

Conception d'un système temps réel LoRa pour compétitions sportives

Travail de Bachelor - Pré-étude



HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

Auteur : Bise Léonard
Conseiller : Bressy Pierre
Filière : Informatique Embarquée (ISEC)
Période de travaille : 9 Mars au 1^{er} Juin 2018

Table des matières

Table des figures	ii
Liste des tableaux	iii
1 Introduction	1
2 Concept	2
3 Objectifs	3
4 Description du système	4
4.1 Fréquence de transmission	5
5 LoRa et LoRaWAN	7
5.1 La couche physique LoRa	7
5.2 La couche MAC LoRaWAN	8
5.3 Utilisation de LoRaWAN	9
6 Description du capteur	11
6.1 Position GPS	12
6.2 Rythme cardiaque	12
6.3 Cadence de pas	13
6.4 Radio LoRa	15
6.5 Matériel	15
6.5.1 Adafruit Feather	15
6.5.2 SODAQ One	17

6.5.3 Multiconnect mDot	18
6.5.4 Comparatif des différents modules	19
7 Description de la passerelle	21
7.1 Types de passerelle	22
7.2 Matériel	22
7.2.1 Raspberry Pi	22
7.2.2 Dragino LoRa Hat	23
7.2.3 IMST iC880A	24
8 Description de l'application mobile	26
9 Cahier des Charges	28
10 Environnement de travail	30
11 Planification	31
11.1 Diagramme de Gantt	33
12 Livrables	34
13 Analyse de risque	35
Bibliographie	37

Table des figures

4.1	ElemSystem	5
5.1	Architecture LoRaWAN	8
5.2	Couches LoRa	9
5.3	Architecture LoRaWAN	9
6.1	Processus Capteur	11
6.2	Antenne GPS	13
6.3	Sangle Pectorale	14
6.4	Adafruit Feather	16
6.5	Adafruit Featherwing GPS	16
6.6	SODAQ One	17
6.7	Multitech mDot	19
7.1	Architecture passerelle	21
7.2	Raspberry Pi	23
7.3	Raspberry Pi	24
7.4	IMST iC880A	25
8.1	Esquisse Application	26
8.2	Architecture Application	27
11.1	Planning - Diagramme de Gantt	33

Liste des tableaux

6.1 Adafruit Feather Caractéristiques	16
6.2 Adafruit Feather List	17
6.3 SODAQ One Caractéristiques	18
6.4 SODAQ One List	18
6.5 Multitech mDot Caractéristiques	19
6.6 Multitech mDot List	19
6.7 Comparatif Microcontrôleur	20
6.8 Comparatif Module	20
7.1 Raspberry Pi Caractéristiques	23
7.2 Dragino Hat Caractéristiques	24
7.3 IMST iC880A Caractéristiques	25
10.1 Environnement de travail	30
12.1 Livrable	34
13.1 Analyse de risque	35

1 Introduction

Ce travail de Bachelor vise à coupler deux disciplines qui m'intéressent particulièrement, d'une part les communications sans-fil dans le monde de l'informatique embarquée et de l'autre la course à pied. Étant moi-même coureur amateur et participant de temps en temps à des compétitions, je me suis demandé s'il serait possible de créer un système qui pourrait contribuer à impliquer davantage les spectateurs de ce genre de manifestation.

Le rythme auquel se déroule une compétition sportive, que ce soit de course à pied ou de cyclisme par exemple, associe la plupart du temps des moments de forte excitation avec des instants plus calmes. On peut découper ce genre d'événement en trois grosses phases. Avant le début de la compétition, il y a généralement beaucoup d'animation dans la zone de départ, les bénévoles s'occupant de l'événement mettent les dernières touches à l'organisation tout en remplissant les tâches requises par leur poste, les sportifs sont en train de s'échauffer et le public arrive peu à peu pour assister au départ de la compétition. Une fois que la course a commencé l'excitation retombe rapidement, les participants quittent la zone de départ pour rejoindre l'arrivée et les spectateurs vont souvent se déplacer le long du parcours pour les regarder passer ou au contraire se diriger directement vers la zone d'arrivée. Enfin, lorsque les sportifs arrivent petit à petit à la fin de la course, de plus en plus de gens se retrouvent dans l'aire d'arrivée, ce qui a pour effet de remettre l'ambiance.

Tout ceci fait que parfois le spectateur vit des moments de temps mort où il ne se passe pas forcément beaucoup de choses car il se trouve loin de l'action, de plus il est rare qu'il ait la possibilité de connaître la situation à un instant donné de la course. Seul les gros événements disposent parfois d'une retransmission télévisuelle, et à ma connaissance très peu proposent un outil utilisable par les spectateurs permettant de visualiser sur une carte la position des coureurs.

C'est sur cet axe que ce projet se positionne : proposer un outil interactif que les spectateurs d'événements sportifs peuvent utiliser à tout moment et de n'importe quel endroit pour connaître l'état actuel de la course.

2 Concept

L'idée du projet est de développer un système de suivi en temps réel de la compétition qui se base sur les informations transmises par un capteur porté par les sportifs. Cela permettrait de pouvoir à tout moment afficher la position actuelle des sportifs le long du parcours ainsi que d'autres informations intéressantes comme la vitesse moyenne, le rythme cardiaque et le nom du coureur par exemple. Le but étant, à l'aide d'une application mobile, de permettre aux spectateurs de pouvoir suivre le déroulement de l'événement plus aisément et de manière globale et ainsi d'essayer de rendre la compétition plus vivante pour lui.

Afin de pouvoir exploiter les données enregistrées par les capteurs, il faut les récupérer et les centraliser sur un serveur. Pour se faire une des solutions les plus communes serait d'utiliser le réseau de téléphonie mobile, ceci comporte un désavantage important qui est le besoin d'acquérir des cartes SIM pour chaque capteur pour pouvoir leur permettre de communiquer sur le réseau. D'autres part, suivant l'endroit où la course se déroule la couverture du réseau peut être faible voire inexistante.

Pour pallier à ses défauts ce projet vise l'utilisation du protocole LoRa qui ne nécessite pas de surcoût pour son utilisation et dont les modules de transmission et réception ont un faible coût. Ensuite un système de borne posé à certains endroits le long du parcours, des passerelles, est utilisé pour garantir une couverture du réseau optimal. Ce système permet la flexibilité de rajouter ou au contraire d'enlever certaines passerelles en fonction de la couverture du réseau souhaitée. Grâce à ses passerelles le système est capable de récupérer les données transmises par les capteurs et de les stocker dans une base de données qui peut ensuite être exploité par une application.

Une autre plus-value de ce système par rapport aux solutions existantes actuellement est qu'il vise à proposer une mise à jour en temps réel des informations, c'est-à-dire que les données produites par les capteurs sont disponibles pratiquement instantanément aux utilisateurs au travers de l'application mobile associée, cela permet de visualiser l'évolution de la course avec plus de précision et d'interactivité. La majorité des solutions existantes actuellement enregistrent des données durant la course mais il faut attendre la fin de l'activité pour pouvoir les télécharger sur un site web afin de visualiser le parcours et les statistiques.

3 Objectifs

L'objectif principal du projet est le développement d'un prototype du système dans son entier, une preuve de concept, c'est-à-dire d'un capteur, d'une passerelle et d'une application mobile permettant la récupération et l'affichage des données. Le projet comprendra le développement et la mise en place de tous les logiciels et équipements qui sont nécessaire à son bon fonctionnement. Même si le but est de développer un prototype fonctionnel les réflexions nécessaires pour en faire un produit seront apportées au mémoire, en particulier en ce qui concerne le capteur.

Afin de permettre de contrôler et appréhender complètement le concept, au lieu d'utiliser une solution déjà existante sur le marché, la passerelle sera également développée dans le cadre du travail de Bachelor à partir de module existant sur le marché. Cela rendra possible une étude plus fine au niveau des débits de transferts des informations produite par le capteur.

Un autre élément très important du projet est l'utilisation des communications sans fil de type LoRa, il sera évalué si l'utilisation de ce protocole pour ce genre d'application est possible ou si au contraire les contraintes qui y sont liées sont trop fortes.

4 Description du système

Le but principal du système est de pouvoir récupérer des informations relatives à plusieurs sportifs puis de centraliser les données. Par la suite une application mobile exploitera ses informations pour montrer à l'utilisateur la situation globale de la course au moyen d'une carte.

Pour se faire le système est composé de trois éléments principaux :

- Un capteur contenant une antenne GPS, une sonde rythme cardiaque et un accéléromètre
- Un gateway ou passerelle qui récupère les données transmises par le ou les capteurs
- Une application mobile qui est chargée d'analyser et d'afficher les informations de chaque sportif

Les sportifs seront équipés d'un capteur qui permettra l'acquisition des différentes données, à intervalles réguliers les capteurs vont transmettre les informations récoltées, une passerelle est ensuite chargée de récupérer les données puis de les stocker dans une base de données relationnelles.

Une ou plusieurs passerelles seront placées le long du parcours afin de pouvoir garantir une récupération optimale des données des capteurs et ainsi limiter les zones d'ombres ce qui péjorera l'efficacité du système. C'est principalement la topologie de l'endroit où se tient la course qui déterminera le nombre de passerelles qui devront être employé, cependant le nombre de participant sera également un facteur à prendre en compte. Dans le cadre du travail de Bachelor et à cause des contraintes de temps qui y sont relatives, une seule passerelle ainsi qu'un capteur seront assemblés, la démonstration du système se tiendra donc dans un endroit permettant l'utilisation d'une seule passerelle dans des conditions permettant le fonctionnement optimal du système avec un seul coureur.

Une fois que les données des coureurs ont été reçues, traitées et stockées, l'application mobile récupère les informations depuis la base de données et les analyse pour pouvoir les restituer au spectateur au moyen d'une carte géographique ainsi que de tableaux de statistiques.

Le système de capteur utilisera la technologie LoRa afin de transmettre les informations récoltées à la passerelle. C'est un protocole de télécommunication radio à bas-débit destiné à des objets connectés à basse consommation électrique et de petite taille. Un des avantages principaux du protocole LoRa pour le genre d'application visé par ce projet de diplôme est qu'il ne nécessite pas d'autorisation spéciale pour son utilisation car la bande de fréquence qu'il utilise fait partie du domaine libre. Ces bandes de fréquences peuvent être utilisées dans un espace réduit pour des applications diverses dans les domaines industrielles, scientifiques, médicales domestiques ou similaires. Elle permet aussi l'envoi de messages à des distances entre 2 km dans un milieu urbain jusqu'à 15 km dans un environnement dépourvu d'obstacle grâce à des mécanismes qui rendent les messages résistants aux interférences externes. De plus il ne nécessite aucun surcoût, à contrario du GSM (Global System for Mobile Communications) par exemple, qui requiert une carte SIM pour pouvoir communiquer sur le réseau. On comprend tout de suite l'avantage du LoRa par rapport au GSM si l'on pense au fait que pour un seul événement en théorie plusieurs centaines de capteurs pourraient être utilisés en même temps.

