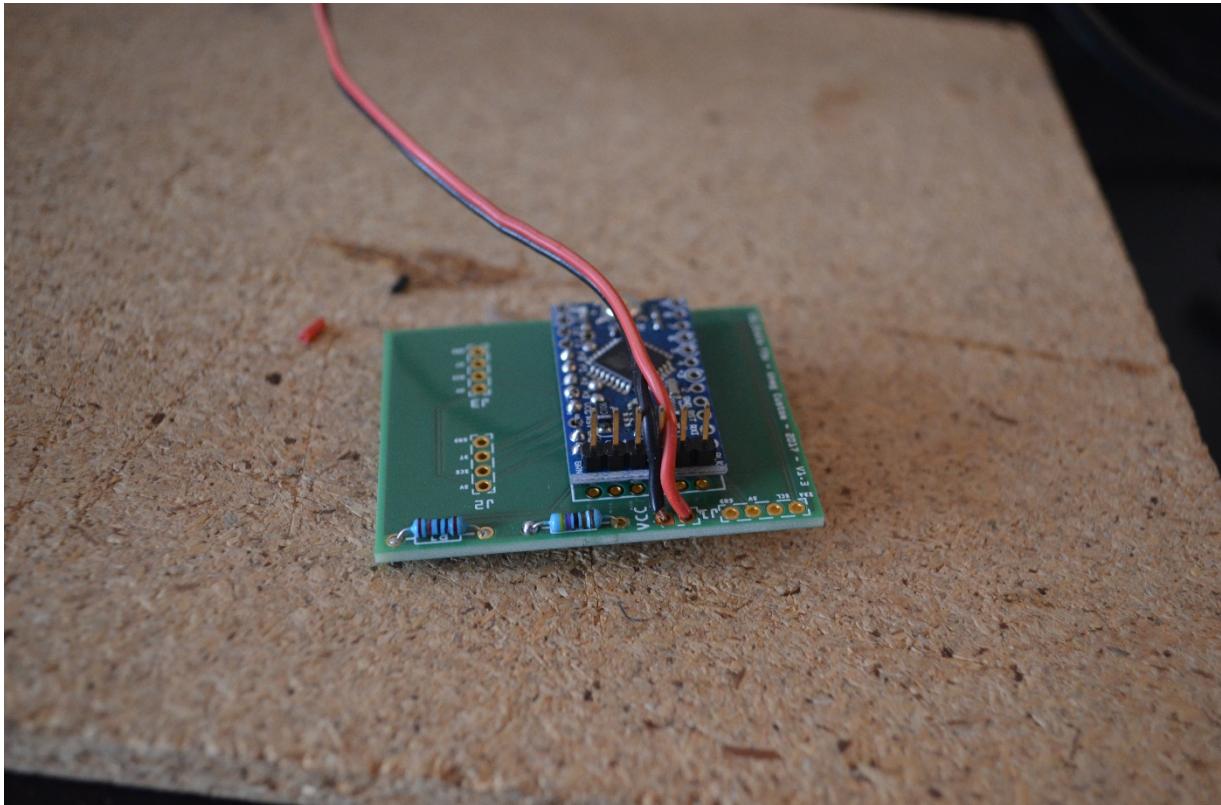
*Figure 18 : Etape 5 – test du bon fonctionnement de la carte Arduino.*

#### 2.4.3     **Etape 6 : Préparer le PCB**

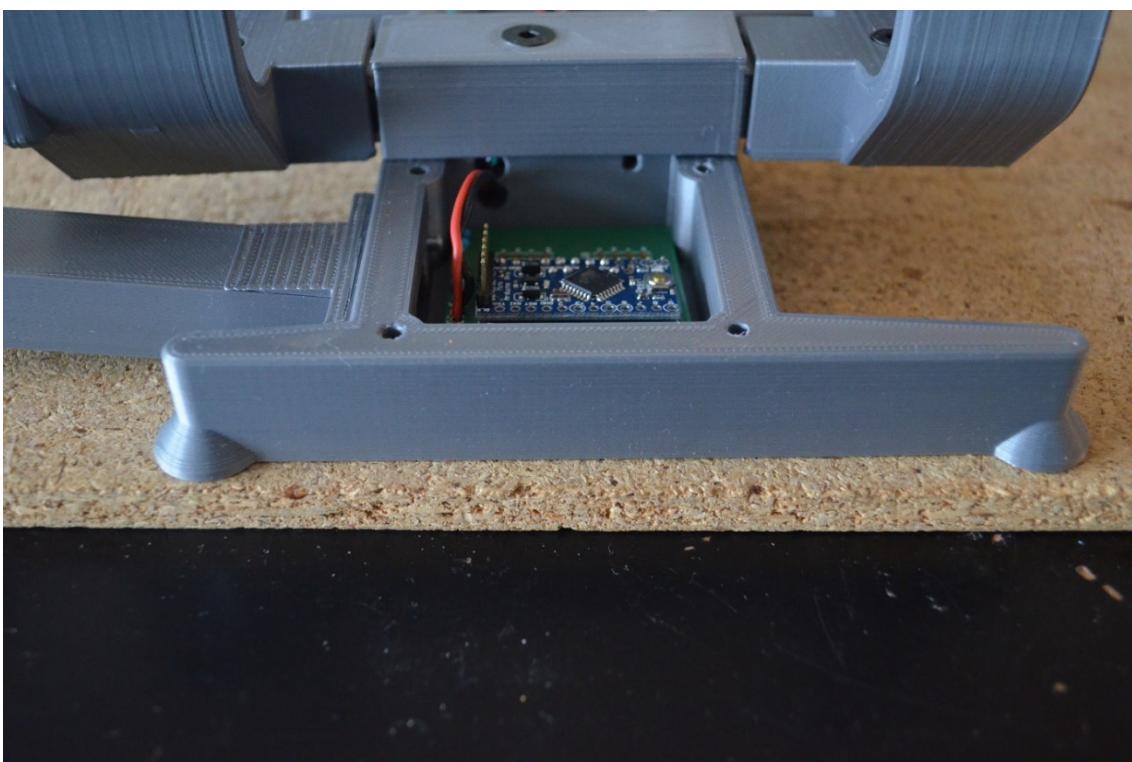
La préparation du PCB consiste à souder une résistance de  $10\text{ k}\Omega$  sur l'emplacement R1 et une résistance de  $4,7\text{ k}\Omega$  sur l'emplacement R2.

Souder également la carte Arduino sur le PCB (pins : raw, gnd, A0, A1, A2, A3, A4, A5, A7). Seuls ces pins sont utilisées.

Souder enfin un fil de masse et un fil d'alimentation sur le connecteur J1 (prévoir un peu de marge pour le fil, 15 cm permet d'adapter par la suite).



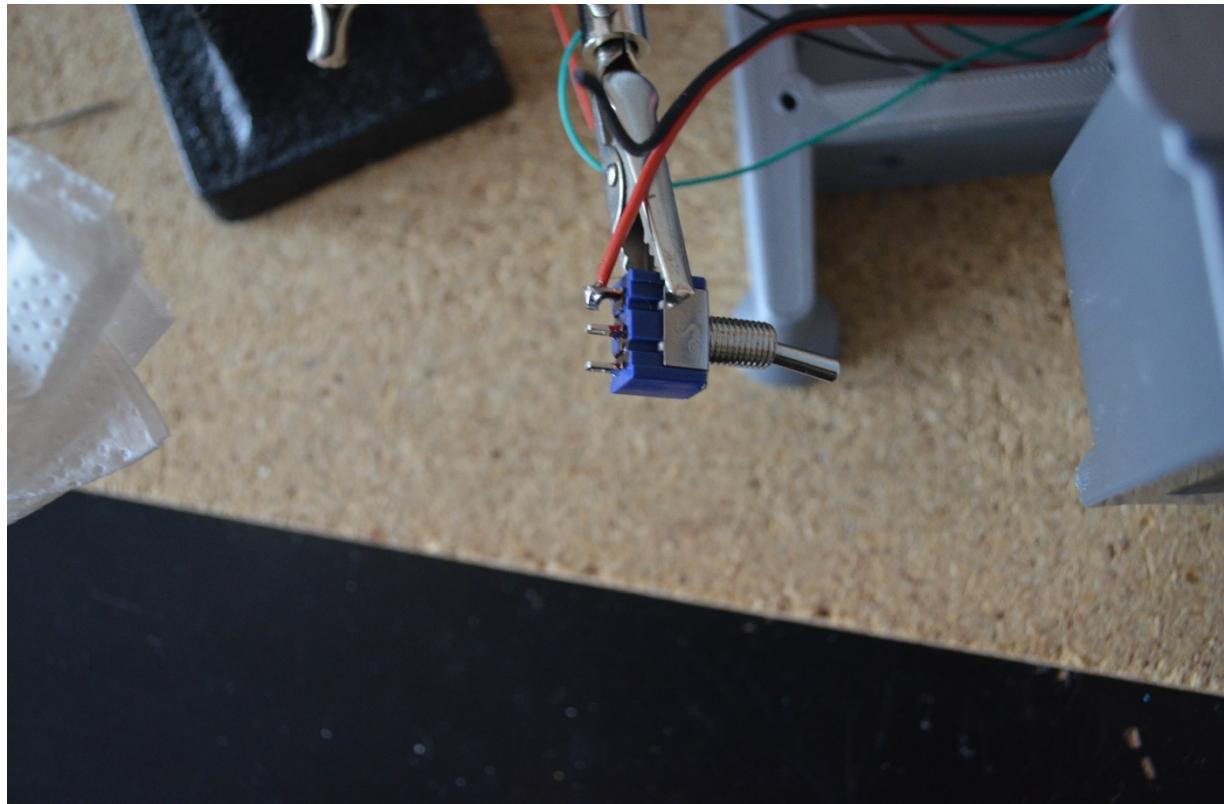
**Figure 19 : Etape 6 – Préparation du PCB**



