



École Polytechnique de l'Université de Tours  
64, Avenue Jean Portalis  
37200 TOURS, FRANCE  
Tél. +33 (0)2 47 36 14 14  
[www.polytech.univ-tours.fr](http://www.polytech.univ-tours.fr)

## Département Informatique

### Cahier de spécification système & plan de développement

Projet :	Roule ma poule sans nid de poule		
Emetteur :	B. Bouchardon	Coordonnées : EPU-DII	
Date d'émission :	11 novembre 2018		
Validation			
Nom	Date	Valide (O/N)	Commentaires

**N MONMARCHE** : 08/11/2018 ; OK ; pour 2018-2019

**E NERON** :

### Historique des modifications

Version	Date	Description de la modification
---------	------	--------------------------------

**00** : 07/11/2018 ; Version initiale non achevée, vérification par l'encadrant

**01** : 11/11/2009 ; Version finale



# Table des matières

---

<b>Cahier de spécification système</b>	<b>5</b>
1.1 Introduction . . . . .	5
1.2 Contexte de la réalisation . . . . .	5
1.2.1 Contexte . . . . .	5
1.2.2 Objectifs . . . . .	5
1.2.3 Hypothèses . . . . .	5
1.3 Description générale . . . . .	6
1.3.1 Environnement du projet . . . . .	6
1.3.2 Caractéristiques des utilisateurs . . . . .	6
1.3.3 Fonctionnalités et structure générale du système . . . . .	7
1.3.4 Contraintes de développement, d'exploitation et de maintenance . . . . .	8
1.4 Description des interfaces externes du logiciel . . . . .	9
1.4.1 Interfaces matériel/logiciel . . . . .	9
1.4.2 Interfaces homme/machine . . . . .	9
1.4.3 Interfaces logiciel/logiciel . . . . .	10
1.5 Architecture générale du système . . . . .	10
1.6 Description des fonctionnalités . . . . .	12
1.6.1 Définition de la fonction Traitement de données (boitier) . . . . .	12
1.6.2 Définition de la fonction communication avec le serveur (depuis le boitier) . . . . .	12
1.6.3 Définition de la fonction visualisation des informations . . . . .	14
1.6.4 Définition de la fonction configuration du système . . . . .	15
1.6.5 Définition de la fonction gestion de l'alimentation et de l'IHM boitier . . . . .	16
1.7 Conditions de fonctionnement . . . . .	17
1.7.1 Performances . . . . .	17
1.7.2 Capacités . . . . .	17
1.7.3 Modes de fonctionnement . . . . .	17
 <b>Plan de développement</b>	 <b>19</b>
2.1 Découpage du projet en tâches . . . . .	19
2.1.1 Tâche Cahier des charges . . . . .	20
2.1.2 Tâche Cahier d'analyse . . . . .	21
2.1.3 Tâche PHASE 1 . . . . .	22
2.1.4 Tâche Études . . . . .	23
2.1.5 Tâche Conceptions . . . . .	24
2.1.6 Tâche Programmations . . . . .	25
2.1.7 Tâche Documentations . . . . .	26
2.2 Planning . . . . .	27

# Cahier de spécification système

## 1.1 Introduction

Ce document spécifie les points techniques et de gestion du projet "roule ma poule sans nid de poule" dirigé par Emmanuel Neron et Nicolas Monmarche. Bastian Bouchardon est le rédacteur de ce document.

## 1.2 Contexte de la réalisation

### 1.2.1 Contexte

Aujourd'hui, de nombreux moyens de déplacement existent pour faire les trajets quotidiens comme se rendre au travail, aller faire des courses, se déplacer à des événements culturels. Il y a les transports en commun comme le bus, le tramway ou le covoiturage et des transports personnels comme la voiture, le skateboard, les appareils électriques type mono wheel, hoverboard et le vélo. Ce dernier est de plus en plus utilisé pour les trajets quotidiens car il permet de garder une bonne forme physique, il est plus rapide que la voiture en ville, peu coûteux et facile d'entretien. Les cyclistes sont sensibles à l'état de la route, en effet, les routes abîmées, écorchées peuvent au mieux endommager, au pire les blesser. En France, d'après plusieurs études, les routes se dégradent, des motards sont décédés à cause de nids de poule, des accidents ont souvent lieu impliquant des cyclistes et des défauts de la route. C'est pour ces raisons que ce projet peut être apprécié des utilisateurs de vélo.

### 1.2.2 Objectifs

Il s'agit de réaliser un système embarqué sur tout type de vélo permettant de capter les vibrations dues à la route et d'envoyer ces informations avec une position GPS sur un serveur. Cela permettra de cartographier l'état des routes et de prévenir les usagers et collectivités. Ce système sera constitué, au final, d'une carte électronique contenant l'intelligence, le capteur, le récepteur GPS, l'alimentation et la communication sans fil. Au départ, il faudra tester et choisir une technologie de capteur qui correspondra à notre utilisation. Il faudra également créer un serveur pour recevoir ces informations.

### 1.2.3 Hypothèses

L'une des contraintes du système est la compatibilité avec tous les types de vélos. En effet, il y a les vélos de route ou de courses, sans suspensions, des roues et pneus très fins et très durs (entre 7 et 15 bars) donc un vélo très sensible aux vibrations et à l'opposé, les VTT (vélo tout-terrain), avec de pneus larges, des fourches avec suspensions non sensibles aux vibrations. Il faut que ce capteur puisse fonctionner sur un vélo de route comme sur un VTT. si le capteur ne peut pas fonctionner sur le cadre, alors il faudra trouver une autre position comme le moyeu d'une des roues mais si ce choix est validé, alors il faudra supprimer la connexion entre ce capteur et une éventuelle dynamo.

