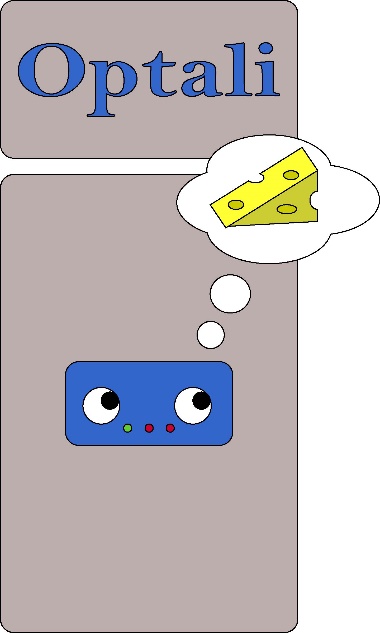
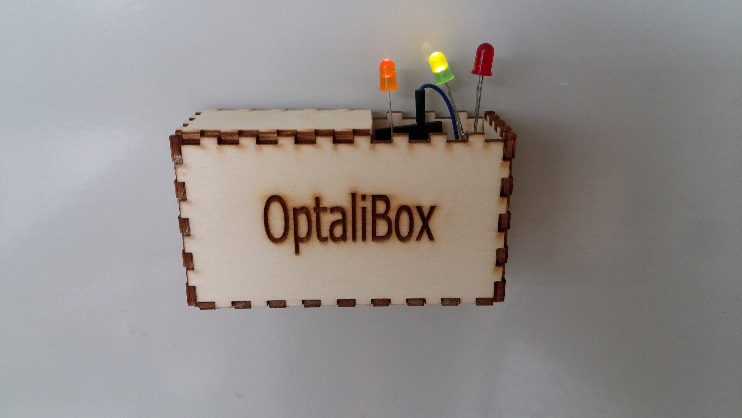


Aide à l’anti-gaspillage alimentaire

Optali

LEFEVRE FRANCOIS / HUET ALEXANDRE



# Remerciements

Premièrement, nous souhaitons remercier l’équipe pédagogique de Polytech Lille pour nous avoir enseigné les connaissances pour mener à bien ce projet de 4ème année au sein du département Informatique, Micro-électronique et Automatique.

Nous tenons aussi à remercier nos tuteurs Mr Xavier Redon, Thomas Vantroys et Alexandre Boé pour leur accompagnement dans l’accomplissement du projet.

Merci à Thierry Flamen pour ses conseils avisés en électronique et dans la conception de notre carte électronique.

Nous souhaitons remercier aussi le Fabricarium de Polytech Lille pour nous avons confié ses équipements pour la réalisation de l’OptaliBox.

Pour finir, un remerciement particulier à Laurent Engels pour le temps consacré au tournage et à la réalisation du montage vidéo.

Table des matières

[Remerciements 2](#_Toc482022044)

[Introduction 4](#_Toc482022045)

[I) Présentation du projet 5](#_Toc482022046)

[1) Objectif et cahier des charges du projet 5](#_Toc482022047)

[2) Choix techniques et matériels 6](#_Toc482022048)

[3) Calendrier prévisionnel 7](#_Toc482022049)

[II) Déroulement du projet 9](#_Toc482022050)

[1) Le serveur Raspberry 9](#_Toc482022051)

[2) La conception de l’OptaliBox 10](#_Toc482022052)

[3) L’application mobile Optali 10](#_Toc482022053)

[III) Analyses personnelles 11](#_Toc482022054)

[1) Difficultés rencontrés 11](#_Toc482022055)

[2) Améliorations possibles 11](#_Toc482022056)

# Introduction

Dans le cadre de notre formation Informatique, Micro-électronique et Automatique à Polytech’lille, nous avons réalisé un projet de 120h tout au long du semestre 8. Nous avons choisi de travailler sur l’aide à l’anti-gaspillage alimentaire car il nous permettait à la fois de travailler sur une application mobile android native, mais aussi de concevoir une carte électronique et de mettre en place un serveur de base de données.

En France, nous jetons en moyenne 21% des aliments que nous achetons, ce qui représente une perte de 100 à 160 euros par an par habitant. La raison principale de ce gaspillage est la date limite de consommation associée à chaque produit, malgré qu’un produit puisse encore être comestible après péremption. Il serait donc judicieux d’aider un utilisateur à gérer les produits dont il dispose dans son frigo pour qu’il pense à les consommer avant qu’ils deviennent périmés.