**Figure 20 : Etape 6 – Test d'installation du PCB dans la base.**

### 2.4.1     Etape 7 : Mise en place de l'interrupteur à bascule.

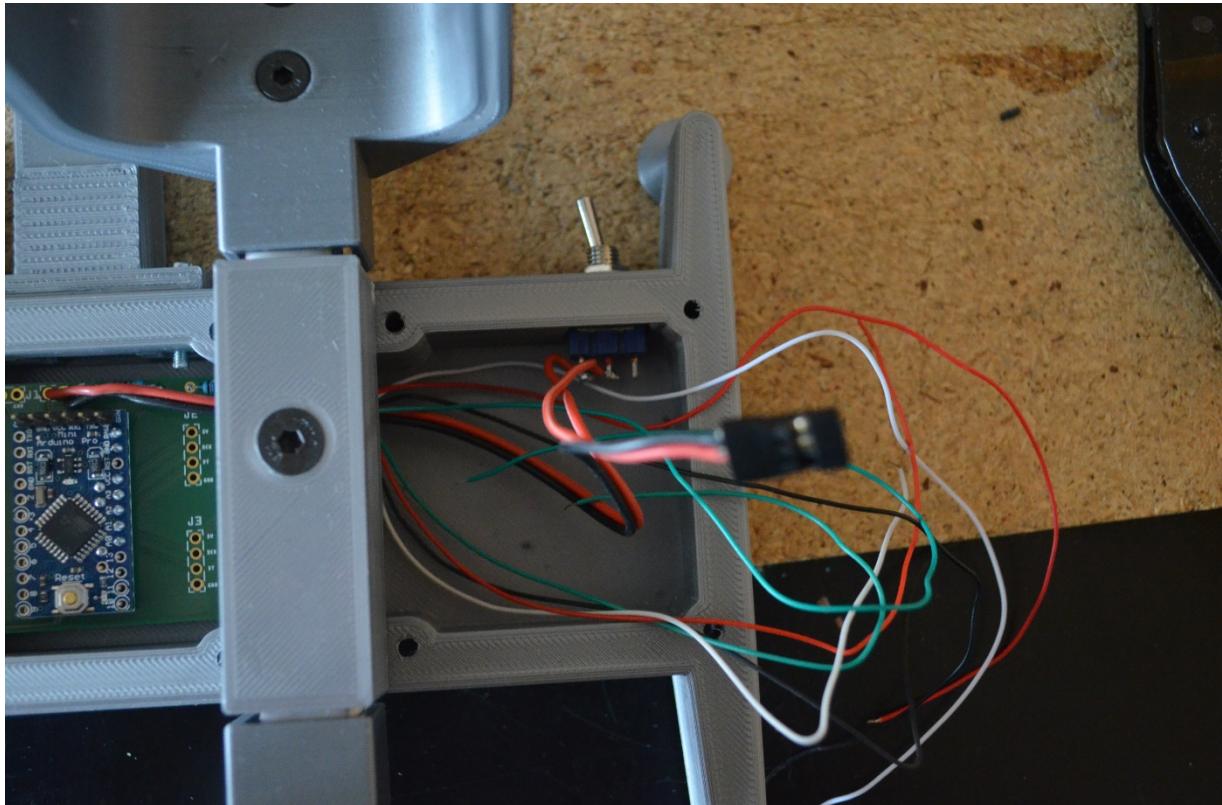
Souder le fil d'alimentation en provenance du PCB sur l'une des pattes de l'interrupteur.



*Figure 21 : Etape 7 – Mise en place de l'interrupteur à bascule.*

Souder un second fil d'alimentation sur la patte adjacente de l'interrupteur. Avec ce fil d'alimentation et le fil de masse en provenance du PCB, installer un connecteur pour connecter la batterie LiPo.

Le type du connecteur est à votre discrétion. La mise en place de ce connecteur peut nécessiter l'utilisation d'une pince à sertir (8€ chez Hobby King).



**Figure 22 :** Etape 7 – L'interrupteur à bascule et la mise en place d'un connecteur pour la batterie LiPo.

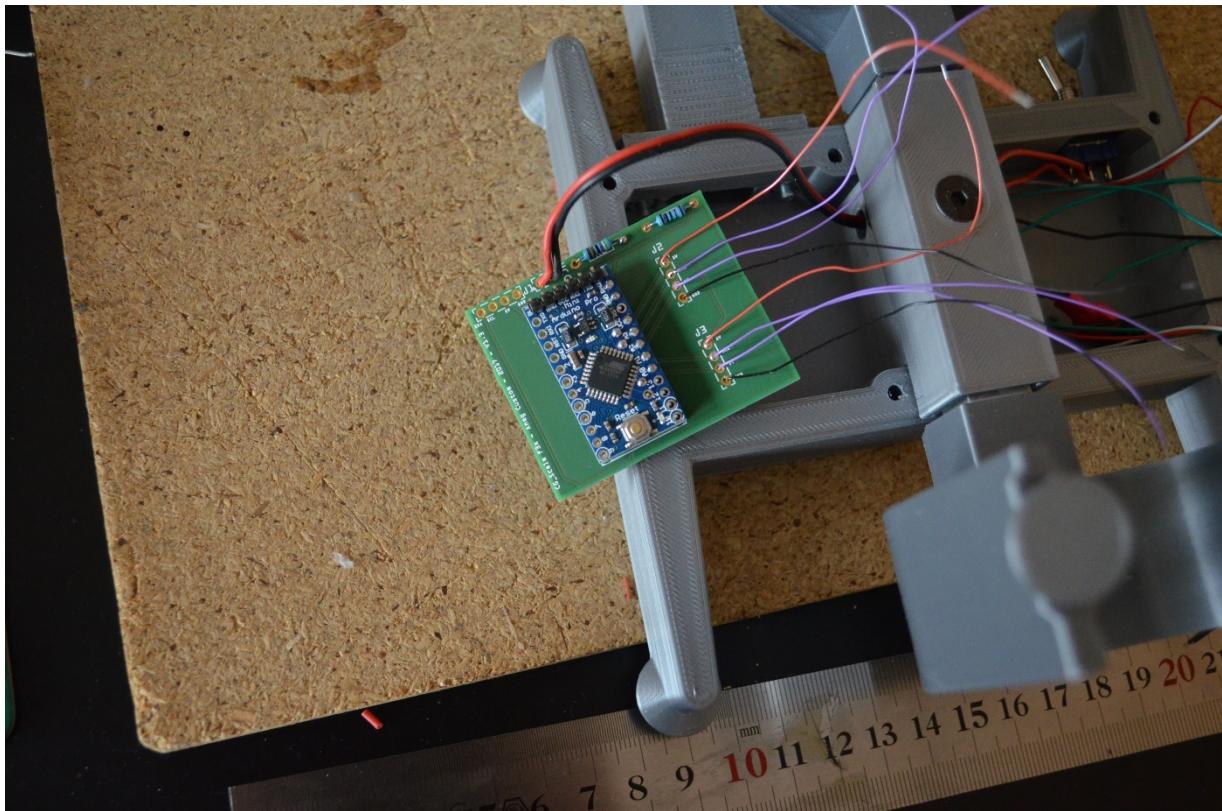
#### 2.4.1     **Etape 8 : Câblage des jauge de contrainte.**

Préparer 8 brins de Kynar ou équivalent (jauge 30 AWG). L'intérêt du Kynar est qu'il est mono filament et relativement peu cassant. Vous pouvez également utiliser du fil de servo.



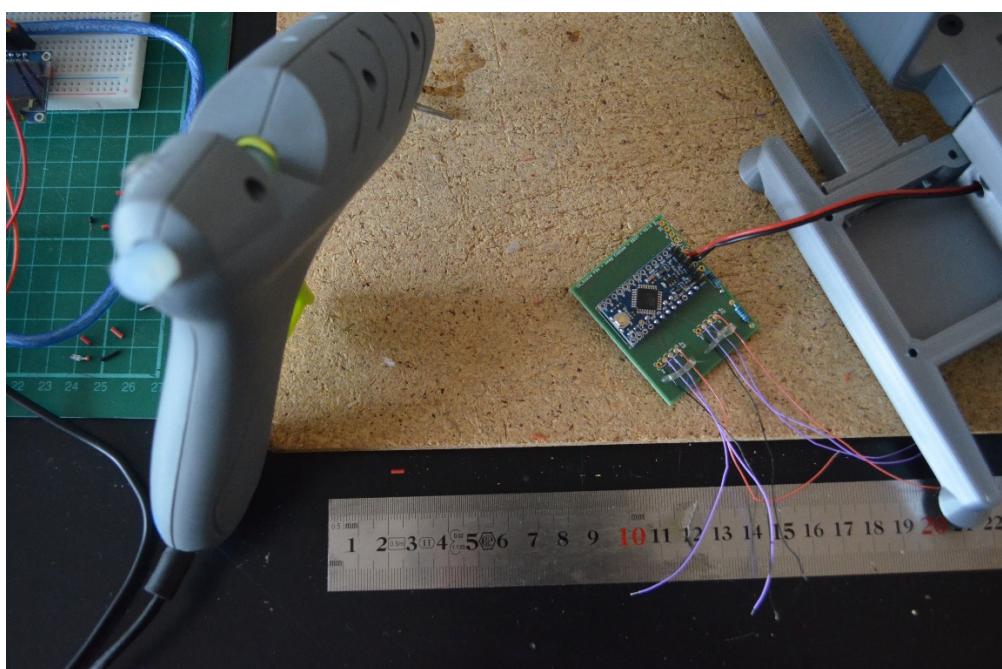
**Figure 23 :** Etape 8 – Préparer 8 brins de Kynar d'environ 12 cm.

Souder ces brins de Kynar sur le PCB comme présenté sur la photo ci-dessous.



**Figure 24 :** Etape 8 – souder les 8 brins de Kynar sur le PCB.

Puis maintenir en place avec un peu de colle à chaud pour éviter qu'une des connexions ne casse durant le montage.

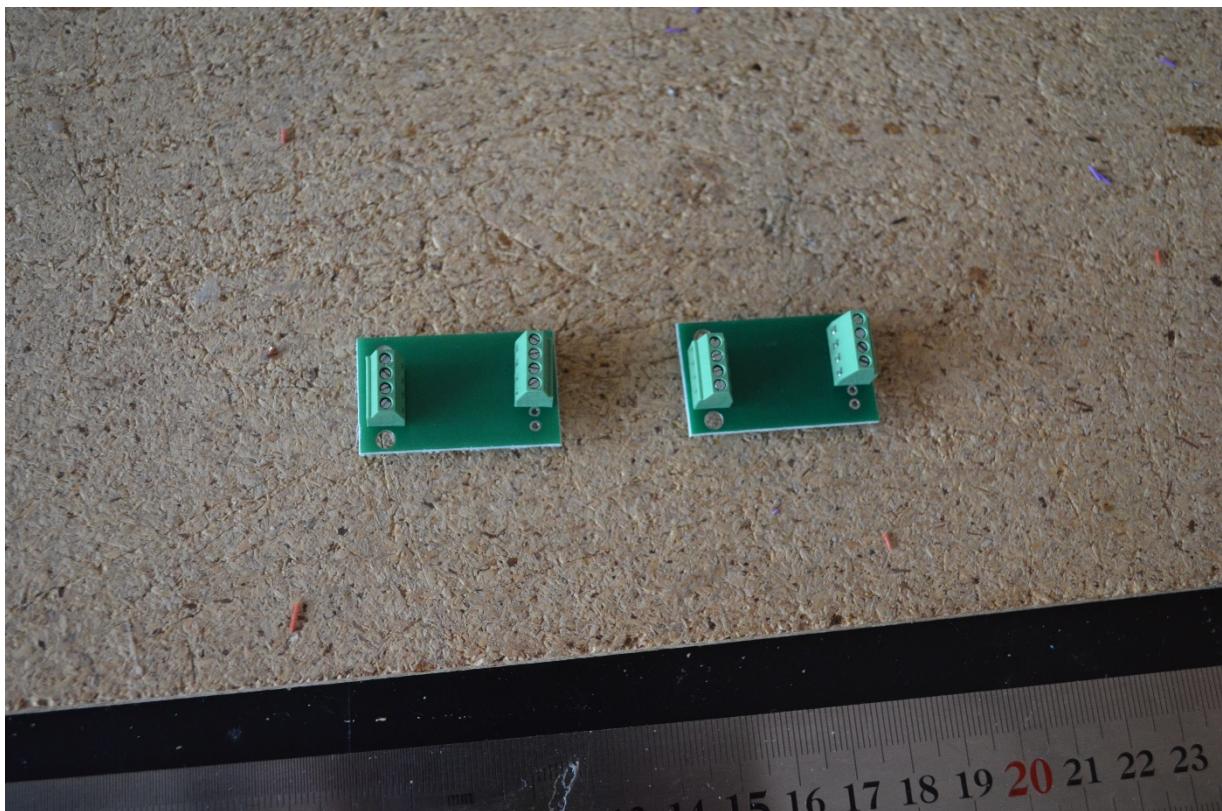


**Figure 25 :** Etape 8 – maintenir les 8 brins de Kynar avec de la colle à chaud.

### 2.4.1      Etape 9 : Préparer les deux cartes HX711

Souder les borniers à vis sur les deux cartes HX711.