Une fois que les données ont été traitées et stockées dans la base de données, l'application mobile doit pouvoir les récupérer. Pour le prototype la base de données sera directement hébergée

sur la passerelle, celle-ci doit être connectée à un réseau qui doit également être accessible à travers l'application mobile. Pour se faire, la solution la plus simple est de connecter la passerelle au réseau en utilisant soit une connexion de type WiFi ou alors de type Ethernet standard. Ceci permettra ensuite, à condition que le téléphone portable ou la tablette exécutant l'application mobile soit connecté à ce même réseau, à l'application de faire des requêtes à destination de la base de données. Pour des raisons de logistiques, il se peut qu'une connexion de type 3G ou 4G soit plus appropriée si par exemple il n'est pas possible à la passerelle de se connecter par d'autres moyens à la base de données. Cet aspect ne sera pas abordé dans le cadre du travail de diplôme mais cela peut être une voie d'évolution pour la suite.

La figure 4.1 montre la situation d'ensemble et les interactions des éléments du système.

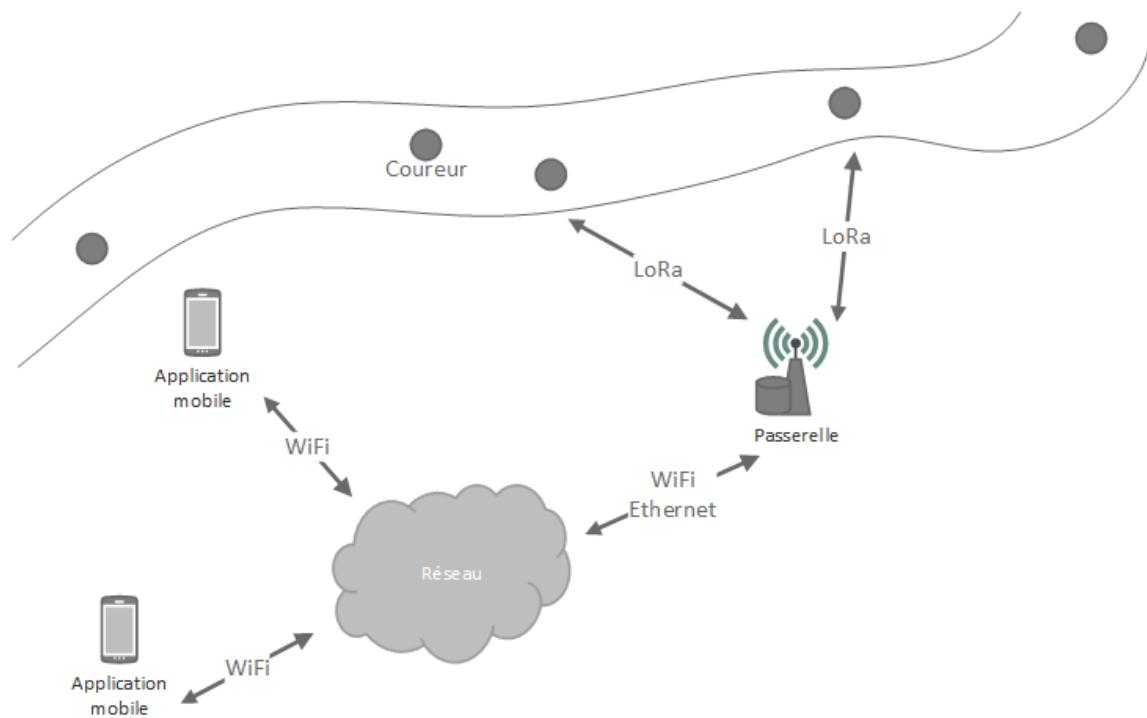


FIGURE 4.1 – Schéma de description du système

4.1 Fréquence de transmission

La fréquence de transmission des données est un des points critique du projet. En effet elle doit être assez élevée pour permettre au système de fonctionner correctement, c'est-à-dire de garantir une réactivité suffisante pour rendre l'application agréable à l'utilisation, mais elle doit également respecter les contraintes liées à la technologie LoRa.

Dans le cadre du projet et parmi les paramètres qui vont être enregistrés par le capteur, l'élément qui va permettre de dimensionner la fréquence de transmission est la position GPS du coureur car c'est cette position qui doit être transmise assez souvent pour permettre une mise à jour fréquente de la carte géographique par l'application mobile. D'un point de vu temporel, les autres éléments, c'est-à-dire le rythme cardiaque et la cadence de pas sont moins critiques.

Afin définir la fréquence de transmission, il est nécessaire de définir la fréquence d'acquisition de la position GPS afin d'obtenir le résultat voulu. L'être humain le plus rapide au monde, Usain

Bolt, peut atteindre la vitesse de 44.72 km h^{-1} , bien sur cette vitesse ne peut être maintenue que sur une courte distance de sprint. Ce projet est destiné à des courses à pied plus longue, de fond, c'est-à-dire à partir de 5 km de distance. Chez les hommes le record de cette distance est détenu par l'Ethiopien Kenenisa Bekele avec un temps de 12 :37.35 minutes ce qui correspond à une vitesse moyenne d'environ 23.8 km h^{-1} et chez les femmes par l'Ethiopienne Tirunesh Dibaba avec un temps de 14 :11.15 minutes c'est-à-dire environ 21.1 km h^{-1} . [1]

Si l'on part du principe qu'un coureur très rapide peu atteindre une vitesse de 20 km h^{-1} ou environ 5.5 m s^{-1} , une fréquence d'acquisition de la position GPS toutes les 30 secondes, ce qui correspond à une distance d'environ 165 m, doit permettre de pouvoir afficher le parcours du coureur sur la carte avec assez de précision.

Il faudra encore approfondir cette question durant la réalisation du projet pour vérifier si cette valeur suffit ou s'il est nécessaire d'augmenter la fréquence d'acquisition. Une analyse devra également être menée pour définir si cette fréquence est compatible avec le duty cycle imposé par l'utilisation de la bande de fréquence ISM. Le duty cycle définit le temps pendant lequel un élément est autorisé à transmettre des données en utilisant cette bande de fréquence, c'est une contrainte que tous les éléments doivent respecter et qui dépend de la quantité de donnée à transférer ainsi que du débit.

5 LoRa et LoRaWAN

Il existe deux couches distinctes dans l'écosystème LoRa, premièrement il y a la couche physique LoRa et ensuite la couche protocolaire LoRaWAN, parfois également appelé LoRa MAC.

La couche LoRa est une couche physique qui définit le type de modulation utilisé pour envoyer les données brute grâce à des ondes électromagnétiques. Cette couche n'a pas la connaissance des hautes couches et peut donc être utilisé avec n'importe quel protocole.

La couche LoRaWAN quant à elle est une couche protocolaire MAC qui s'appuie sur la couche physique LoRa. Elle définit les aspects réseaux comme la topologie employée, le moyen de relayer les messages entre les acteurs ainsi que les aspects sécuritaires comme le chiffrement des données.

Il est possible de n'utiliser que la couche physique LoRa sans le LoRaWAN, ceci simplifie beaucoup le système par contre cela implique la perte d'un certain nombre de fonctionnalité qui sont proposées par le protocole.

Le choix ou non d'utiliser la couche protocolaire LoRaWAN est détaillé à la fin du chapitre.

5.1 La couche physique LoRa

La couche physique LoRa définit de quelle manière les données sont transformées pour être transmise par des ondes radios. La technique de modulation de base pour transformer les informations en signal se porte sur le principe d'étalement de spectre (spread spectrum), l'idée est d'étaler les données sur une largeur de spectre plus grande que le signal en modifiant la phase de la fréquence porteuse du transmetteur en suivant une séquence pseudo aléatoire.

L'avantage de la technique de l'étalement de spectre réside dans le fait que les signaux émis sont par définition plus résistant aux interférences externes que si le signal était transmis normalement. Le Direct Sequence Spread Spectrum est une technique de modulation qui applique ce principe. Cette technique est beaucoup utilisée, notamment par la norme 802.11b (WiFi). Le prix a payé pour utiliser cette méthode est qu'elle nécessite la présence d'une horloge très précise sur le récepteur et le transmetteur afin de pouvoir coder et décoder les signaux correctement. En plus l'opération en elle-même est également un processus qui prend passablement de temps et de ressource et qui dépend de la séquence de codage utilisée. Ses aspects font que le Direct Sequence Spread Spectrum n'est pas approprié pour des applications à bas coût et basse consommation comme ceux requis pour ce projet.

La couche physique LoRa vise à corriger ses défauts en utilisant également une méthode d'étalement de spectre un peu différente que le Direct Sequence Spread Spectrum. Elle se base sur une autre technique nommée Chirp Spread Spectrum. Cette technique a été développée dans les années 1940 pour être employé dans les radars. L'étalement de spectre est toujours un élément central cependant la séquence pseudo aléatoire a été abandonnée ce qui simplifie la gestion de la modulation et permet à des équipements basse consommation de l'utiliser sans contrepartie coûteuse. [2]

La technique de modulation utilisé par la couche physique Lora se nomme logiquement LoRa Spread Spectrum, elle utilise la bande de fréquence libre Industrielle, scientifique et médicale

(ISM) qui, pour l'Europe, utilise la bande de fréquence 868 Mhz. Cette bande de fréquence n'est pas soumise à autorisation pour être utilisée, ce qui est donc un avantage de taille et qui limite les coûts supplémentaires. Cette modulation permet au moyen de paramètre changeable, des débits de transferts allant de 0.3 kbps à 27 kbps, suivant le débit sélectionné on pourra envoyer plus ou moins de données, entre 51 à 222 octets par message. [3]

5.2 La couche MAC LoRaWAN

La couche radio LoRa est une technique propriétaire de la société Semtech, cependant la couche protocolaire LoRaWAN qui fait partie du groupe des Low-Power Wide-Area Network (LPWAN), est entièrement ouverte et son standard ainsi que ses évolutions sont gérés par la LoRa Alliance qui est une association qui regroupe diverses entreprises ou groupes qui sont actifs dans le domaine des communications sans fils.

Le protocole LoRaWAN définit les acteurs qui font partie du réseau. Ils sont décrits dans la figure 5.1 .

