**Compte rendu final de T.I.P.E  :**

Capteur de puissance pour vélo



BESNARD Nicolas   
CHAUVET David

**Table des matières**

[**I) Cahier des charges**](#_e5dprncymtws) **2**

[**II) Choix des solutions**](#_6o36i2bliw0c) **3**

[**III) Fonctionnement**](#_g7g173drn8d2) **4**

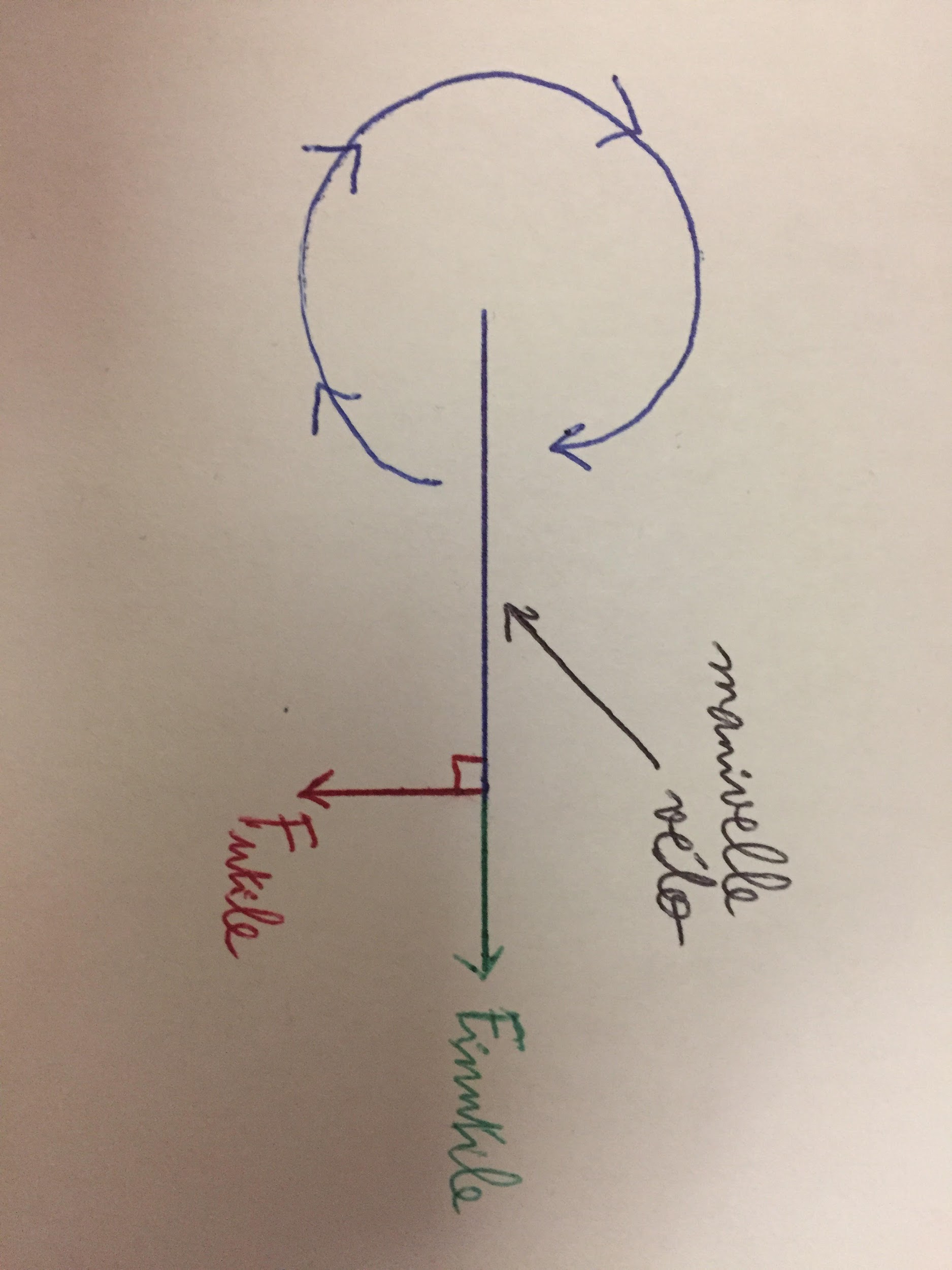
[**IV) Problèmes et solutions**](#_lehrsud23xai) **10**