Sur chaque carte, un bornier doit être soudé sur les pins GND, DT, SCK, VCC et l'autre bornier doit être soudé sur les pins E+, E-, A- et A+.



**Figure 26 : Etape 9 – Préparer les deux cartes HX711.**

### 2.4.1      Etape 10 : Intégrer les deux HX711 et l'alimentation.

La carte HX711 de la jauge de contrainte avant doit être connectée sur les brins de Kynar en provenance de J2 en respectant :

- Fil noir sur GND,
- Fil rouge sur VCC,
- A3 sur SCK,
- A2 sur DT.

La carte HX711 de la jauge de contrainte arrière doit être connectée sur les brins de Kynar en provenance de J3 en respectant :

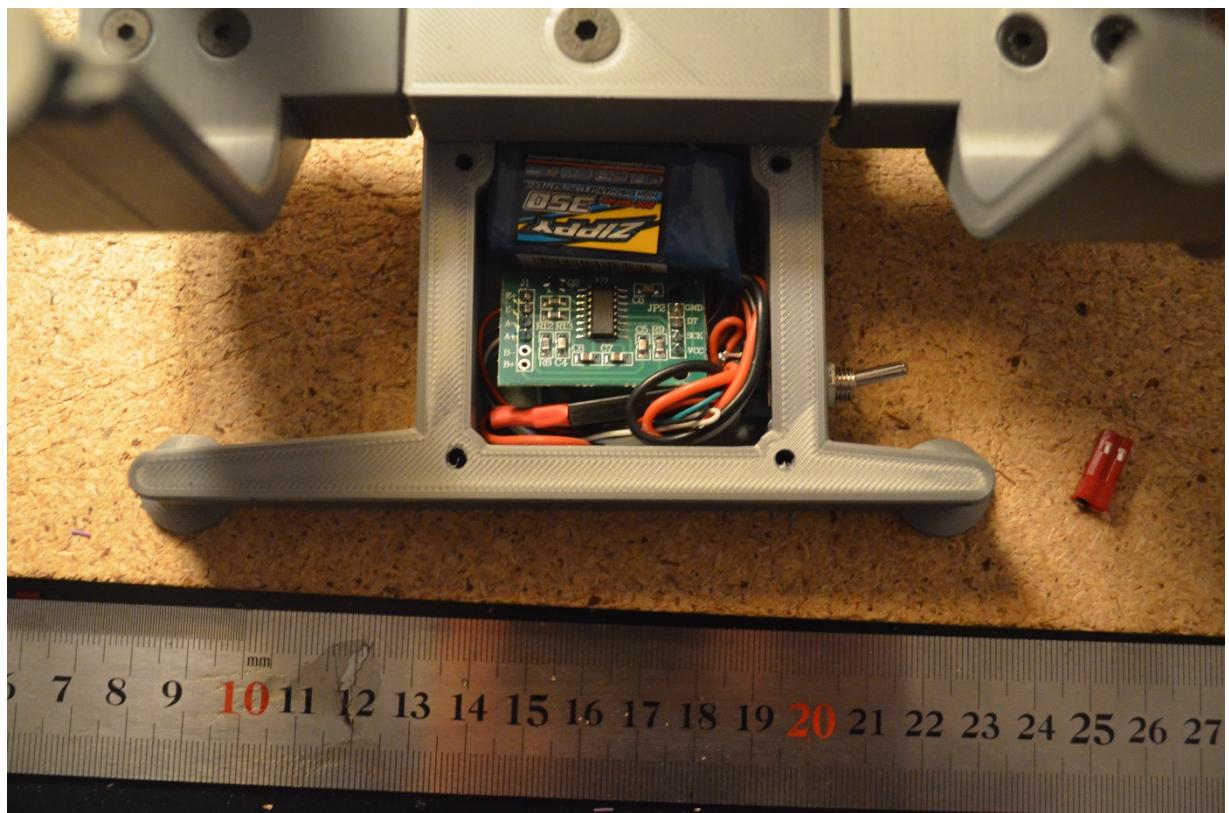
- Fil noir sur GND,
- Fil rouge sur VCC,
- A1 sur SCK,
- A0 sur DT.

**Attention à bien respecter GND et VCC. Les cartes HX711 ne sont pas protégées contre les inversions de polarité.**

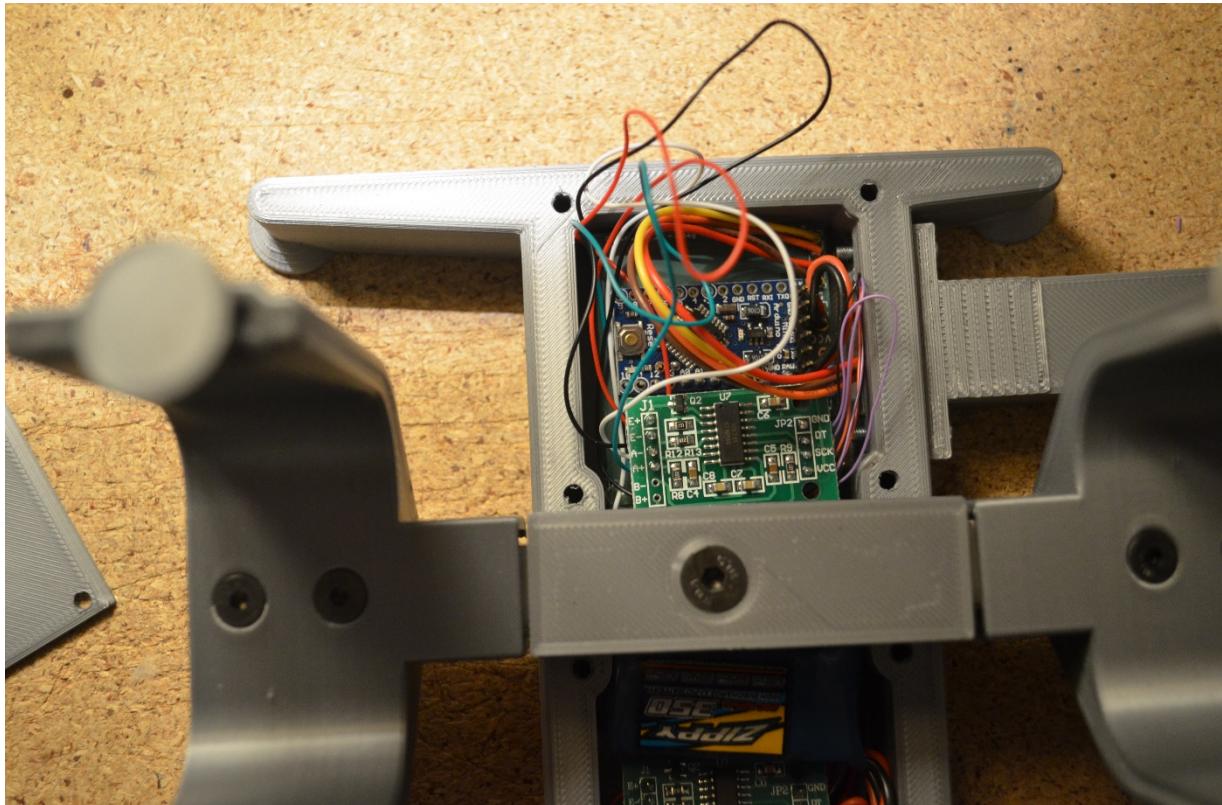
Les jauge de contrainte sont généralement équipées de 4 fils de couleurs noir, rouge, blanc et vert. Sur chaque carte connecter les fils en provenance des jauge respectives (jauge avant sur la HX711 câblées sur J2 et jauge arrière sur la HX711 câblées sur J3) en respectant :

- Fil noir sur E+,
- Fil rouge sur E-,
- Fil blanc sur A-,
- Fil vert sur A+.

Les deux photos suivantes présentent l'intégration des deux jauge dans les compartiments de la base de la balance. La droite et la gauche s'entendent en regardant la balance en positionnant le support arrière de la balance face à soi.



**Figure 27 : Etape 10 – Intégration HX711 et alimentation dans le compartiment de droite.**

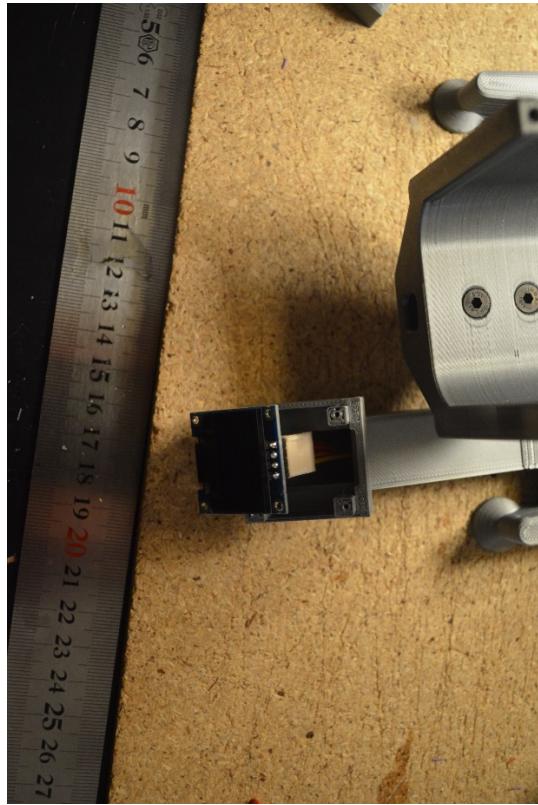


**Figure 28 : Etape 10 – Intégration HX711 et Arduino dans le compartiment de gauche.**

#### 2.4.1      **Etape 11 : Intégrer l'écran OLED 0,96”**

Souder le connecteur 4 fils sur le PCB et relier le connecteur à l'écran OLED.