## 1.3 Description générale

### 1.3.1 Environnement du projet

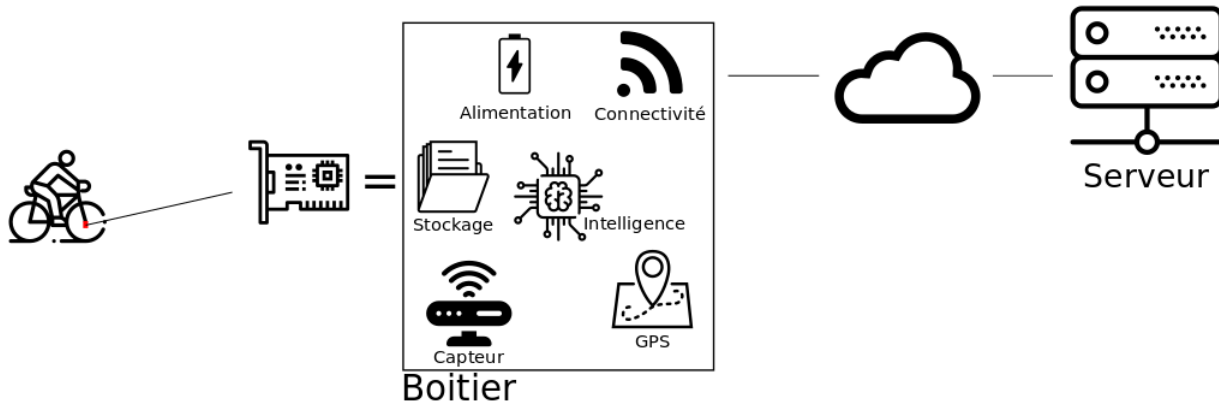


FIGURE 1.1 – Environnement du projet

Le projet sera sous la forme d'un boîtier placé sur un vélo, Il contiendra une carte électronique comprenant :

- Un système de communication sans fil type Wifi, bluetooth ou sur un réseau LPWAN.<sup>1</sup> type LoRa, Sigfox.
- Un capteur de vibration qui sera défini lors du premier livrable. il pourra être de type piezzo-électrique, résistif capacitif, ...
- Un système de positionnement tel que le GPS permettant de connaître la position du véhicule.
- Une intelligence permettant de récupérer une valeur correspondant à la vibration, l'interpréter, récupérer la position GPS en interprétant les trames, mettre en forme ces données et envoyer ces informations sur le serveur.
- Un stockage des données si il n'y a pas de connexion permanente avec le serveur.
- Un module d'alimentation électrique avec une source d'énergie : une batterie et un système de conversion pour correspondre aux différents modules ci-dessus.

Le système sera le moins énergivore possible, il devra activer les systèmes de communication et de réception de la position seulement lorsque cela est nécessaire. Il pourra être connecté à un serveur externe, comme celui d'une société. Il y aura également une interaction entre le client et le boîtier par le biais d'une application ou d'un site internet lui permettant de visualiser ses informations comme ses parcours, son profil, configurer le boîtier en matière d'identité (anonymer les informations le concernant ou non, la récurrence d'envoi de ces informations sur le serveur, ...).

### 1.3.2 Caractéristiques des utilisateurs

Le capteur connecté sera utilisé par tous types de cyclistes. Des cyclistes quotidiens, sportifs, cyclistes occasionnels, ... Le système devra donc être le plus facile d'accès possible, pour tous les âges et toutes les utilisations. Éviter au maximum les paramétrages du système et se poser la question quant à la nécessité de posséder un smartphone sur soi ou non.

1. Low Power Wide Area Network

### 1.3.3 Fonctionnalités et structure générale du système

Voici un diagramme des principaux cas d'utilisations du système, du point de vue de l'utilisateur. Ce système est la plupart du temps autonome, l'utilisateur joue un rôle mineur dans le fonctionnement. On y retrouve les quatre principales fonctionnalités :

- Le traitement des données provenant du capteur de vibration et du GPS.
- La configuration du système par l'utilisateur. La récurrence d'envoi des données, la connexion, ...
- La connexion du système au serveur.
- La visualisation des parcours et de l'état de la route.

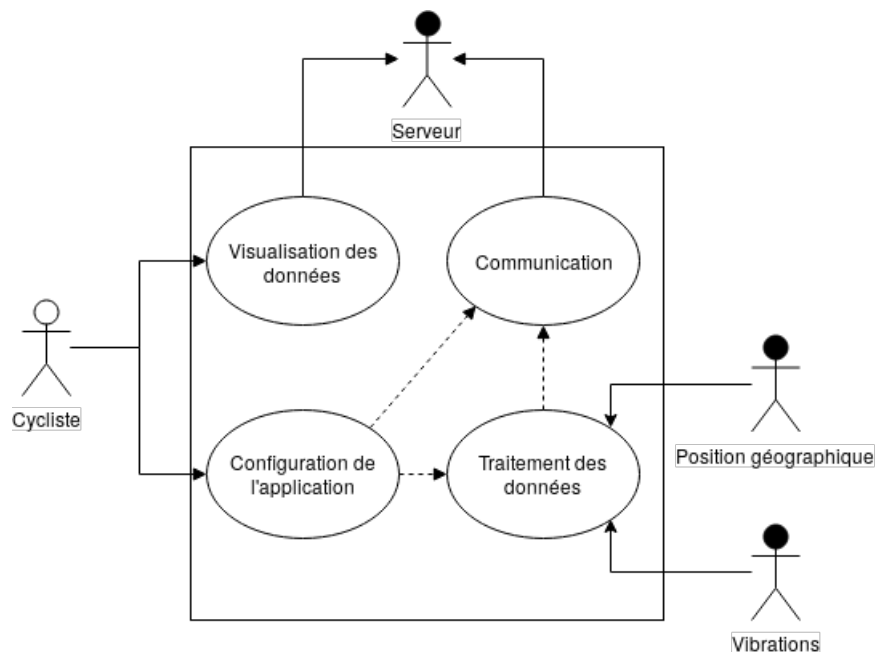


FIGURE 1.2 – diagramme de cas d'utilisation général

Voici une description des cas d'utilisation :