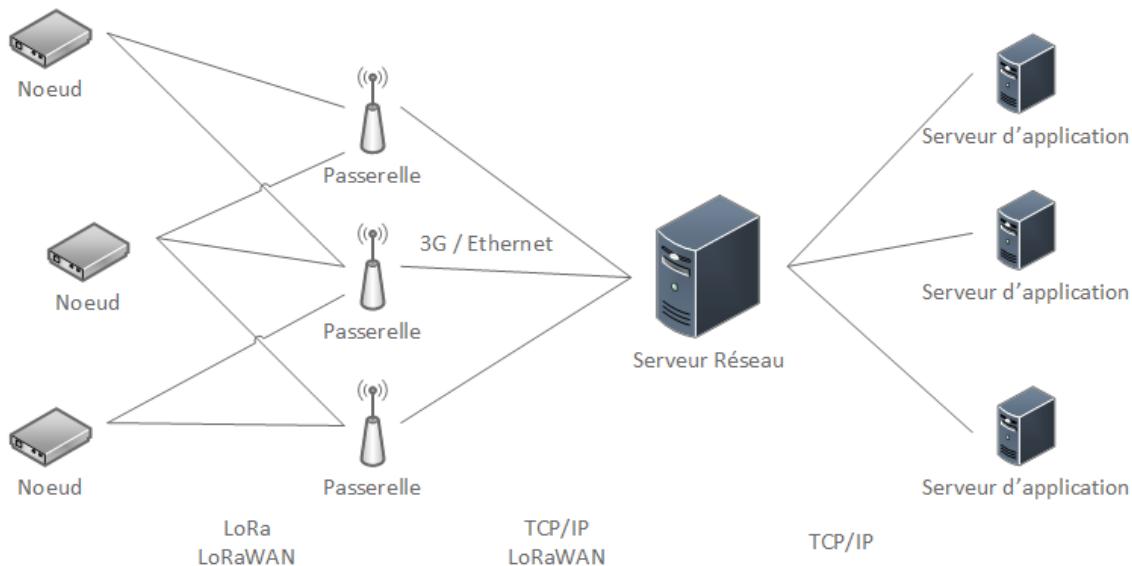


FIGURE 5.1 – Architecture des réseaux LoRaWAN

Les aspects sécuritaires, à travers l'utilisation de la cryptographie symétrique, la définition des procédures pour rejoindre le réseau et la gestion des taux de transferts des nœuds sont également des éléments abordés dans ce standard. Un des aspects intéressant du LoRaWAN est qu'il peut employer une technique appelé Adaptive Data Rate (ADR) qui est une façon de maximiser la vie des batteries des nœuds et la capacité du réseau en donnant la possibilité à l'infrastructure du réseau LoRaWAN de changer le taux de transfert et la fréquence de transmission pour chaque nœud individuellement. Cela permet au réseau de diminuer le taux de transferts d'un nœud très éloigné tout en augmentant son facteur d'étalement, ce qui fera qu'il prendra plus de temps à envoyer les données mais elles auront plus de chance de parvenir jusqu'à la passerelle, inversement les nœuds proches auront leur taux de transfert augmenté mais leur facteur d'étalement diminué.

La figure 5.2 montre l'architecture d'une application utilisant le LoRaWAN.

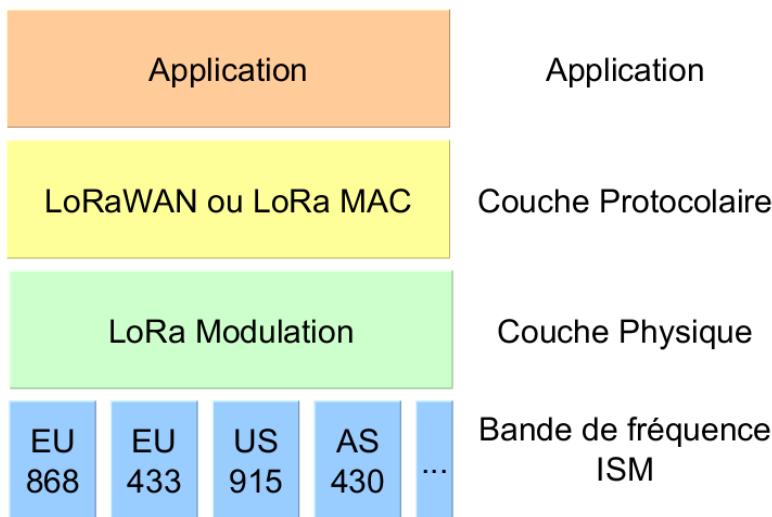


FIGURE 5.2 – Couches LoRa – Document LoRaWAN Specification v1.1

C'est également dans ce standard qu'est définie l'architecture du réseau dans son ensemble, c'est-à-dire comment les données sont acheminées d'un nœud jusqu'au serveur réseau qui traite et stock les données. Les données sont acquises par le nœud et ensuite transmise à la couche LoRaWAN qui les envoie grâce à la couche radio LoRa. Une fois le paquet reçu par la passerelle, un packet forwarder est en charge de l'envoyer à destination du serveur réseau, généralement en utilisant le protocole UDP. Le serveur réseau est ensuite responsable de la gestion du protocole LoRaWAN, c'est lui, si nécessaire, qui génère les acquittements des paquets et les retransmet à une passerelle pour être envoyé au nœud. Enfin le serveur réseau transmet les données utiles reçus des nœuds aux serveurs d'application qui vont les exploiter pour par exemple montrer l'évolution de la température dans une région ou encore la position d'un élément mobile. [4]

La figure 5.3 montre l'architecture d'un réseau LoRaWAN dans son ensemble ainsi que le détails de chaque éléments.

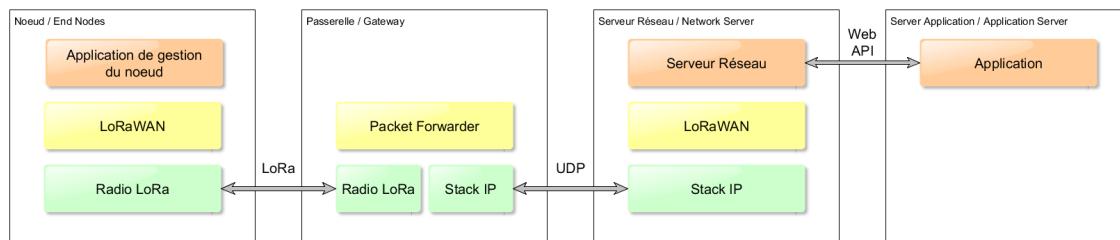


FIGURE 5.3 – Architecture des réseaux LoRaWAN

5.3 Utilisation de LoRaWAN

La couche protocolaire LoRaWAN offre beaucoup d'avantage cependant elle nécessite énormément de travail pour être entièrement mise en place pour un réseau privé comme c'est le cas dans ce projet. Dans le cas où un nombre important de nœuds ainsi que plusieurs passerelles sont exploités, cette couche prend tout son sens car elle offre des moyens de gestion de l'en-

semble du réseau et permet de centraliser l'information. Elle offre également une interface web moderne qui permet la récupération des données produite par les nœuds afin de les exploiter dans de diverses applications. Elle s'occupe aussi de la gestion du chiffrement des données transmises par les nœuds jusqu'aux passerelles par exemple.

Dans le cadre du travail de Bachelor et dans un esprit de simplification, la couche LoRaWAN ne sera pas utilisée. Comme expliqué elle nécessite la mise en place d'une infrastructure réseau complexe ce qui complique passablement la mise au point du projet. Il est clair que dans une optique de développement d'un produit, ce protocole prendrait tout son sens mais le développement du prototype mais l'accent sur la définition du système, principalement du capteur, ainsi que des logicielles associés. L'utilisation d'un serveur hébergé sur internet rajoute également de la complexité au niveau de la passerelle, puisque le système sera testé en extérieur, cela nécessite l'ajout d'une interface de communication par le réseau mobile, 3G par exemple, afin de pouvoir transmettre les données récoltées par la passerelle au serveur. Enfin, puisque le prototype ne sera composé que d'un seul capteur et d'une seule passerelle, l'utilisation du protocole LoRaWAN n'apporte que très peu d'avantage au prix d'une complexité accrue.

Afin de pouvoir concentrer le travail sur le développement du capteur, de la passerelle et de l'application mobile, il semble judicieux de ne pas se lancer dans la mise en place longue et coûteuse que représente l'utilisation du LoRaWAN. Néanmoins, dans une optique d'évolution du projet, le développement durant le travail de Bachelor sera pensé de façon à faciliter l'éventuelle modification ultérieure du système afin de pouvoir utiliser le protocole LoRaWAN.

6 Description du capteur

Les informations relatives aux sportifs sont acquises par le biais du capteur. Il sera composé d'un assemblage de différents éléments déjà existants dans le commerce. On veillera à minimiser au maximum son poids ainsi que sa taille pour éviter de gêner le sportif durant sa course. Une fois assemblé, il sera placé dans une boîte étanche qui disposera d'un bracelet élastique ce qui permettra à l'athlète de le porter facilement. Enfin il doit également disposer de sa propre source d'énergie sous la forme d'une batterie, celle-ci doit lui conférer une autonomie assez grande pour pouvoir fonctionner durant l'entièreté d'une course.

Trois types de données différentes seront récupérées, la position GPS du coureur (longitude/latitude), son rythme cardiaque en battement par minute ainsi que la cadence de ses pas en pas par minute. Régulièrement le capteur va effectuer les acquisitions, puis il va traiter les données et créer un paquet de données. Une fois que le paquet est prêt à être transmis, le capteur utilise son module de transmission LoRa pour les envoyer à la passerelle. Quand ce processus est terminé, le capteur se met en attente jusqu'au début du cycle suivant et recommence le processus. Ce processus est illustré par la figure 6.1.

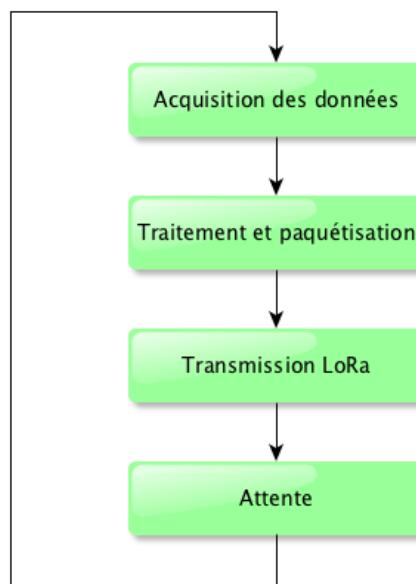


FIGURE 6.1 – Processus du capteur

Le microcontrôleur est la partie centrale du capteur, c'est lui qui exécute le programme d'acquisition des différents paramètres décrit ci-dessus, qui fait le traitement des données pour ensuite les envoyer en utilisant la transmission LoRa. Le cœur du capteur sera donc un module disposant d'un microcontrôleur ainsi que tout ce qui lui est nécessaire pour fonctionner (oscillateur, mémoire, interfaces...).

