

# BRACELET CONNECTÉ

## ShakeUrArms



*MASTER 1 MIAGE*  
*UE Découverte de la Recherche*

Année 2016/2017

*RESPONSABLE:*  
*M. DEQUEN Gilles*

## REMERCIEMENTS

*Avant de commencer ce rapport, nous tenons tout d'abord à exprimer nos remerciements à notre enseignant: **M. DEQUEN Gilles**, de nous avoir offert l'opportunité de travailler sur ce projet. Nous avons prit plaisir à découvrir des domaines qui étaient inconnus pour nous.*

*Ensuite, nous remercions également toute l'équipe de la **Machinerie** (FabLab et Coworking) et plus particulièrement **M. LEMAY Benjamin**, pour avoir contribué au bon déroulement et à l'aboutissement de notre projet grâce à leur aide et leurs conseils.*

## SUJET CHOISI

Initialement, l'intitulé du projet était: *ShakeUrShoes*.

Le but était de mettre en place un GPS piéton, essentiellement pour les malvoyants, avec des semelles connectées à usiner.

Imprimer une semelle 3D est plus compliqué: cette dernière doit être confortable, rigide et souple à la fois pour préserver le pied et protéger les composants. Aussi, nous étions limité en « temps » pour mener à bien l'étape de conception d'une semelle 3D.

C'est pour cela, qu' après concertation avec l'équipe de la Machinerie, nous avons opté pour ***ShakeUrArms***: un GPS piéton avec bracelets connectés. Le bracelet gauche vibre lorsqu'on doit tourner à gauche et le droit lorsqu'on doit tourner à droite.

# SOMMAIRE

- |                                 |     |
|---------------------------------|-----|
| 1. L'équipe projet              | P5  |
| 2. Prototypage Hardware         | P6  |
| 3. Application mobile           | P13 |
| 4. Perspectives d'améliorations | P20 |