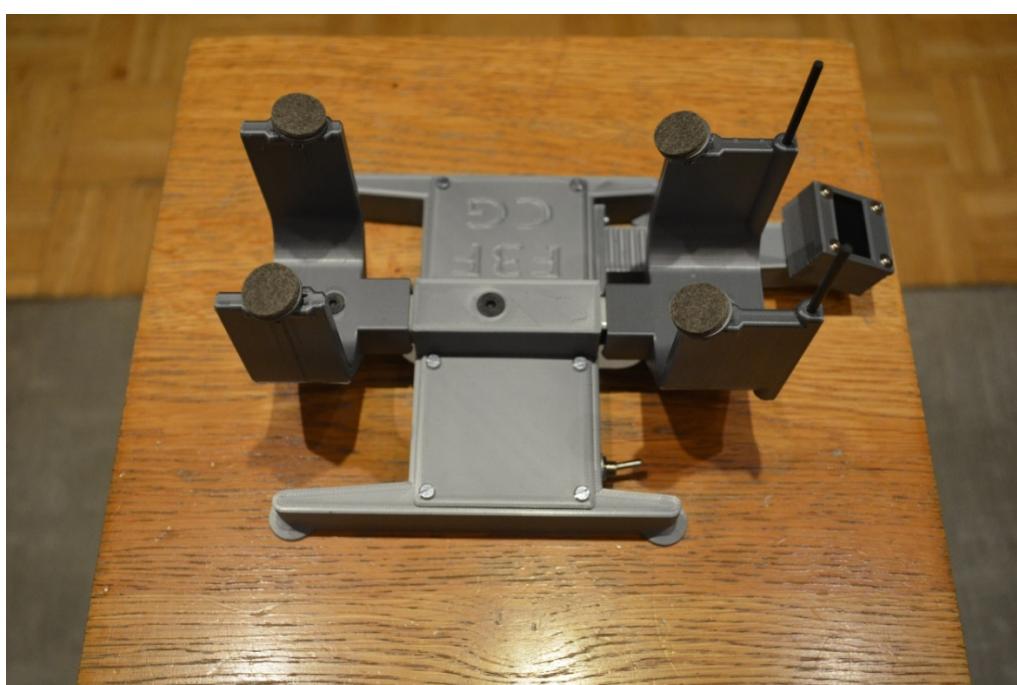
**Attention à bien respecter GND et VCC. L'écran OLED n'est pas protégé contre les inversions de polarité.**



**Figure 29 : Etape 11 – Intégration de l'écran OLED 0,96"**

Fermer le boitier écran à l'aide de 4 vis 1 mm à tête arrondie.

Et voilà la balance et totalement montée. Nous pouvons passer à l'installation de l'environnement Arduino, au téléchargement du logiciel et au calibrage de la balance.

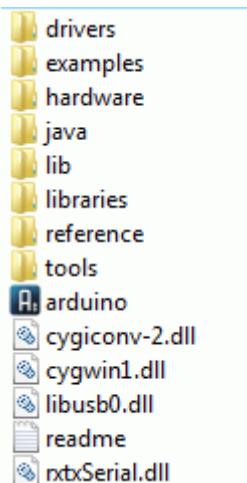


**Figure 30 : Etape 11 – La balance totalement montée.**

### 3 Prise en main de l'environnement Arduino.

Téléchargez la dernière version du logiciel Arduino sur cette page [https://www.arduino.cc/download\\_handler.php](https://www.arduino.cc/download_handler.php)

Le téléchargement est au format compressé zip. Quand le téléchargement est terminé, dézippez le fichier téléchargé : vous obtenez ainsi un répertoire Arduino-00XX **dont il faut garder la structure**. Ouvrez ce dossier : vous devriez voir quelques fichiers et sous répertoires.



Repérer le fichier avec l'icône Arduino qui correspond au logiciel Arduino. A noter que ce répertoire est "portable" et peut-être mis où vous voulez sur votre ordinateur, voire même sur une clé USB ou un disque dur externe. Le logiciel Arduino s'exécutera sans problème.

#### 3.1 Connecter la carte Arduino à l'ordinateur.

A présent, connectez votre carte Arduino à votre ordinateur en utilisant votre programmeur USB. La LED (verte ou rouge en fonction de la carte Arduino) d'alimentation (notée PWR) doit s'allumer.

#### 3.2 Installer les drivers USB du port Série virtuel.

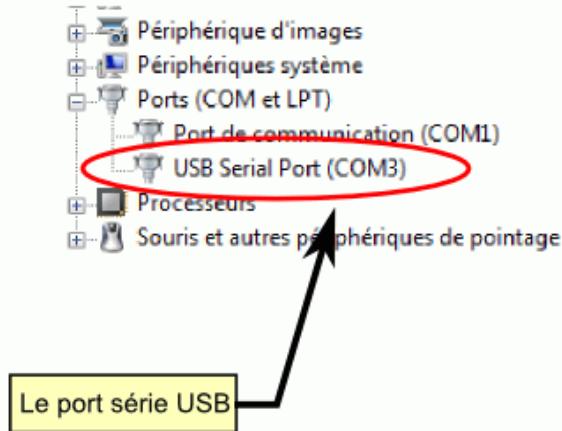
Quand vous connectez la carte à l'ordinateur pour la première fois, Windows devrait démarrer le processus d'installation du driver.

Sous Windows Vista et supérieur, le driver devrait normalement être installé automatiquement. Si ce n'est pas le cas :

- Aller dans Démarrer > Panneau de Configuration > Système > Gestionnaire de périphérique
- Aller dans Port Com et LPT > Repérer le port USB série > Clic droit > Mettre à jour le pilote
- Sélectionner alors le pilote dans le sous-répertoire /drivers/FTDI\_USB du répertoire Arduino-00xx précédemment téléchargé.
- Valider les différentes étapes.

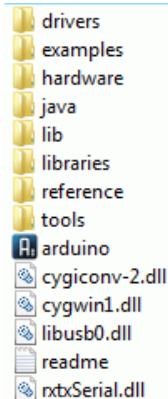
Vous pouvez à présent vérifier que les drivers ont bien été installés en ouvrant le Panneau de Configuration > Système > Gestionnaire de Périphériques. Vous devriez retrouver dans la section

Ports LPT et COM un "USB Serial Port" : c'est le port USB de la carte Arduino. Noter au passage le numéro du port.

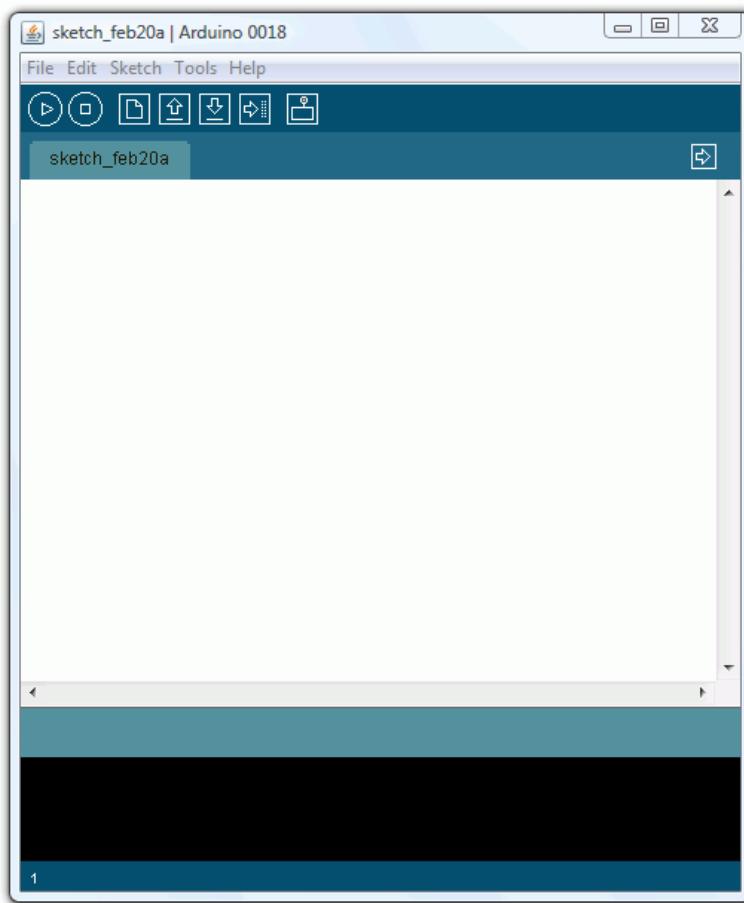


### 3.3 Lancer le logiciel Arduino :

A présent, lancez le logiciel Arduino en double-cliquant 2 fois sur l'icône Arduino dans le répertoire téléchargé précédemment :

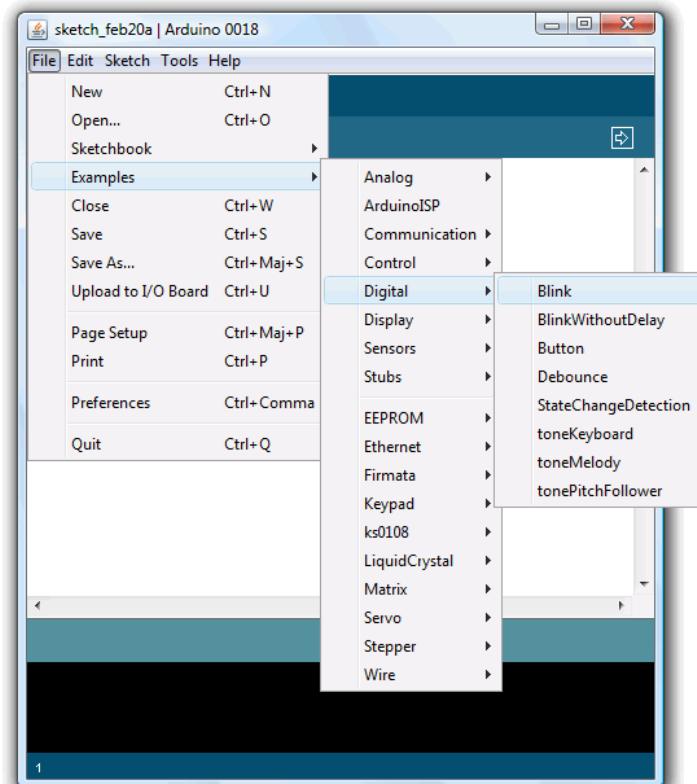


La fenêtre du logiciel doit s'ouvrir :



### 3.4 Ouvrir le programme d'exemple Blink

Ouvrez à présent le programme d'exemple "Blink" qui fait clignoter la LED de la carte connectée à la broche 13. Pour se faire, aller dans le menu File > Examples > Digital > Blink



Le code du programme doit à présent apparaître dans la fenêtre de l'éditeur. Noter que vous auriez pu écrire vous-même un programme de test mais pour commencer, ce n'est pas le plus simple...

```

Blink | Arduino 0018
File Edit Sketch Tools Help
[Icons]
Sketch
Blink
/*
int ledPin = 13;      // LED connected to digital pin 13

// The setup() method runs once, when the sketch starts

void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}

// the loop() method runs over and over again,
// as long as the Arduino has power

void loop()
{
  digitalWrite(ledPin, HIGH);    // set the LED on
  delay(1000);                 // wait for a second
  digitalWrite(ledPin, LOW);     // set the LED off
  delay(1000);                 // wait for a second
}