Viendront se coupler au microcontrôleur les quatre modules suivants :

- Un module GPS permettant le positionnement du coureur
- Un module permettant la mesure du rythme cardiaque
- Un module accéléromètre qui donnera la possibilité de déterminer la cadence du coureur
- Un module radio LoRa avec antenne qui se chargera de la communication avec la passe-

relle

Le microcontrôleur devra communiquer avec chacun de ses modules d'une manière ou d'une autre afin de pouvoir récupérer les informations et les transmettre. Les modules n'ont pas les mêmes contraintes, il est donc impératif de sélectionner un environnement dans lequel chacun d'eux aura à disposition le canal de communication approprié que ce soit une interface série, un I/O ou un bus SPI par exemple.

Les prochaines sections de ce chapitre décrivent chaque module ainsi que leurs interfaces.

6.1 Position GPS

Un module GPS est responsable de déterminer la position du coureur. Le système GPS se base sur une constellation d'au moins 24 satellites qui sont en orbite moyenne (Altitude de 20'200 km) autour de la Terre. Afin qu'un utilisateur puisse déterminer sa position sur le globe, il est nécessaire qu'il ait la vue sur au moins 4 satellites. A une fréquence très précise, donné par des horloges atomiques, les données de position orbitales ainsi que de temps sont envoyées en direction de la planète, grâce à ses informations le receveur est en mesure de connaître la distance de chacun des satellites en vue et donc de calculer sa position géographique ainsi que de déterminer le temps actuel, c'est ce que l'on appelle le GPS « lock » ou « fix ». [5]

Cette tâche compliquée est entièrement à la charge du module GPS lui-même, il dispose d'une antenne qui lui permet de recevoir les données envoyées par les satellites, de les analyser et de calculer les informations requises pour les mettre à disposition du microcontrôleur. La grande majorité des modules GPS transmettent les informations en utilisant une interface série UART sur lequel des messages de type NMEA 0183 sont envoyés vers le récepteur. Ce protocole définit le format de plusieurs messages qui contiennent diverses informations de position, de statuts et de temps. [6]

Dans le cadre du projet de Bachelor l'information principale qui sera exploité sera la longitude et latitude du capteur qui permettra de déterminer la position du coureur. D'autres informations intéressantes pourront éventuellement être utilisées comme le temps ou l'altitude.

Le composant peut être directement placé sous l'antenne ou alors il existe également des solutions avec antenne branché par un câble comme montré sur la figure 6.2.

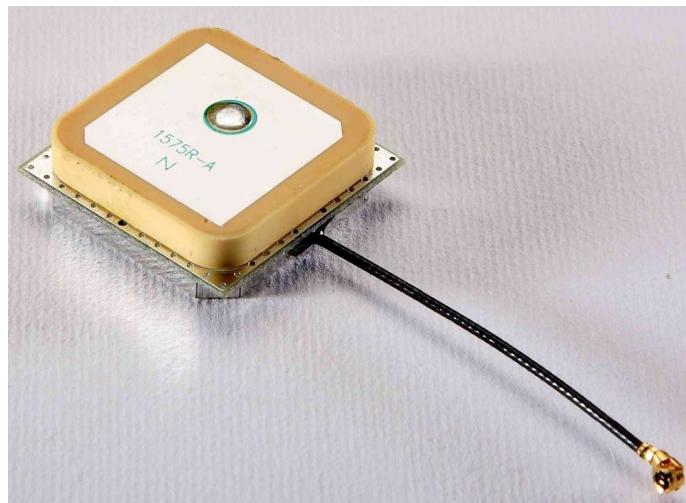


FIGURE 6.2 – Antenne GPS - © SODAQ

6.2 Rythme cardiaque

Il existe deux moyens principaux de déterminer le rythme cardiaque d'une personne. La première solution consiste à utiliser une ou plusieurs électrodes qui détecteront directement les signaux électriques émis par le cœur, c'est le principe utilisé par les électrocardiogrammes. L'autre solution est basée sur l'utilisation de LEDs placées contre la peau et qui vont émettre de la lumière à une fréquence régulière, la lumière réfléchie par réfraction et qui dépend de la dilatation des vaisseaux sanguins est ensuite détecté par le capteur et permet de déterminer le rythme cardiaque de l'utilisateur. Cette deuxième technique est très utilisée dans les montres GPS utilisés pour le sport car elle a l'avantage de pouvoir être directement intégrée dans le boîtier et ne nécessite que peu de moyen pour être mis en place.

Pour le capteur de ce projet, les deux solutions existent sur le marché, cependant la solution de l'électrocardiogramme semble plus appropriée et pratique car elle permet d'utiliser un cardiofréquencemètre sans fil du commerce et évite ainsi au porteur du capteur de devoir s'accommoder des fils qui sont nécessaire pour le fonctionnement de la solution à base de LED.

Dans les deux cas, l'utilisation par le microcontrôleur est semblable, la sonde sera connectée sur une entrée digitale et elle émettra une impulsion à chaque fois qu'un battement sera détecté. Il suffit donc de compter le nombre d'impulsion et de mesurer le temps pendant lesquels elles ont eu lieu pour pouvoir déterminer la fréquence cardiaque en BPM (Battement par minute).

La formule 6.1 permet de calculer le rythme cardiaque rc en battement par minutes (BPM).

$$rc = \frac{60 * n}{t} \quad (6.1)$$

Avec n le nombre de battement accumulé pendant un temps t en secondes.

Les battements du cœur peuvent être détecté grâce à une sangle pectorale munie d'un cardiofréquencemètre. Adafruit propose ce genre de dispositif, il émet une impulsion chaque fois qu'un battement est détecté, un petit module connecté sur une ligne d'entrée du microcontrôleur permet ensuite au logiciel de les compter. Ce type de dispositif est illustré sur la figure 6.3.

Du fait de la gêne occasionnée par une électrode cette solution sera privilégiée.



FIGURE 6.3 – Sangle pectorale - © Garmin

6.3 Cadence de pas

Le nombre de pas effectué pendant un certain temps, c'est-à-dire la cadence, est une information intéressante pour les coureurs. Tout un tas de facteur peuvent l'influencer, comme le poids, la taille ou la longueur des jambes. Les coureurs amateurs se trouvent en général dans la tranche de 160 à 170 pas par minute alors que des coureurs d'élites à 180 ou même 200 pas par minute lorsqu'ils sont au maximum de leur vitesse. La cadence permet de juger de la bonne posture d'un coureur, plus les pas seront courts et plus la cadence sera élevée.

De manière générale, en course à pied la préférence est portée sur les petits pas, d'une part car c'est uniquement lorsque le coureur touche le sol qu'il peut donner une impulsion vers l'avant, ce qui lui permet d'aller plus vite, et d'autre part car le fait de faire des grands pas augmente le risque de blessure, le pied a tendance à taper plus fort sur le sol. Une cadence plus élevée permet d'améliorer la position où le pied touche le sol, idéalement cela devrait être directement en dessous du centre de gravité ou lieu de plus en avant.

Afin de déterminer la cadence de pas, un module accéléromètre est utilisé. Ce type de composant permet de mesurer l'accélération linéaire soit en m s^{-2} ou en g sur un ou plusieurs axes, en général les axes X, Y et Z. Plusieurs interfaces de communication existent pour ce type de composant, analogique ou les niveaux sont transmis à travers des valeurs de voltage, digitale avec un bus I2C ou SPI ou alors grâce à une modulation de largeur d'impulsion (PWM). Pour ce projet une communication digitale de type I2C ou SPI sera privilégiée afin de simplifier la récupération des données de l'accéléromètre.

Beaucoup d'application qui utilise ce composant existe, il permet par exemple de détecter l'orientation d'un téléphone portable (paysage ou portrait) ou de savoir si un objet est en mouvement.

C'est également l'élément central des podomètres, qui permettent de compter le nombre de pas effectué, gadget qui est très à la mode en ce moment. C'est justement cette application qui sera utilisée dans le cadre du projet, il faudra compter le nombre de pas effectué par le coureur en un certain temps pour déterminer sa cadence.

Afin de détecter quand la personne fera un pas, il est nécessaire de pouvoir détecter un pic d'accélération dans une ou plusieurs directions. Il faudra déterminer un pallier au-dessus duquel on considérera que l'intensité de l'accélération est suffisamment élevée pour la considérer comme un pas. Pour se faire il faudra que le programme lise les informations des différents axes de l'accéléromètre régulièrement et qu'il les analyse, ceci lui permettra de garder un compte du nombre de pas. Il faudra également mesurer le temps durant lequel les pas ont été comptés pour pouvoir calculer la cadence à un certain moment.

Au lieu que le programme fasse lui-même les acquisitions, certains accéléromètres peuvent être configuré pour déclencher une interruption lorsqu'un certain seuil est dépassé, cette méthode dépend entièrement du composant utilisé cependant dans la mesure du possible elle sera privilégiée car elle facilite le traitement de cette activité et évite une surcharge des opérations requises par le logiciel.

De manière similaire au rythme cardiaque, la cadence de pas cp en pas par minute pourra être calculée avec la formule 6.2.

$$cp = \frac{60 * n}{t} \quad (6.2)$$

Avec n le nombre de pas accumulé pendant un temps t en secondes.

6.4 Radio LoRa

La partie radio LoRa, c'est-à-dire la transformation des données en ondes radio est à la charge d'un composant spécifique. Il existe deux principaux types de composant, le premier intègre uniquement la couche physique LoRa, le deuxième en plus de cette couche propose également la couche protocolaire LoRaWAN. Comme expliqué dans ce document, dans le cadre du projet seule la couche radio LoRa sera employée, les composants intégrant la couche LoRaWAN offre toujours la possibilité d'utiliser uniquement la couche radio, il sera donc possible d'utiliser l'un ou l'autre de ses types de composant pour le projet.

La majorité des modules radio LoRa existant sur le marché proposent en règle générale uniquement la gestion de la couche physique, cette partie étant la propriété de la société Semtech, les composants embarqués sont toujours les mêmes, soit le composant SX1272 ou le SX1276. Ils communiquent les deux avec le microcontrôleur au moyen d'un bus SPI.

6.5 Matériel

Ce chapitre décrit une série de composants qui pourront être utilisé dans l'assemblage final du capteur. La sélection des composants exacts utilisés pour le travail de Bachelor se fera au début de la phase de réalisation.

Il est possible que certains de ses modules soit directement intégré au module microcontrôleur,

il existe sur le marché des solutions qui en plus du microcontrôleur dispose également du module radio LoRa ou d'un GPS par exemple. Ce genre de configuration permet un gain de place et de temps car les composants sont montés sur un seul et même PCB.