# L'ÉQUIPE PROJET

Chef de projet Hardware + Software: ***Mlle. BOUTIAB Meriem Sarah***

Equipe Software: ***M. ANDUJAR Stanislas***

***M. EL MARSSI Tarek***

***M. ZHOURI Zakaria***

Equipe Hardware: ***M. ANDUJAR Stanislas***

***M. ASTAR Mohamed***

***Mlle. BENABDELHAK Chaimae***

***Mlle. BOUTIAB Meriem Sarah***

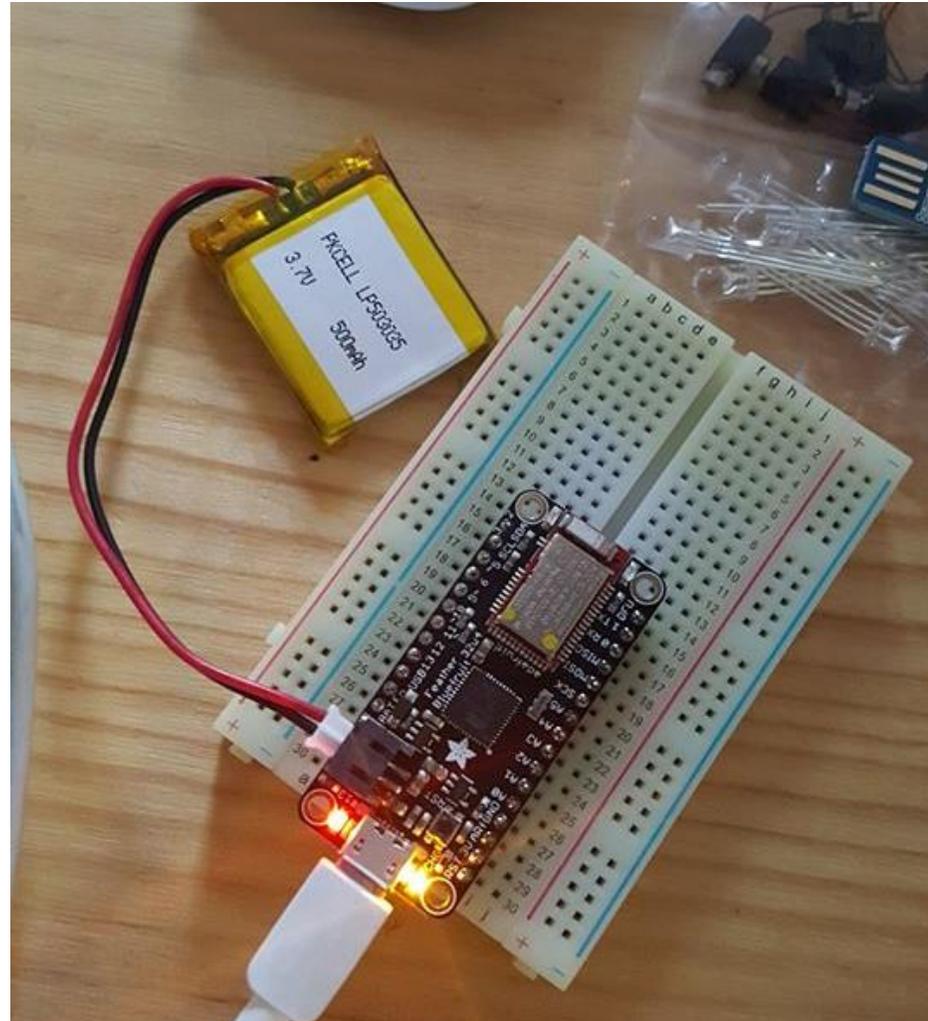
***M. CALLEBERT Maxime***

***M. DJELIL AHMED***

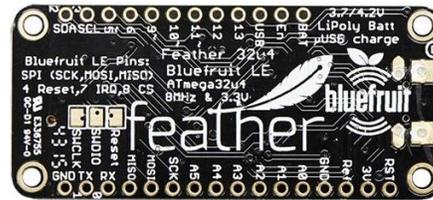
***Mlle. KOURAICHI Oumayma***

***M. YOUNESS Med Amine***

# PROTOTYPE PAGE HARDWARE



- Feather BlueFruit 32u4 (Bluetooth intégré). Cependant, nous étions contraints d'utiliser l'application qui allait avec le Feather. Cela a posé problème pour la communication Bluetooth entre notre application et le Feather. Nous avons donc opté pour des Feather 32u4 sans Bluetooth



