Vélo intelligent

Projet SI

2015-2016

Goureau Thomas, Duarte Sylvain, Cotard Guillaume, Salifou Nguetcheu, Houchard Corentin, Gael Herbrecht, Ali Mohamed.

Contexte du projet

L' ESIR intègre dans le programme de deuxième année un module projet. Le but de ce projet est la mise en pratique aussi bien des connaissances techniques acquises que des connaissances humaines en matière de travail d'équipe et de gestion de projet.
C'est dans ce cadre que nous nous sommes regroupés en une équipe projet de 7 personnes.

Parmi les trois projets qui nous ont été soumis, nous avons choisi de prendre le projet du vélo intelligent. Le but de ce projet est de réaliser une application de monitoring d'un vélo électrique.

Dans ce rapport nous vous présenterons notre application ainsi que les différentes fonctionnalités que nous avons développées. Nous vous présenterons dans un premier temps l'architecture globale de l'application, dans un second temps nous feront ressortir les aspects matériels et logiciels (Android et web) ainsi que les différents choix techniques que nous avons réalisés. Nous finirons par une conclusion qui fera ressortir les évolutions possibles de l'application.

I. Architecture de l'application

Le projet est divisé en deux grandes parties :

- Une partie applicative : Avec la création d'une application Android et la mise en place d'une base de donnée.
- Une partie matérielle : Avec l'utilisation d'un Arduino pour contrôler la roue motorisée.

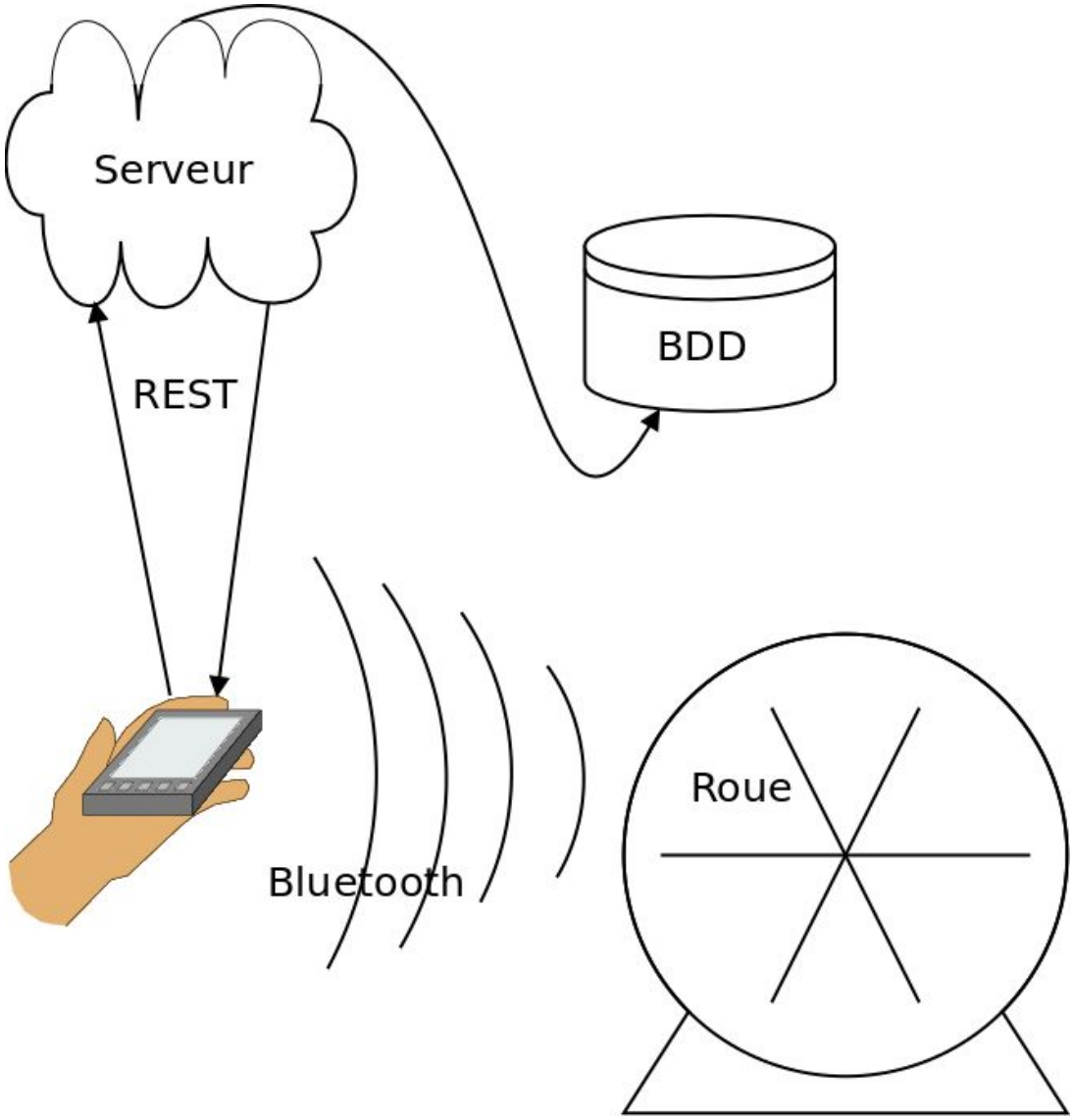
L'application Android permet à l'utilisateur de contrôler via Bluetooth son vélo. Un utilisateur enregistré sur l'application peut modifier le niveau d'assistance de son vélo une fois appareillé avec ce dernier. Il peut également consulter des informations concernant son vélo comme par exemple la vitesse, le niveau de batterie ou encore le niveau de l'assistance.

