

# RAPPORT DU SEMESTRE 2

## *SMART BIKE*

+ALACHE KASSI  
+AVOGNAN OLIVIA  
+SOUMAGNIN AKOUA  
+YASSI FREDERIC

# SOMMAIRE

SOMMAIRE .....	2
TABLE DES FIGURES.....	3
INTRODUCTION .....	4
I- DESCRIPTION DU PROJET.....	5
1) Contexte.....	5
2) Objectif général du projet.....	5
II- ETUDE DU PROJET .....	6
1) Les spécifications fonctionnelles.....	6
2) Idées conservées de nos prédécesseurs.....	7
3) Idées apportées.....	7
4) Architecture fonctionnelle .....	8
5) Comparaisons des technologies Bluetooth.....	14
6) Comparaison des langages de programmation.....	15
7) Architecture logicielle .....	19
III-PRESENTATION DE L'EQUIPE ET ORGANISATION .....	19
1) Pyramide des tâches.....	20
2) Diagramme des responsabilités .....	20
IV-REALISATION DU PROJET.....	21
1) Partie électronique .....	21
2) Partie informatique .....	29
IV-DIFFICULTES RENCONTRES ET PERSPECTIVES ENVISAGEES.....	34
CONCLUSION.....	35
WEBOGRAPHIE .....	36
BIBLIOGRAPHIE.....	37

## TABLE DES FIGURES

Figure 1:Architecture fonctionnelle.....	8
Figure 2: leds de 3 mm .....	9
Figure 3 : leds de 5 mm .....	9
Figure 4 : leds de 10 mm .....	9
Figure 5: leds de20 mm.....	10
Figure 6 : exemple 1 d'utilisation des leds .....	10
Figure 7: exemple 2 d'utilisation des leds.....	10
Figure 8 : capteur de vitesse .....	12
Figure 9: capteur de vitesse FEBI BILSTEIN .....	13
Figure 10: accelerometre .....	14
Figure 11: architecture matérielle .....	18
Figure 12 : montage .....	22
Figure 13: tet pour éteindre la led.....	22
Figure 14 : detection de freinage .....	23
Figure 15 : test allumage de la led .....	24
Figure 16 : firebase authentication.....	30
Figure 17: firebase firestore collection users .....	30
Figure 18 : firebase firestore collection parcours .....	31
Figure 19: page d'accueil	Figure 20: page d'authentification .....32
Figure 21: page de la map	Figure 22 : détail sur le parcours .....33

## INTRODUCTION

L'utilisation de capteurs dans les accessoires permet d'enregistrer des données retransmises sur des applications mobiles dédiées : vitesse, nombre de mouvements, fréquence, intensité de l'activité... De plus en plus d'accessoires connectés sont créés pour faciliter le quotidien des usagers dont le smart bike. Ce dernier étant un élément essentiel de l'écomobilité, de la transition durable que nous sommes en train de vivre, plus de 50% de la population préfère se déplacer en véhicule pour éviter les persécutions avec les voitures. Nous comprenons donc l'urgence de mettre en place une solution qui protégera et le cycliste et les usagers de la route. Ainsi, comment se prémunir et protéger les autres de tout danger lié à la circulation de la route ? Comment traçabiliser ses parcours ?

Afin de répondre à cette interrogation, il nous a été soumis le thème « Mise en place d'un smart bike ». L'objectif étant la mise en place d'une application mobile permettant de contrôler son vélo et récupérer l'historique de ses parcours. Pour mieux comprendre ce projet et d'en cerner tous les rouages, nous avons jugé nécessaire de le scinder en trois (03) grandes parties. D'abord la première partie intitulée « Etude préalable » met le projet dans son contexte à travers la présentation du cadre de référence et du projet. Ensuite dans la seconde partie appelée « Architecture du système », nous présentons les différentes étapes de la conception du système. Enfin la dernière partie intitulée « Mise en œuvre du système » est consacrée à la réalisation du système avec certains aspects inhérents à cette réalisation.

## I- DESCRIPTION DU PROJET

### 1) Contexte

Les vélos personnels ou ceux en location (Star) ne disposent pas d'un langage commun permettant de communiquer de façon gestuelle entre eux mais aussi entre les autres usagers. Ceci dit, les mouvements du cycliste ne sont connus que par lui seul. Par ailleurs, il n'est pas possible de retracer ses parcours pour des fins utiles (améliorer sa conduite, atteindre des objectifs à vélo, connaître des itinéraires déjà empruntés etc.). Notre application propose des systèmes de vélos connectés basés sur une technologie GPS permettant à la ville de mettre à disposition des vélos capables d'alerter les usagers de ses actions sans mettre en place d'infrastructures spécifiques, de suivre et sauvegarder ses trajets. L'utilisateur a besoin d'un smartphone muni de l'application pour avoir la traçabilité de son parcours. Il s'agira d'ajouter des feux de direction et de freinage pour une meilleure sécurité routière puis à les connecter avec une application mobile pour en tirer des informations tels que sa vitesse, le trajet parcouru, etc...

### 2) Objectif général du projet

Contrôler les indicateurs de son vélo à partir d'une application mobile et récupérer certaines informations du vélo grâce à celle-ci.

Cet Objectif se décline en plusieurs objectifs spécifiques, il s'agira ici de mettre en place une application mobile capable de :

- Contrôler l'indicateur avec des boutons
- Afficher la vitesse instantanée, la vitesse moyenne sur le trajet en cours ainsi que la distance parcourue (à partir du gps du téléphone ou du vélo connecté)
- Avoir l'historique des trajets

## II- ETUDE DU PROJET

### 1) Les spécifications fonctionnelles

Il s'agit ici d'énumérer toutes les fonctions dont disposera le Smart Bike.

<b>En tant que</b>	<b>Je dois pouvoir</b>	<b>Afin de</b>
Cycliste	Indiquer que je freine	Protéger et renseigner les autres usagers
Cycliste	Indiquer dans quelle direction je tourne à partir de l'application	Me protéger et renseigner les autres usagers
Cycliste	Connaître ma vitesse instantanée	Me protéger l'excès de vitesse tue
Cycliste	Connaître ma vitesse moyenne	Savoir
Cycliste	Connaitre mon trajet par l'application	Arriver à bon port
Cycliste	Connaitre mon historique de trajet	Savoir où je suis allé

## 2) Idées conservées de nos prédécesseurs

En tant que	Je dois pouvoir	Afin de
Cycliste	Consulter ma position	Me repérer sur la carte
Cycliste	Consulter mon trajet	Savoir où aller
Installateur	Me procurer du matériel et l'attacher facilement	Rendre le vélo connecté
Cycliste	Configurer l'application	Pouvoir me connecter à mon vélo
Installateur	Comprendre le matériel	L'expliquer au client

## 3) Idées apportées

En tant que	Je dois pouvoir	Afin de
Cycliste	Indiquer ma direction de virage par un bouton sur le guidon	Prévenir les autres usagers
Cycliste	Connaitre mon trajet en se référant à un indicateur lumineux sur le guidon	Faciliter la conduite et se concentrer sur la conduite
Cycliste	Connaitre mon trajet en se référant à la voix du GPS de notre application	Me repérer

#### 4) Architecture fonctionnelle



Figure 1: Architecture fonctionnelle

### 5-Etude des outils et protocoles

#### 5.1-Module affichage

Pour le module d'affichage, nous allons utiliser des Leds selon leurs caractéristiques et l'objectifs du projet.

##### Les types de led

Il existe différents types de led avec des formes, tension et courant de fonctionnement différents, nous disposons des Leds suivantes:



- Les led de 3mm



*Figure 2: leds de 3 mm*

- Les led de 5mm



*Figure 3 : leds de 5 mm*

- Les led de 10mm



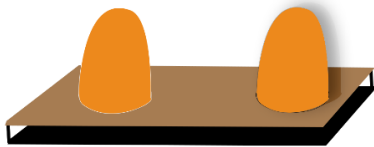
*Figure 4 : leds de 10 mm*

- Les led de 20mm

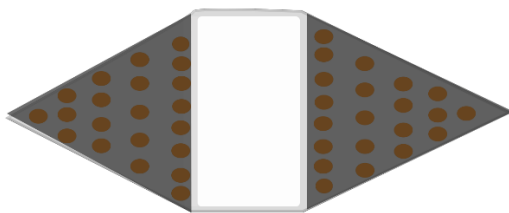


*Figure 5: leds de 20 mm*

Dans le but de réaliser un indicateur de direction sur le vélo, nous utiliserons les différentes led vu précédemment. On les associe pour obtenir, soit le modèle de gauche, soit le modèle de droite. Pour chaque modèle effectué un dimensionnement de l'alimentation sera fait au préalable pour une meilleure conception. Nous opterons pour les leds ci-dessous en raison de l'intensité forte de leur éclairage et aussi pour leur disponibilité chez le professeur encadreur.