6.5.1 Adafruit Feather

La société Adafruit a développée toute une suite de produits différents dont les Feather et les Featherwing. L'objectif des Feather est de proposer des cartes électroniques légères, de petite taille, pouvant être alimenté par batterie et proposant tout un panel de configuration et périphérique différents. Les Featherwing quant à elle sont des cartes d'extension qui viennent se fixer par-dessus un Feather et qui permettent d'ajouter les fonctions additionnelles désirées.

Dans cette gamme il existe plusieurs modules pouvant être utilisé pour le projet, le plus approprié est le Feather M0 with RFM95 LoRa Radio, le module intègre directement un microcontrôleur ainsi qu'un module radio LoRa couche physique. Il est également possible d'y brancher une batterie lithium-ion.

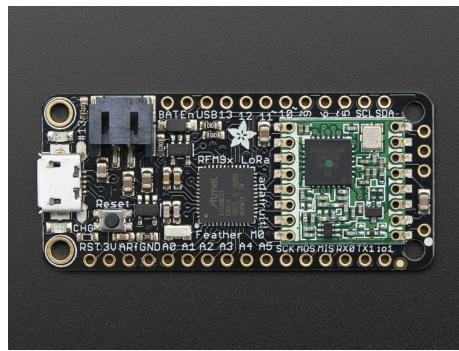


FIGURE 6.4 – Feather M0 with RFM95 LoRa Radio - © Adafruit

Les caractéristiques du microcontrôleur sont résumées dans la table 6.1.

TABLE 6.1 – Caractéristiques de la carte Adafruit Feather M0

Dimensions	51mm x 23mm x 8mm
Microcontrôleur	ATSAMD21G18 – ARM Cortex M0
Oscillateur	48 Mhz
Flash	256 kB
RAM	32 kB
LoRa	Semtech SX1272
Prix	34.50 CHF

Pour la gestion du GPS, on pourra s'appuyer sur le Adafruit Ultimate GPS Featherwing illustré par la figure 6.5, qui propose un module pouvant se fixer sur le module de base avec un composant de gestion de la position. Une connexion série UART sera utilisée pour récupérer les messages NMEA et ainsi pouvoir en extraire la position du module.

En ce qui concerne la partie logiciel, quelques systèmes d'exploitation temps réel existent pour ce type de processeur ARM. Pour ce développement l'utilisation de Zephyr sera privilégiée, il

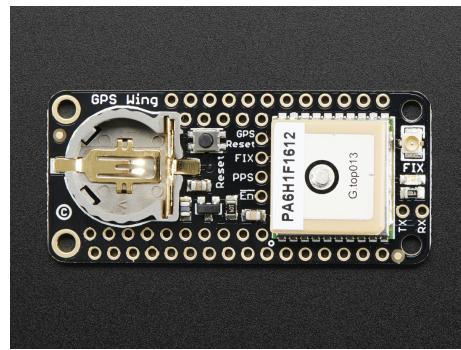


FIGURE 6.5 – Adafruit Ultimate GPS Featherwing - © Adafruit

propose toutes les fonctionnalités standard que l'on pourrait attendre de ce type de système d'exploitation et sa communauté est très active.

La table 6.2 propose une liste de l'ensemble des modules nécessaire pour l'assemblage du capteur.

TABLE 6.2 – Liste de module pour Adafruit Feather

Module	Prix	Num. produit
Adafruit Feather M0	34.05 CHF	ADAFRUIT-3178
Adafruit Heart Rate Start Pack	65 CHF	ADAFRUIT-1077
Adafruit Triple-Axis Accelerometer	8 CHF	ADAFRUIT-2019
Adafruit Ultimate GPS FeatherWing	40 CHF	ADAFRUIT-3133
Prix Total	147.50 CHF	

6.5.2 SODAQ One

La société Néerlandaise SODAQ propose un module nommé SODAQ One qui contient un microcontrôleur, le même que sur la solution Adafruit, un module LoRa, un accéléromètre ainsi qu'un module GPS sur le même PCB. Cette solution a l'avantage de contenir directement pratiquement tous les modules nécessaires au projet, reste la sonde de rythme cardiaque à intégrer.

Il est possible d'alimenter le SODAQ One avec une batterie lithium-ion, un petit panneau solaire est également proposé. Autre avantage de taille de cette carte est le fait qu'elle embarque le module LoRa de Microchip RN2483 qui propose la gestion de la couche physique LoRa ainsi que de la couche protocolaire LoRaWAN. Une interface série de type UART en utilisé pour envoyer des commandes à la puce de gestion du LoRa, elle permet aussi l'utilisation de la couche radio uniquement ou alors en consort avec la couche protocolaire. La figure 6.6 montre l'allure du module SODAQ One.

Les caractéristiques du microcontrôleur sont résumées dans la table 6.3.

C'est cette solution qui requiert le moins de travail de modification ou d'ajout pour pouvoir couvrir tous les aspects requis par le projet. Sa petite taille la rend également idéale pour l'utilisation pour ce travail de Bachelor.

Puisque le microcontrôleur utilisé par ce capteur est le même que la solution Adafruit, en termes



FIGURE 6.6 – SODAQ One v3 - © SODAQ

TABLE 6.3 – Caractéristiques de la carte SODAQ One v3

Dimensions	45mm x 25mm
Microcontrôleur	ATSAMD21G18 – ARM Cortex M0
Oscillateur	48 Mhz
Flash	256 kB
RAM	32 kB
LoRa	Microchip RN2483
GPS	uBlox EVA 8M
Accéléromètre	STMicroelectronics LSM303AGR
Prix	114 CHF

de logiciel le même système d'exploitation temps réel sera utilisé, c'est-à-dire Zephyr.

La table 6.4 propose une liste de l'ensemble des modules nécessaire pour l'assemblage du capteur.

TABLE 6.4 – Liste de module pour SODAQ One

Module	Prix	Num. produit
SODAQ One v3	114 CHF	ONE-EU-RN2483-V3
Adafruit Heart Rate Start Pack	65 CHF	ADAFRUIT-1077
Prix Total	179 CHF	

6.5.3 Multiconnect mDot

La société Multitech propose plusieurs modules de taille et de spécifications différentes, l'un d'eux est le mDot, un module de petite taille qui incorpore un microcontrôleur ainsi qu'un module radio LoRa. Il n'embarque pas d'autre module utile au projet cependant son avantage réside dans le fait qu'il donne la possibilité d'utiliser le système d'exploitation temps réel mBed qui offre toutes les fonctionnalités généralement proposées pour la gestion du temps réel comme la création de tâches, de sémaphores ou d'évènements. Cet environnement de développement est également composé d'une couche LoRaWAN logiciel permettant l'envoie de donnée à une passerelle.

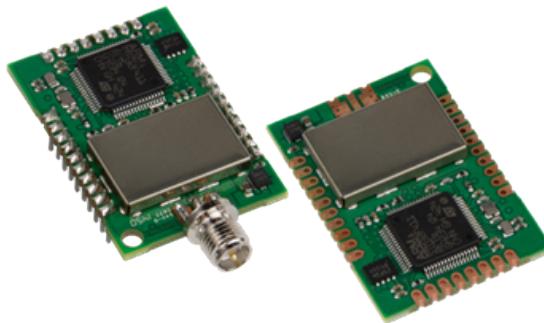


FIGURE 6.7 – Multitech mDot - © Multitech

Les caractéristiques du microcontrôleur sont résumées dans la table 6.5.

TABLE 6.5 – Caractéristiques de la carte Multitech mDot

Dimensions	37.3mm x 25.5mm
Microcontrôleur	STM32F411RET – ARM Cortex M4
Oscillateur	96 Mhz
Flash	512 kB
RAM	128 kB
LoRa	Semtech SX1272
Prix	53 CHF

L'avantage principal de cette solution est que le module vient accompagné par une suite logicielle complète et bien documenté permettant le développement d'une application temps réel. Par contre il nécessite la mise en place d'une carte supplémentaire afin de pouvoir y placer tous les autres composants nécessaires, c'est-à-dire le module GPS, l'accéléromètre et la sonde de rythme cardiaque, ce qui rajoute une quantité de travail non négligeable ainsi que des risques accrus de rencontrer des problèmes.

La table 6.6 propose une liste de l'ensemble des modules nécessaire pour l'assemblage du capteur.

TABLE 6.6 – Liste de module pour Multitech mDot

Module	Prix	Num. produit
Adafruit Feather M0	34.05 CHF	MTDOT-868-X1-SMA
Adafruit Heart Rate Start Pack	65 CHF	ADAFRUIT-1077
Adafruit Triple-Axis Accelerometer	8 CHF	ADAFRUIT-2019
Adafruit Ultimate GPS Breakout	40 CHF	ADAFRUIT-746
Prix Total	166 CHF	

6.5.4 Comparatif des différents modules

Cette section vise à comparer les modules entre eux pour pouvoir évaluer les avantages et inconvénients de chacun d'entre eux, ce qui facilitera la sélection du module qui sera utilisé durant le

travail de Bachelor.

La table suivante résume les caractéristiques de chacun des modules présentés dans cette section.

TABLE 6.7 – Comparatif des caractéristiques des modules

Module	Prix	Microcontrôleur	Fréquence	Flash	RAM
Adafruit Feather M0	34.50 CHF	ATSAMD21G18	48 Mhz	256 kB	32 kB
SODAQ One	114 CHF	ATSAMD21G18	48 Mhz	256 kB	32 kB
Multiconnect mDot	53 CHF	STM32F411RET	96 Mhz	512 kB	128 kB

TABLE 6.8 – Comparatif des différents modules nécessaire

Module	LoRa	LoRaWAN	GPS	Accéléromètre	Rythme Cardiaque
Adafruit Feather M0	SX1272	Librairie tierce	x	x	x
SODAQ One	RN2483	RN2483	✓	✓	x
Multiconnect mDot	SX1272	Librairie mDot	x	x	x

Si l'on analyse le contenu des tableaux, on remarque que le module SODAQ One se détache des deux autres solutions. Il propose déjà deux des trois fonctionnalités demandées par le projet, le GPS et l'accéléromètre, reste le rythme cardiaque à intégrer qui ne requiert à priori qu'une simple ligne I/O. De plus ce module propose une solution LoRa intéressante grâce à la puce RN2483 produite par Microchip commandé par un lien série UART, ce qui rend la gestion de la communication et l'envoie de donnée plus simple.

Le module mDot de la société Multitech représente une solution intermédiaire avec plus de flexibilité dans le choix des composants utilisé, de base elle ne propose aucun des modules requis pour le projet. Son avantage principal réside dans le fait que le module est fourni avec une suite logicielle complète comprenant un système d'exploitation temps réel, mBed, ainsi que plusieurs librairies pour la gestion des communications. Pour cette solution un travail plus substantiel sera nécessaire pour ajouter les modules nécessaires au projet.