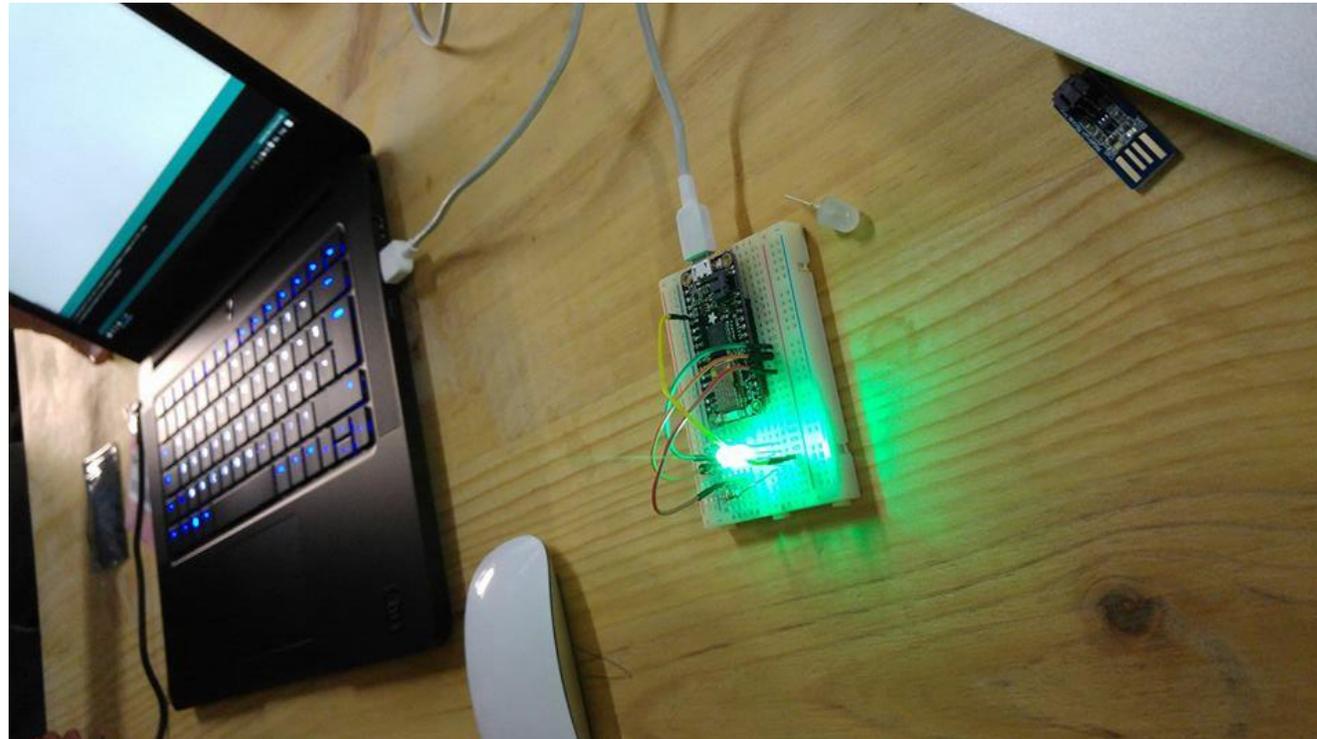
- Module Bluetooth HC 05



- Des LED 4 broches pour faire des tests et s'exercer
- Des modules vibrants Vm 390

Le Feather BlueFruit étant compatible Arduino, nous avons effectués des tests afin d'allumer une LED et d'en apprivoiser le fonctionnement.

**Lien vers la documentation:** <https://learn.adafruit.com/adafruit-feather-32u4-bluefruit-le/overview>



Dans un premier temps, nous étions partis sur un prototype incluant une LED qui indique le niveau de batterie ainsi qu'un haut parleur en plus du vibreur. Ceci a demandé de nombreuses soudures et a entraîné une mésaventure avec le Feather. Finalement et pour gagner de la place, nous avons décidé de ne garder que le module vibrant pour cette V1 de *ShakeUrArms*.

### Côté Arduino:

Pour faire communiquer le bracelet connecté et l'application, on envoie un caractère:

- « ? » : liste des commandes
- « 1 » : tourner à gauche ou à droite
- « 2 » : indiquer le niveau de batterie

Le module Bluetooth HC 05 relié directement au Feather traite l'information caractère par caractère, effectue une action en conséquence et attend une nouvelle instruction.

Attention, il ne faut pas utiliser le PIN 12 qui est déjà pris par une autre communication. Tout est spécifié dans le code côté Feather.



Nous avons expérimenté 2 méthodes pour concevoir notre bracelet:

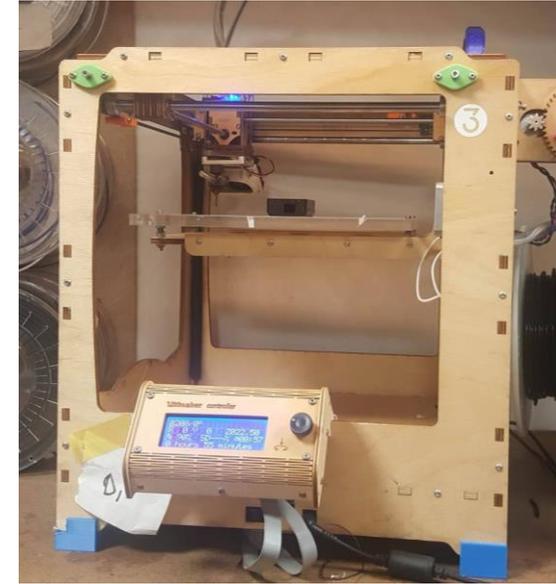
**1. Méthode découpe Laser PMMA (Poly méthacrylate de méthyle):**