*Figure 6 : exemple 1 d'utilisation des leds*



*Figure 7: exemple 2 d'utilisation des leds*

## *5.2-Module Alimentation*

Dans cette partie, il s'agit de faire un inventaire des moyens qui pourraient être utilisés pour alimenter tout le système électronique qui sera utilisé.

a) Batterie

La solution la plus simple et celle qui est réalisable demeure l'utilisation d'une batterie. Selon les besoins du système en courant et tension, l'on choisira la batterie adéquate

b) Téléphone

Vu que dans le projet, un téléphone portable intervient, on pourrait l'utiliser pour alimenter tout le système avec un câble. Nous ne savons pas si elle est faisable ou pas dans la réalité et à l'aide de quel moyen. Mais utiliser un téléphone de cette manière peut le détériorer, ce qui ne serait pas intéressant pour le projet.

c) Energie généré à partir des pédales

On pourrait également utiliser l'Energie mécanique généré par le mouvement des pédales pour la convertir en Energie électrique. C'est une solution très intéressante mais elle est en dehors de notre formation et nos compétences actuelles.

d) Energie solaire

Une perspective aussi serait d'utiliser de l'Énergie renouvelé à savoir l'Energie solaire pour alimenter notre système

Au vu des différentes analyses, nous choisirons l'utilisation d'une batterie comme moyen d'alimentation.

### 5.3 - Module de communication et Traitement de données

Pour la communication et le traitement des données une étude comparative décrite dans le tableau ci-dessous nous aidera dans le choix d'un des microcontrôleurs existants mais respectant les exigences du projet.

	ESP8266 NodeMCU V3	ESP32 NodeMCU	ESP8266 WeMos D1 Mini	Arduino UNO WIFI R2
Microcontrôleur	ESP8266	ESP32	ESP8266	ATmega4809
Tension d'exploitation	3.3V	3.3V	3.3V	5V
Alimentation(caractéristique)	7V – 12V	7V – 12V	4V – 6V	7V – 12V
Consommation actuelle	15 µA – 400 mA	20 mA – 240 mA		50 mA – 150 mA
Broches numériques I/O	16	36	11	14
Broches numériques I/O avec PWM	16	36	11	5
Broches d'entrée analogiques	1	15	1	6
SPI/I2C/I2S/UART	2/1/2/2	4/2/2/2	1/1/1/1	1/1/1/1
Courant DC par broche I/O	12 mA	20 mA		40 mA
Mémoire flash	4 MB	4 MB	4 MB	48 KB
SRAM	64 KB	520 KB		6 KB
EEPROM	512 bytes	–		256 bytes
Disponibilité en France	oui	oui	non	oui
Vitesse de l'horloge	80 MHz	80 MHz / 160 MHz	80 MHz / 160 MHz	16 MHz
Bluetooth	oui	oui	oui	non
Ethernet MAC Interface	Non	Oui	Non	Non
Prix	6 euros	9.02euros	5.92 euros	36.82euros

Nous choisirons pour le cadre de notre projet, la carte ESP8266 NodeMCU V3 en particulier pour son module Bluetooth, sa performance et son coût mais aussi sa disponibilité en France. Vu sa capacité à se connecter en réseau, elle agira également en tant qu'unité de communication.

#### 5.4-Module acquisition de données

##### a. Capteur de vitesse

Cette partie consiste à exposer les différents outils permettant d'avoir la vitesse de notre vélo.

- Capteur de vitesse LM393 H2010



Figure 8 : capteur de vitesse

- 2 sorties : 1 numérique, 1 analogique
- LED d'alimentation
- LED pour impulsions à D0
- Tension de service : de 3.3 à 5 VCC
- Largeur de rainure : 5 mm
- Masse : 8 g
- Dimensions : 32 x 14 x 7 mm

- Capteur FEBI BILSTEIN



## FEBI BILSTEIN Capteur, vitesse

Réf.: 109728

Nombre de connexions:	3
Poids [kg]:	0,024
FEBI BILSTEIN:	Capteur, vitesse
Numéro d'article:	109728
Notre prix:	15,35 €
Numéro de pièce du fabricant:	109728
Fabricant:	FEBI BILSTEIN
Numéro de EAN:	4054224097281

N° de châssis à partir de : Cette caractéristique varie en fonction du modèle de voiture.

L'état Neuf

Utilisez ce numéro: FEBI BILSTEIN 109728

EAN: FEBI BILSTEIN 4054224097281

Figure 9: capteur de vitesse FEBI BILSTEIN

- Vitesse grâce au GPS

Il est aussi possible de récupérer la vitesse instantanée du vélo grâce au GPS du téléphone.

### Récapitulatif des différents outils pour l'acquisition des données

Outils	Avantages	Inconvénients	Choix
Capteur LM393 H2010	- Précision de la vitesse élevée	- Compétence technique élevée - Temps de prise en main de l'outil - Cout	2
Capteur FEBI BILSTEIN	- Précision de la vitesse élevée	- Compétence technique élevée - Temps de prise en main de l'outil - Cout - Ce module nous est inconnu	3
Vitesse grâce au GPS	- Compétence technique faible - Pas de cout	- Précision de la vitesse moins élevée - Dépendance à internet	1

#### b. Capteur de freinage

Accéléromètre

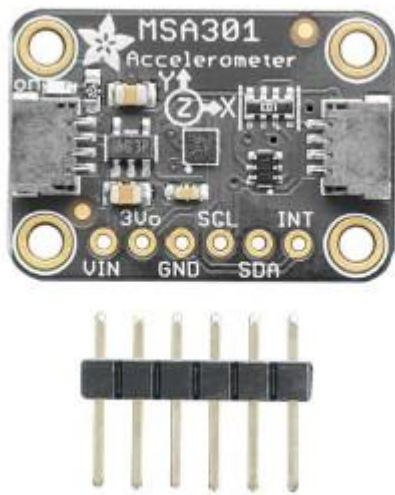


Figure 10: accelerometre

Décélération grâce aux informations du GPS

Lors d'une décélération normale (sans appuyer le frein ou du uniquement aux forces de frottement entre le sol et les pneus), la vitesse diminue de manière quasi linéaire. En utilisant ce constant, nous considérons que le cycliste appuie sur le frein s'il y'a une chute brusque de vitesse entre l'instant  $t$  et l'instant  $t+1$ . Dans ce cas, nos phares arrière s'allumeront.

## 5) Comparaisons des technologies Bluetooth

Pour communiquer entre l'objet connecté et le smartphone, nous allons avoir besoin D'utiliser un protocole communiquant sans fil courte distance. Nous avons donc un Choix à faire entre le Wi-Fi, le Bluetooth, Zigbee, Z-Wave...

Technologies Bluetooth	Avantages	Inconvénients
Zigbee	Nécessite l'utilisation d'un pont ou concentrateur	Les appareils ne se connectent pas individuellement à Internet
	<b>Faible énergie consommation</b>	<b>Portée réduite</b> , entre 10 et 20 mètres.
	C'est un protocole ouvert.	Vitesses de transfert de données très faibles, vitesse de <b>250 kbit/s.</b>
	La collecte de données ne nécessite pas de vitesses élevées	
Bluetooth Low Energy	la latence pour la connexion et le transfert de données est réduite.	Il fonctionnera uniquement avec un smartphone compatible Bluetooth 4.0 LE
	Basse consommation	Portée et débit réduit Par rapport au

		Bluetooth classique
	Déjà implémenté dans Le smartphone	Temps de reconnexion Plus élevé que le Zigbee après "sommeil" Zigbee
NFC	Aucun besoin de paramétrage et d'appariement	Portée réduite entre 4 et 10 centimètre
	Réduction des interférences et des complications de transfert de données dû à la faible distance	
	Moins énergivore (économie de batterie)	
	Œuvre pour une fréquence de 13.56Mhz	
Z-WAVE	Nécessite l'utilisation d'un <b>pont</b>	Vitesses de transfert de données très faibles, vitesse maximale de <b>100 kbit/s.</b>
	<b>Faible</b> énergie de <b>consommation</b>	Portée maximale 100m.
	Portée maximale 100m	C'est un protocole fermé.
		Les appareils ne se connectent pas individuellement à Internet

On opte pour Bluetooth Low Energy, en raison de la portée et le débit réduit du BLE bien que la portée soit courte mais dans le cadre de notre projet, le smartphone et le vélo ne sont pas très distants donc nous convient, ensuite pour son implémentation déjà sur smartphone cela facilitera le paramétrage et nous n'aurons plus besoin de passer par une passerelle comparativement aux autres technologies Bluetooth.