Cas d'utilisation	Traitement par le système
<b>Description</b>	Réception et traitement des informations du capteur et du GPS puis mise en forme des données à envoyer
<b>Acteurs</b>	GPS et capteur de vibration
<b>Pré-conditions</b>	Avoir encore assez de stockage disponible si nous n'avons pas accès au serveur
<b>Post-conditions</b>	Données mises en forme
<b>Flux d'activité</b>	1 - Le système récupère la vibration 2 - Il interprète ces vibrations 3 - Il récupère la position GPS 4 - Mise en forme des données GPS et des vibrations

Cas d'utilisation	Communication
Description	Envoi des données sur le serveur
Acteur	Serveur
Pré-conditions	Données correctement mises en forme et accès au serveur
Post-conditions	Bonne réception des données par le serveur
Flux d'activité	1 - Réception des données par le système de traitement 2 - Connexion au serveur 3 - Envoi des données au serveur 4 - Déconnexion du serveur

Cas d'utilisation	Visualisation des données (parcours)
Description	Visualisation des données utilisateur stockées sur le serveur par le biais d'une application
Acteur	Utilisateur, Serveur
Pré-conditions	Accès au serveur
Post-conditions	Affichage des informations
Flux d'activité	1 - Demande des informations concernant l'utilisateur suivant des paramètres 2 - Réception de ces informations 3 - Affichage des informations sur l'application

Cas d'utilisation	Configuration du système
Description	Configuration du système par l'utilisateur sans fil
Acteur	Utilisateur
Pré-conditions	Utilisateur connecté au système
Post-conditions	Application de la configuration par le système
Flux d'activité	1 - Modification de certains paramètres 2 - Enregistrement des paramètres

### 1.3.4 Contraintes de développement, d'exploitation et de maintenance

#### Contraintes de développement

Comme le système entier est à développer, il n'y a pas de contraintes entre le matériel et le logiciel. Par contre le système doit avoir la meilleure autonomie possible. Cela passe par le choix des composants, le choix des systèmes de communication, la programmation embarquée.

Il faut également que le serveur, le système embarqué et le téléphone du cycliste puissent fonctionner en écosystème. Le serveur doit pouvoir enregistrer les informations provenant des vélos et puisse envoyer les informations demandées par les cyclistes. Il faudra également développer une application pour les cyclistes, pour qu'ils puissent configurer le système embarqué sur leur vélo et qu'ils puissent visualiser leurs informations. Cette application peut être développée sur une cible type Android ou iOS. Un site internet n'est pas pertinent car l'application permettra au système embarqué d'utiliser la connexion au réseau mobile du smartphone pour envoyer ses informations au serveur en passant par une connexion bluetooth si l'utilisateur en donne la permission.

Ce développement logiciel est secondaire, la priorité est sur le choix du capteur et la réalisation du système embarqué.

Il va falloir également mettre en place un simulateur pour tester et choisir un capteur de vibration répondant à notre besoin. Ce simulateur sera une route étalon, un ou plusieurs type de vélo étalons pour pouvoir mettre en évidence les différences et comparer les différentes technologies de capteurs.

Je n'ai pas d'algorithmes imposés, cependant il faut que mon algorithme de détection des nids-de-poule



puisse s'adapter au type de route, aux vibrations fréquentes, ... Je n'ai pas non plus de bibliothèques de programmes imposées ou de protocole de communication particulier.

Le projet, c'est-à-dire le choix du capteur, le système embarqué terminé et un prototype de serveur doit être terminé à la date de fin du projet de fin d'études.

### Contraintes d'exploitation

L'envoi des parcours des utilisateurs sur le serveur demandera une réflexion particulière, sur le format des trames, les informations présentes ainsi que le chiffrement de ces données ou non. En effet, cela peut représenter un problème si il y a des données personnelles présentes dans les trames. Par contre si il n'y a pas de moyen pour un éventuel hacker d'identifier les informations transmises par le système embarqué, un chiffrement des informations n'est éventuellement pas nécessaire.

## 1.4 Description des interfaces externes du logiciel

### 1.4.1 Interfaces matériel/logiciel

Le système nécessitera une connexion directe ou indirecte à internet pour transmettre des informations à un serveur.

Directe par une connexion LPWAN ou par une connexion Wi-Fi permettant avec un minimum de configuration au système d'envoyer ses informations.

Indirect par le biais du bluetooth, le système envoie les informations au téléphone qui se charge de les envoyer sur le serveur par le biais de l'application.

Le système (le boîtier sur le vélo et l'application sur téléphone) communiquera en TCP/IP avec le serveur sur internet.

Concernant le boîtier, le GPS sera connecté en UART sur l'élément programmable du système. En effet plusieurs modules GPS dans le commerce sont commandés par UART, c'est un système de communication très répandu et il est très fiable.

### 1.4.2 Interfaces homme/machine

**Application :** L'interface n'est pas encore définie mais il faudra au moins 3 pages :

- Une page de connexion permettant de connecter le boîtier au téléphone. En arrière-plan, l'application activerait le bluetooth.
- Une page de visualisation des parcours avec une carte, les parcours effectués ainsi que l'état des routes.
- Une page de configuration permettant au cycliste de configurer son boîtier, son nom d'utilisateur, ...

Une interface très simple, épurée avec de simples boutons type sliders à l'image du boîtier.