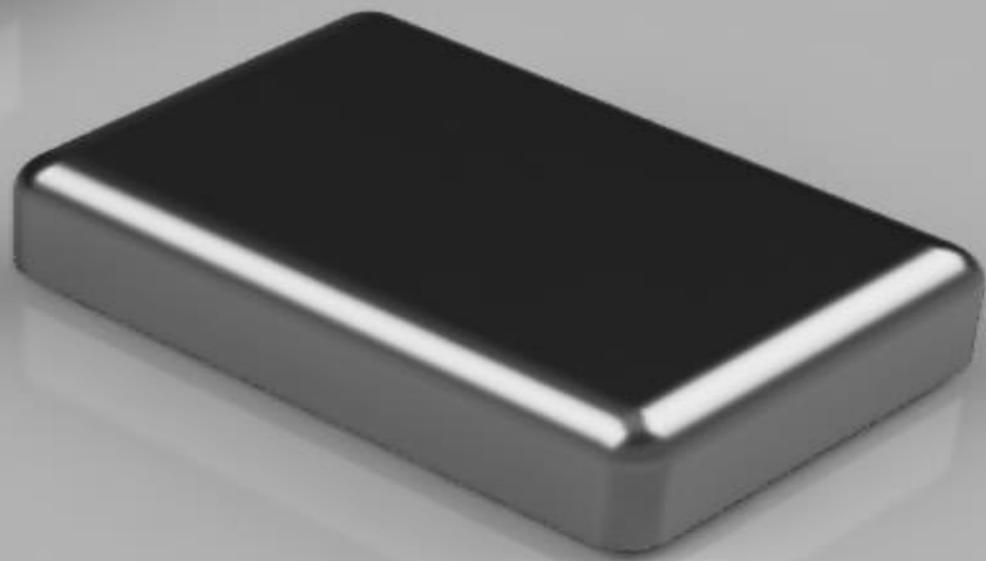
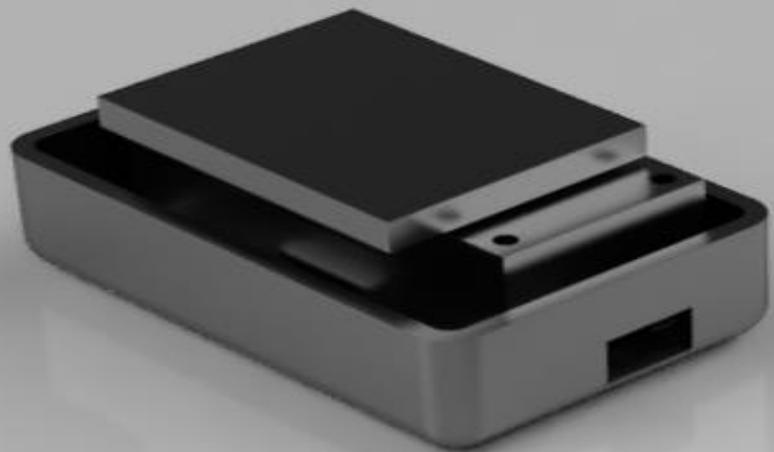
Le principe est la superposition de plaques de plexiglas de différentes épaisseurs ce qui donne le résultat ci-contre. Pour modéliser ce boitier, nous avons utilisé le logiciel Inkscape qui est un logiciel libre de dessin vectoriel avec l'aide d'un coworker au FabLab.



## 2. Impression 3D :

Imprimer le boîtier grâce aux imprimantes 3D en modélisant le boîtier sur le logiciel Fusion 360. Etant donné qu'il faut être un peu expérimenté, Benjamin Lemay s'est occupé de la modélisation. Nous avons finalement décidé d'imprimer nos deux boîtiers (gauche et droit) en 3D en utilisant . Le matériau utilisé est le PLA (acide polylactique) qui est sensible à l'eau et à la chaleur.





# APPLICATION MOBILE



En ce qui concerne le développement de l'application, nous avons choisi Ionic Framework qui s'appuie sur AngularJS et permet de créer des applications mobile hybrides (programmation iOS , Android et Windows Phone).

Nous avons également décidé d'intégrer l'API de Google Maps qui est à disposition des développeurs. Pour se faire il suffit d'avoir un compte Google puis générer une clé d'autorisation.

**Lien pour utiliser l'API de Google Maps:**

<https://developers.google.com/maps/documentation/javascript/get-api-key?hl=Fr>

1. Pouvoir récupérer les données en métrique sur l'API Google entre toutes les étapes car nous avons besoin de la direction à suivre et de la distance.