## 6) Comparaison des langages de programmation

- Dart

Dart est un langage de programmation optimisé pour les applications sur plusieurs plateformes. Il est développé par Google et est utilisé pour créer des applications mobiles, de bureau, de serveur et web.

Dart est un langage orienté objet, basé sur la classe, récupérateur de mémoire avec une syntaxe de type C. Dart peut se compiler en code natif ou en JavaScript. ...

### Utilisation

Malgré sa relative jeunesse, le langage de programmation DART a déjà su s'imposer, notamment dans la programmation d'applications mobiles. L'entreprise Google a lancé Flutter, son kit de développement de logiciel (SDK) qui utilise DART et favorise l'utilisation et la diffusion de ce langage de programmation. Dart est un langage de programmation optimisé pour les applications sur plusieurs plateformes. Il est développé par Google et est utilisé pour créer des applications mobiles, de bureau, de serveur et web.

- Java

Le langage Java reprend en grande partie la syntaxe du langage C++. Néanmoins, Java a été épuré des concepts les plus subtils du C++ et à la fois les plus déroutants, tels que les pointeurs et références, ou l'héritage multiple contourné par l'implémentation des interfaces. De même, depuis la version 8, l'arrivée des interfaces fonctionnelles introduit l'héritage multiple (sans la gestion des attributs) avec ses avantages et inconvénients tels que l'héritage en diamant. Les concepteurs ont privilégié l'approche orientée objet de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.) qui ont cependant leurs variantes qui héritent de l'objet Object (Integer, Float, ...).

#### Utilisation

Java est l'un des langages de programmation les plus polyvalents et les plus flexibles au monde. Il est utilisé par des entreprises du monde entier pour créer des applications de bureau et pour le développement de sites Web back-end.

- Kotlin

Kotlin est un langage de programmation orienté objet et fonctionnel, avec un typage statique qui permet de compiler pour la machine virtuelle java, javascript, et vers plusieurs plateformes en natif (grâce à llvm). Son développement provient principalement d'une équipe de programmeurs chez JetBrains basé à Saint-Petersbourg en Russie

#### Utilisation

Kotlin est un langage de programmation à typage statique, open source développé par JetBrains. Tout comme Java, Kotlin s'exécute dans la JVM. Kotlin peut être utilisé pour développer des applications multi plateforme pour ordinateur, Android, web ETC...

### ➤ TYPE DE BASE DE DONNÉES

#### Base de données relationnelle

Une base de données relationnelle est une base de données où l'information est organisée dans des tableaux à deux dimensions reliés entre eux par des clefs.

Différents systèmes de gestion de base de données relationnelle

- MySQL

MySQL est un serveur de bases de données relationnelles SQL qui fonctionne sur de nombreux systèmes d'exploitation différents incluant Linux, Mac OS X, Windows.



- PostgreSQL

PostgreSQL est un système de gestion de base de données relationnelle, objet et libre. Il est un système de base de données hybride SQL/NoSQL qui trouve un terrain d'entente entre ces deux options.

- Oracle

Oracle Database est un système de gestion de base de données relationnelle (SGBDR) qui depuis l'introduction du support du modèle objet dans sa version 8 peut être aussi qualifié de système de gestion de base de données relationnel-objet (SGBDRO).

- Base de données Non Relationnelle ou NoSQL

Le NoSQL est un type de bases de données, dont la spécificité est d'être non relationnelles. La principale particularité des bases de données NoSQL est qu'elles ne suivent pas le modèle relationnel et ne présentent pas de tableaux sous forme de colonnes fixes. Ces bases de données ne nécessitent pas de normalisation de données ou de mapping relationnel. Il est possible d'interagir sans utiliser de langages de requête complexe.

- MongoDB

PostgreSQL est un système de gestion de base de données orienté objet » (SGBD), et pas simplement un » système de gestion de base de données relationnel » (SGBDR). Cela signifie qu'il sert d'hybride entre un modèle strictement relationnel (SQL) et un modèle strictement orienté objet (NoSQL).

Firebase

Firebase est un ensemble de services d'hébergement de google pour n'importe quel type d'application (Android, iOS, javascript, node.js, java, unity, PHP, c++ ...). Il propose d'héberger en nosql et en temps réel des bases de données,

➤ CHOIX DE LA BASE DE DONNEES (AVANTAGE ET INCONVENIENT)

TYPE DE BASE DE DONNEES	AVANTAGE	INCONVENIENT	CHOIX
DB SQL	<ul style="list-style-type: none"> <li>- BASE DE DONNEES STRUCTUREE ET SEGMENTEE</li> <li>- STOCKAGE OPTIMISE ET STABILITE</li> </ul>	<ul style="list-style-type: none"> <li>- REQUETE COMPLEXE</li> </ul>	2
DB NOSQL	<ul style="list-style-type: none"> <li>- GRANDS ENSEMBLE DE DONNEES</li> <li>- LES DONNEES NECESSAIRE PEUVENT ETRE RECUPERE D'UN COUP SANS JOINTURE PARTICULIERE</li> <li>- TECHNOLOGIE NATURELLEMENT COMPATIBLE AVEC FLUTTER</li> <li>- REQUETE SIMPLE ET PERFORMANTE</li> <li>- GESTION AUTOMATIQUE DE NOTIFICATION AVEC FLUTTER</li> <li>- FACILITE ET FLEXIBILITE DU STOCKAGE</li> </ul>	<ul style="list-style-type: none"> <li>- PARFOIS TROP PERMISSIF</li> </ul>	1

6 – Architecture matérielle

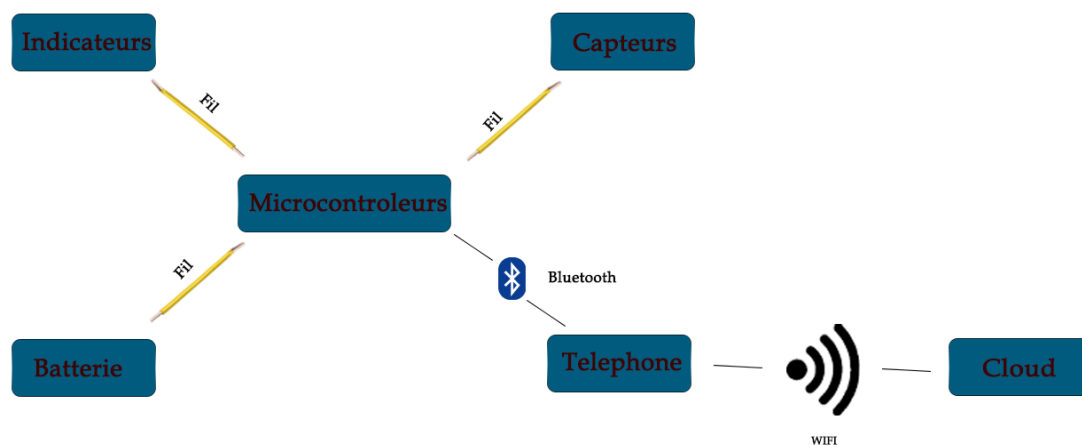


Figure 11: architecture matérielle

Ci-dessus présenté l'architecture matérielle avec ses différentes composantes et la manière dont elles communiquent entre elles. Les microcontrôleurs sont pratiquement au centre de l'architecture où intervient le système électronique. Ils communiquent avec les capteurs, les indicateurs de direction et la batterie de façon filaire. Les microcontrôleurs sont commandés avec le téléphone par le Bluetooth et le téléphone interagit avec la base de données par le wifi.

## 7) Architecture logicielle

\*Protocoles de communication

-Bluetooth

Le protocole BLE est celui que nous utiliserons dans notre projet pour la communication entre le téléphone portable et le système électronique .

-Http

\*Base de données

Nous avons choisi une base de données noSQL qui semble plus adaptée à notre projet .

Elle sera en ligne et hébergée par firebase .

\*Langage de programmation

Nous avons choisi Flutter pour réaliser notre application mobile qui est basé sur le langage dart .