Certaines données se devant d'être persistantes, une base de données MySQL se trouvant sur un serveur distant a été mise en place. Les données qui y sont stockées sont des informations concernant les utilisateurs, leur records de vitesse et

l'historique des précédents parcours. L'aspect réseau social est également présent puisqu'un forum est également mis en place.

La roue motorisée sera fixée sur la fourche avant du vélo et sera reliée à sa batterie qui se trouvera fixée au support de la selle. Le contrôleur et la carte que nous avons réalisé seront également présents sur le vélo et connecté à la roue.

Lorsque l'utilisateur voudra se servir de son vélo, il lancera l'application Android et allumera la batterie. Une fois l'appareillage lancé, l'utilisateur sera libre de se déplacer selon la vitesse souhaitée.



II. Les grandes articulations du projet

A. Partie matérielle

1) Fonctionnement

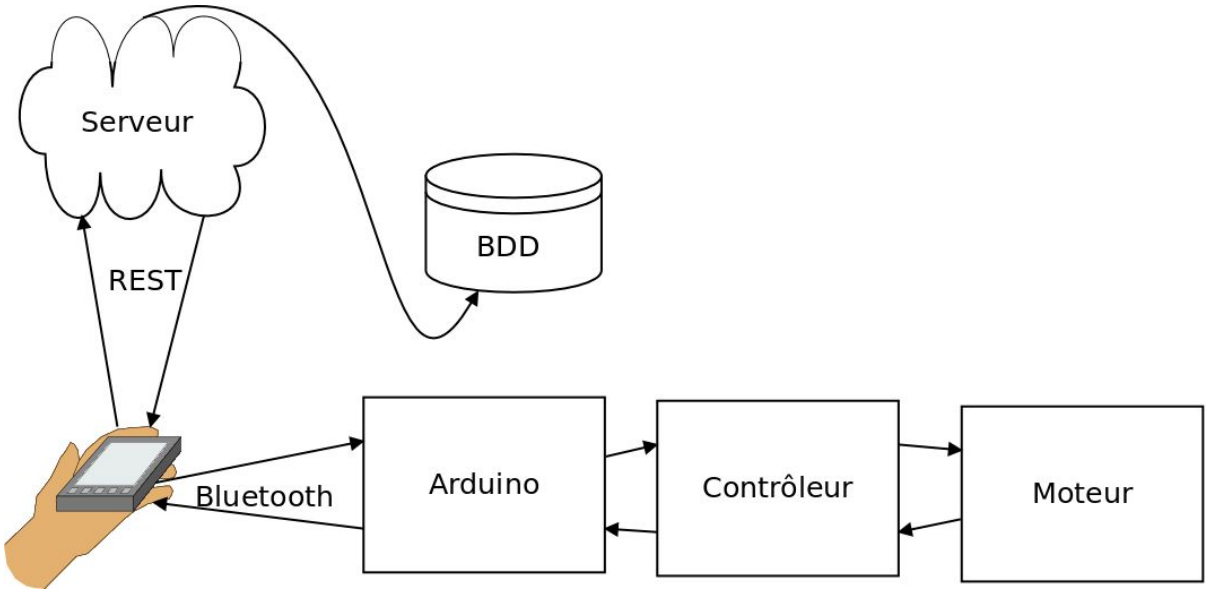
Les composants nous permettant le contrôle du vélo :