[**V) Conclusion**](#_u7nlhw6k7wdq) **13**

[**VI) Bibliographie**](#_4j45zv4dcc80) **14**

# I) Cahier des charges

Le but du projet est de réaliser un dispositif d’analyse de puissance pour cycliste. L’objectif principal est de connaître la puissance transmise lors du pédalage en temps réel. Cela revient à collecter la cadence et le couple de pédalage pour ensuite calculer la puissance.



Pour répondre à tous les objectifs nous devons mesurer le vecteur force (intensité + direction) exercée par le cycliste ainsi que la cadence. Les informations doivent ensuite être traitée pour calculer la puissance.

Nous souhaitons également enregistrer toutes ces données pour pouvoir ensuite (après une session de vélo par exemple), avoir un historique de la puissance au fil du temps.

# II) Choix des solutions

Les mesures de ce vecteur force a été le sujet de nos recherches lors des premières séance de TIPE. Nous avons imaginé plusieurs solutions, plus ou moins proche de choses existantes.

Nous avons d’abord pensé à fabriquer une pédale entière. Cependant nous n’avons pas trouvé de solution viable suite au faible espace disponible dans une pédale et les différentes contraintes mécaniques imposées. Nous nous sommes finalement rabattus sur un collage des jauges de contrainte directement sur la manivelle du vélo qui subit des déformations lors du pédalage (une flexion). Pour ce qui concerne la cadence, nous avons choisi d’utiliser un gyroscope plutôt qu’un comptage au tour (par exemple avec un capteur magnétique). Cela nous permet une plus grande précision et une mise en place mécanique plus facile bien que l’interprétation soit légèrement plus complexe.

Pour la réalisation globale du projet nous avons donc deux “boîtiers” composé d’un arduino nano chacun :

* Le premier est situé sur la manivelle et a pour rôle de transmettre les données des capteurs tel que les capteurs de déformations et le gyroscope.
* Le second est situé sur le guidon et a pour rôle de recevoir les informations, les enregistrer et les afficher.

# 

# III) Fonctionnement

Le fonctionnement de notre projet est décomposé en deux parties distinctes qui communiquent entre elles : le “boitier manivelle” et le “boitier guidon”.

1. **Boîtier manivelle**

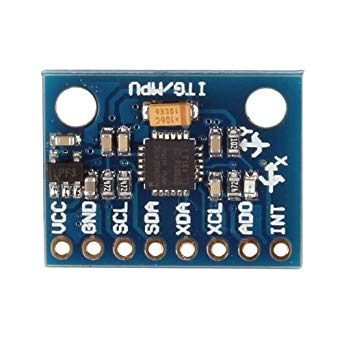
Le boitier manivelle à pour objectif de traiter les données des différents capteurs afin de calculer la puissance et la cadence de pédalage pour ensuite les transmettre au boîtier guidon.



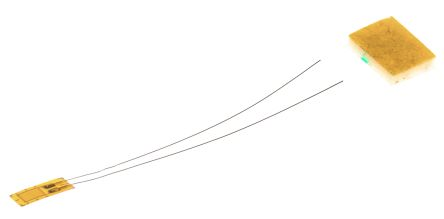


*Schéma fonctionnel du boîtier manivelle*

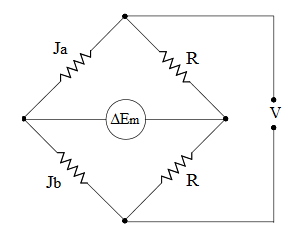
Pour réaliser ces différentes fonctions nous avons utilisé les composant suivant :

* **Capteurs :** 
  + **Gyroscope MPU-6050 :**

Nous avons choisi d’utiliser ce gyroscope pour mesurer la vitesse de rotation de la manivelle. Il possède une précision d’environ ± 3° et nous l’avons configuré pour mesurer une vitesse jusqu'à 1000°/s. Ainsi nous pouvons calculer la cadence de pédalage, soit le nombre de tours de pédale par minute.