## III-PRESENTATION DE L'EQUIPE ET ORGANISATION

L'équipe est constituée de trois étudiants en 2eme année IOT à l'ESIR et dont les compétences sont :

Membres	AVOGNAN	SOUMAGNIN	YASSI	ALACHE
Compétence				
Programmation Arduino	**	**	***	***
Programmation ESP	**	*	*	**

Câblage	*	**	**	***
Conception de Base de Données	**	**	*	*
Modélisation logiciel	*	*	*	*
Dev Mobile	*	*	*	*

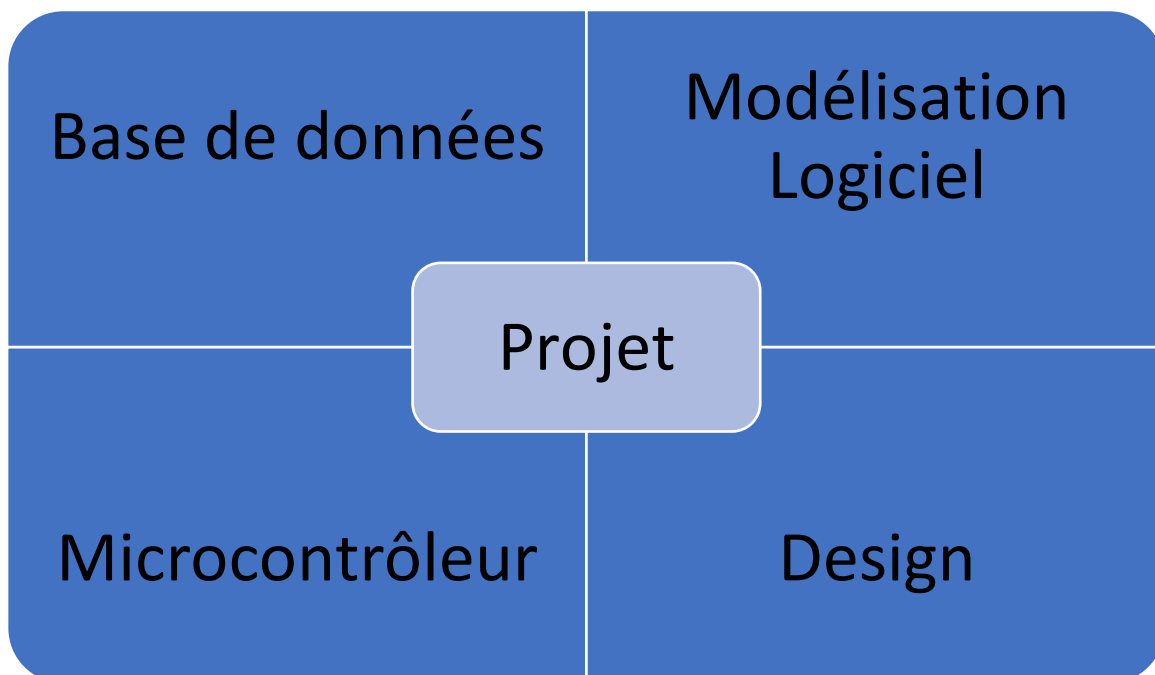
« \* » : débutant

« \*\* » : assez bon

« \*\*\* » : bon

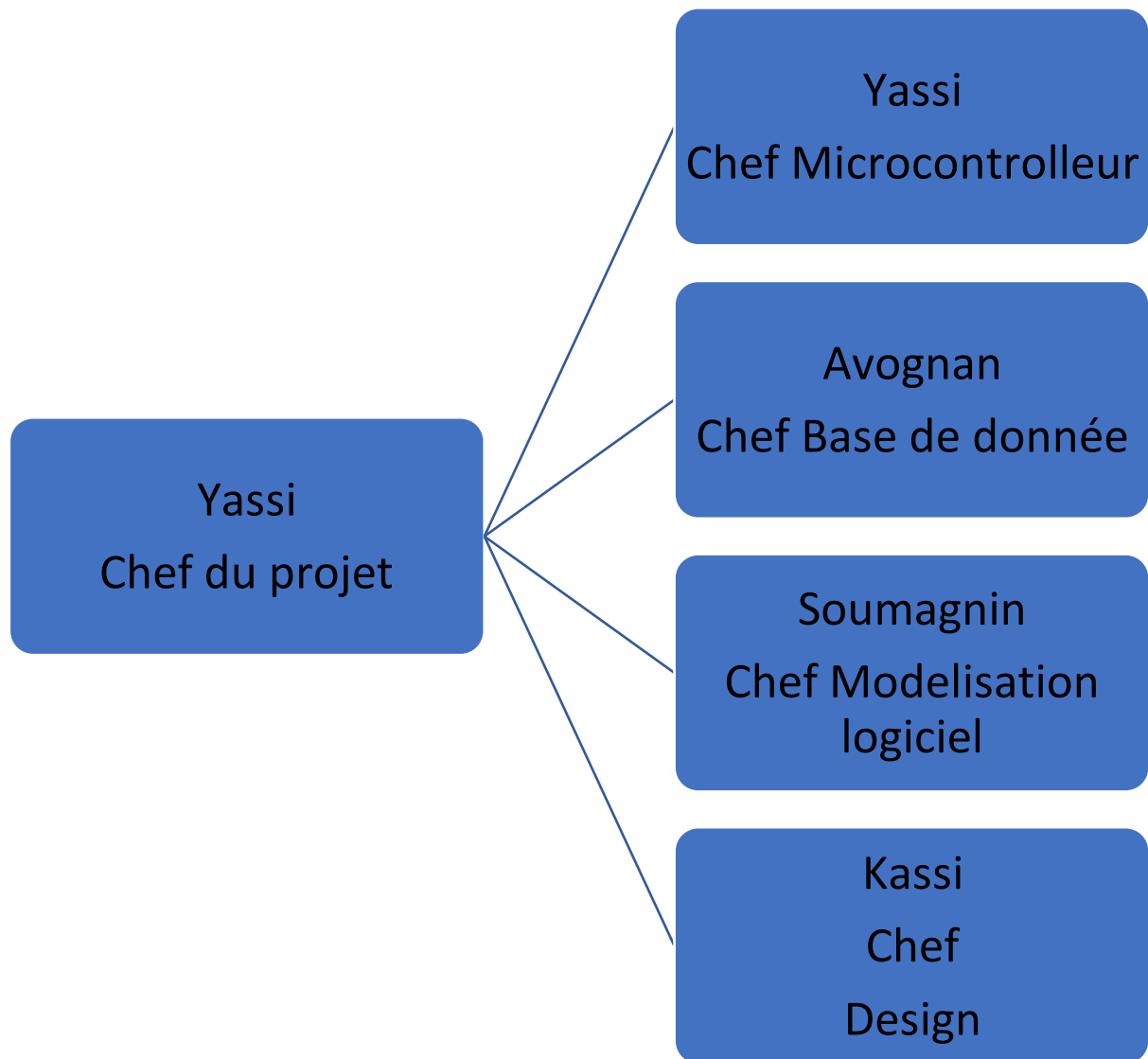
### 1) Pyramide des tâches

La réalisation du projet se fera à travers la réalisation de plusieurs sous tâches exposées ci-dessous, dans la pyramide des tâches de notre projet



### 2) Diagramme des responsabilités

Après avoir scindé notre projet en plusieurs sous-tâches, nous affectons ces différentes tâches à des sous-groupes gérées par des chefs de groupe.