- Notre carte comprenant une carte Arduino munie d'un module Bluetooth 4.0
- Une roue électrique munie de son contrôleur.
- Une batterie cycloboost de 36V (11,6 a/h).

La roue possède en son sein un moteur capable de la faire tourner à une vitesse maximale d'environ 33 km/h. La roue est alimentée par une batterie Cycloboost de 36V. Le moteur de la roue est relié à un contrôleur "S06PW Torque Simulation controller", alimenté lui aussi par la batterie. Le contrôleur permet de modifier l'assistance en envoyant une commande au moteur et de récupérer les informations telles que la vitesse, le niveau de batterie ou encore le niveau d'assistance.

La carte Arduino a pour objectif de remplacer les périphériques tel que le potentiomètre et le LCD. Elle va recevoir les informations du contrôleur que lui même avait reçu du moteur. Elle va également envoyer des commandes au contrôleur tel que la modification du niveau d'assistance qui se chargera de l'envoyer au moteur.

L'application smartphone enverra, en fonction des actions de l'utilisateur, des requêtes au module bluetooth pour que l'Arduino envoie les bonnes commandes au contrôleur. Le module bluetooth renverra au smartphone les données que l'Arduino reçoit depuis le contrôleur.

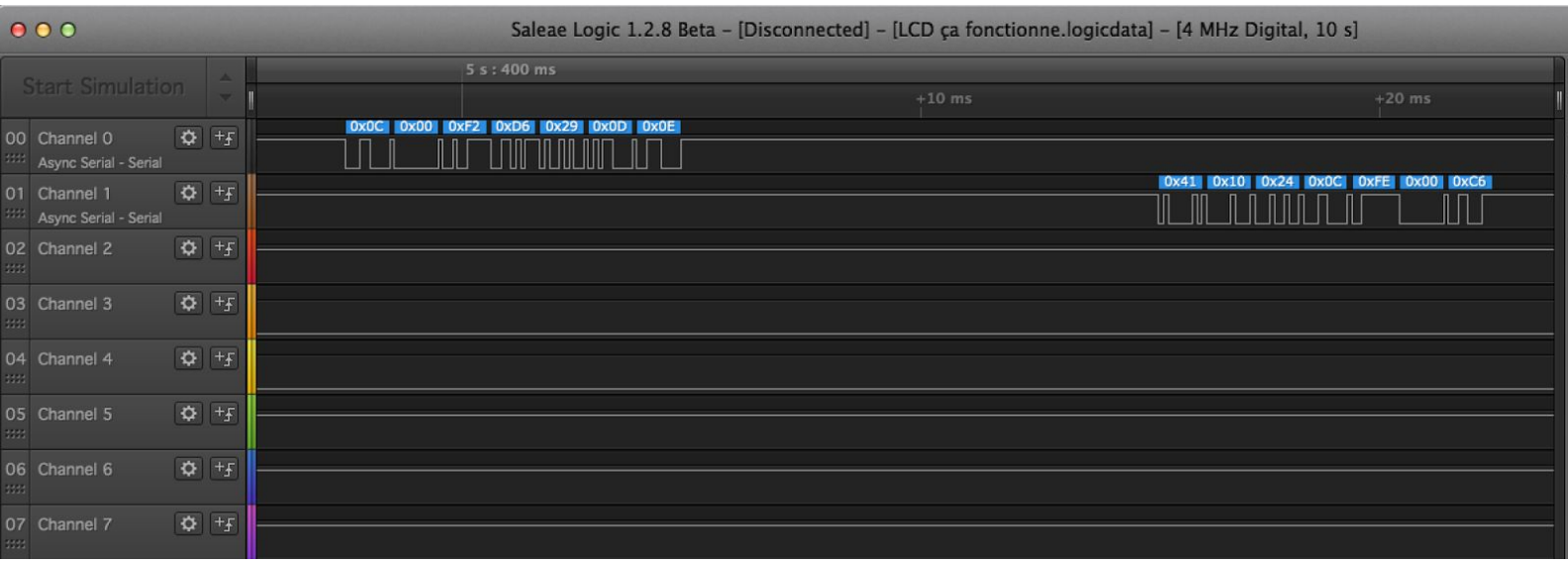


2) Conception

En plus de la roue, de son contrôleur et de la batterie, nous avions trois périphériques connectés aux contrôleur permettant d'interagir directement sur la roue.

- Un potentiomètre qui permet de modifier la vitesse de la roue.
- Un écran LCD capable d'afficher le vitesse et le niveau de batterie de la roue.
- Un frein.