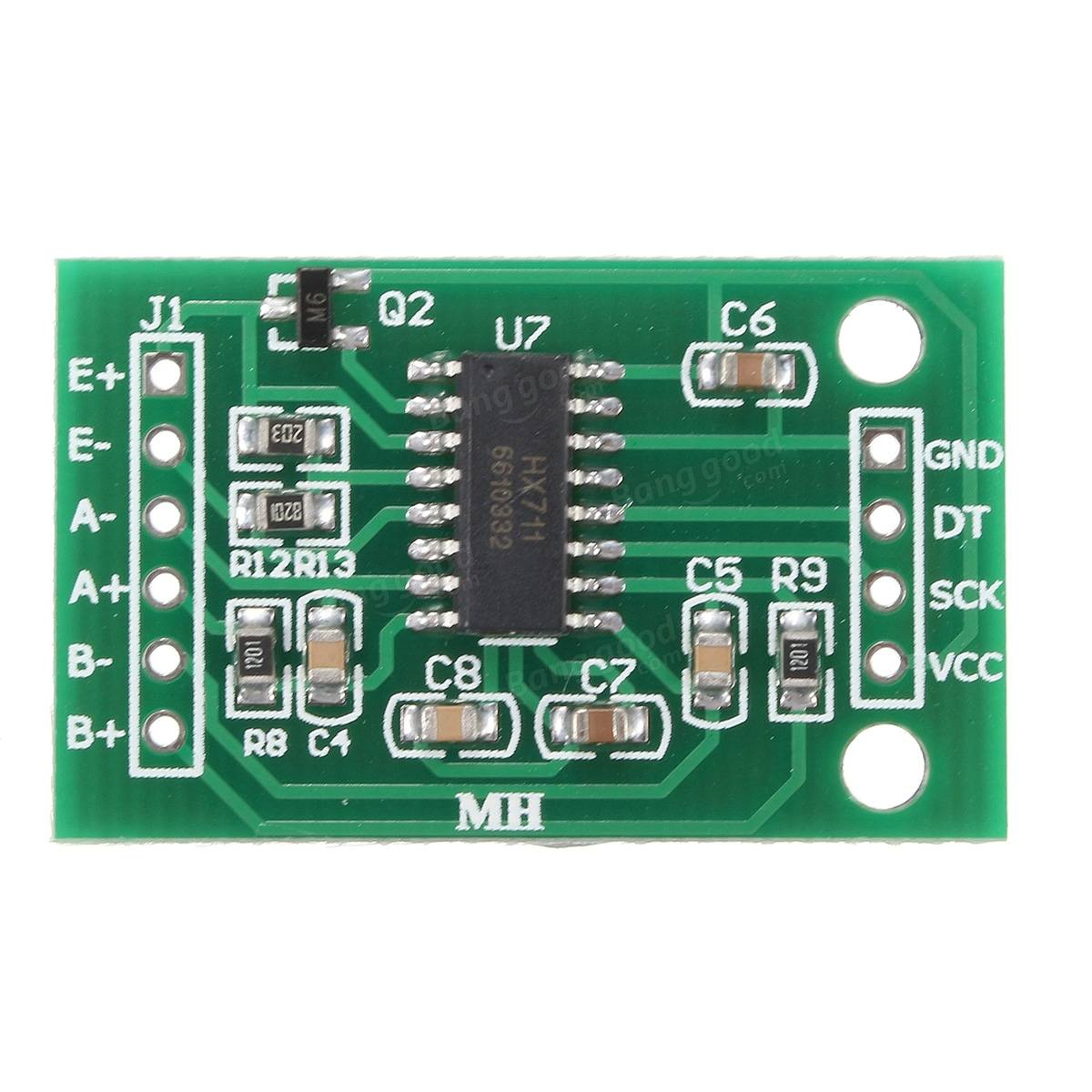


* **Jauge de déformation :**

Nous avons choisi d’utiliser des jauges de déformation afin de mesurer la force appliquée. Les jauges de déformations sont des résistances piézoélectriques donc leurs résistances varient en fonction de la déformation. Cependant cette variation est infime et nous devons convertir cette variation de résistance en une variation de tension à l’aide d’un pont de Wheatstone. 