## IV-REALISATION DU PROJET

### 1) Partie électronique

#### 1-1-Montage

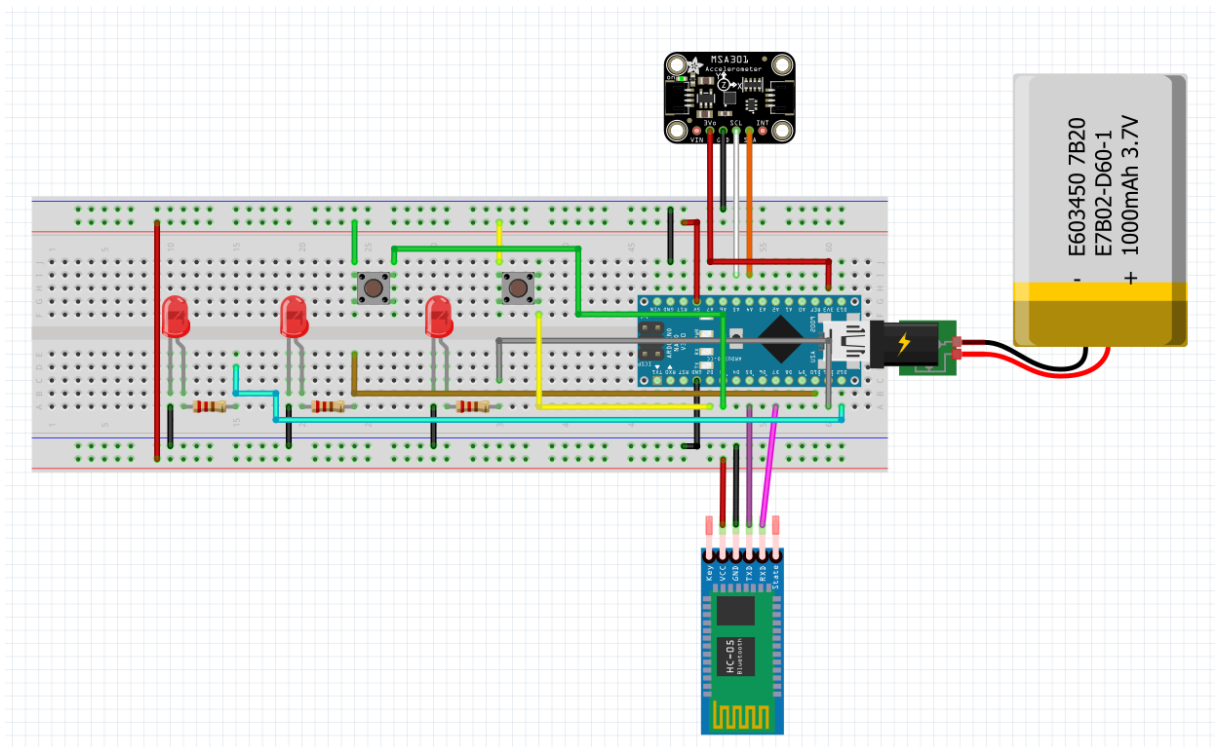


Figure 12 : montage

A l'appui du bouton du gauche de notre vélo, la Led de couleur rouge s'éteint comme le montre la figure ci dessous

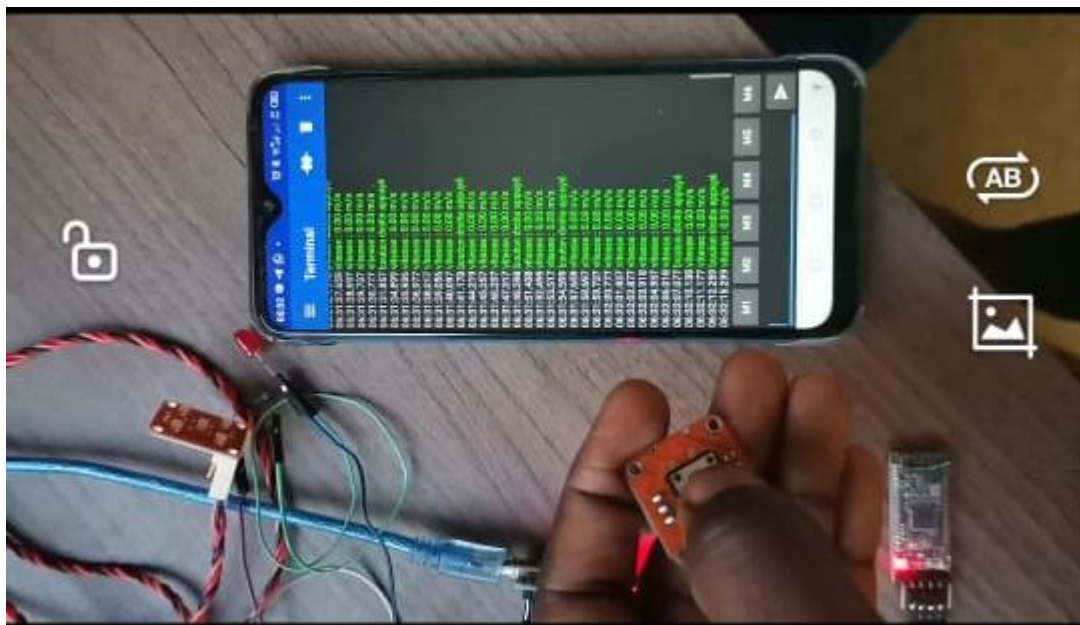


Figure 13: tet pour éteindre la led

A l'appui d'un bouton, nous pouvons accélérer comme décélérer, sur le téléphone nous voyons une variation de la vitesse selon qu'elle augmente lorsqu'on appuie sur le bouton pour accélérer et inversement. Par ailleurs lors d'un freinage, la détection est signalée au niveau de l'application mobile mais aussi la led rouge qui s'allume pour signaler aux autres usagers.

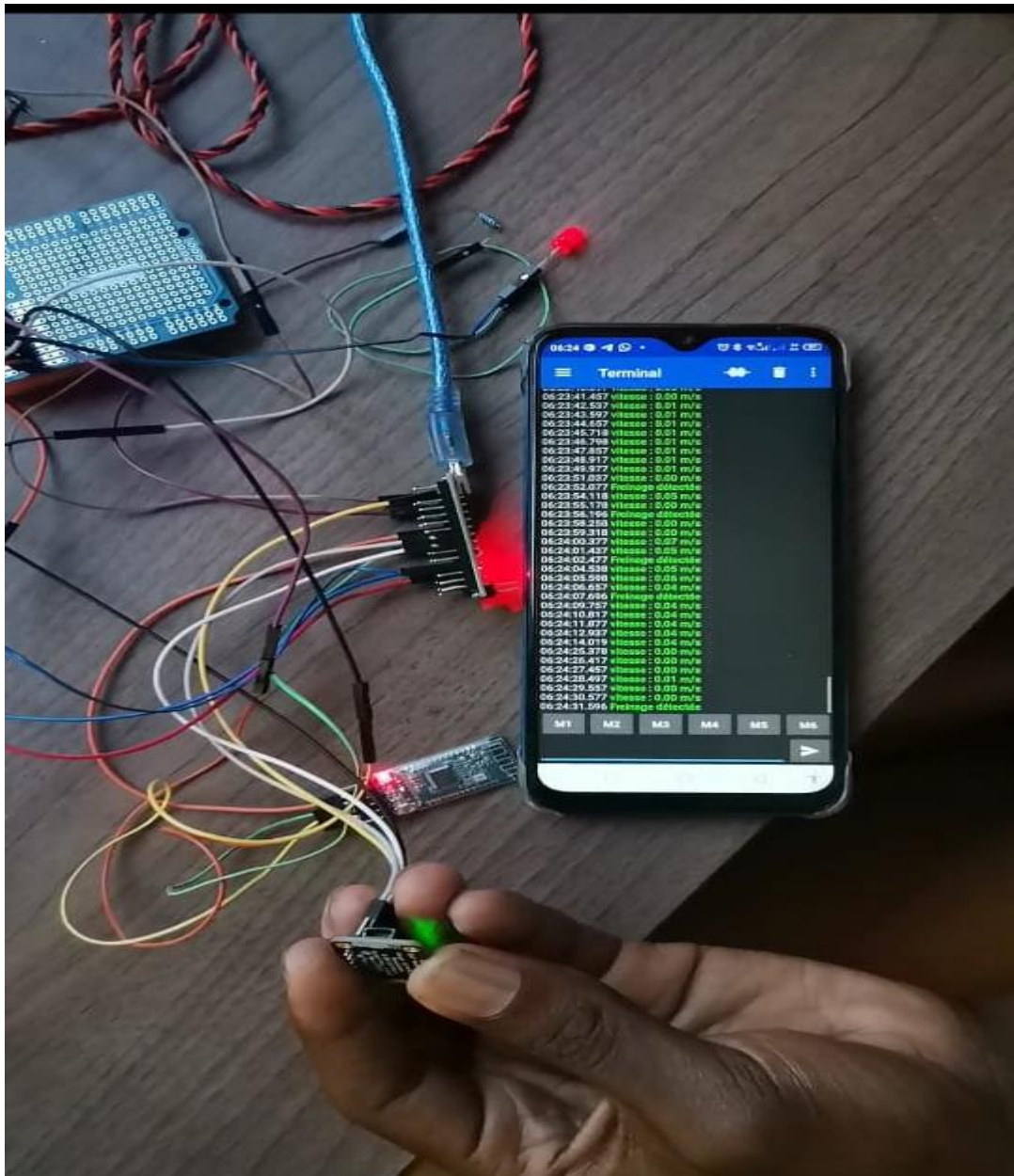


Figure 14 : detection de freinage



Pour tourner à gauche ou à droite, le cycliste se doit d'appuyer sur le bouton de droite, le leds blanche s'allume et alerte les usagers de ton action, aussi sur l'application mobile du cycliste apparaît de mouvement de celui-ci.