Finalement la solution Adafruit Feather représente une somme de travail conséquente, il faudra prévoir l'ajout d'un accéléromètre ainsi que d'une sonde pour le rythme cardiaque. L'avantage principal de cette solution est son prix bas ainsi que la possibilité de ficher le module GPS par-dessus le module de base ce qui permet un gain en termes de place ce qui est intéressant dans une optique de minimiser la taille du capteur.

7 Description de la passerelle

Comme expliqué dans la section précédente le capteur transmet les informations à la passerelle en utilisant le protocole LoRa. L'objectif principal de la passerelle est la récupération des données transmises par les capteurs, de leur traitement pour finalement procéder au stockage des informations dans une base de données.

La passerelle est composée de deux parties, d'une part le récepteur LoRa également appelé un concentrateur, et d'autre part un ordinateur chargé du traitement des paquets reçus. Le concentrateur LoRa est branché sur un bus SPI relié à l'ordinateur de traitement, lorsqu'un paquet est reçu l'ordinateur hôte récupère ce paquet et effectue le traitement approprié.

Le logiciel qui fait le traitement des paquets est un « packet forwarder », son travail consiste à récupérer les paquets reçus par le concentrateur LoRa au travers du bus SPI puis de les transférer à un serveur par le biais d'un paquet de type UDP. Le paquet UDP contient des métadonnées diverses, comme le temps de réception du paquet ou encore l'adresse de la passerelle qui l'a reçu ainsi que les données brutes du paquet. Une implémentation de « packet forwarder » open source développée par Semtech est disponible sur internet gratuitement, une multitude d'autres version ont également été créées sur cette base couvrant différent type de cas d'utilisations.

Dans le cadre du travail de Bachelor il s'agira de déterminer quel est le forwarder adapté au matériel et de l'installer sur la passerelle. Un logiciel sera développé dans le cadre du projet, le serveur de paquet, qui réceptionnera les paquets UDP reçu par le packet forwarder, son travail sera l'extraction des données des paquets reçus depuis le capteur pour enregistrer le tout dans une base de données. Un protocole de communication simple sera défini qui régira le format dans lequel seront transmis les données des capteurs et, si le besoin se manifeste, la gestion d'éventuelles acquittement de transmission des données. Ce protocole sera également géré par le serveur de paquet.

Pour des raisons de simplification le forwarder, le logiciel de traitement des paquets et la base de données seront tous hébergé sur la même machine hôte, c'est-à-dire l'ordinateur de traitement connecté au concentrateur.

La figure 7.1 montre l'architecture de la passerelle ainsi que du capteur.

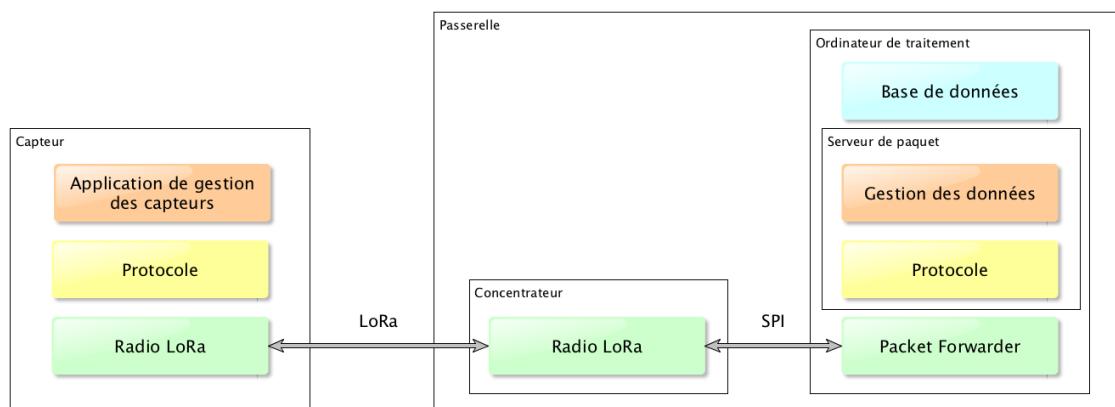


FIGURE 7.1 – Architecture logiciel de la passerelle

7.1 Types de passerelle

Les passerelles sont divisées en deux groupes, multi canaux (multi channel) et simple canal (single channel). Les simples canal sont des passerelles qui se basent sur les mêmes composants LoRa radio que les modules LoRa utilisé pour les nœuds. Puisque ces composants sont destinés aux end-devices ils ne sont capables de gérer qu'un seul canal en réception à la fois, cela vient du fait que la réception qui est prévu uniquement pour les paquets d'acquittements provenant des passerelles, sont toujours envoyé sur le même canal que le paquet envoyé par le nœud, il n'y a donc pas besoin de pouvoir faire de la réception sur plus d'un canal.

L'utilisation d'une passerelle simple canal implique la sélection d'une seule bande de fréquence sur laquelle les paquets de données seront transmis, il faudra veiller à configurer le concentrateur de la passerelle pour qu'il écoute sur cette même fréquence.

Dans le cadre du travail de Bachelor, et dans la mesure où un seul capteur sera développé, une passerelle simple canal sera probablement suffisante pour pouvoir tester le système dans de bonne conditions, il est bien sûr également possible d'utiliser une passerelle multi canaux si le besoin se fait sentir.

7.2 Matériel

Ce chapitre décrit une série de composants qui pourront être utilisé dans l'assemblage final de la passerelle. La sélection des composants exacts utilisés pour le travail de Bachelor se fera au début de la phase de réalisation.

7.2.1 Raspberry Pi

L'élément de base pour la passerelle est un ordinateur de traitement de type Raspberry Pi, mondialement connu, c'est un petit ordinateur à bas coût et proposant des performances très intéressantes. Il a été conçu par la Raspberry Pi Foundation originaire d'Angleterre. Cet ordinateur qui est proposé en différents modèle est utilisé pour tout un tas de projet à travers le monde et du fait de sa petite taille et de ses performances est tout à fait adapté à mon travail de Bachelor. De plus il propose une connexion Ethernet et également WiFi intégrée ainsi que suffisamment de mémoire et d'un processeur capable d'exécuter un système d'exploitation à base de Linux.

Dans le cadre du projet, il faudra installer une distribution Linux sur le Raspberry Pi pour y faire tourner les éléments suivants :

- Le packet forwarder qui sera en charge de la récupération des paquets LoRa et de leur envoie vers le processus serveur sous forme de paquet UDP
- Le serveur de paquets LoRa qui traitera les données reçues du packet forwarder afin d'extraire les informations relatives au coureur (Position, rythme cardiaque et cadence) pour ensuite les écrire dans une base de données. Le serveur de paquet sera également responsable de la gestion, s'il y en a, du protocole de communication spécialement développé pour ce projet.
- Le système de gestion de base de données

Les caractéristiques du Raspberry Pi sont résumées dans la table 7.1.



FIGURE 7.2 – Raspberry Pi - © Raspberry Pi Foundation

TABLE 7.1 – Caractéristiques du Raspberry Pi 3 Model B+

Dimensions	85mm x 49mm
Microcontrôleur	Broadcom BCM2837B0 – Cortex-A53 64-bit
Oscillateur	1.4 Ghz
Stockage	Carte SD
RAM	1 GB SDRAM
WiFi	802.11 b/g/n/ac
Prix	34.50 CHF

7.2.2 Dragino LoRa Hat

Cette configuration est idéale pour une passerelle de type simple canal facile à mettre en place. Au Raspberry Pi viendra se coupler un module qui va gérer la communication LoRa, ces modules d'extension se nomment souvent des hats, ils sont conçus pour pouvoir se fixer au-dessus du Raspberry Pi comme un chapeau. Il en existe plein de type différent proposant tout un panel de fonctionnalité.

Le module d'extension en l'occurrence le Dragino LoRa Hat contient un module HopeRF RFM95W qui propose une interface SPI à un module radio LoRa SX1276, un composant similaire proposé sur la plupart des solutions décrites dans la section 6 des capteurs. De plus ce module d'extension est également équipé d'un module GPS dont l'exploitation pourrait être intéressante dans une optique produit, afin de pouvoir localiser également les passerelles, cependant dans le cadre du travail de Bachelor cet aspect ne sera pas traité.

Les caractéristiques du Dragino Hat sont résumées dans la table 7.2.

L'avantage de la solution Dragino LoRa Hat est principalement son coût peu élevé. Le désavantage étant qu'il ne peut être utilisé uniquement en mode simple canal et donc limite le nombre de canal qu'il est possible d'utiliser en parallèle.



FIGURE 7.3 – Raspberry Pi et son Dragino LoRa hat - © Dragino

TABLE 7.2 – Caractéristiques du Dragino LoRa Hat

Dimensions	60mm x 53mm x 25mm
LoRa	SX1276
Type de passerelle	Simple canal
Prix	38.90 CHF

7.2.3 IMST iC880A

L'iC880A est un concentrateur capable de fonctionner en mode multi canaux. Il peut recevoir des paquets utilisant des facteurs d'étalement différents sur 8 canaux au maximum, le tout en parallèle. Un de ses avantages principaux est qu'il est capable de gérer différents étalements et largeur de bandes ce qui rend possible l'utilisation de l'adaptive data rate (ADR) proposé par le protocole LoRaWAN et qui permet au serveur réseau d'adapter le taux de transferts des nœuds de manière dynamique et ainsi rendre le réseau plus performant. Dans la mesure où le projet ne va utiliser que la couche radio LoRa, cet argument n'entre donc pas en ligne de compte, cependant dans les réflexions amenées pour le développement d'un produit ce module pourrait se rendre intéressant.

De manière similaire au Dragino LoRa Hat, ce concentrateur travail conjointement avec un ordinateur de traitement, le Raspberry Pi par exemple qui traitera les paquets LoRa qui sont reçus et transférés par le packet forwarder.

Les caractéristiques du Dragino Hat sont résumées dans la table 7.2.

Le principal avantage de cette passerelle est le fait qu'elle propose une solution multi canal, avec la gestion de 8 canaux en parallèle il devient donc possible de gérer plusieurs communications avec différents nœuds à la fois. On notera cependant son prix très élevé comparé à une solution simple canal.



FIGURE 7.4 – IMST iC880A -© IMST

TABLE 7.3 – Caractéristiques du IMST iC880A

Dimensions	79.8mm x 67.3mm
LoRa	SX1301
Type de passerelle	Multi canaux
Prix	142 CHF

8 Description de l'application mobile

L'application mobile est le moyen de visualiser les données acquises par les capteurs des coureurs. C'est une application à base Android qui permet de visualiser une course, passée ou qui est en train de se dérouler, et d'autre part d'administrer les courses.

Dans le mode de visualisation de course, l'application propose une carte géographique, sur laquelle est dessiné le parcours de la course. Vient se rajouter sur la carte des points qui désignent la position actuelle des coureurs munis d'un capteur.