Notre premier objectif était de trouver le protocole de communication du contrôleur. Malheureusement aucune documentation du constructeur n'a pu être trouvée sur internet. C'est pourquoi il a fallu faire du reverse engineering. En utilisant un analyseur logique branché entre le contrôleur et les différents périphériques. Cet analyseur récupère les signaux électriques qui circulent entre les périphériques et le contrôleur, puis il les affichent sur l'ordinateur. Nous avons donc analysé les signaux reçus et envoyés lorsque les périphériques étaient branchés.



À l'aide d'un logiciel, les différents signaux étaient plutôt simples à analyser car déjà converti en hexadécimal. Sur l'image ci-dessus, on peut observer deux channels différents. Channel 0 sert à la transmission des périphériques vers le contrôleur. Channel 1 sert à transmettre dans le sens inverse.
La description du protocole décodé est présenté en annexe.

Une fois le protocole décodé nous avons commencé le programme Arduino.
Assez rapidement nous avons été en mesure d'envoyer depuis l'Arduino au contrôleur une commande pour modifier l'assistance. Petit à petit, nous avons retiré les périphériques lorsque nous en avions plus besoin c'est à dire quand le programme Arduino était capable de les remplacer (sauf les freins). A ce stade du projet, nous étions en capacité de récupérer le niveau de batterie, le niveau d'assistance en cours et la vitesse de la roue depuis l'Arduino.

Les derniers objectifs consistaient finalement à faire fonctionner le module bluetooth pour que l'application Android et l'Arduino communiquent ensemble. Puis enfin que le potentiomètre soit remplacé par le programme Arduino. En rajoutant un condensateur de 1000 uf on envoi désormais une tension entre 1.3v et 4V au contrôleur pour remplacer le potentiomètre physique.

3) Evolution possible

En terme d'évolution possible, il serait bien de mettre un tracker sur le vélo pour pouvoir le retrouver si il a été volé, un cadenas électronique serait aussi le bienvenu pour empêcher toute tentative.
On peut également penser à un support de smartphone permettant de le recharger depuis la batterie du vélo durant le trajet.