Figure 15 : test allumage de la led

## 1-2-Programmation

```
#include <Wire.h>
#include <Adafruit_MSA301.h>
#include <Adafruit_Sensor.h>
/* nous utilisons une liaison série virtuelle pour le module bluetooth */
#include <SoftwareSerial.h> // Software Serial Port
#define RxD 5                // Pin 5 pour la réception dans l'arduino
#define TxD 7                // Pin 7 pour l'émission de l'arduino
SoftwareSerial BTSerial(RxD, TxD);
// définition des broches 10 11 12 de la carte en tant que variable des led
const int led_gauche = 11;
const int led_droite = 12;
const int led_arriere = 10;
// port numérique lié au bouton poussoir
const int button_gauche = 3;
const int button_droite = 2;
Adafruit_MSA301 msa;
//SDA = 4;
//SCL = 5;
float V1 = 0;
float Ai = 0;
```

On définit dans cette partie chaque variables qui seront utilisées

Dans le code ainsi que les différentes broches à utiliser.



```

float vitesse(float vi, float An) {
    delay(10);
    return (An*0.01)+vi;
}

float getnorme(float Ax, float Ay, float Az) {
    return sqrt( (Ax*Ax) + (Ay*Ay) + (Az*Az) );
}

float sustract(float a, float b) {
    return (a-b);
}

void clignotant(int led) {
    for(int i = 0; i<10; i++) {
        digitalWrite(led, HIGH);
        delay(150);
        digitalWrite(led, LOW);
        delay(150);
    }
}

```

On définit les fonctions

- vitesse() : utilisée pour calculer la vitesse à partir de la vitesse initiale et de l'accélération
- getnorme(): calcule la norme d'une entité grâce à ces trois composantes
- sustract(): effectue la soustraction entre deux variables
- clignotant(): active le clignotement de la led indiquée

```

void setup() {
  // initialisation de la liaison série du module à la vitesse de 9600 bauds
  Serial.begin(115200);
  while (!Serial) delay(10);    // will pause Zero, Leonardo, etc until serial console opens

  // Configuration du module bluetooth à 9600 bauds, la vitesse du module par défaut
  pinMode(RxD, INPUT);
  pinMode(TxD, OUTPUT);
  BTSerial.begin(9600);

  // réglage du port des boutons en mode ENTREE
  pinMode(button_gauche, INPUT_PULLUP);
  pinMode(button_droite, INPUT_PULLUP);
  // configuration des broches des leds en sortie
  pinMode(led_gauche, OUTPUT);
  pinMode(led_droite, OUTPUT);
  pinMode(led_arriere, OUTPUT);
  // éteindre toutes les leds
  digitalWrite(led_gauche, LOW);
  digitalWrite(led_droite, LOW);
  digitalWrite(led_arriere, LOW);
}

```

## Configuration des broches en entrée et sortie des broches

```

// Initialisation!
if (! msa.begin()) {
  Serial.println("Failed to find MSA301 chip");
  while (1) { delay(10); }
}
Serial.println("MSA301 Found!");

//msa.setDataRate(MSA301_DATARATE_31_25_HZ);
Serial.print("Data rate set to: ");
switch (msa.getDataRate()) {
  case MSA301_DATARATE_1_HZ: Serial.println("1 Hz"); break;
  case MSA301_DATARATE_1_95_HZ: Serial.println("1.95 Hz"); break;
  case MSA301_DATARATE_3_9_HZ: Serial.println("3.9 Hz"); break;
  case MSA301_DATARATE_7_81_HZ: Serial.println("7.81 Hz"); break;
  case MSA301_DATARATE_15_63_HZ: Serial.println("15.63 Hz"); break;
  case MSA301_DATARATE_31_25_HZ: Serial.println("31.25 Hz"); break;
  case MSA301_DATARATE_62_5_HZ: Serial.println("62.5 Hz"); break;
  case MSA301_DATARATE_125_HZ: Serial.println("125 Hz"); break;
  case MSA301_DATARATE_250_HZ: Serial.println("250 Hz"); break;
  case MSA301_DATARATE_500_HZ: Serial.println("500 Hz"); break;
  case MSA301_DATARATE_1000_HZ: Serial.println("1000 Hz"); break;
}

```

```

//msa.setPowerMode(MSA301_SUSPENDMODE);
Serial.print("Power mode set to: ");
switch (msa.getPowerMode()) {
    case MSA301_NORMALMODE: Serial.println("Normal"); break;
    case MSA301_LOWPOWERMODE: Serial.println("Low Power"); break;
    case MSA301_SUSPENDMODE: Serial.println("Suspend"); break;
}

//msa.setBandwidth(MSA301_BANDWIDTH_31_25_HZ);
Serial.print("Bandwidth set to: ");
switch (msa.getBandwidth()) {
    case MSA301_BANDWIDTH_1_95_HZ: Serial.println("1.95 Hz"); break;
    case MSA301_BANDWIDTH_3_9_HZ: Serial.println("3.9 Hz"); break;
    case MSA301_BANDWIDTH_7_81_HZ: Serial.println("7.81 Hz"); break;
    case MSA301_BANDWIDTH_15_63_HZ: Serial.println("15.63 Hz"); break;
    case MSA301_BANDWIDTH_31_25_HZ: Serial.println("31.25 Hz"); break;
    case MSA301_BANDWIDTH_62_5_HZ: Serial.println("62.5 Hz"); break;
    case MSA301_BANDWIDTH_125_HZ: Serial.println("125 Hz"); break;
    case MSA301_BANDWIDTH_250_HZ: Serial.println("250 Hz"); break;
    case MSA301_BANDWIDTH_500_HZ: Serial.println("500 Hz"); break;
}

//msa.setRange(MSA301_RANGE_2_G);
Serial.print("Range set to: ");
switch (msa.getRange()) {
    case MSA301_RANGE_2_G: Serial.println("+2G"); break;
    case MSA301_RANGE_4_G: Serial.println("+4G"); break;
    case MSA301_RANGE_8_G: Serial.println("+8G"); break;
    case MSA301_RANGE_16_G: Serial.println("+16G"); break;
}

//msa.setResolution(MSA301_RESOLUTION_14 );
Serial.print("Resolution set to: ");
switch (msa.getResolution()) {
    case MSA301_RESOLUTION_14: Serial.println("14 bits"); break;
    case MSA301_RESOLUTION_12: Serial.println("12 bits"); break;
    case MSA301_RESOLUTION_10: Serial.println("10 bits"); break;
    case MSA301_RESOLUTION_8: Serial.println("8 bits"); break;
}

sensors_event_t event;
msa.getEvent(&event);
Ai = getnorme(event.acceleration.x,event.acceleration.y,event.acceleration.z);//accélération au démarrage
V1 = vitesse(0,0);//vitesse initiale
//X1=0;
}

```

Détection de l'accéléromètre ,affichage de ses configurations et initialisation de la vitesse et de l'accélération

```

void loop() {
//Partie accéléromètre
msa.read();          //get X Y and Z data at once
/* Or....get a new sensor event, normalized */
sensors_event_t event;
msa.getEvent(&event);

float Av = getnorme(event.acceleration.x,event.acceleration.y,event.acceleration.z);
V1 = max(0,vitesse(V1,sustract(Av,Ai)));//estimation de la vitesse
Serial.print(" \tAcceleration: "); Serial.print(sustract(Av,Ai));Serial.print(" \tm/s^2 ");//affichage accélération
Serial.print(" \tVitesse: "); Serial.print(V1);Serial.print(" \tm/s ");//affichage vitesse
Serial.println();|
BTSerie.print("vitesse : ");BTSerie.print(V1);BTSerie.println(" m/s");
delay(1000);
if(sustract(Av,Ai)<~-0.5){
    BTSerie.println("Freinage détectée");
    digitalWrite(led_arriere, true);
    delay(2000);
}
digitalWrite(led_arriere, false);
}

```

Allumage de la led arriere lorsqu'un freinage est détecté. Cette détection est effectuée suite à un changement brusque d'accélération

```

//Partie bluetooth
char recvChar;
//On lit caractere par caractere sur le BTSerie et on affiche sur le Terminal Serie
if (BTSerie.available()) {
    Serial.println("blue connect");
    recvChar = BTSerie.read();
    Serial.write(recvChar);          //cette ligne n'est pas indispensable,
                                     //mais elle peut aider au debugage pour
                                     //vérifier que le module Android envoie correctement les caractères 0 et 1

// teste du caractère reçu
if (recvChar=='1'){                // s'il est à 1 on allume/éteint la led de gauche
    clignotant(led_gauche);
}
if (recvChar=='0'){                // s'il est a 0 on allume/éteint la led de droite
    clignotant(led_droite);
}
}
}

```

Clignotement des indicateurs suivant le caractère envoyé

```

//Partie bouton poussoir
if (digitalRead(button_gauche) == HIGH) { // si le bouton est pressé ...
    delay(50);
    BTSerie.println("bouton gauche appuyé");
    Serial.println("bouton gauche appuyé");
    clignotant(led_gauche);
}
if (digitalRead(button_droite) == HIGH) { // si le bouton est pressé ...
    delay(50);
    BTSerie.println("bouton droite appuyé");
    Serial.println("bouton droite appuyé");
    clignotant(led_droite);
}
}
}

```

L'appui d'un bouton poussoir déclenche le clignotement de la led correspondante.