Les jauges sont placées dans un montage en demi-pont de Wheatstone afin de mesurer uniquement la flexion de la manivelle. Pour compléter ce demi-pont nous utilisons deux résistances fixes.

* **Traitements :**
* **Amplificateur HX711**



L’amplificateur HX711 permet d’amplifier la tension au centre du pont de Wheatstone (ΔEm sur le schéma précédent). En effet, cette tension est très petite et il est impossible de l’utiliser directement. Ce module permet aussi de calibrer (scale et offset) les jauges. De plus, il convertis en numérique les données avant de les transmettre à l’arduino.

* **Arduino Nano**



Nous avons fait le choix d’utiliser un Arduino Nano comme microcontrôleur pour des raisons de taille et de facilité de programmation.

Sa petite taille nous a permis de le souder sur un PCB avec tous les autres composants pour pouvoir diminuer la taille du dispositif au maximum.

* **Transmission :**

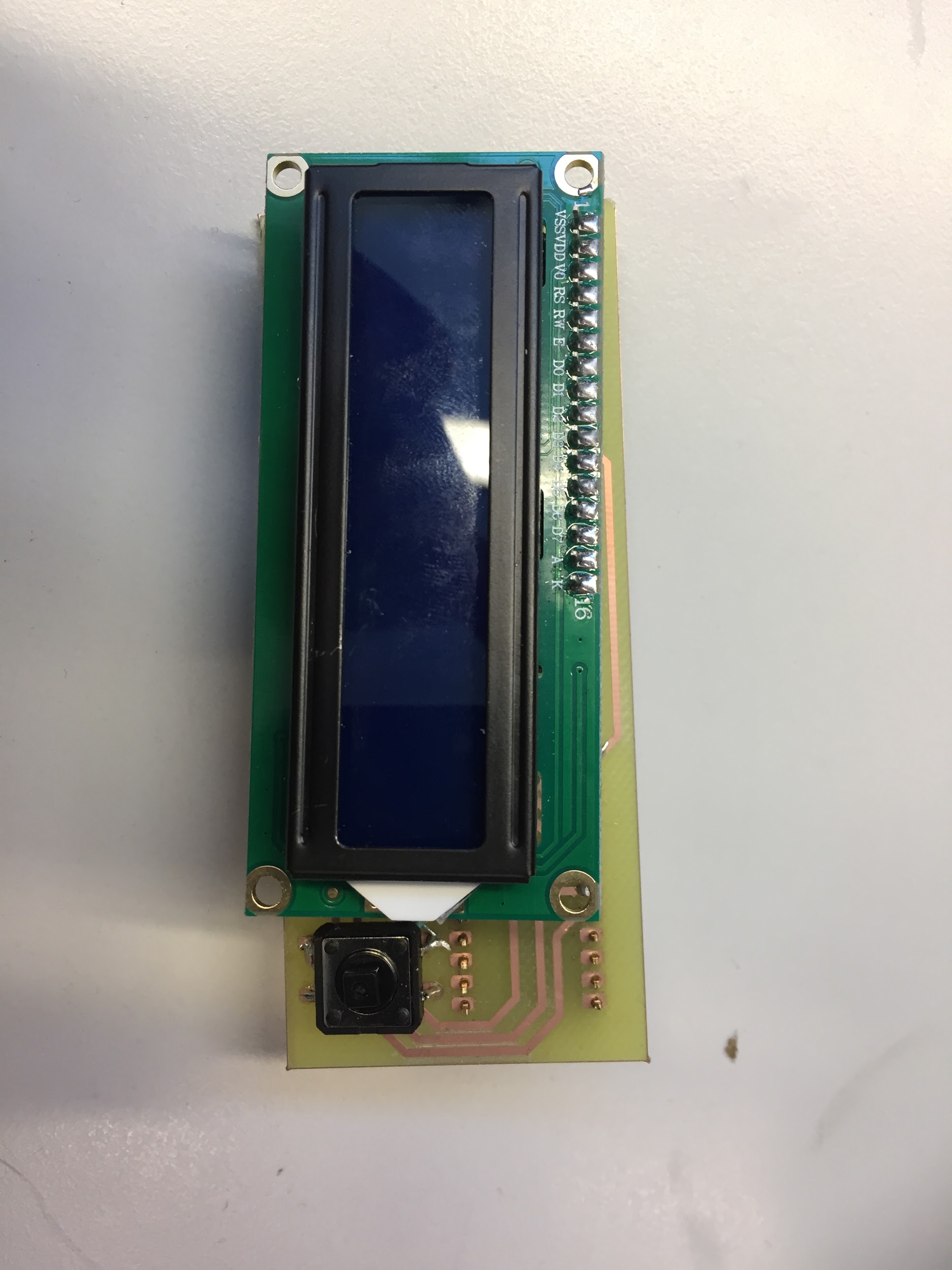
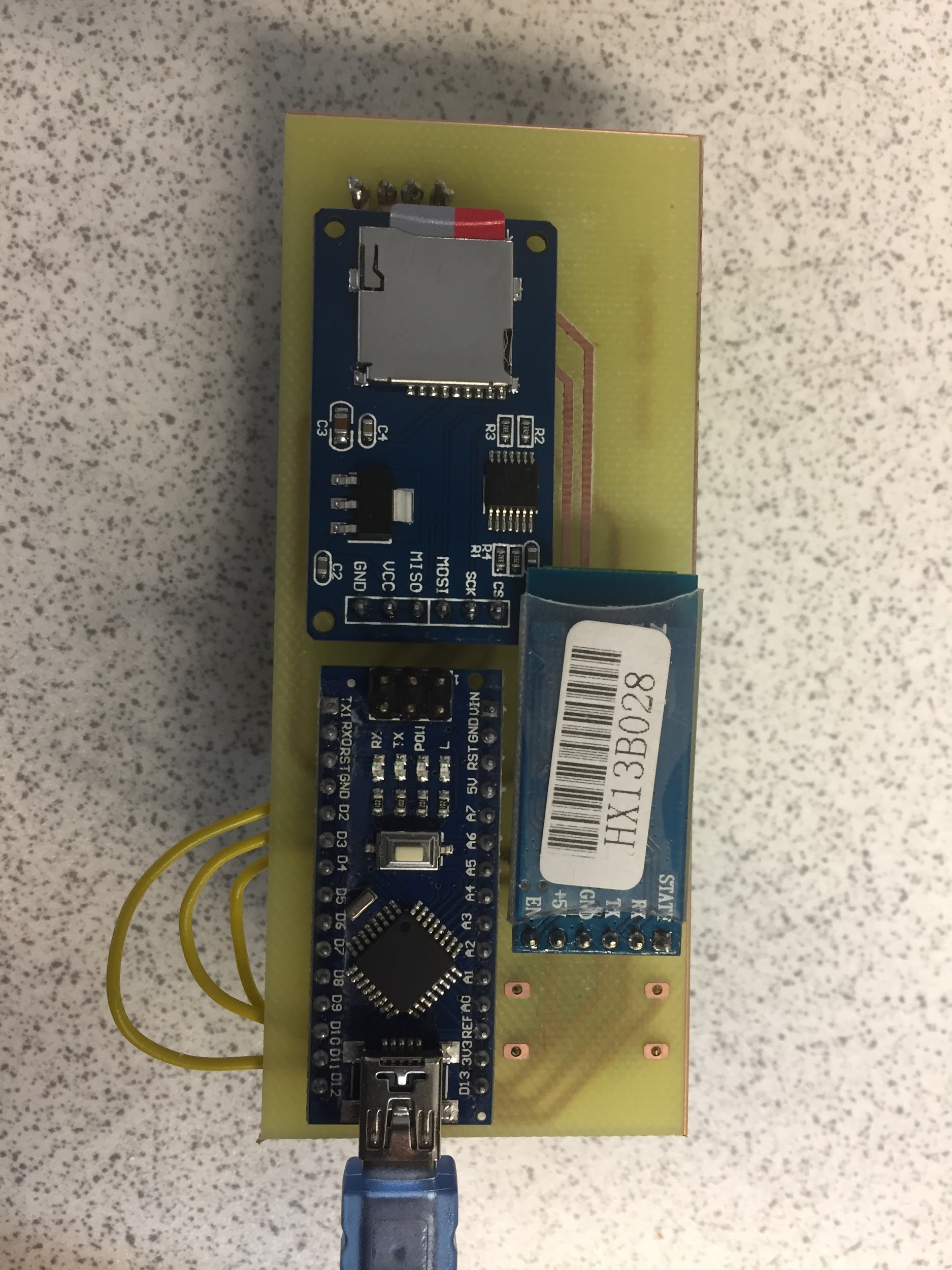