**Boîtier :** Le boîtier, suivant sa position sur le vélo, ne sera pas forcément accessible. Les interactions avec l'utilisateur doivent être minimales. J'ai donc choisi de n'avoir qu'un seul bouton sur le boîtier, de préférence sensitif pour que le boîtier reste étanche (interrupteur sensitif derrière une paroi du boîtier) permettant d'appairer le dispositif au téléphone du cycliste et de servir d'interrupteur marche / arrêt suivant la longueur ou le nombre d'appui. Cette fonctionnalité peut également être implémentée par un simple interrupteur mécanique placé sur le flanc du boîtier. Ajoutez à cela quelques voyants indiquant le niveau de batterie ou l'état du boîtier permettant de savoir dans quel état est le système (connecté, en phase d'appairage, problème de connexion, envoi de données, ...) et d'avoir un diagnostic rapide si le boîtier est défaillant. Il y aura également un connecteur placé sur le côté du boîtier permettant de recharger ce dernier. Voici une maquette de l'interface ci-dessous :

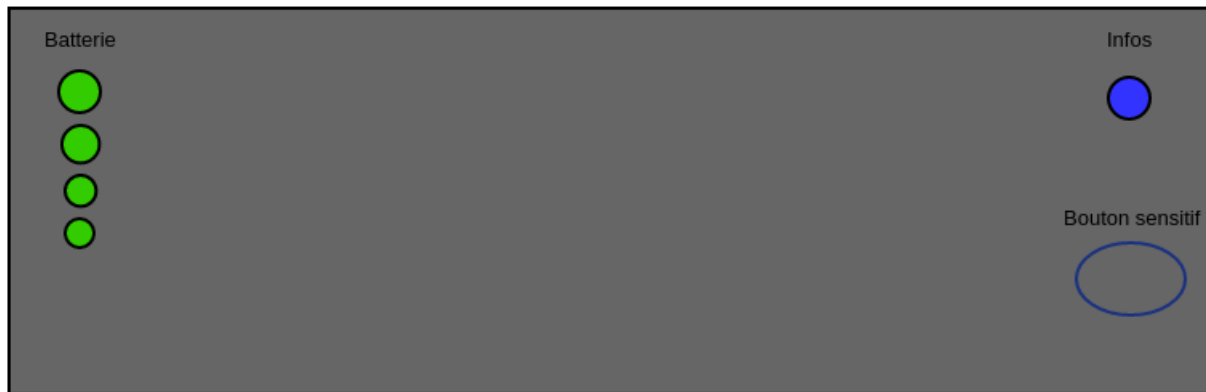


FIGURE 1.3 – design de l'IHM sur le boîtier

### 1.4.3 Interfaces logiciel/logiciel

Les interfaces entre boîtier/serveur et serveur/application seront les mêmes. Les données seront sous la forme de trames binaires simples permettant d'avoir de petite quantité de données à échanger et donc des envois et réceptions rapides sur le serveur. En effet, si la communication est difficile entre le boîtier et le serveur, cela permettra de s'affranchir d'un format type JSON qui rajoute des octets dont on peut s'affranchir dans notre cas. Les accès au serveur par le boîtier peuvent être un envoi après chaque parcours ou un envoi après plusieurs parcours. Des envois réguliers pendant un parcours ne sont pas nécessaires car nous ne pouvons pas accéder à ces données pendant notre trajet.

## 1.5 Architecture générale du système

Dans cette partie on retrouve les principaux composants/éléments du système ainsi que leurs relations.

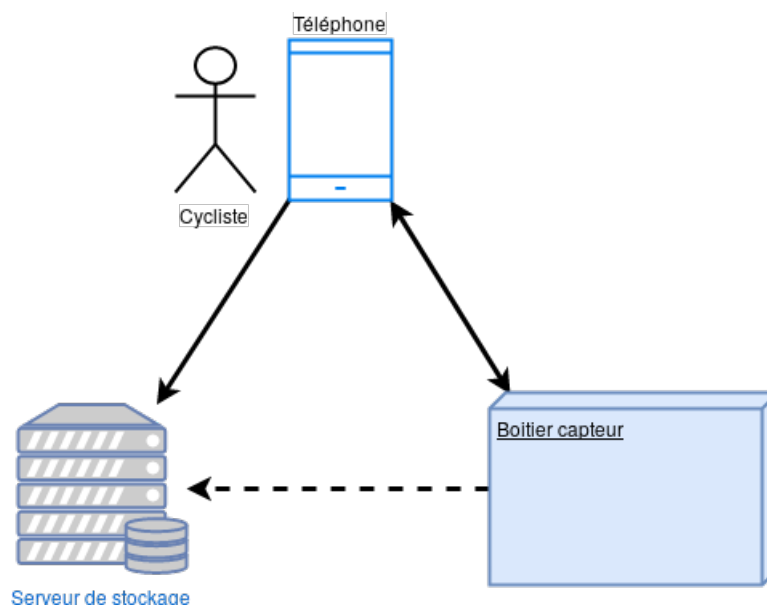


FIGURE 1.4 – Diagramme de blocs fonctionnels global

L'utilisateur par le biais de son téléphone, fera la passerelle entre le boîtier et le serveur et lui permettra d'interagir avec le boîtier en le configurant et en visualisant les informations concernant son/ses parcour(s). On pourra également imaginer que le boîtier puisse envoyer les informations directement au serveur sans passer par le téléphone du cycliste (route discontinue).