## 2) Partie informatique

### a - langages et base de données

Nous avons utilisé le Framework flutter pour réaliser notre application mobile. Le langage Dart est celui qui nous a permis de tout implémenter, allant du front end au backend de l'application.

Dans notre application, nous avons eu besoin d'une solution open source de géolocalisation afin de pouvoir effectuer plusieurs fonctionnalités essentielles. Open Street Maps nous a permis de faire ce travail.

Pour utiliser les services de firebase ainsi que la connexion via Bluetooth de l'application et du système électronique, nous avons importé dans notre projet des dépendances telles que :

- \*cloud firestore

- \*firebase auth

- \*provider

- \*serial Bluetooth

Avec les dépendances citées, firebase authentication et firebase firestore ont été implémentés dans l'application.

Firebase Authentication permet de faire de l'authentification pour que seul l'utilisateur concerné puisse avoir à son compte, ainsi qu'aux parcours qu'il a déjà réalisés. Pour cela, on utilise juste un email et un mot de passe qui sont sauvegardés sur firebase de cette manière :

Recherchez par adresse e-mail, numéro de téléphone ou ID utilisateur					
Ajouter un utilisateur					
Identifiant	Fournisseurs	Date de création	↓	Dernière connexion	UID utilisateur
akouaa@gmail.com	✉	24 mai 2022		24 mai 2022	yajd9IWNQYZKKu4CIIntu549B4c2
akoua@gmail.com	✉	24 mai 2022		24 mai 2022	2qLu43dWKW0dfg2A7mqDCFba...
wilfried@gmail.com	✉	24 mai 2022		24 mai 2022	JttegnTYyVxK3Ac1IPYDQaewHc12
dzjdj@gmail.com	✉	24 mai 2022		24 mai 2022	5Axq5xRhSOVwzz4xGhzNyL7ey73
yassifrederic@gail.com	✉	23 mai 2022		23 mai 2022	Zx2ANqPTu2T50BuolbvkmbI7Af13
wilfriedsoumagnin@gmail...	✉	23 mai 2022		24 mai 2022	QybJSFCHFGS5zYWNmuNwI4Te1...
zfcsc@hy.lmo	✉	22 mai 2022		22 mai 2022	55i5aiwPf3X7a1BRr4ZAIdGgl9s2
tred@gmail.com	✉	22 mai 2022		22 mai 2022	mU4M6JkFSRavxpV5kpBKkKrxMa...
yassifrederic@gmail.com	✉	20 mai 2022		24 mai 2022	U5YH91ErhRNIHMBLSg4EiojQ7Ez1
yushihnncloud@gjj.com	✉	2 mai 2022		2 mai 2022	4g2KFhacZ7O0fZlhkbThHowxE8q1
zfsc@gzfz.nl	✉	2 mai 2022		2 mai 2022	48pii33QMiaPEHar7NH8yu1iyx2
vjbuni@dhh.pp	✉	2 mai 2022		2 mai 2022	FEyxlGuduOJm39uc5KK1mMhu...
tontin@gmail.com	✉	8 janv. 2022		8 janv. 2022	96atda4tmrZrhPecWuppzMZhu0j2
Lignes par page : 50 1 - 13 of 13					

Figure 16 : firebase authentication

En ce qui concerne firebase firestore, nous avons utilisé deux collections, une qui serait réservée à toutes les informations personnelles de l'utilisateur et pour stocker les parcours.

Pour la collection user, les id des documents correspondent aux id des users au niveau de l'authentification, ce qui permet d'identifier toutes les informations appartenant à un utilisateur facilement. La seule information que nous avons jugé utile pour le moment est le pseudo ou nom de l'utilisateur.

🏠 > users > yajd9IWNQYZK...		
flutterfirebase-89cbb	users	yajd9IWNQYZKKu4CIIntu549B4c2
+ Commencer une collection	+ Ajouter un document	+ Commencer une collection
parcours	mQbHznWnLk0s64dDtZcXg0ZH5Fg1	+ Ajouter un champ
users >	yajd9IWNQYZKKu4CIIntu549B4c2 >	name : "akoua"

Figure 17: firebase firestore collection users

Pour la collection parcours, les id des documents de manière automatique car il n'y pas besoin d'un id en particulier. A chaque parcours, nous avons défini des attributs tels que précisés

dans l'image ci-dessous permettant de caractériser les parcours et de savoir à quel utilisateur elles appartiennent.

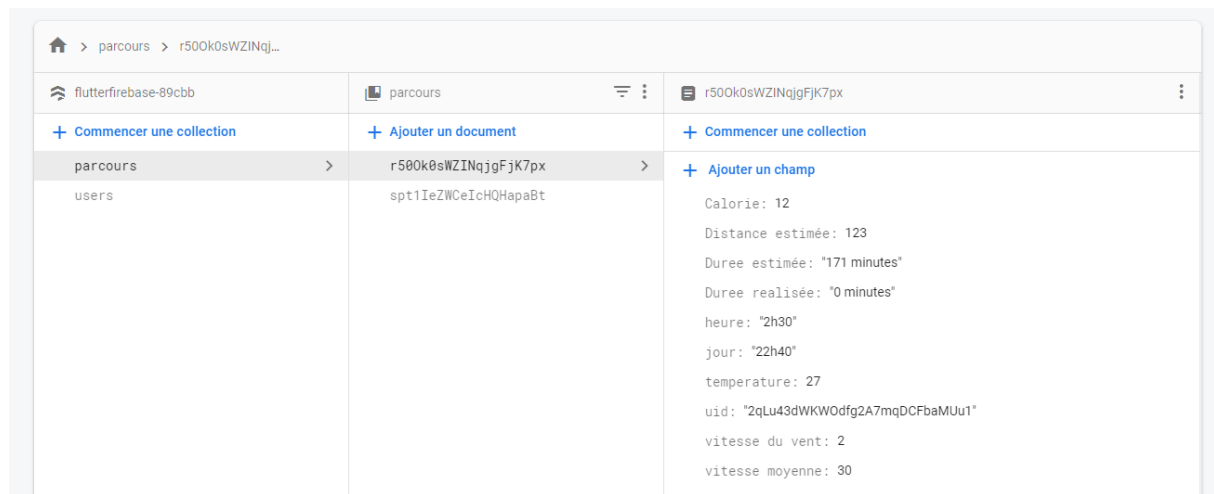


Figure 18 : firebase firestore collection parcours

## b - Interfaces

La page d'authentification (figure) est la page qui s'affiche au démarrage de notre application. Elle comporte un champ de saisie pour le login de l'utilisateur et un champ pour son mot de passe. Vous validez l'authentification en cliquant sur le bouton connexion (bouton orange sur la page).