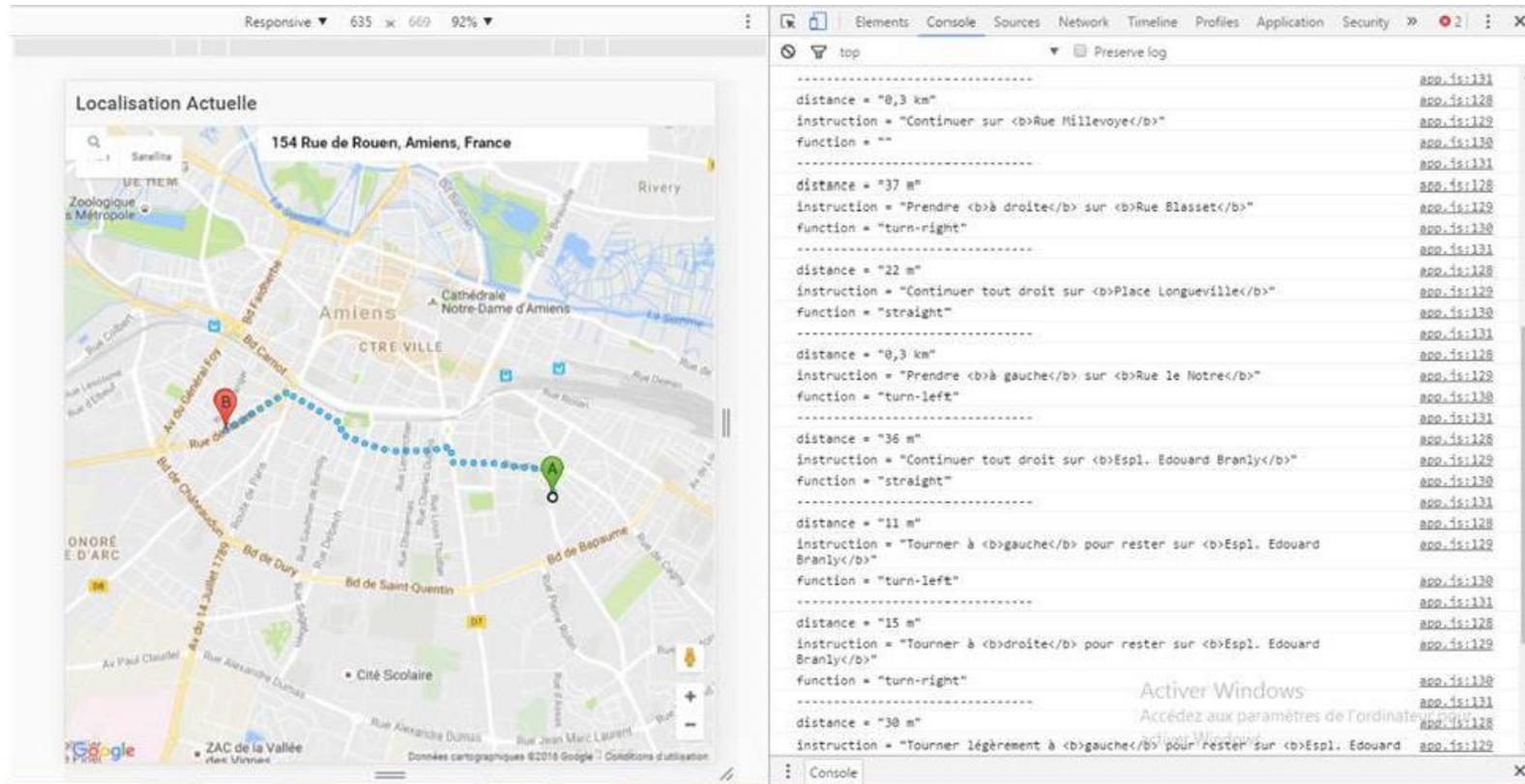
### **Solution:**

- Récupérer la localisation actuelle fournie par le plugin Ionic.
- Calculer l'itinéraire.
- Afficher la route et la récupérer en mode console (fichier JSON) puis la parser en String.
- Récupération de toutes les étapes.

L'API utilise des termes prédéfinis pour indiquer les directions et récupère de nombreuses données telles que la distance restante en temps réel (sachant que nous nous connectons à Google Maps en temps réel), le nombre de pas ...

Notre application est liée au GPS du téléphone et détecte le changement de position donc, à chaque changement, elle envoie la position actuelle à l'API de Google Maps qui fournit les nouvelles manœuvres sans prendre en compte les anciennes.

L'image ci-dessous montre les données que nous avons décidé donc de garder pour les instructions.



The image displays a Google Maps interface on the left and a browser's developer console on the right. The map shows a route starting at '154 Rue de Rouen, Amiens, France' (marked with a red pin 'B') and ending at a green pin 'A'. The route is highlighted in blue and passes through several streets including Rue Millevoye, Rue Blasset, Place Longueville, Rue le Notre, Espl. Edouard Branly, and Rue de Bapume.