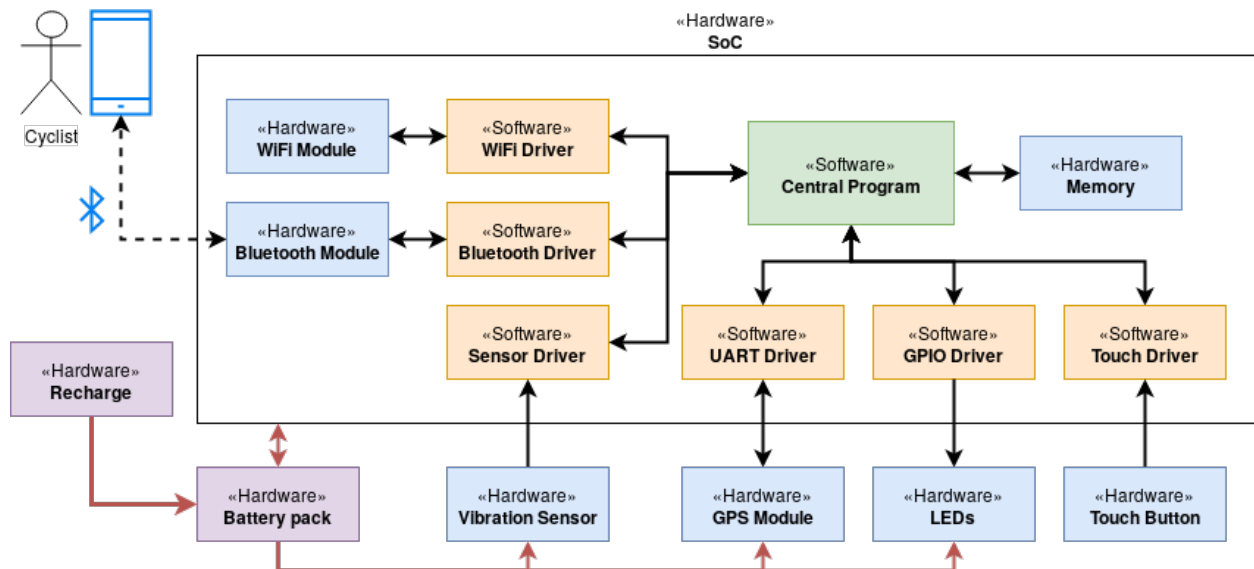


FIGURE 1.5 – Diagramme de blocs fonctionnels - Boîtier

Le boîtier sera constitué d'une intelligence, celle-ci aura des modules de communication bluetooth et Wi-Fi. Elle commandera les LEDs de visualisation pour l'utilisateur, elle recevra les informations du GPS, du capteur de vibration et du bouton sensitif. C'est le coeur du système. En plus de cela, elle recevra une valeur analogique image de la tension de la batterie permettant de signaler au cycliste le niveau de batterie restant. Il y aura également une mémoire présente dans le système pour enregistrer les données d'état de la route suivant les paramètres de l'utilisateur et de la connexion disponible.

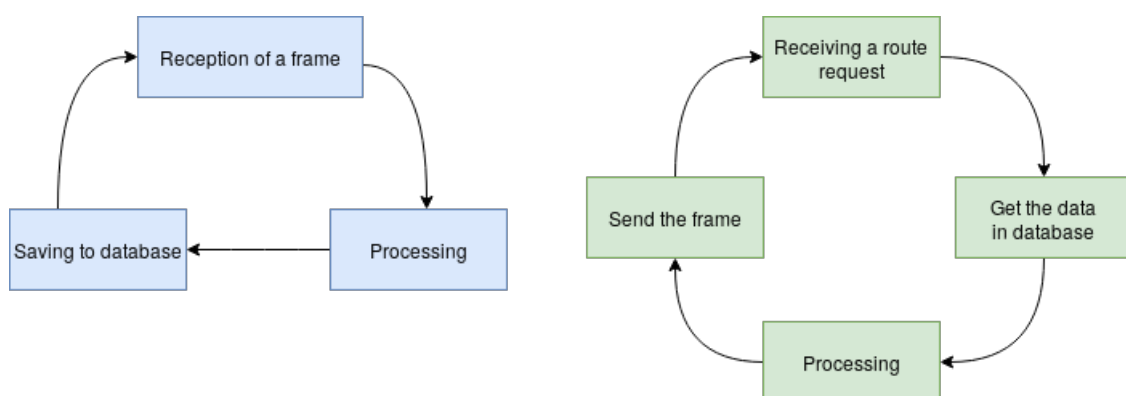


FIGURE 1.6 – Diagramme de blocs fonctionnels - Serveur

Sur la partie serveur, Deux modes coexistent, le mode réception de données où un boîtier/smartphone envoie des données au serveur sous forme de trames binaires, le serveur traite ces données et les enregistre dans une base de données. Deux processus seront exécutés sur le serveur, un processus recevra les informations et les transformera pour les enregistrer dans la base de données qui sera le second processus. Le second mode permettra une communication entre un client (smartphone) et le serveur. Lors d'une

demande d'un téléphone, le serveur récupérera les informations correspondant à cette demande dans la base de données et un second processus transformera ces données en trames binaires pour les envoyer à l'application qui les mettra en forme pour être visualisées par l'utilisateur.

Un dernier mode permettra à l'utilisateur d'envoyer ses informations le concernant permettant ainsi qu'il puisse voir ses trajets personnels.

## 1.6 Description des fonctionnalités

Dans cette partie sont développées les fonctionnalités du système. Nous rentrons plus en détail dans les fonctionnalités de la partie 1.3.3. Des détails sont apportés aux fonctionnalités suivantes :

- Traitement des données (boitier)
- Configuration du système (boitier)
- Visualisation des informations
- Communication avec le serveur (depuis le boitier)

Et d'autres fonctionnalités correspondant aux diagrammes des blocs fonctionnels 1.5 :

- Création d'un utilisateur dans le serveur
- Ajout d'un parcours sur le serveur
- Envoi des parcours d'un utilisateur depuis le serveur
- Gestion de l'alimentation et de l'IHM boitier

### 1.6.1 Définition de la fonction Traitement de données (boitier)

#### Identification de la fonction Traitement de données

Présenter la fonction :

- Traitement des données ;
- Cette fonction permet de récupérer les vibrations et la position GPS par intervalles réguliers et traite ces données ;
- Cette fonction est primordiale pour les livrables du projet. Une partie de cette fonction (acquisition) sera également mise en place lors des phases de tests du capteur de vibration nécessaire au premier livrable.