* **HC-06 Bluetooth :**

Nous utilisons un module HC-06 pour pouvoir communiquer en bluetooth avec le dispositif au niveau du guidon. C’est un module bluetooth fonctionnement uniquement en mode esclave, donc il ne peut pas initier l’appairage lui même et doit être associé avec un HC-05 (présent sur le boitier guidon).

1. **Boîtier guidon**

Le boitier guidon à pour objectif d’afficher et enregistrer la puissance et la cadence de pédalage qui lui sont transmise.





*Schéma fonctionnel du boîtier guidon*

Pour réaliser ces différentes fonctions nous avons utiliser les composant suivant :

* **Réception :** 
  + **HC-05 Bluetooth :**

Nous utilisons un module HC-05 pour pouvoir communiquer en bluetooth avec le dispositif au niveau de la manivelle. C’est un module bluetooth pouvant fonctionner en mode maître, ainsi il peut initier et contrôler l’appairage avec le module de la manivelle.



* **Traitements :**
* **Arduino Nano**

Nous avons fait le choix d’utiliser un Arduino Nano comme microcontrôleur pour des raisons de taille et de facilité de programmation.

Sa petite taille nous a permis de le souder sur un PCB avec tous les autres composants pour pouvoir diminuer la taille du dispositif au maximum.

* **Affichage / Enregistrement :**



* **Module Micro-SD :**

Nous utilisons ce pour pouvoir enregistrer la puissance et la cadence sur une carte Micro-SD. La communication se fait en SPI.

Afin de ne pas ralentir toutes l'exécution de notre code nous écrivons sur la carte seulement une fois toutes les 60 secondes.

* **Ecran LCD 16x2 avec I2C**

Pour afficher toutes les informations du cycliste, nous avons fait le choix d’utiliser un écran LCD de taille 16x2. La taille est suffisante pour la quantité d’information à afficher puisque nous utilisons également un bouton permettant de naviguer entre plusieurs affichages différents.

L’écran est accompagné d’un module I2C qui permet de diminuer grandement le nombre de fils à relier à l’arduino et de simplifier la communication avec l’écran.

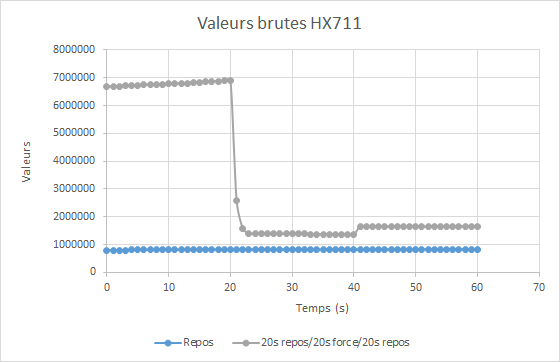
# IV) Problèmes et solutions

Au cours de notre projet nous avons été confrontés à différentes difficultés qui nous ont obligé à improviser et s’adapter pour trouver des solutions. Certaines sont liés à des erreurs de notre part, d’autre à des problèmes matériels ou logiciels, mais elles ont toutes occupés une grande partie de notre temps. Nous allons présenter dans cette parties ces différents problèmes et qu’elles sont les solutions que nous avons apportées.

1. **Jauges de déformation**

Les jauges de déformations ont été la plus grande source de difficulté dans notre projet car nous avons eu de nombreux soucis lié à leurs sensibilité et fragilité.