Si l'authentification réussie on accède alors à la page suivante avec le menu principale (figure)

NB : Il est également possible de créer un nouveau compte grâce à l'option 'SIGNUP' présent sur la figure

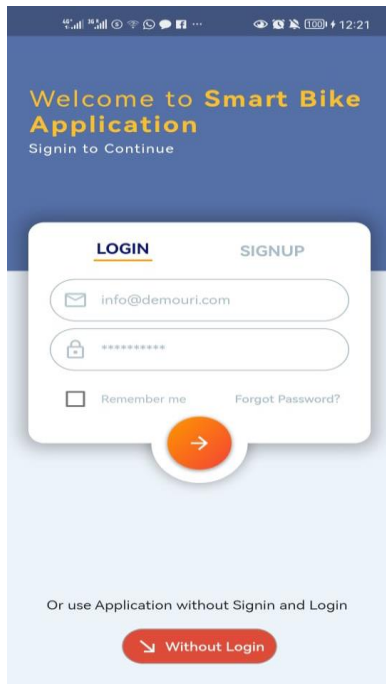


Figure 19: page d'accueil

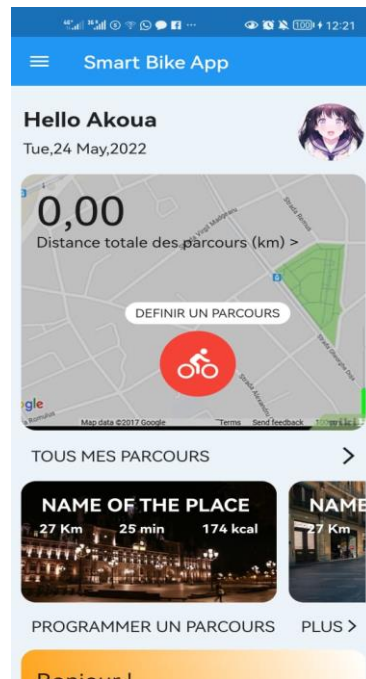


Figure 20: page d'authentification

Une fois dans le menu principal, il est possible de définir un parcours en appuyant sur le bouton avec l'icône du vélo puis de rechercher le nom de lieu où l'on veut aller, le chemin le plus long se trace alors sur notre map (figure).

Une fois que le parcours est terminé, certains détails sur le parcours tel que la distance, la durée, la vitesse moyenne et l'énergie dépensée en calorie nous est proposé par l'application (figure).



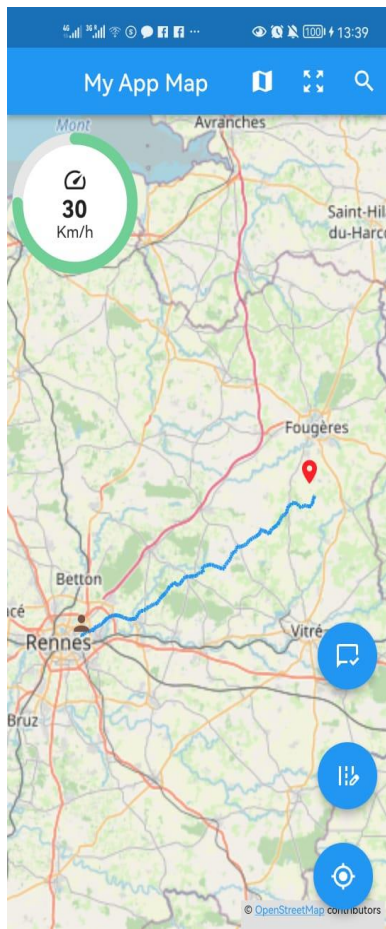


Figure 21: page de la map

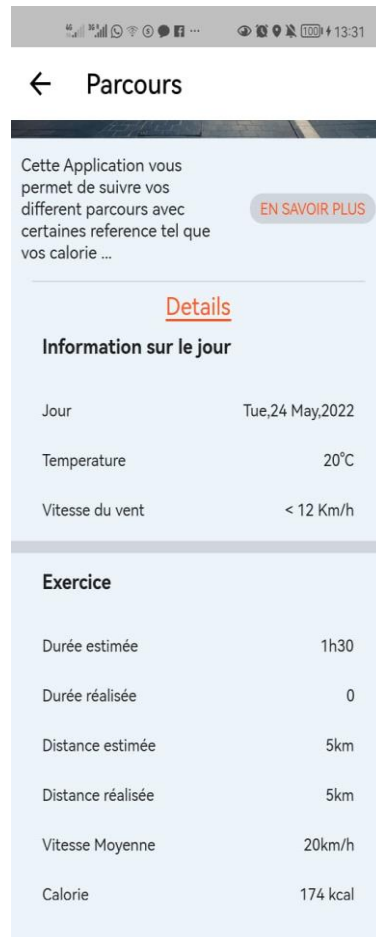


Figure 22 : détail sur le parcours

### c - Fonctionnalités réalisées

Au terme de notre travail, les différentes fonctionnalités que nous avons pu réaliser sont :

- La détection automatique d'un freinage et allumage de la led de frein ;
- Allumage d'une Led lorsqu'on appuie l'un des deux boutons (bouton gauche ou droite) afin d'indiquer qu'on tourne ;
- Authentification d'un utilisateur ;
- Création d'un nouveau compte ;
- Définir un parcours par nom du lieu ou en cliquant sur la map ;
- Calcul de la distance du parcours ;
- Estimation de la durée du parcours ;
- Calcul de la durée réalisée sur le parcours ;
- Voir l'historique des parcours ;
- Se déconnecter de l'application

## IV-DIFFICULTES RENCONTRES ET PERSPECTIVES ENVISAGEES

Tout au long de ce projet, nous avons rencontré des difficultés techniques et managériales.

Au niveau des difficultés techniques, d'abord nous nous sommes efforcé d'utiliser Google Maps pour notre solution de géolocalisation, qui ne nous permet pas d'effectuer les fonctionnalités que nous désirons à travers sa version gratuite et le nombre de requêtes est limité.

La solution alternative à Google Maps qui est OpenStreetMaps, elle aussi ne nous permet pas d'implémenter le guide vocal et d'avoir la vitesse instantanée.

Le module GPS avait une portée faible il fallait se mettre en extérieur pour pouvoir recevoir les données

Avec L'accéléromètre, il n'est possible d'obtenir la vitesse instantanée automatiquement, nous devons nous même passé par des calculs de physique pour la déduire en prenant soin de retirer l'accélération du au champ gravitationnelle terrestre.

Le module ESP32 qui nous a été donné avait des problèmes matériels , il ne s'allume pas lorsqu'on le connecte ,donc nous avons dû passer à un Arduino Nano comme microcontrôleur en plus du module Bluetooth hc 05. Ensuite le module Bluetooth n'arrive pas à se connecter directement au téléphone à cause de son mot de passe de 6 chiffres.

## CONCLUSION

Le projet sur lequel nous avons travaillé au cours du semestre 7 et semestre 8 consiste à mettre en place un système électronique et informatique afin de contrôler un vélo à partir d'une application mobile et récupérer de l'historique des trajets parcourus. Pour ce faire, nous avons d'abord procédé à la description contextuelle du projet ; cela nous a permis de mieux comprendre le fonctionnement du système à mettre en place et des besoins des utilisateurs finaux. Ensuite nous avons fourni grâce à l'étude des différentes technologies, des langages et outils permettant de concevoir le système global.

Par ailleurs le projet a été d'un grand apport non seulement à notre formation personnelle en matière de système électronique et informatique mais aussi à notre formation de gestion de projet. En outre, il a été le bienvenu car il nous a permis d'approfondir nos connaissances Dart et en firebase, les systèmes de détection avec des capteurs de vitesse, le module GPS etc.

Finalement, nous nous réjouissons de l'état d'avancement et d'avoir fait une bonne analyse pour faciliter la tâche aux concepteurs qui aimeraient mettre en place une telle innovation sur les vélos existants. Nous pensons qu'un tel système faciliterait la circulation et minimiserait ses différents risques au quotidien.

## WEBOGRAPHIE

1. <https://leblogduscooter.fr/gofaststaycool-smart-bike-angell/>
2. <http://igm.univ-mlv.fr/~dr/XPOSE2012/dart/fonctions.html>
3. <https://www.youtube.com/watch?v=hRco5Ufp9m4>
4. <https://phmarduino.wordpress.com/2020/07/11/comment-ajouter-rapidement-un-gps-a-un-montage/>
5. <https://cloud.google.com/firestore/docs/client/get-firebase?hl=fr>
6. <https://firebase.google.com/docs/auth/web/start>
7. <https://docs.arduino.cc/hardware/nano>
8. <https://dart.dev/tools>
9. <https://docs.microsoft.com/fr-fr/cpp/c-language/source-files-and-source-programs?view=msvc-170>
10. <https://docs.flutter.dev>

## BIBLIOGRAPHIE

Ancien rapport de projet SMART BIKE (2020\_2021) de l'équipe KILLIAN GIRAUD, HUGO MASQUILIER et FRANÇOIS DESHAYES en IOT 2 à l'ESIR.