The console log on the right contains the following instructions:

```
----- app.js:131
distance = "0,3 km" app.js:128
instruction = "Continuer sur <b>Rue Millevoye</b>" app.js:129
function = "" app.js:130
----- app.js:131
distance = "37 m" app.js:128
instruction = "Prendre <b>à droite</b> sur <b>Rue Blasset</b>" app.js:129
function = "turn-right" app.js:130
----- app.js:131
distance = "22 m" app.js:128
instruction = "Continuer tout droit sur <b>Place Longueville</b>" app.js:129
function = "straight" app.js:130
----- app.js:131
distance = "0,3 km" app.js:128
instruction = "Prendre <b>à gauche</b> sur <b>Rue le Notre</b>" app.js:129
function = "turn-left" app.js:130
----- app.js:131
distance = "36 m" app.js:128
instruction = "Continuer tout droit sur <b>Espl. Edouard Branly</b>" app.js:129
function = "straight" app.js:130
----- app.js:131
distance = "11 m" app.js:128
instruction = "Tourner à <b>gauche</b> pour rester sur <b>Espl. Edouard Branly</b>" app.js:129
function = "turn-left" app.js:130
----- app.js:131
distance = "15 m" app.js:128
instruction = "Tourner à <b>droite</b> pour rester sur <b>Espl. Edouard Branly</b>" app.js:129
function = "turn-right" app.js:130
----- app.js:131
distance = "30 m" app.js:128
instruction = "Tourner légèrement à <b>gauche</b> pour rester sur <b>Espl. Edouard Branly</b>" app.js:129
----- app.js:131
```

2. Établir la liaison Bluetooth avec les deux bracelets et communiquer via une trame de message.

**Solution:**

Sachant qu'il existe 2 technologies de Bluetooth : low energy et serial, notre module HC 05 est en sérial. Nous avons utilisé un plugin open source sur Github « BluetoothSerial » qui nous offre la possibilité de communiquer avec notre module HC 05 et d'ailleurs, ce plugin là est utilisé également pour la communication entre Android ou iOS et Arduino. Cependant ce plugin a des limites: l'appairage passe obligatoirement par notre téléphone.

**Lien Github vers le plugin:** <https://github.com/don/BluetoothSerial#readuntil>

Le plugin BluetoothSerial offre plusieurs méthodes qui vont faciliter le processus communication entre la chipset Bluetooth du téléphone et les bracelets par exemple lister les périphériques, les appairer , se connecter à un périphérique à l'aide d'un Id, envoyer/recevoir des données.

Comment l'application communique-t-elle avec le bracelet?

D'abord, nous devons appairer le téléphone avec nos 2 bracelets. Nous avons défini un système de sécurité pour les bracelets avec un mot de passe qui est 3006.

Ensuite, côté application, on liste tous les appareils qui sont appariés avec la fonction *bluetoothSerial.list* ensuite on cherche notre périphérique par son nom avec *device.name*. Une fois trouvé, on se connecte à ce dernier grâce à son Id, on envoie notre caractère (« 1 » pour tourner à gauche ou à droite) puis on se déconnecte.

On répète l'opération à chaque fois qu'on doit envoyer une donnée. Pour l'instant c'est la seule solution qui répond à notre problématique.

Nous avons utilisé divers autres plugin dont whitelist qui permet, en général de donner l'accès aux sites Internet, et dans notre cas donner l'accès à Google Maps dans le fichier « config.xml ».

Vous trouverez les code de l'application et côté Feather en suivant le lien ci-dessous:

<https://github.com/standujar/M1Miage/tree/master/ShakeUrArms>

## PERSPECTIVES D'AMÉLIORATIONS

Pour une V2, nous voulons réduire la taille du bracelet et donc des composants et forcément avoir un meilleur design.

Pour cela nous avons pensé à utiliser d'autres composants, plus petits par exemple on peut remplacer le module Bluetooth HC 05 par un ESP3212 qui est plus petit qui est compatible WiFi et Bluetooth. On peut souder directement ce Bluetooth sur notre Feather mais nous n'avons pas encore réfléchi au fonctionnement.

Aussi, nous aimerions repartir sur notre idée de base et intégrer une LED ainsi qu'un haut parleur et bien sûr utiliser un matériau opaque pour la lumière. Nous aimerions également créer un design spécifique pour notre bracelet.

Côté application, l'amélioration majeure serait de pouvoir se connecter aux 2 bracelets en même temps.

## CONCLUSION

Ce sujet, en plus d'être très intéressant, nous a offert la liberté de pouvoir plus ou moins choisir vers quoi on veut se pencher est vraiment agréable.

Ce projet de recherche nous a permis d'aborder le prototypage Feather et de découvrir le fonctionnement de la communication Bluetooth à travers un objet connecté. Nous avons également imprimé notre premier objet en 3D .

En plus d'avoir fait de superbes rencontres, le projet s'est déroulé dans une ambiance très sympathique et nous en gardons un souvenir fort agréable. Des liens se sont créés et nous espérons que ce n'est que le début d'une grande aventure.