En effet, même après avoir suivi le protocole d’application des jauges de déformation sur la manivelle, celle-ci nous renvoie des valeurs incohérentes lorsque nous relâchons l’effort. Ci-dessous vous pouvez voir des essais effectués pour mieux visualiser notre problème :

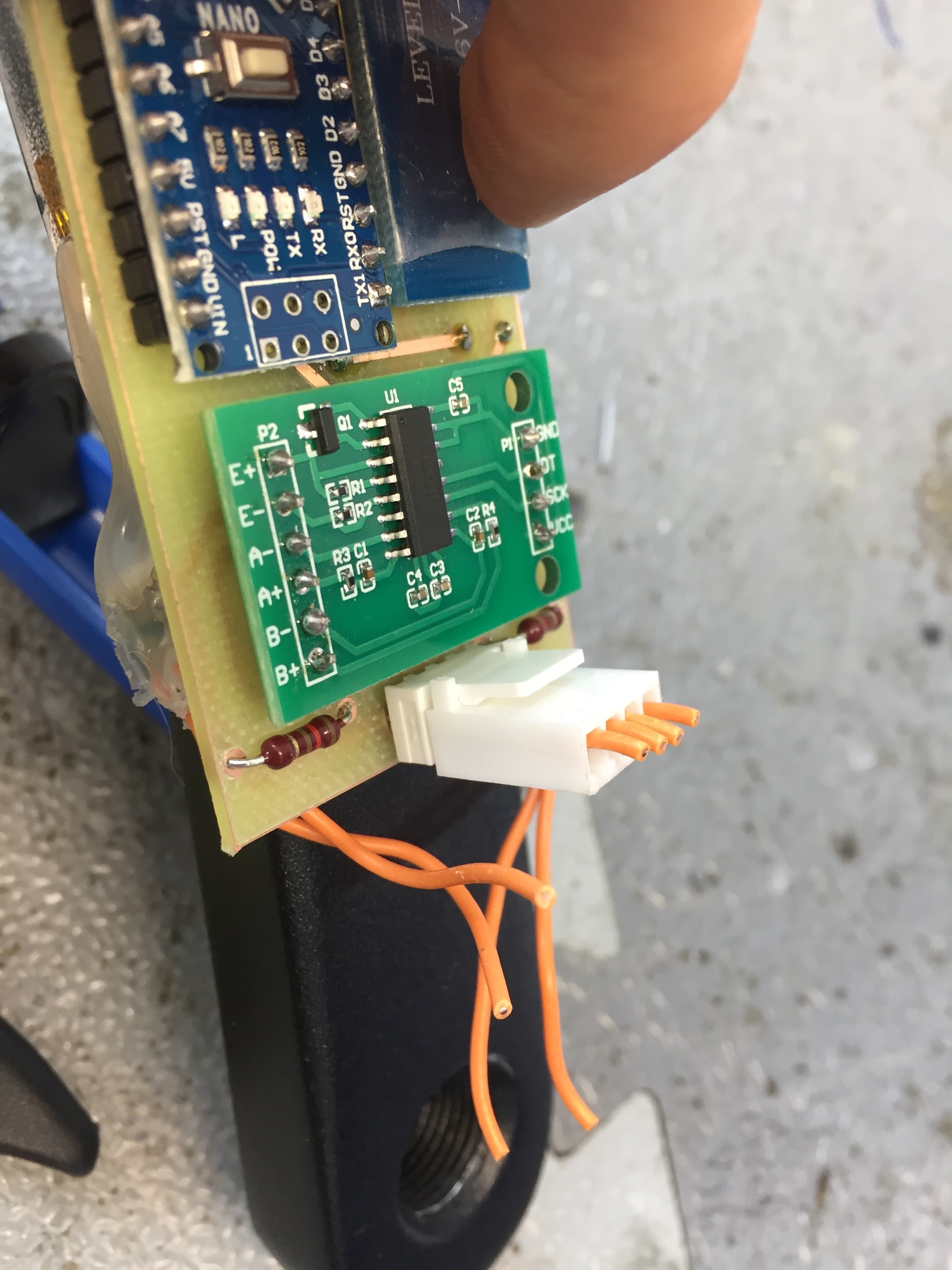
****

La courbe bleu représente les valeurs lorsque nous appliquons aucune force sur la manivelle. Jusqu’ici les valeurs semblent cohérentes puisqu’elles restent à peu près stable.