B. Partie logicielle

1) Android

Afin d'interagir avec la partie matérielle de notre application, nous avons décidé de programmer une application sur Android. Ce choix peut se justifier par le fait qu'Android représente actuellement l'OS le plus utilisé dans les smartphones. Notre application devant essentiellement tourner sur les smartphones, il était donc judicieux pour nous de choisir cet OS.

a) Fonctionnalités implémentées

Les fonctionnalités mises en place sur Android sont les suivantes:

- Un système de géo-localisation permettant à un utilisateur de calculer son itinéraire. Cette partie est essentiellement basée sur l'API Google Maps de Google. L'utilisateur est appelé à saisir son point de départ et son point d'arrivée afin de rechercher son itinéraire.

- Le contrôle de la vitesse se fait via le smartphone de l'utilisateur qui a la possibilité de changer le niveau d'assistance sur son smartphone. il peut donc augmenter ou diminuer sa vitesse. Ceci est possible grâce au contrôleur qui récupère la vitesse au niveau du moteur.
- La surveillance de la batterie : Le tableau de bord permet également à l'utilisateur d'avoir un aperçu du niveau de la batterie du vélo. Ces informations sont également envoyées par le contrôleur.

- L'utilisateur a aussi la possibilité de créer un compte utilisateur avec lequel il pourra s'authentifier afin de consulter ses enregistrements de vitesse, ses parcours, ses records ou encore son historique.
- La connexion entre le vélo et le smartphone se fait via Bluetooth.

b) Évolution possible

Des fonctionnalités additionnelles peuvent être implémentées. Nous citerons par exemple :

- L'affichage de la météo.
- Le nombre de calories dépensées par celui qui utilise le vélo.
- Blocage et déverrouillage du vélo avec son smartphone.

2) Partie web

Nous disposons également d'une interface web faîte en PHP. Cette partie web représente la partie "Réseau social" de notre application. Notre application web est construit sur le modèle MVC. Nous disposons d'une base de donnée MySQL qui stocke les informations des utilisateurs comme le nom, le mot de passe, les records de vitesse ou encore les distances parcourues. Le choix de ce modèle se justifie par le fait que nous avons voulu créer une application facilement maintenable et pouvant évoluer facilement.

Pour l'aspect graphique nous avons utilisé Boostrap qui est un framework CSS pour composer les pages web.

a) Intérêt de la partie web

L'intérêt de cette partie de l'application réside essentiellement dans notre volonté de créer un aspect social où les utilisateurs peuvent discuter de leurs records de vitesse, leur distances parcourues et éventuellement les aspects mécaniques du vélo.

b) Fonctionnalités implémentées

Les fonctionnalités web dont nous disposons sont les suivantes :

- La création d'un compte utilisateur : grâce à ce compte, l'utilisateur peut, tout comme dans la partie Android, voir ses performances antérieures et les partager avec les membres de sa communauté.
- Nous avons également un aspect forum où les utilisateurs peuvent échanger entre eux sur des aspect liés à leurs performances aussi bien sur les vitesses atteintes que sur les distances parcourues.
- Un système de demande d'amitié est également en place, de ce fait, les membres de la communauté choisiront de façon libre avec qui partager leurs informations.

Conclusion

Ce projet a regroupé plusieurs aspects, à savoir :

- L'aspect matériel avec l'utilisation des composants électroniques
- L'aspect software divisé en deux parties

- Une partie Android pour l'interaction avec la partie matérielle.
- Une partie web sur PHP pour l'aspect "réseau social" de l'application.