```

The screenshot shows the Arduino IDE with the title bar "Blink | Arduino 0018". The code editor window is open, displaying the "Blink" sketch. The code is as follows:

```

int ledPin = 13;      // LED connected to digital pin 13

// The setup() method runs once, when the sketch starts

void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}

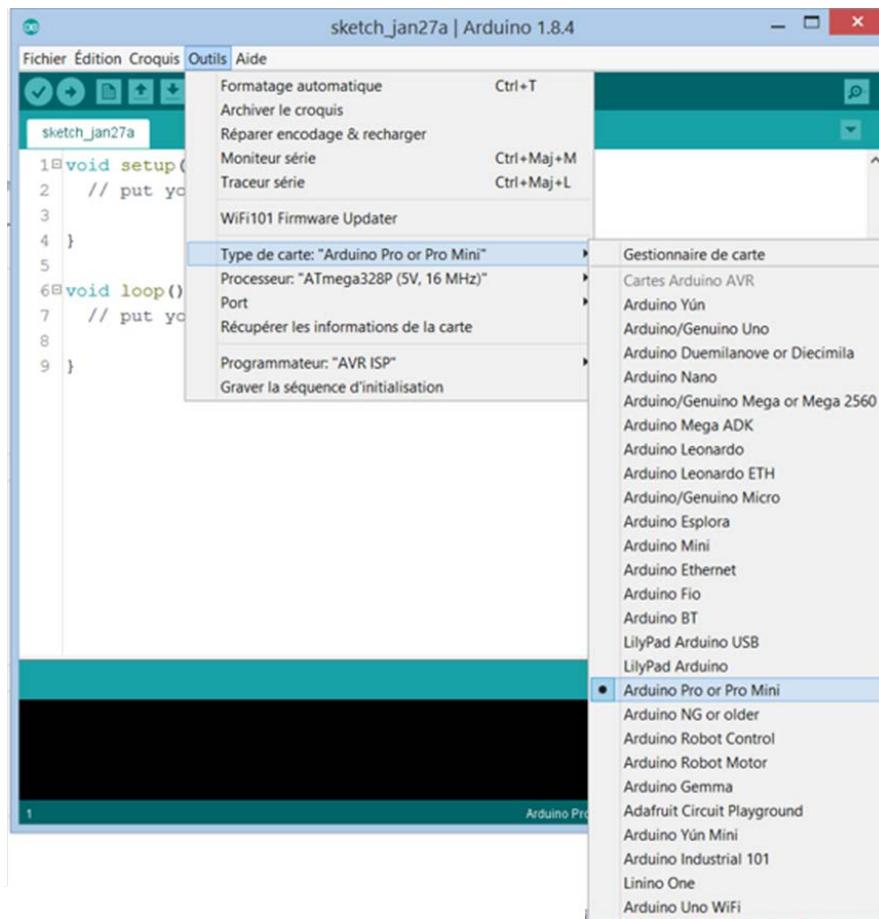
// the loop() method runs over and over again,
// as long as the Arduino has power

void loop()
{
  digitalWrite(ledPin, HIGH);    // set the LED on
  delay(1000);                 // wait for a second
  digitalWrite(ledPin, LOW);     // set the LED off
  delay(1000);                 // wait for a second
}

```

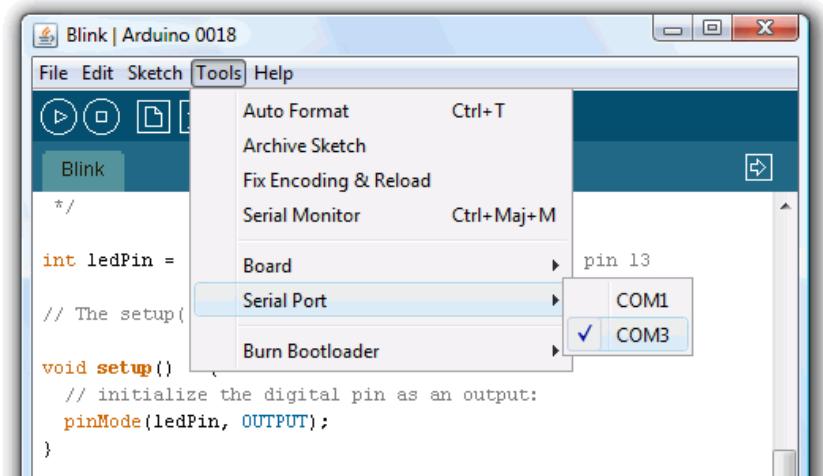
### 3.5 Sélectionner votre carte Arduino

Vous devez à présent sélectionner votre carte Arduino dans le menu Tools>Board : pour les nouvelle cartes avec un ATmega328, sélectionner Arduino Pro or Pro Mini.



### 3.6.8. Sélectionner votre port série.

A présent, vous devez sélectionner le port série utilisée pour la communication avec la carte Arduino depuis le menu Tools>Serial Port (Outils>Port Série). Ce sera probablement le port COM 3 ou supérieur (les ports COM1 et COM2 sont habituellement réservés pour les ports série matériel). Pour trouver de quel port il s'agit, vous pouvez déconnecter votre carte Arduino et réouvrir le menu Tools>Serial Port (Outils>Port Série) : l'entrée qui a disparue est probablement celle du port de la carte Arduino. Reconnecter la carte, réouvez le menu Tools>Serial Port (Outils>Port Série) et sélectionner le port série qui a du réapparaître :



### 3.79. Transférer le programme vers la carte Arduino

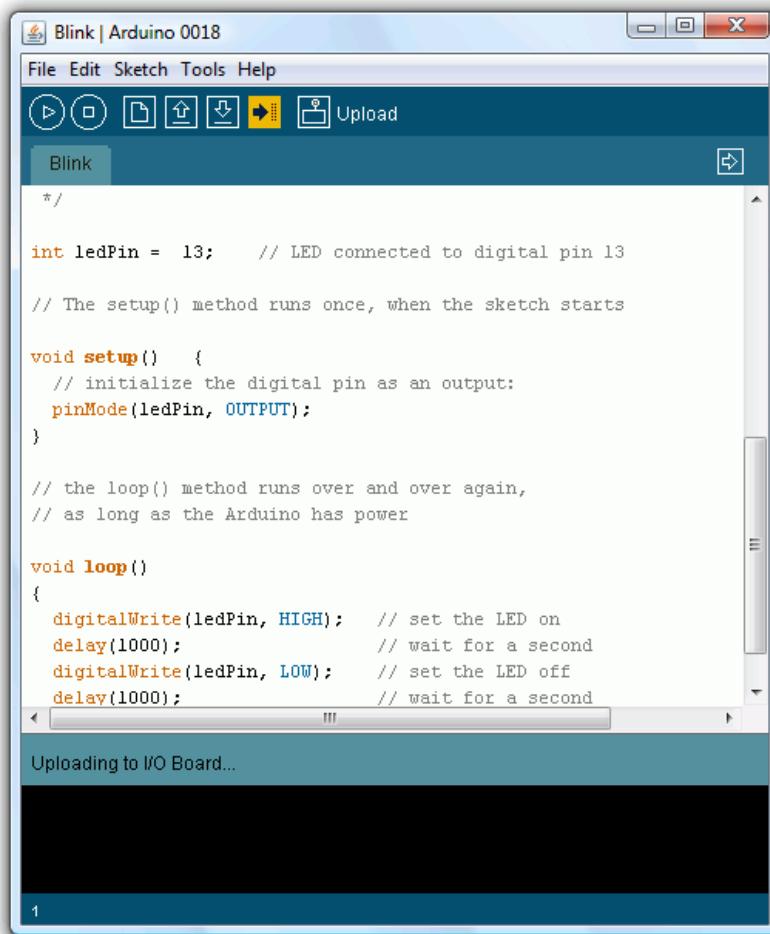
A présent, une fois que vous avez bien sélectionné le bon port série et la bonne carte Arduino, cliquez sur le bouton UPLOAD (Transfert vers la carte) dans la barre d'outils, ou bien sélectionner le menu File>Upload to I/O board (Fichier > Transférer vers la carte). Cliquer simplement sur le bouton "UPLOAD" de la barre d'outils du logiciel Arduino.

A noter que le transfert du programme démarre par une recompilation du croquis.

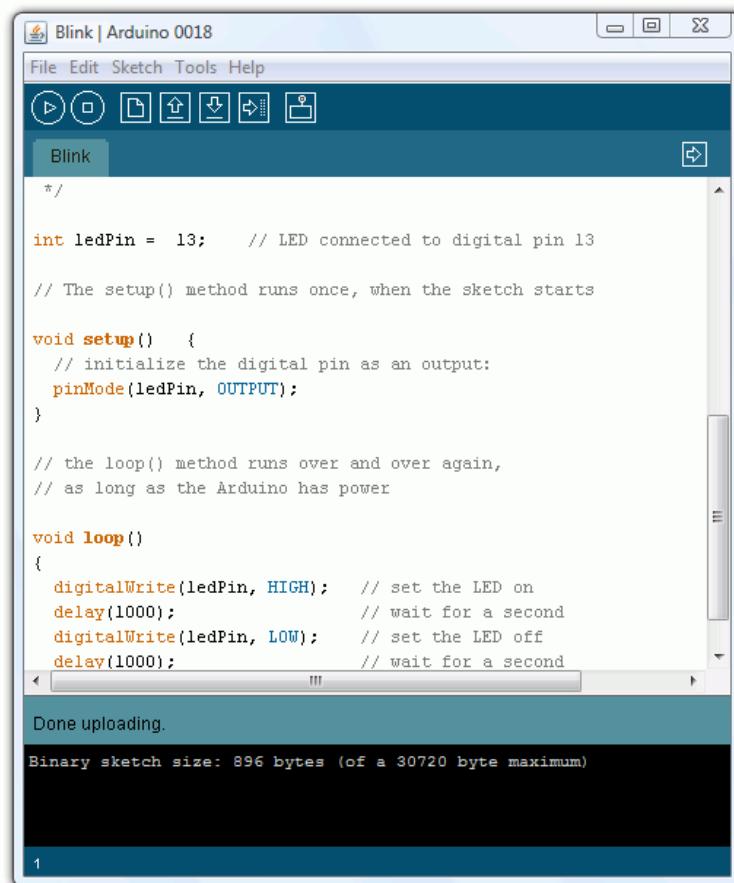


Vous devez appuyer sur le bouton "reset" de la carte juste avant de démarrer le transfert !  
A ce moment précis : l'INSTANT MAGIQUE !!!

Vous devez voir les LEDs des lignes RX et TX clignoter rapidement, témoignant que le programme est bien transféré. Durant le transfert, le bouton devient jaune et le logiciel Arduino affiche un message indiquant que le transfert est en cours :



Une fois le transfert terminé, le logiciel Arduino doit afficher un message ("Done uploading") indiquant que le transfert est bien réalisé, ou montrer des messages d'erreurs...



The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 0018". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with various icons. The main window displays the "Blink" sketch code. The code defines a variable `ledPin` set to 13, initializes it as an output pin in `setup()`, and sets it to HIGH and LOW alternately in `loop()`. The status bar at the bottom indicates "Done uploading." and "Binary sketch size: 896 bytes (of a 30720 byte maximum)".

```
/*
int ledPin = 13;      // LED connected to digital pin 13

// The setup() method runs once, when the sketch starts

void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}

// the loop() method runs over and over again,
// as long as the Arduino has power

void loop()
{
  digitalWrite(ledPin, HIGH);    // set the LED on
  delay(1000);                 // wait for a second
  digitalWrite(ledPin, LOW);    // set the LED off
  delay(1000);                 // wait for a second
}

Done uploading.

Binary sketch size: 896 bytes (of a 30720 byte maximum)
```

Une fois le transfert terminé, le bootloader est actif une petite seconde ("écoute" pour voir si un nouveau programme arrive...) une fois que la carte est réinitialisée à la fin du transfert; puis le dernier programme programmé dans la carte s'exécute. Vous devriez donc voir la LED sur la broche 13 clignoter (couleur orange).