Il est possible à l'utilisateur de l'application de choisir parmi tous les participants à la course lesquels ils désirent suivre en particulier, les informations relatives à ces coureurs seront ensuite affichées sur l'interface. En plus de l'information de la position GPS une présentation des informations relatives au coureur choisis sera présentée sur l'interface. Elles sont décrites dans la liste ci-dessous.

- Le nom et prénom du coureur
- Sa nationalité
- Son numéro de dossard
- Son rythme cardiaque
- Sa cadence (pas par minute)
- La distance totale parcourus
- Son temps de course actuel
- Sa vitesse moyenne en minutes par kilomètre

Dans le mode de visualisation, l'application sera en charge de faire des requêtes périodiques à la base de données afin de détecter l'ajout d'une nouvelle position. Lorsque c'est le cas, l'application met à jour la position du coureur sur la carte et le cycle recommence.

La figure 8.1 montre une esquisse de l'écran de visualisation de l'application mobile.

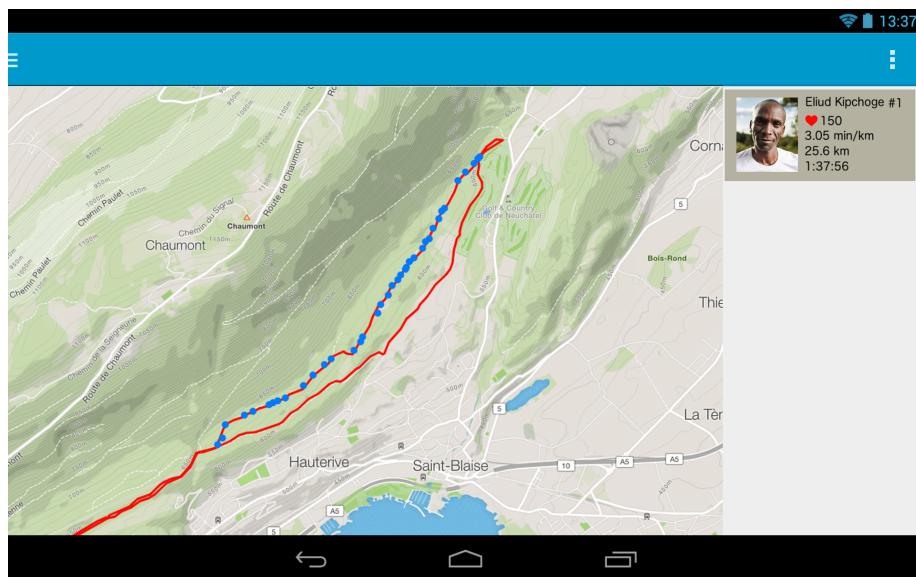


FIGURE 8.1 – Concept application mobile – Carte acquise depuis Strava

En plus de cela l'application proposera un menu de gestion des événements qui permettra à

l'administrateur de la course d'effectuer diverses opérations de maintenances.

- Création de nouvelle course
- Modification de course existante
- Définition du parcours de la course
- Ajout de coureur dans la base de données
- Gestion de l'attribution des capteurs aux coureurs
- Démarrage et fin d'une course

L'architecture générale de l'application mobile est décrite sur la figure 8.2

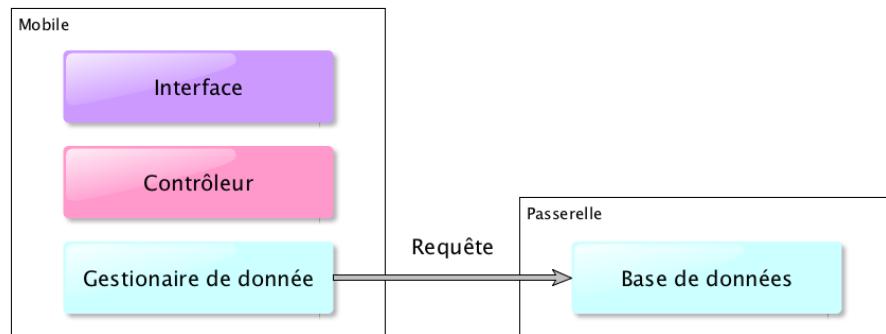


FIGURE 8.2 – Architecture de l'application mobile

9 Cahier des Charges

Cette section détail le cahier des charges à appliquer pour le travail de Bachelor. Il liste les fonctionnalités qui seront nécessaire à chaque élément afin que le système dans son ensemble puisse fonctionner.

Famille	Numéro	Description
Capteur		
	1.1	Permet l'acquisition de la position GPS (Longitude et latitude)
	1.2	Permet la mesure du rythme cardiaque de la personne portant le capteur en battement par minute
	1.3	Équipé d'un accéléromètre permettant de compter le nombre de pas par minute
	1.4	L'acquisition et la transmission de la position GPS, du rythme cardiaque et de la cadence de pas doit être fait au moins toutes les 30 secondes
	1.5	Utilise la couche radio LoRa sur la bande de fréquence 868 Mhz pour transmettre les données acquises
	1.6	Codage du programme en langage C/C++
	1.7	Poids maximal de 200g
	1.8	Le capteur doit être capable de transmettre les données à une passerelle situé jusqu'à 5 km de distance en espace libre
	1.9	Alimenté par une batterie avec une autonomie de 10h
	1.10	Placé dans un boîtier étanche munis d'une sangle afin d'être porté sur le bras du coureur

Famille	Numéro	Description
Passerelle		
	2.1	La passerelle héberge une base de données accessible par le réseau qui permet de stocker toutes les informations relatives aux courses
	2.2	La base de données est de type PostgreSQL
	2.3	La passerelle utilise la couche radio LoRa sur la bande de fréquence 868 Mhz pour la réception des données acquises
	2.4	Une application serveur de paquet, traite les paquets reçus afin d'en extraire les données des coureurs
	2.5	L'application serveur de paquet reçoit les paquets UDP au moyen d'un socket
	2.6	L'application serveur de paquet est codée en C++
	2.7	L'application serveur de paquet fait la gestion du protocole (Format des données reçu, génération des acquittements...)
	2.8	L'application serveur de paquet, stock les données des coureurs extraites des paquets dans la base de données hébergée sur la passerelle
	2.9	La passerelle est connectée au réseau au moyen du WiFi ou d'un cable Ethernet

Famille	Numéro	Description
Application Mobile		
3.1		Application mobile exécutée sur le système d'exploitation Android
3.2		Un menu permet de sélectionner parmi les deux modes de fonctionnement, visualisation ou administration.
3.3		Permet la visualisation d'une course choisie par l'utilisateur (Mode visualisation)
3.3.1		Le mode visualisation permet de sélectionner une course en cours ou une course déjà terminée
3.3.2		Si l'utilisateur sélectionne une course déjà terminé, l'application mobile rejouera la course comme elle s'est passée en temps réel
3.3.3		En mode visualisation, l'application affiche une carte avec le parcours de la course dessiné
3.3.4		En mode visualisation, l'application affiche la position des coureurs sur la carte au moyen d'un point par coureur
3.3.5		En mode visualisation, l'application vérifie régulièrement si de nouvelles positions sont disponibles dans la base de données et les mets à jour si c'est le cas
3.3.6		En mode visualisation, l'application permet à l'utilisateur de sélectionner des coureurs favoris
3.3.7		En mode visualisation, l'application affiche les informations détaillées des coureurs favoris
3.3.8		En mode visualisation, les points désignant les coureurs favoris auront chacun une couleur distincte afin de pouvoir les remarquer plus facilement sur la carte
3.4		Permet l'administration des courses (Mode administration)
3.4.1		En mode administration, un menu permet de choisir parmi les différentes options possibles
3.4.2		En mode administration, l'utilisateur peut créer une nouvelle course et rentrer les informations relatives (Nom, date, lieu, points GPS désignant le parcours de la course). L'application se chargera de créer la course dans la base de données
3.4.3		En mode administration, l'utilisateur peut modifier une course qu'il a déjà créée
3.4.4		En mode administration, une fois une course créée, l'utilisateur pourra créer les coureurs qui y participeront et entrer les informations relative (Nom, prénom, nationalité, numéro de dos-sard, numéro de capteur associé)
3.4.5		En mode administration, une fois une course créée, l'utilisateur peut signaler le début et la fin de la course

10 Environnement de travail

Le tableau 10.1 contient la liste des éléments faisant parti de l'environnement de travail. C'est-à-dire les différents outils et logiciels nécessaire au développement du travail de Bachelor.

TABLE 10.1 – Liste des outils utilisés durant le travail de Bachelor

Nom	Description
Eclipse	Environnement de développement open source pour C et C++
Zephyr ou mBed	Système d'exploitation temps réel utilisé par le capteur. Il dépendra du choix final du microcontrôleur.
GNU Compiler Collection (GCC)	Compilateur C et C++
Cross Compilateur ARM	Compilateur C et C++ pour la compilation du code source exécuté sur le capteur
Make	Exécute les fichiers makefile utilisés pour compiler le code source
Android Studio	Environnement de développement pour la plate-forme Android
pgAdmin	Outils de développement pour les bases de données PostgreSQL
git	Outils de configuration
Semtech LoRa Packet Forwarder	Programme qui s'exécute sur la passerelle et transfère les paquets LoRa reçu au serveur de paquet.

11 Planification

L'idée générale de la planification de la réalisation du projet consiste à pouvoir au plus vite valider le fonctionnement de la chaîne complète de communication. Pour se faire le développement sera découpé en 3 phases.

- Phase 1 : Validation de la transmission des données avec LoRa
- Phase 2 : Stockage des données dans la base de données et application mobile simple
- Phase 3 : Finalisation des fonctionnalités

Dans la phase 1, un capteur avec uniquement la fonctionnalité d'acquisition de la position GPS ainsi que de la transmission radio ainsi qu'une passerelle capable de récupérer les données et les afficher simplement sur un terminal seront en priorité développées puis testées. Le test veillera à reproduire les conditions finales d'utilisation du capteur, cela veut dire vérifier que la réception des données est bonne lorsque le capteur est en mouvement.

Au terme de la première phase, un jalon est placé, il implique la validation de la solution choisie en termes de matériel pour le capteur et la passerelle. Si le test de transmission de paquet LoRa avec le capteur en mouvement est concluant alors la solution sera validée. Si au contraire des problèmes sont rencontrés il faudra peut-être changer de matériel.

La phase 2 consiste à développer la base de données, dans son ensemble, ainsi qu'une application mobile simple qui se contentera d'afficher simplement des positions GPS sur une carte. Cela implique également la modification du serveur de paquet qui devra aller écrire les informations dans la base. Une fois implanté cela permet de faire le test de la chaîne complète et vérifier si la fréquence de transmission des données est suffisante pour garantir une mise à jour en temps réel de la position.