#### Description de la fonction Traitement de données

- Cette fonction est connectée au capteur de vibration et au capteur GPS. Le capteur de vibration est à choisir pour le premier livrable donc on ne connaît pas encore quel type d'entrée nous aurons mais elle sera de type, ADC, GPIO, I<sup>2</sup>C, SPI, ... Pour le GPS, l'entrée sera une entrée série UART. La sortie de cette fonction sera connectée à la fonction communication ;
- Cette fonction interagit avec le capteur de vibration et le GPS. Elle agira également avec la mémoire du microcontrôleur ou SoC choisi ;

### 1.6.2 Définition de la fonction communication avec le serveur (depuis le boitier)

#### Identification de la fonction communication avec le serveur

Présenter la fonction :

- Communication avec le serveur ;
- Cette fonction envoie les données au serveur par le biais du téléphone du cycliste ;
- Cette fonctionnalité est primordiale pour tester l'envoi au serveur et avoir un stockage sur le serveur permettant de qualifier le système entier. Cette fonction fait partie du système de base.

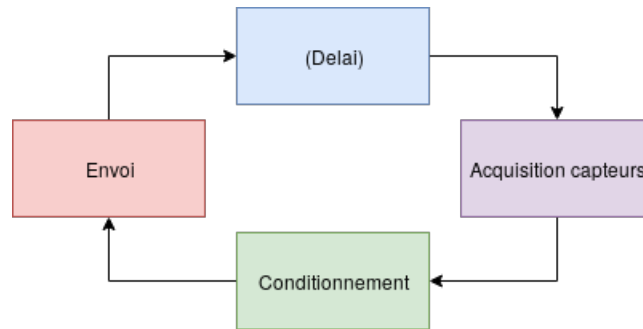


FIGURE 1.7 – Fonction Traitement des données

### Description de la fonction communication avec le serveur

- Les entrées proviennent de la fonction Traitement des données ci-dessus et la fonction gestion de l'alimentation et de l'IHM boîtier permettant d'appairer un téléphone avec le boîtier. Cette fonction possède deux sorties, l'une sur le module Bluetooth et Wi-Fi dans un second temps (facultatif). La seconde est une mémoire permettant de stocker les données si il n'y a pas de connexion ou si l'utilisateur à choisit de n'envoyer les données qu'à chaque fin de trajet ou autres ;
- Cette fonction intéragie avec la mémoire du système et les modules Wi-Fi/bluetooth ;
- Si le smartphone n'a pas d'accès au réseau, le boîtier enregistrera les données dans sa mémoire pour les envoyer lorsque le réseau sera de nouveau disponible.

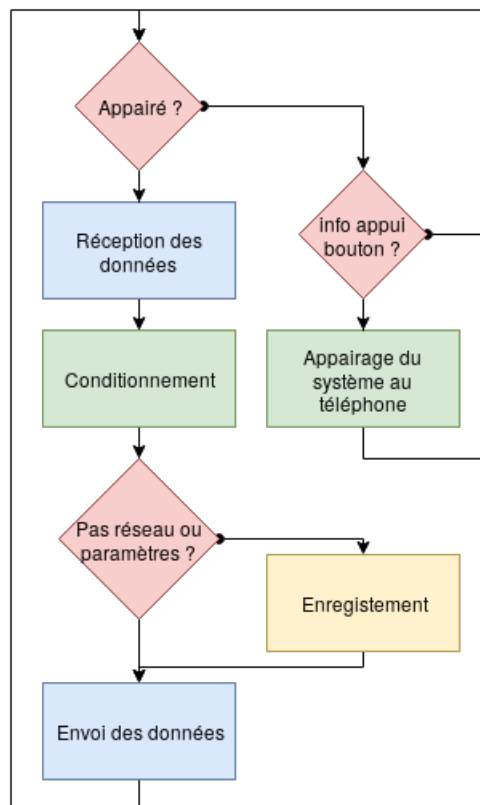


FIGURE 1.8 – Fonction Communication avec le serveur

### 1.6.3 Définition de la fonction visualisation des informations

#### Identification de la fonction visualisation des informations

Présenter la fonction :

- Visualisation des informations ;
- Cette fonction permet à l'utilisateur de visualiser ses parcours sur l'application de son téléphone ;
- Cette fonctionnalité est secondaire car il faut d'abord capter et enregistrer les données sur un serveur avant de pouvoir les visualiser.

#### Description de la fonction visualisation des informations

- En entrée nous avons le serveur permettant de récupérer les informations stockées à propos de l'utilisateur et en sortie L'affichage de l'application sur le téléphone du cycliste ;
- Cette fonction interagit avec le serveur de stockage et le téléphone de l'utilisateur ;

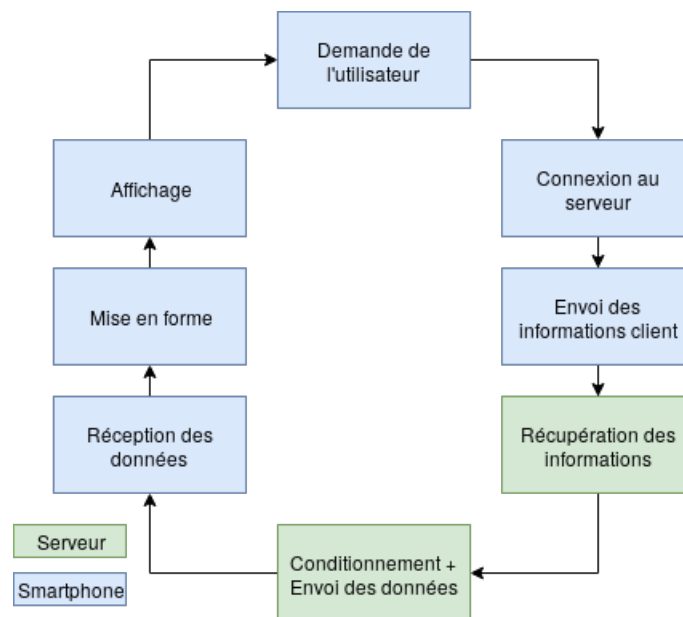


FIGURE 1.9 – Fonction visualisation des informations

### 1.6.4 Définition de la fonction configuration du système

#### Identification de la fonction configuration du système

Présenter la fonction :

- configuration du système ;
- Elle permet à l'utilisateur de configurer le boîtier et son profil dans le serveur. Cela lui permet de configurer la récurrence d'envoi des données, la connexion, et de récupérer les parcours de l'utilisateur sur le serveur ;
- Cette fonction est secondaire, l'utilisateur peut utiliser le boîtier avec des valeurs par défaut avant que cette fonctionnalité soit implémentée et les données peuvent être enregistrées sur le serveur sans identification.