Si c'est bien le cas, félicitations ! Vous avez réussi à programmer et à faire fonctionner votre carte Arduino.

## 4 Installer les bibliothèques supplémentaires

Deux librairies ou trois librairies (en fonction de la version que vous souhaitez construire) sont à installer pour pouvoir compiler le fichier .ino qui sera téléchargé dans la carte arduino :

- La librairie HX711\_ADC, qui permet de s'interfacer avec les ... HX711,

Si vous construisez la version OLED I2C :

- Adafruit\_SSD1306
- Adafruit\_GFX

Si vous construisez la version 1602 I2C :

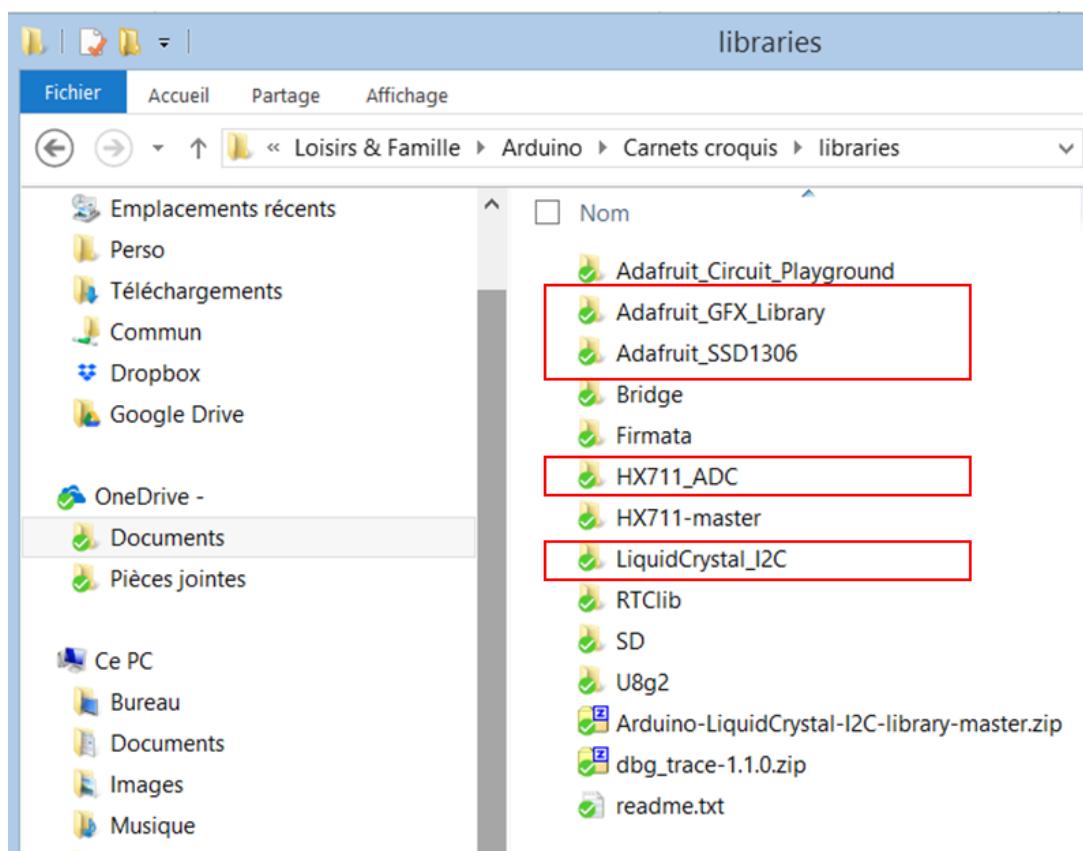
- LiquidCrystal\_I2C

Si vous construisez la version 1602 de base (installée par défaut dans l'environnement arduino).

- LiquidCrystal

Toutes ces bibliothèques s'installent à partir du Gestionnaire de bibliothèques dans le menu Croquis/Inclure une bibliothèque/Gérer les bibliothèques... Il permet de retrouver une liste détaillée des bibliothèques présentes, de les mettre à jour lorsqu'une évolution a été détectée chez son créateur, et d'en installer de nouvelles proposées automatiquement. Cette liste peut être triée par thème, ou par état (installées, à mettre à jour...).

Par défaut, ces librairies sont installées à l'emplacement de votre carnet de croquis.



## 5 Télécharger le logiciel et calibrer les ADC

A noter que pour toutes les opérations suivantes, l'interrupteur à bascule doit être en position « off » car c'est votre PC qui alimente la carte Arduino.

### 5.1 Initialiser les paramètres de calcul du CG.

En ligne 77 et 78 (cas du croquis CG\_scale\_I2c\_Oled) du croquis deux valeurs sont à modifier en fonction de votre balance :

```
74
75 //*** configuration:
76 //*** set dimensional calibration values:
77 const long WingPegDist = 1214; //calibration value in 1/10mm,
78 const long LEstopperDist = 306; //calibration value 1/10mm, I
```

WingPegDist correspond à la distance en 1/10 de mm projetée entre les deux axes des pads du support avant et du support arrière.

LEstopperDist correspond à la distance en 1/10 mm projetée entre l'axe du support avant et la tige d'arrêt des ailes du planeur.

Ces deux distances sont à mesurer au pied à coulisse.

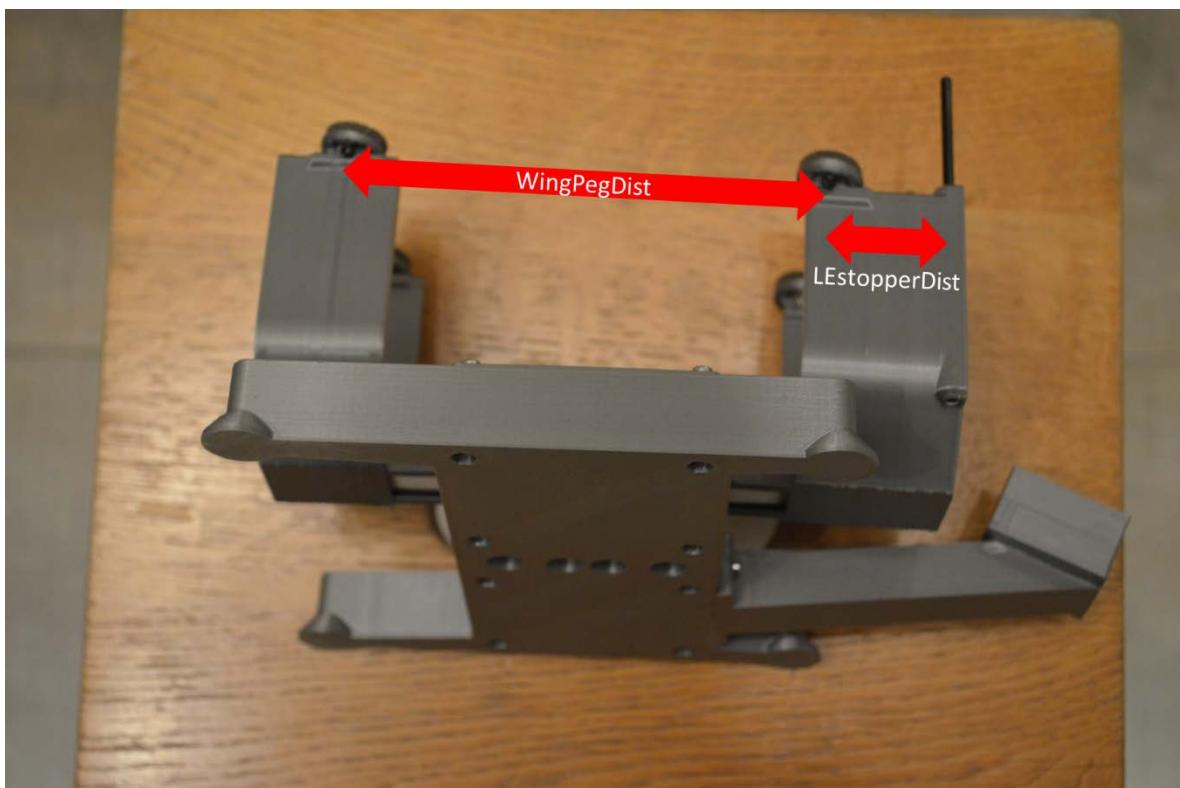
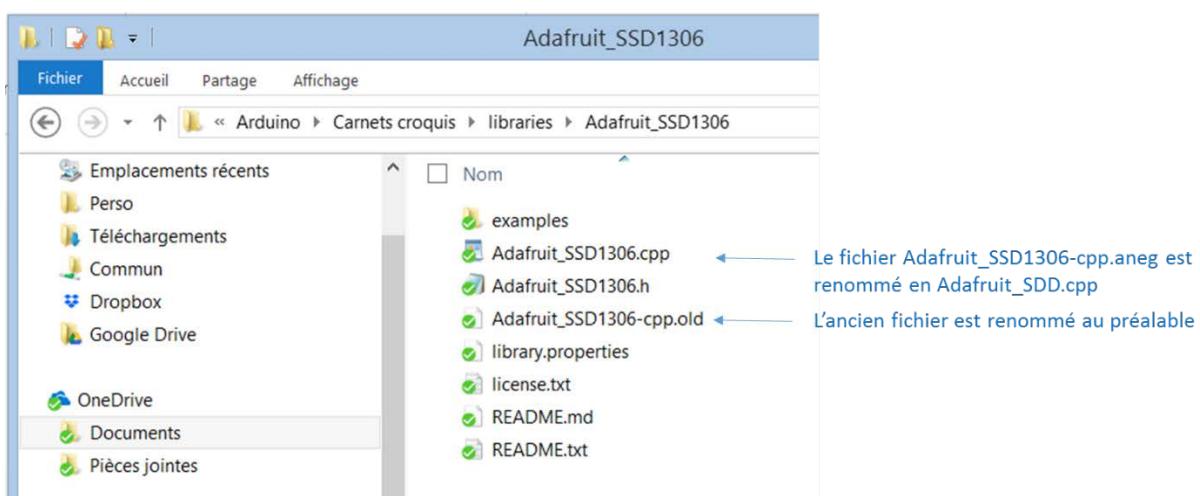


Figure 31 : Etape 11 – Définition de WingPegDist & LEstopperDist.

## 5.1 Télécharger le logiciel dans l’Arduino.

Une fois votre environnement Arduino configuré et les bibliothèques nécessaires installées, il ne vous reste plus qu'à télécharger le croquis CG\_scale\_I2c\_Oled (ou CG\_scale\_I2c\_1602 ou CG\_scale suivant votre version) dans la carte Arduino. Pour ce faire procéder comme pour le croquis Blink et attendez la fin de l'opération. Après le téléchargement, la carte va redémarrer et un écran d'accueil doit apparaître sur l'écran Oled 0,96".