La courbe grise représente les valeurs lorsque nous appliquons une force constante pendant 20 secondes puis relâchons pendant 20 secondes. On remarque que les valeurs renvoyées lorsque la manivelle est au repos juste après un effort ne sont plus identiques à celle au repos 20 secondes auparavant. Cette différence s’accentue de plus en plus à chaque cycle ce qui rend l’utilisation de ces valeurs impossibles.

Nous avons donc cherché à comprendre l’origine du problème :

* Nous avons d’abord pensé que ceci venait des fils qui étaient volant et donc impacté les valeurs.
  + Solution potentielle : Nous avons collé les fils des jauges sur la manivelle.
  + Conclusion : Le problème était toujours présent.

* Ensuite nous avons pensé que ceci venait de la longueurs des fils et du connecteur reliant les jauges au pont de Wheatstone. 
  + Solution potentielle : Nous avons remplacé le connecteur par une soudure directement sur le PCB. Ainsi nous avons aussi pu raccourcir les fils.
  + Conclusion : Le problème était toujours présent.

*Photo de la connexion entre les jauges et le pont*

*avant de remplacer le connecteur*

* Nous avons voulu être certains que le problème ne provenait pas du module HX711.
  + Solution potentiel : Nous avons soudé des nouveaux fils pour tester un nouveau module HX711 sur une plaquette d’essai. De plus nous essayé le module avec des potentiomètres pour simuler les jauges. Au cours de ces manipulations, nous avons accidentellement arrachés une jauge. Nous avons donc dû répéter le protocole d’application de la jauge pour la recoller et la reconnecter au pont de Wheatstone.
  + Conclusion : Les valeurs renvoyées par le nouveau module présentait le même problème. Avec les potentiomètres, les valeurs étaient cohérentes. Le problème provient donc des jauges.

1. **Synchronisation des arduinos**

Une autre difficulté rencontrée était liée à la communication entre les Arduinos. En effet, si l’arduino au niveau de la manivelle était déconnecté, l’arduino au niveau du guidon s'arrêtait de fonctionner correctement. Nous devions alors réinitialiser manuellement l’arduino pour reprendre un cycle classique.

La solution apportée a été de créer une étape de synchronisation qui permet aux deux cartes de pouvoir être prévenu dès que l’une des deux cartes n’est plus connecté ou ne peut plus transmettre d’informations. Ainsi, si la carte de la manivelle s'arrête de transmettre, la carte du guidon affiche un message d’erreur et attend la reprise de la transmission pour reprendre son cycle classique automatiquement.

1. **Lecture du gyroscope**

La lecture des valeurs du gyroscope nous a aussi posé quelques difficultés. En effet pour pouvoir configurer le gyroscope et lire les valeurs nous devons passer par les registres du module. Nous avons donc dû communiquer en I2C et apprendre à accéder et écrire dans les registres.

1. **Module Micro-SD**

Nous avons fait une erreur lors du routage du PCB du guidon. Nous avons connecté le module micro SD sur des pins où la communication SPI est impossible. Or le module micro-SD communique uniquement en SPI, nous avons donc souder des nouveaux fils sur les bons pins de l’arduino.

# V) Conclusion

Au terme de ce projet, nous sommes satisfaits de notre organisation et de notre travail malgré les problèmes rencontrés. Même si les jauges de déformations nous ont posés de grandes difficultés, nous ne regrettons pas notre choix de sujet. Bien au contraire, nous avons pu apprendre à identifier les problèmes, et chercher des solutions. Ceci nous a aussi permis d’enrichir notre savoir technique en utilisant des nouveaux types de capteurs (piézo-électrique).

# VI) Bibliographie

Nous avons pu trouver plusieurs projets similaires déjà réalisés sur internet. Cela nous a permis de nous donner des pistes de solutions technologiques, notamment pour la mesure du vecteur force exercée par le cycliste.

<http://keithhack.blogspot.com/2013/01/v3power-meterthe-complete-how-to.html>

<https://www.instructables.com/id/Homemade-Cycling-Powermeter/>

<https://hackaday.com/2016/04/03/bike-power-meter-with-crank-mounted-wifi-strain-gauges/>

<https://www.joshajohnson.com/diy-bicycle-power-meter-part-1/>