#### Description de la fonction configuration du système

- Le smartphone et les choix de l'utilisateur sont les entrées de la fonction. En sortie nous avons l'intelligence du boîtier et une table d'une base de données du serveur pour l'identification de l'utilisateur ;
- Cette fonction interagit avec l'utilisateur, le boîtier et le serveur ;

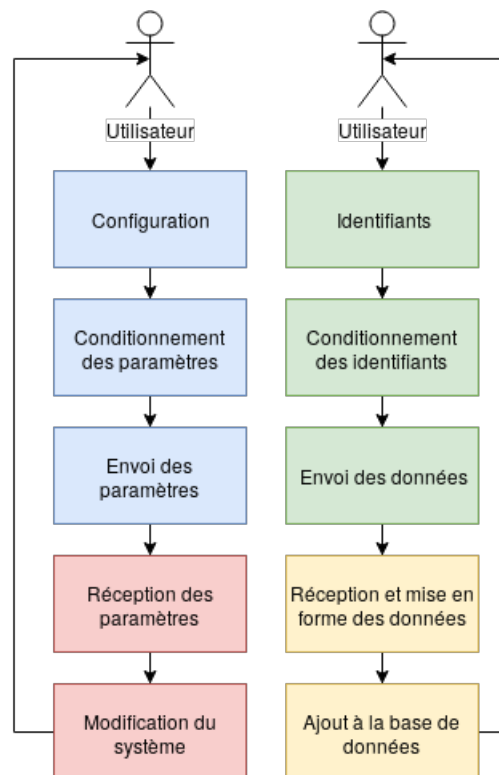


FIGURE 1.10 – Fonction configuration des informations

Les couleurs bleues et vertes représentent la partie téléphone, la partie rouge représente le boîtier et enfin la partie jaune représente le serveur.

### 1.6.5 Définition de la fonction gestion de l'alimentation et de l'IHM boitier

#### Identification de la fonction gestion de l'alimentation et de l'IHM boitier

Présenter la fonction :

- gestion de l'alimentation et de l'IHM boitier ;
- Cette dernière fonction permet de contrôler l'alimentation du boitier, informer et alerter l'utilisateur du niveau de batterie, de l'état actuel du boitier et enfin appairer le boitier au téléphone ;
- cette fonctionnalité est secondaire dans le projet.

#### Description de la fonction gestion de l'alimentation et de l'IHM boitier

- Les entrées de cette fonction sont la tension image du niveau de batterie et la touche sensitive. En sortie nous avons les LEDs de visualisations permettant d'indiquer le niveau de batterie et l'état du système et enfin la fonction communication avec le serveur permettant d'appairer le boitier avec un téléphone en bluetooth ;
- Cette fonction interagit avec les LEDs de visualisation et l'interrupteur ;
- Le niveau de batterie n'a pas besoin d'être vérifié et affiché en permanence. Cela peut être fait seulement lors du début du parcours par le cycliste (lorsqu'il démarre le système par exemple). Par contre, la valeur de l'interrupteur a besoin d'être scruté en permanence. Pour la LED de visualisation d'état, elle doit être allumée lors d'événements comme un envoi de données, une erreur de connexion, un appairage en cours, ...



## 1.7 Conditions de fonctionnement

### 1.7.1 Performances

Le système doit fonctionner tout le long des trajets, à scruter l'état de la route, la fonction de traitement des données doit fonctionner en permanence. Nous n'avons pas de problème temps réel sur le projet mais ce traitement doit être prioritaire sur le logiciel embarqué dans le boîtier. Il doit également scruter l'interrupteur sensitif mais seulement lorsque le boîtier n'est pas appairé à un téléphone, dans ce cas le système ne fonctionne pas (aucun parcours possible).

Sur la communication, le téléphone ou le serveur peuvent être inaccessibles durant un ou plusieurs parcours suivant la taille de la mémoire embarquée et le format des trames envoyées, leur taille. Côté serveur, il doit être toujours disponible, aucune indisponibilité n'est acceptée.

### 1.7.2 Capacités

Il n'y a pas de capacité maximum sur le système complet identifié. Il y aura une capacité maximum d'utilisateurs et de données stockée sur le serveur et cela correspondra à la capacité du serveur (la taille de stockage). Également un nombre maximum client se connectant au serveur simultanément.

Il y aura également un nombre maximum de parcours pouvant être stocké sur la mémoire du boîtier avant de faire une mise à jour sur le serveur mais tout cela n'est pas encore fixé.

### 1.7.3 Modes de fonctionnement

Pour démarrer le système il faudra allumer le boîtier à l'aide d'un long appui sur le bouton sensitif ou bien un interrupteur marche / arrêt placé sur le boîtier. ensuite le système commencera à enregistrer les vibrations. Un autre type d'appui sur le bouton permettra d'appairer le boîtier à un téléphone.

# Plan de développement

## 2.1 Découpage du projet en tâches

Voici la liste des tâches du projet Roule ma poule sans nid de poule provenant du gantt prévisionnel :



FIGURE 2.11 – Liste des tâches du projet

Les paragraphes suivants présentent les grandes tâches listées.

Les durées sont données en semaine / homme, nous considérons une semaine par 1h/jour pendant 5 jours + 3 heures/jour pendant 2 jours (week-end), soit un total de 11 heures par semaine.



### **2.1.1 Tâche Cahier des charges**

#### **Description de la tâche**

Il s'agit d'écrire ce document.

#### **Livrables**

À la fin de cette tâche, il s'agit de rendre ce document pour contrôle.