**Par défaut, l'écran d'accueil de la balance est configuré avec le logo Adafruit. Si vous souhaitez avoir sur l'écran d'accueil le logo de l'ANEGR, il vaut faut recopier le fichier Adafruit\_SSD1306.cpp.aneg que vous trouverez dans la directoire « Arduino files » du Github, en le renommant Adafruit\_SSD1306.cpp au niveau de la librairie Adafruit\_SSD1306, en prenant soin de sauvegarder au préalable l'ancien fichier en Adafruit\_SSD1306.cpp.old. Vous pouvez également concevoir votre propre logo en suivant la procédure indiquée au chapitre 6.**



## 5.2 Calibrer les convertisseurs des HX711

Munissez-vous de deux poids en plombs ou de deux ballasts dont vous connaissez les poids en les ayant pesés sur une autre balance.

En ligne 94 du croquis (cas du croquis CG\_scale\_I2c\_Oled), commencer par positionner la variable output à 0 et téléverser le croquis dans la carte Arduino.

```
92 void setup() {
93     // ***
94     output = 1; //change to 1 for OLED, output = 0 for Serial terminal
95     // ***
96 }
```

Ouvrez le moniteur série dans le menu Outils. L'écran affiche périodiquement les poids mesurés par chaque HX711 via les jauge de contrainte (weight\_Ldcell\_1 et weight\_Ldcell\_2).

Nous allons décrire la procédure pour le HX711 relié au support avant. Poser un des vos poids de référence sur le support avant et lire la valeur mesurée par la balance.

Si la mesure affichée à l'écran du moniteur série est supérieure au poids de référence, en ligne 80 du croquis, modifier la valeur affectée à ldcell\_1\_calfactor en lui affectant une valeur inférieure (de -50 par exemple) et recharger le croquis et constater le résultat de votre modification. Répéter l'opération jusqu'à ce que la valeur lue sur le moniteur série soit proche de la valeur de votre poids de référence.

```
80 | const float ldcell_1_calfactor = 897.0;
81 | const float ldcell_2_calfactor = 745.0;
```

Recommencer l'opération en plaçant votre second poids de référence sur le support arrière en modifiant la valeur affectée à ldcell\_2\_calfactor.

Lorsque les deux HX711 sont calibrés, vous pouvez déconnecter votre programmeur de la carte Arduino, refermer les deux caches des deux bases à l'aide de 8 vis à tête ronde M3 x 20 mm et votre balance est prête à être utilisée.

## 6 Mode d'emploi de la balance

Le mode d'emploi est extrêmement simple :

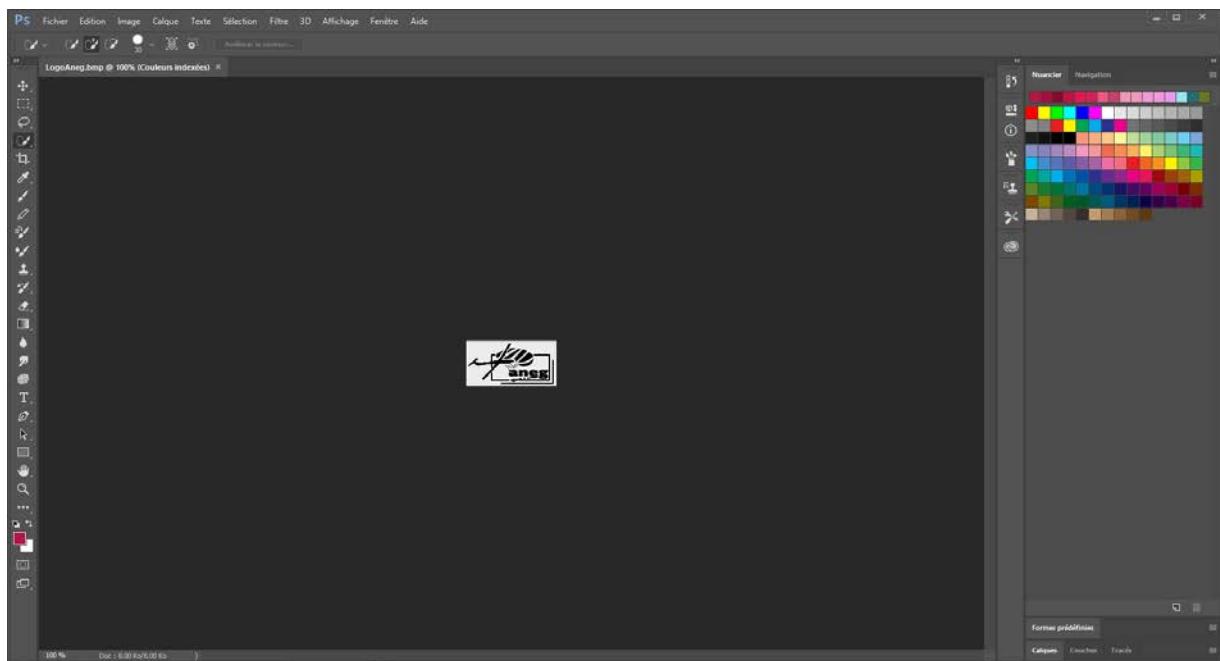
1. Allumer la balance et attendre l'affichage de « Batt : X.YY V » et « Wt : 0000 » sur l'écran.
2. Si « Wt : » n'est pas à zéro, éteindre la balance et repasser au 1°.
3. Positionner le planneur au centre des supports, les ailes en butée sur les tiges sur support avant.
4. Le poids total et la localisation du CG (distance par rapport au bord d'attaque) vont s'afficher.



**Figure 32 :** La balance en condition d'utilisation.

## 7 Modifier le logo qui apparaît sur l'écran à l'initialisation de la balance.

### 7.1 Etape 1 : créer votre image d'accueil



Utiliser l'éditeur graphique de votre choix pour créer une image 8 bits de 128 x 64 pixels.

**Vérifier que le fond de votre bitmap est blanc et que le premier plan (votre image) soit noir. Cette étape est importante car le logiciel utilisé pour convertir le bitmap en une matrice de code hexa ne fonctionne pas correctement si d'autres couleurs sont utilisées.**

Sauvegarder votre image obligatoirement au format bitmap 8 bits.

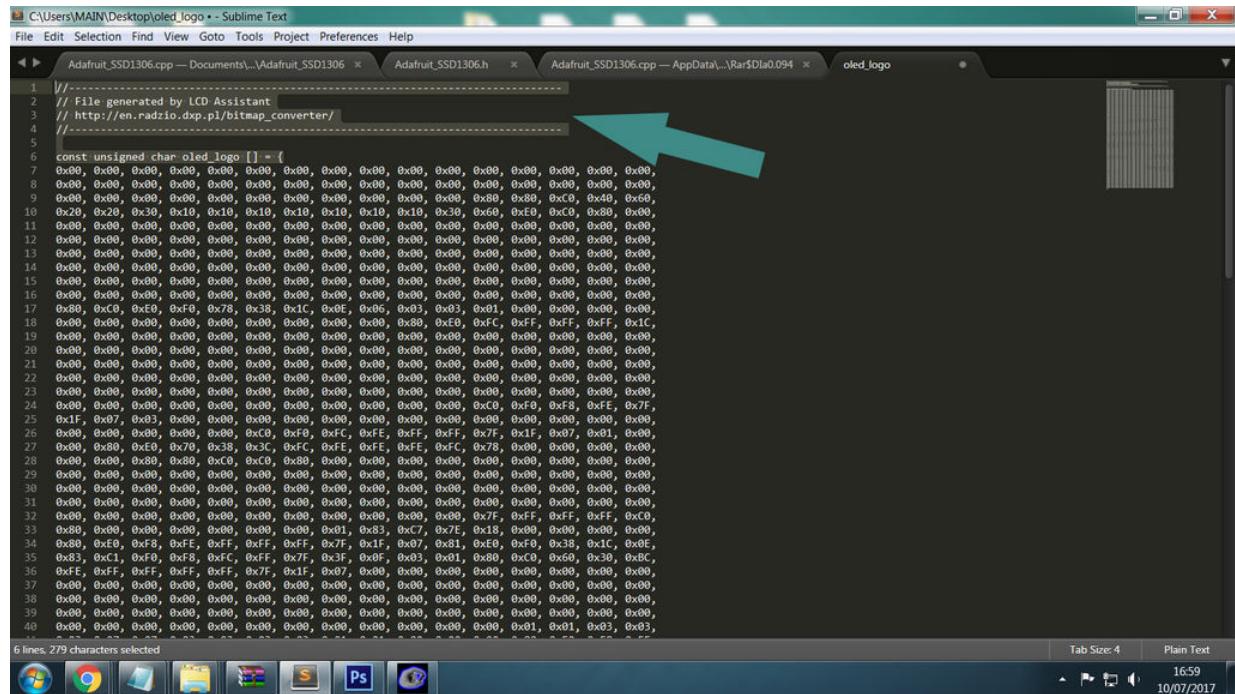
### 7.2 Step 2: Utiliser LCD Assistant pour convertir le bitmap.

Télécharger le logiciel gratuit LCD Assistant à l'adresse [http://en.radzio.dxp.pl/bitmap\\_converter/](http://en.radzio.dxp.pl/bitmap_converter/).

Ouvrir LCD Assistant et à partir du menu « fichier », charger l'image 8 bits précédemment sauvegardée à l'étape précédente. LCD Assistant va automatiquement détecter la taille de votre image (128 x 64) et laisser tous les autres paramètres à leurs valeurs par défaut.

Puis à partir du menu « fichier », procéder à la conversion en sélectionnant l'emplacement de sauvegarde et valider.

### 7.3 Etape 3 : Editer le fichier sauvegardé.



```

C:\Users\MAIN\Desktop\oled_logo • Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
Adafuit_SSD1306.cpp — Documents..., \Adafuit_SSD1306 ...
Adafuit_SSD1306.h ...
Adafuit_SSD1306.cpp — AppData...\Rar$Dta0.094 ...
oled_logo ...

1 ///////////////////////////////////////////////////////////////////////////////
2 // File generated by LCD Assistant
3 // http://en.radzio.dxp.pl/bitmap_converter/
4 //////////////////////////////////////////////////////////////////////////////

5
6 const unsigned char oled_logo [] = {
7     0x00, 0x00,
8     0x00, 0x00,
9     0x00, 0x00,
10    0x20, 0x20, 0x30, 0x10, 0x10, 0x10, 0x10, 0x10, 0x10, 0x10, 0x10, 0x00,
11    0x00, 0x00,
12    0x00, 0x00,
13    0x00, 0x00,
14    0x00, 0x00,
15    0x00, 0x00,
16    0x00, 0x00,
17    0x00, 0xC0, 0xF0, 0x78, 0x38, 0x1C, 0x0F, 0x05, 0x03, 0x01, 0x00, 0x00, 0x00,
18    0x00, 0x00,
19    0x00, 0x00,
20    0x00, 0x00,
21    0x00, 0x00,
22    0x00, 0x00,
23    0x00, 0x00,
24    0x00, 0x00,
25    0x1F, 0x07, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
26    0x00, 0x00,
27    0x00, 0x00, 0xE0, 0x70, 0x30, 0x30, 0x30, 0x30, 0x30, 0x30, 0x30, 0x30, 0x30,
28    0x00, 0x00,
29    0x00, 0x00,
30    0x00, 0x00,
31    0x00, 0x00,
32    0x00, 0x00,
33    0x88, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x83, 0x00, 0x00, 0x00, 0x00,
34    0x00, 0xE0, 0xF8, 0x00, 0x00,
35    0x00, 0x00,
36    0x00, 0x00,
37    0x00, 0x00,
38    0x00, 0x00,
39    0x00, 0x00,
40    0x00, 0x00
};

6 lines, 279 characters selected

```

Tab Size: 4 Plain Text

16:59 10/07/2017

Ouvrir le fichier généré par LCD Assistant à l'étape précédente dans n'importe quel éditeur de texte, comme par exemple Notepad..