Mais surtout, ce projet nous a permis d'en apprendre encore un peu plus sur la gestion d'un projet avec plusieurs collaborateurs, en effet, nous avons expérimenté la répartition des tâches, le respect des délais et le fait de surmonter les difficultés en demandant de l'aide et pas seulement en cherchant chacun de son côté.
Des compétences de divers horizons étaient requises pour mener à bien ce projet de JXS, c'est pourquoi 7 personnes n'étaient pas de trop pour travailler.
Un projet complet et complexe qui a su mettre nos compétences à l'épreuve mais qui nous a aussi permis d'en apprendre un peu plus sur le hardware ou des outils auxquels nous n'avions jamais eu recours pour développer.
Après notre mois de développement et de casse-tête, ce projet de vélo intelligent est un apport important dans notre liste de projets, de part son hétérogénéité et l'intérêt qu'il a pu susciter.

Annexes

1) Protocole du contrôleur

Envoyé par le contrôleur pour retourner les informations (en hexadécimal) :

0x41 0x(*battery*) 0x24 0x(*wheeltime high*) 0x(*wheeltime low*) 0x(*error*) 0x(*checksum*)

À envoyer au contrôleur (en hexadécimal) :

0x0C 0x(*valeur niveau d'assistance*) 0xF2 0xD6 0x29 0x(*clé niveau d'assistance*) 0x(*checksum*)

Informations :

battery : valeur entre 0 et 16. 16 = 100% de batterie.
wheeltime : valeur renvoyer sur deux octets représentant le nombre de révolution par minute de la roue

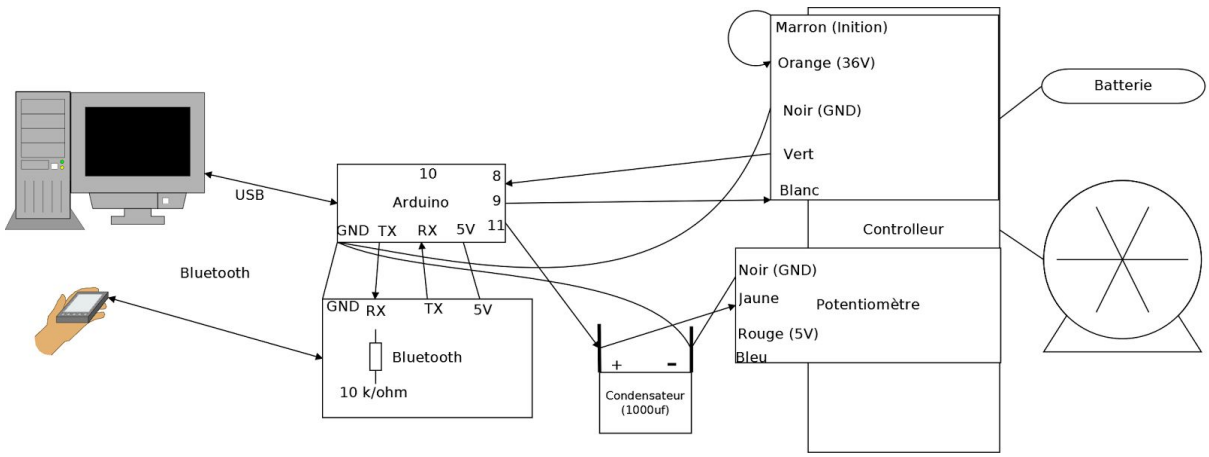
Map <Clé niveau d'assistance,Valeur niveau d'assistance> = [0,0x0D],[1,0x0C],[2,0x0F],[3,0x0E],[4,0x9],[5,0x08],[6,0x0B]

Exemple assistance 0
0x0C 0x00 0xF2 0xD6 0x29 0x0D 0x0E

Code d'erreur renvoyé

0x00 no error;
0x01 error Throttle Signal Abnormality;
0x03 error Motor Hall Signal Abnormality;
0x04 error Torque Sensor Signal Abnormality;
0x05 error Speed Sensor Signal Abnormality (Suitable for torque system);
0x06 error Motor or Controller Short Circuit Abnormality;

2) Schéma du montage matériel lors de la conception de l'application



3) Schéma de la carte finale

Carte réalisée afin de faciliter et miniaturiser le montage (réalisé par Pierre Le Corre)