#### **Estimation de charge**

Cette tâche est estimée à 3 semaines.

#### **Contraintes temporelles**

La tâche doit être terminée avant le 11 / 11 / 2018 à 23 h.

### **2.1.2 Tâche Cahier d'analyse**

#### **Description de la tâche**

Il s'agit d'écrire le cahier d'analyse permettant de supprimer toutes les incertitudes techniques du projet.

#### **Livrables**

Un cahier d'analyse sera rendu.

#### **Estimation de charge**

Cette tâche sera réalisée pendant une bonne partie de la conception du système, elle s'étend donc sur 6 semaines mais avec très peu d'heures par semaine.

#### **Contraintes temporelles**

Cette tâche doit être réalisée avant le 13 janvier 2019.



### 2.1.3 Tâche PHASE 1

#### Description de la tâche

Cette tâche constitue le premier livrable. Elle inclut le choix et l'achat de plusieurs capteurs de vibration différents, la conception d'un système prototype de test et les tests de ces capteurs pour ne choisir que celui qui correspond à notre besoin.

#### Cycle de vie

Cette tâche possédera une longue phase de test. Elle aboutira sur le choix d'un capteur avec les raisons de ce choix.

#### Livrables

Un rapport sera apporté aux encadrants du projet.

#### Estimation de charge

L'estimation de cette tâche principale est de 6 semaines.

#### **2.1.4 Tâche Études**

##### **Description de la tâche**

Cette tâche regroupe toutes les études techniques concernant le fonctionnement du système, les choix techniques, ...excepté la Phase 1 décrite précédemment.

##### **Livrables**

Les éléments choisis apparaîtront dans le cahier d'analyse.

##### **Estimation de charge**

Nous avons estimé cette tâche à 4 semaines en sachant qu'elle est pratiquement terminée.

### **2.1.5 Tâche Conceptions**

#### **Description de la tâche**

Elle consiste à concevoir la carte électronique finale, le boîtier, la création du serveur et les délais de livraison et de fabrication.

#### **Livrables**

Il n'y aura pas de livraison à proprement parlé sur cette tâche.

#### **Estimation de charge**

Nous avons estimé ce regroupement de tâches à 7 semaines.

#### **Contraintes temporelles**

Il y aura des contraintes temporelles sur l'achat et la livraison de la carte électronique pour ne pas engendrer des retards sur les autres tâches.



### **2.1.6 Tâche Programmmations**

#### **Description de la tâche**

Cette tâches regroupe la programmation du boitier (système embarqué), de l'application téléphone et du serveur.

#### **Cycle de vie**

Cette tâche, spécifiquement sur la programmation orientée objet sera découpée en sprint de 1 semaine. Des phases de tests pour ces programmes seront également mises en place tout au long de la réalisation.

#### **Estimation de charge**

Cette tâche est estimée à 10 semaines pour tous les programmes cités ci-dessus.



### **2.1.7 Tâche Documentations**

#### **Description de la tâche**

Cette tâche regroupe toute la documentation finale : les manuels utilisateur / maintenance, le rapport, la soutenance et les propositions pour la suite du projet.

#### **Livrables**

Les livrables seront tous les documents et la présentation du projet en soutenance.

#### **Estimation de charge**

Cette tâche est estimée à 5 semaines en fin de projet.

#### **Contraintes temporelles**

Ces documents font partie des livrables donc ont de fortes contraintes temporelles.

## 2.2 Planning

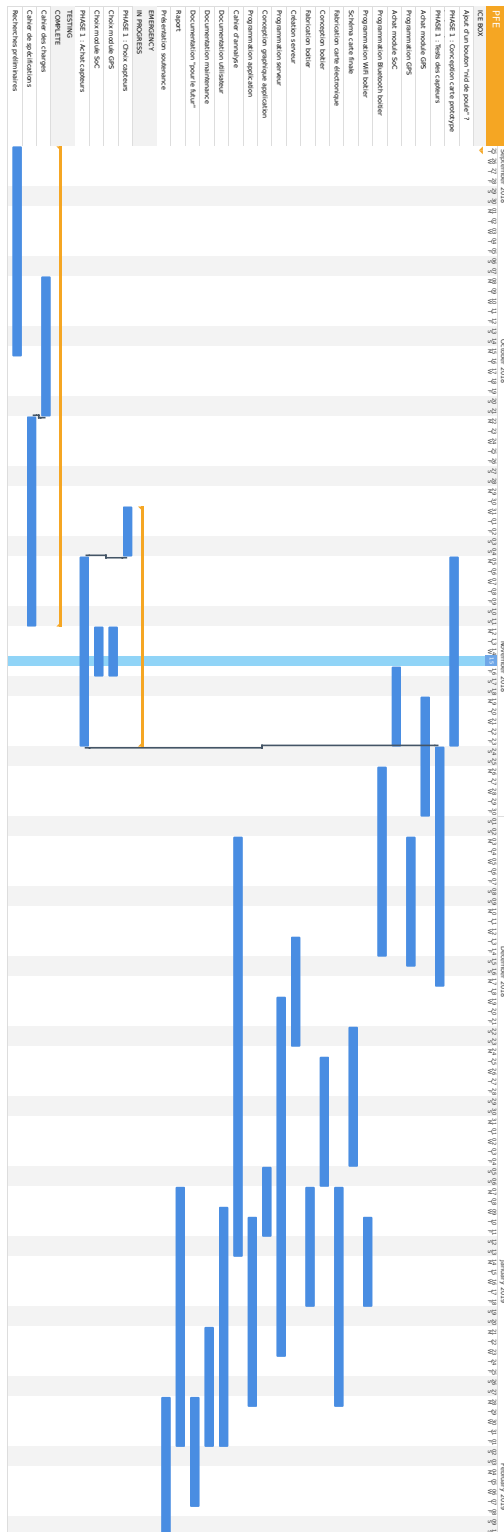


FIGURE 2.12 – Gantt prévisionnel