A la fin de la deuxième phase, la solution choisie pour la fréquence de transmission des données sera validée. Elle permettra de définir définitivement la configuration afin d'avoir une réactivité adéquate du système. Enfin dans la phase 3 toutes les fonctionnalités seront finalisées afin de pouvoir compléter le projet dans son entier. Cela comporte l'ajout de l'acquisition de la fréquence cardiaque ainsi que de la cadence de pas au capteur, la modification du serveur de paquet afin de pouvoir stocker ces nouvelles informations et la finalisation de l'application mobile afin de pouvoir afficher toutes les informations relatives aux coureurs ainsi que les fonctionnalités de gestion d'événements.

Pour terminer, lorsque le développement du système sera achevé, une démonstration de son fonctionnement en extérieur dans des conditions similaire à une course sera fait. Une personne sera équipée du capteur et elle se lancera sur un parcours à définir, d'une longueur entre 5 et 10 km, afin de pouvoir générer des données et les afficher grâce à l'application. Cela permettra de vérifier le bon fonctionnement du système.

Le planning se base sur les heures de travail suivantes.

- Lundi à Jeudi – 18h00 à 22h00 (4h de travail)
- Vendredi – 8h00 à 12h00 et 13h00 à 17h00 (8h de travail)
- Samedi et Dimanche – 14h00 à 18h00 (4h de travail)

		Name	Duration	Start	Finish
1	✓	Séance d'information	0.25 days	11/13/17 8:00 AM	11/13/17 10:00 AM
2	█	Recherche de sujet	40 days	1/8/18 8:00 AM	3/2/18 5:00 PM
3	█	Pré-étude	65 days	3/5/18 8:00 AM	6/1/18 5:00 PM
4	█	■ Développement & Réalisation	62.125 days	7/9/18 6:00 PM	9/28/18 4:00 PM
5		■ Phase 1	15 days	7/9/18 6:00 PM	7/29/18 1:00 PM
6		Assemblage capteur	0.5 days	7/9/18 6:00 PM	7/9/18 10:00 PM
7		Assemblage passerelle	0.5 days	7/10/18 6:00 PM	7/10/18 10:00 PM
8		Codage capteur (GPS + Transmission)	6 days	7/11/18 8:00 AM	7/18/18 10:00 PM
9		Installation passerelle (Linux + Packet Forwarder)	2 days	7/19/18 6:00 PM	7/21/18 1:00 PM
10		Codage serveur de paquet - affichage	4 days	7/21/18 1:00 PM	7/27/18 1:00 PM
11		Test transmission LoRa	2 days	7/27/18 1:00 PM	7/29/18 1:00 PM
12		Jalon capteur + passerelle	0 days	7/29/18 1:00 PM	7/29/18 1:00 PM
13		■ Phase 2	12 days	7/29/18 1:00 PM	8/13/18 10:00 PM
14		Codage base de données	4 days	7/29/18 1:00 PM	8/3/18 5:00 PM
15		Codage serveur de paquet - stockage BD GPS	3 days	8/4/18 8:00 AM	8/7/18 10:00 PM
16		Codage application mobile - Position GPS	3 days	8/8/18 8:00 AM	8/11/18 1:00 PM
17		Test système complet	2 days	8/11/18 1:00 PM	8/13/18 10:00 PM
18		Jalon fréquence transmission	0 days	8/13/18 10:00 PM	8/13/18 10:00 PM
19		■ Phase 3	17 days	8/14/18 6:00 PM	9/5/18 9:00 AM
20		Codage capteur (Rythme cardiaque + cadence)	4 days	8/14/18 6:00 PM	8/18/18 5:00 PM
21		Codage serveur de paquet - finalisation	4 days	8/19/18 8:00 AM	8/24/18 1:00 PM
22		Codage application mobile - finalisation	6 days	8/24/18 1:00 PM	8/31/18 5:00 PM
23		Démonstration du système	3 days	9/1/18 8:00 AM	9/5/18 9:00 AM
24		Création fiche de résumé projet	2 days	9/5/18 9:00 AM	9/7/18 2:30 PM
25	█	Remise fiche de résumé projet	0 days	9/7/18 2:30 PM	9/7/18 2:30 PM
26		Rédaction du mémoire	6 days	9/7/18 2:30 PM	9/15/18 11:00 AM
27		Marge	1 day	9/15/18 11:00 AM	9/16/18 11:00 AM
28		Création affiche du projet	2 days	9/16/18 11:00 AM	9/19/18 1:00 PM
29	█	Remise affiche du projet	0 days	9/19/18 1:00 PM	9/19/18 1:00 PM
30		Rédaction du mémoire	5 days	9/19/18 1:00 PM	9/26/18 9:30 AM
31	█	Marge	2 days	9/26/18 9:30 AM	9/28/18 3:00 PM
32	█	Remise Travail de Bachelor	0 days	9/28/18 4:00 PM	9/28/18 4:00 PM
33	█	Soutenance	16.125 days	10/8/18 4:00 PM	10/30/18 5:00 PM

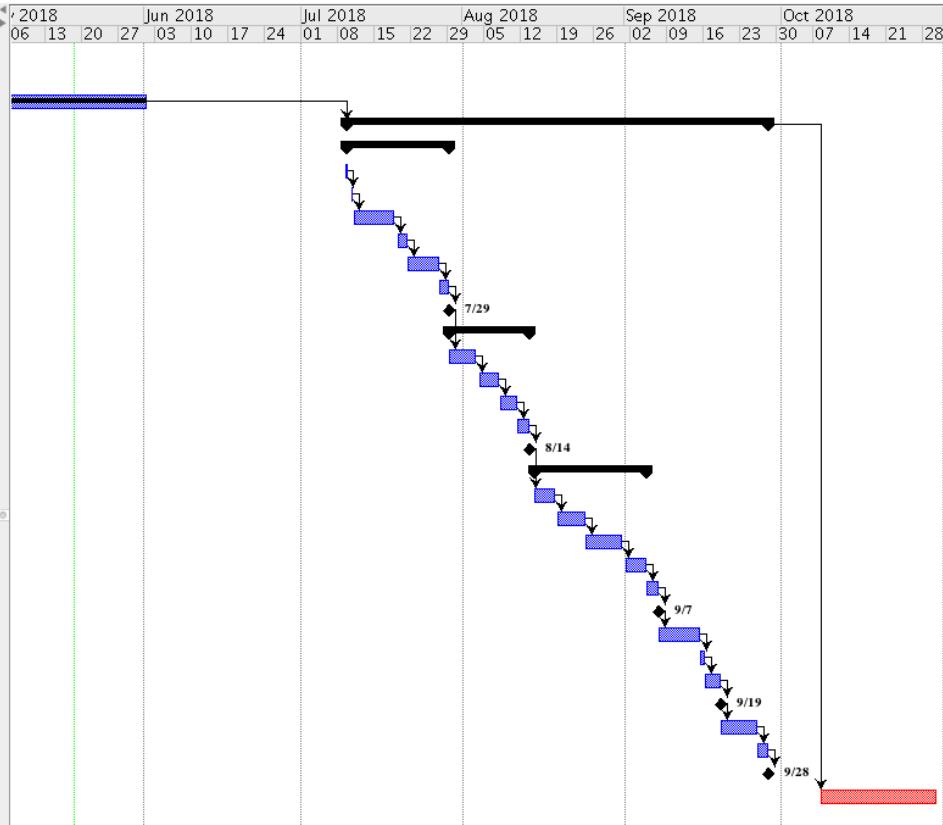


FIGURE 11.1 – Planning - Diagramme de Gantt

12 Livrables

La table 12.1 définit les objets qui forment le système dans son ensemble et qui seront évalués au terme du projet.

TABLE 12.1 – Liste des livrables

Description
1 x Capteur complet (GPS, rythme cardiaque, accéléromètre et module radio LoRa)
Logiciel embarqué de gestion du capteur
1 x Passerelle complète (Concentrateur LoRa + Ordinateur de traitement)
Logiciel « Packet Forwarder » adapté pour la passerelle
Logiciel serveur de paquet
Base de données
Application mobile Android pour visualiser et administrer les courses

13 Analyse de risque

La table 13.1 détails les risques liés au développement du projet de travail de Bachelor. Chaque risque est décrit, une probabilité qu'il se déclenche et une gravité s'il se déclenche sont mentionné. De plus la criticité est déterminée, elle définit l'impact sur le projet dans le cas où le risque se déclencherait. Une ou plusieurs mesures qui peuvent être mise en place pour réduire la probabilité que le risque se réalise sont également mentionnés dans le cas où cela est possible.

Les probabilités et les gravités sont notés de 1 Faible à 10 Haute. La criticité est la multiplication de la probabilité que le risque se déclenche avec la gravité.

TABLE 13.1: Analyse de risque

Description	Probabilité	Gravité	Criticité	Mesure de prévention
Mauvaise compréhension des activités	1	10	10	<ul style="list-style-type: none"> — Bien définir le cahier des charges — Validation du cahier des charges par l'école
Objectifs trop ambitieux, manque de temps	5	5	25	<ul style="list-style-type: none"> — Bonne définition du cadre du projet — Discussions avec le conseiller
Capteur incompatible avec la passerelle	1	10	10	<ul style="list-style-type: none"> — Recherche sur internet de système compatible — Demande de renseignements à des experts (Monsieur Tongolini et son équipe HEIG-VD)
Débit de transfert LoRa trop faible pour projet à grande échelle	7	8	56	<ul style="list-style-type: none"> — Analyse de la quantité de donnée produite par le capteur

Nouvelles technologies utilisées	10	2	20	<ul style="list-style-type: none"> — Recherches approfondies pour s'approprier ces nouvelles technologies — Demande de renseignements à des experts (Monsieur Tognolini et son équipe HEIG-VD)
Maladie	2	8	16	
Retard de livraison du matériel	4	5	20	<ul style="list-style-type: none"> — Trouver des fournisseurs en Suisse — Commander les composants dès que possible
Livraisons tardives (Affiche, résumé, mémoire)	1	10	10	<ul style="list-style-type: none"> — Définition d'un planning précis — Contrôle de l'évolution du planning durant la phase de développement — Anticipation des livraisons

Bibliographie

- [1] IAFF. All time top lists, 05 2018.
- [2] Semtech. Lora modulation basics. Application Note AN1200.22, Semtech, 05 2015.
- [3] Ferran Adelantado, Xavier Vilajosana, Pere Tuset-Peiro, Borja Martinez, Joan Melià-Seguí, and Thomas Watteyne. Understanding the limits of lorawan. 55, 06 2017.
- [4] LoRa Alliance Technical Committee. Lorawan™ 1.1 specification. Specification, LoRa Alliance, Inc, 10 2017.
- [5] U.S. government. Gps overview.
- [6] National Marine Electronics Association. Standards nmea (2000, 0183), 05 2018.