Supprimer l'entête du fichier insérée par LCD Assistant et supprimer la ligne ***const unsigned char oled\_logo[] = {***

Supprimer les caractères ***} ; en fin de fichier.***

## 7.4 Etape 4: Découper le code de votre logo en trois parties.

```

Adafruit_SSD1306.cpp — Documents\...\Adafruit_SSD1306 — Adafruit_SSD1306.h — Adafruit_SSD1306.cpp —
1 0x00, 0x00,
2 0x00, 0x00,
3 0x00, 0x00,
4 0x20, 0x20, 0x30, 0x10, 0x10,
5 0x00, 0x00,
6 0x00, 0x00,
7 0x00, 0x00,
8 0x00, 0x00,
9 0x00, 0x00,
10 0x00, 0x00,
11 0x80, 0xC0, 0xE0, 0xF0, 0x78, 0x38, 0x1C, 0x0E, 0x06, 0x03, 0x03, 0x01, 0x00, 0x00, 0x00, 0x00,
12 0x00, 0xF0, 0xFF, 0x00, 0x00, 0x00, 0x00,
13
14 0x00, 0x00,
15 0x00, 0x00,
16 0x00, 0x00,
17 0x00, 0x00,
18 0x00, 0x00,
19 0x00, 0x00,
20 0x1F, 0x07, 0x03, 0x00, 0x00,
21 0x00, 0x00, 0x00, 0x00, 0x00, 0xC0, 0xF0, 0xFC, 0xFE, 0xFF, 0x7F, 0x1F, 0x07, 0x01, 0x00,
22 0x00, 0x80, 0xE0, 0x70, 0x38, 0x3C, 0xFC, 0xF0, 0xF0, 0x78, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
23 0x00, 0x00, 0x80, 0x80, 0xC0, 0xC0, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
24 0x00, 0x00,
25 0x00, 0x00,
26 0x00, 0x00,
27 0x00, 0x00,
28 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x83, 0xC7, 0x7E, 0x18, 0x00, 0x00, 0x00, 0x00,
29 0x80, 0xE0, 0xF8, 0xFE, 0xFF, 0x7F, 0x1F, 0x07, 0x81, 0xE0, 0xF0, 0x38, 0x1C, 0x0E,
30 0x83, 0xC1, 0xF0, 0xF8, 0xFC, 0xFF, 0x7F, 0x3F, 0x0F, 0x03, 0x01, 0x80, 0xC0, 0x60, 0x30, 0x8C,
31 0xFE, 0xFF, 0xFF, 0xFF, 0x1F, 0x07, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
32 0x00, 0x00,
33 0x00, 0x00,
34
35 0x00, 0x00,
36 0x00, 0x01, 0x01, 0x03, 0x03,
37 0x03, 0x07, 0x03, 0x03, 0x03, 0x03, 0x03, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
38 0xFF, 0xFF, 0x8F, 0x8F, 0x3C, 0xF1, 0x38, 0x1C, 0x03, 0x0F, 0x03, 0x01, 0x80, 0xE0, 0x00, 0x00,
39 0xFF, 0x3F, 0x0F, 0x03, 0x00, 0x00,
40 0xFF, 0x3F, 0x0F, 0x03, 0x00, 0x00,
```

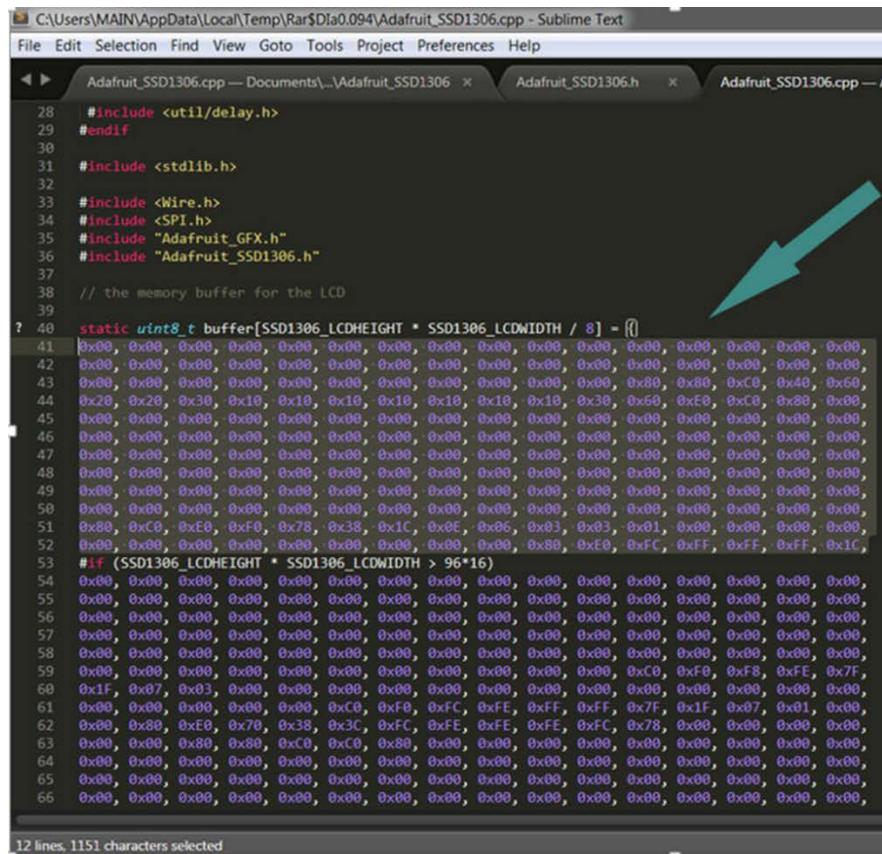
1<sup>er</sup> section  
contient 12  
lignes de code

2<sup>ième</sup> section  
contient 20  
lignes de code

3<sup>ième</sup> section  
contient 32  
lignes de code

Dans cette étape, nous allons découper le code du logo en trois sections en insérant un saut de ligne. La première section correspond au 12 premières lignes, la seconde section correspond aux 20 lignes suivantes et la troisième et dernière section correspond aux 32 lignes suivantes, ce qui fait un total de 64 lignes de code, conformément à la taille de l'écran (128 x 64).

## 7.5 Etape 5 : Copier le code de votre logo à la place du code du logo Adafruit installé par défaut dans la librairie Adafruit\_SSD\_1306.



```

C:\Users\MAIN\AppData\Local\Temp\Rar$Dla0.094\Adafruit_SSD1306.cpp - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
Adafruit_SSD1306.cpp — D Adafruit_SSD1306.h — A Adafruit_SSD1306.cpp — A
28 #include <util/delay.h>
29 #endif
30
31 #include <stdlib.h>
32
33 #include <Wire.h>
34 #include <SPI.h>
35 #include "Adafruit_GFX.h"
36 #include "Adafruit_SSD1306.h"
37
38 // the memory buffer for the LCD
39
40 static uint8_t buffer[SSD1306_LCDHEIGHT * SSD1306_LCDWIDTH / 8] = {
41 0x00, 0x00,
42 0x00, 0x00,
43 0x00, 0x00,
44 0x20, 0x20, 0x30, 0x10, 0x10,
45 0x00, 0x00,
46 0x00, 0x00,
47 0x00, 0x00,
48 0x00, 0x00,
49 0x00, 0x00,
50 0x00, 0x00,
51 0x80, 0xC0, 0xE0, 0xF0, 0x78, 0x38, 0x1C, 0x0E, 0x06, 0x03, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
52 0x00, 0x00,
53 #if (SSD1306_LCDHEIGHT * SSD1306_LCDWIDTH > 96*16)
54 0x00, 0x00,
55 0x00, 0x00,
56 0x00, 0x00,
57 0x00, 0x00,
58 0x00, 0x00,
59 0x00, 0x00,
60 0x1F, 0x07, 0x03, 0x00, 0x00,
61 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xC0, 0xF0, 0xFC, 0xFE, 0xFF, 0x7F, 0x1F, 0x07, 0x01, 0x00,
62 0x00, 0x00, 0x00, 0x70, 0x38, 0x3C, 0xFC, 0xFE, 0xFF, 0x7C, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
63 0x00, 0x00,
64 0x00, 0x00,
65 0x00, 0x00,
66 0x00, 0x00,
}

```

12 lines, 1151 characters selected

**Copier/Coller le code pour les trois sections, une section à la fois.**

Pour cette étape, vous devez déjà avoir téléchargé et installé la bibliothèque Adafruit\_SSD1306.

Dans la bibliothèque Adafruit\_SSD1306, copier le fichier Adafruit\_SSD1306.cpp, en Adafruit\_SSD1306.old.

**Important : ne pas renommer le fichier en <quelque-chose.cpp>, car à la compilation l'IDE arduino essaye de compiler tous les fichiers .cpp contenus dans une bibliothèque et vous auriez de multiples redéfinitions et donc une impossibilité de compiler.**

Ouvrez le fichier Adafruit\_SSD1306.cpp à l'aide d'un éditeur de texte comme Notepad. Faites défiler vers le bas pour trouver la section suivante commençant par cette ligne:

**static uint8\_t buffer[SSD1306\_LCDHEIGHT \* SSD1306\_LCDWIDTH / 8] = {**

Vous allez ensuite copier et coller en remplaçant le code par le nouveau code de votre logo dans chaque section:

**static uint8\_t buffer[SSD1306\_LCDHEIGHT \* SSD1306\_LCDWIDTH / 8] = {**

Le code de la première section est à coller ici.

```
#if (SSD1306_LCDHEIGHT * SSD1306_LCDWIDTH > 96*16)
```

Le code de la seconde section est à coller ici.

```
#if (SSD1306_LCDHEIGHT == 64)
```

Le code de la troisième et dernière section est à coller ici.

```
#endif #endif };
```

Laissez le reste du code totalement intact sous peine d'avoir une erreur de compilation. Une fois terminé, sauvegarder le fichier.

Recompiler le croquis CG\_scale\_I2C\_Oled et téléversez-le. En supposant que tout a été fait correctement votre logo va maintenant remplacer celui d'Adafruit Industries.



## 8 Dépannage

Symptôme	Résolution
Odeur de bakélite au branchement de la balance.	Vous avez probablement inversé le branchement du GND et du VCC, soit sur un des HX711, soit sur l'écran OLED. Remplacer le composant.
Impossible de détecter le programmeur TTL-USB par le PC.	Votre driver USB n'est probablement pas correctement installé. Le driver FTDI est livré avec l'IDE arduino et est localisé sur le chemin C:\<Votre chemin d'installation>\arduino-1.8.4\drivers\FTDI USB Drivers. Essayez de le réinstaller.
Votre carte arduino pro mini ne possède pas le pin A7 au bon endroit.	Wrapper le pin A7 de votre carte, vers le pin A7 du PCB avant de souder la carte arduino sur le PCB.