Pour résumer tout le travail accompli, nous commencerons par présenter globalement notre projet en donnant l’objectif et le cahier des charges défini, les différents choix que nous avons effectué ainsi qu’un calendrier prévisionnel sur la répartition des tâches. Nous développerons ensuite les différentes parties du projet, à savoir le serveur Raspberry, l’application mobile Optali et la réalisation de l’OptaliBox. Nous terminerons par donner notre ressenti sur le projet, en évoquant les problèmes rencontrés mais également les améliorations possibles.

# Présentation du projet

## Objectif et cahier des charges du projet

Notre but est d'aider les utilisateurs d'une part à mémoriser les denrées périssables qu'ils disposent dans leur réserve alimentaire et d'autre part d'avertir l'utilisateur de divers problèmes. Ceci peut être une notification qu'un aliment va bientôt dépasser sa date de péremption, que la porte du frigidaire est restée ouverte ou encore que le nombre d'aliments dans la réserve devient critique et qu'il faudra de nouveau le remplir. Enfin, le système aidera l'utilisateur à trouver des recettes permettant de consommer les aliments en fin de vie.

Notre système sera composé de 3 éléments.

Premièrement, un smartphone appartenant aux utilisateurs. Ces derniers permettront de scanner grâce à son appareil photo, le code barre du produit ainsi que sa date de péremption. Il servira aussi d'interface utilisateur pour notre système. En effet, c'est par cet outil que l'utilisateur pourra lister tous les aliments qu’il possède dans sa réserve en ayant la possibilité de les trier sous différentes conditions, ou encore voir les recettes possibles pour consommer des aliments bientôt périmés.

Deuxièmement, un serveur, de type Raspberry, qui permettra de stocker la base de donnée contenant les aliments de la réserve. Ce serveur sera connecté par wifi.

Troisièmement, un micro-contrôleur (RFduino) qui interagira avec une LED pour indiquer la mise sous tension du contrôleur, une LED pour prévenir l'utilisateur qu'un aliment va bientôt passer sa date de péremption et une dernière LED pour prévenir l'utilisateur que son niveau de réserve est critique. Il y aura également un buzzer et un accéléromètre pour émettre un signal sonore quand la porte du frigo est restée trop longtemps ouverte. Ce dernier sera fixé au frigo par l'intermédiaire d'un dispositif adapté qui sera un système aimanté qui se pose sur le frigo.

Nous implémenterons donc une application mobile Android qui fera le lien entre les 2 autres systèmes.

Pour la partie électronique que nous aurons à réaliser, nous utiliserons une Breadboard pour faire des premiers tests puis nous créerons une carte PCB pour notre montage final.

## Choix techniques et matériels

On utilisera une Raspberry Pi en guise de serveur pour pouvoir s'occuper du stockage de toutes les données dans notre projet puisque notre contrôleur sera incapable de les contenir, ce n'est d'ailleurs pas son but ici. En effet, même si la base de données commence à prendre de la place, nous n’aurons aucun souci à la stocker sur Raspberry. Ce n’est pas forcément le cas sur des microcontrôleurs où la mémoire reste limitée.

Notre smartphone fera donc le lien entre notre base de données et notre système embarqué par l'intermédiaire de l'application mobile. De plus, nous implémenterons un service web qui pourra gérer les différentes requêtes SQL entre l’application et notre base de données sur Raspberry.

Nous prendrons ici un RFduino comme micro-contrôleur pour notre OptaliBox puisque celui-ci nous permet d'ajouter des modules comme bon nous semble afin d'adapter le système aux besoins de notre application, comme le fait d’ajouter un module batterie pour que le contrôleur puisse être autonome. Il est également parfaitement compatible avec les applications Android. De plus, nous avons un Bluetooth directement intégré au RFduino pour s'assurer que les liaisons RFduino-application mobile se fassent correctement. On notera également que ce contrôleur consomme peu de courant et ne demande qu'une faible tension d'alimentation, critères importants pour ce genre de projet. C'est la raison principale de son utilisation pour être utilisé comme interface physique sur le frigo, en ajoutant également sa taille réduite comparée à la Raspberry Pi. Il possède aussi des pins programmables à souhait qui nous permettent d’interagir aisément avec des composants externes.

En fonction des objectifs que nous nous sommes fixés et des solutions proposées, nous avons pu définir une liste de matériel dont nous aurons besoin :

* Rfduino DIP board + USB shield (Mouser)
* Rfduino Dual AAA shield (Mouser)
* 2 piles de 1,5V AAA
* Raspberry Pi équipé du Wifi
* 1 buzzer (Mouser)
* 1 accéléromètre (Mouser)
* 3 LEDs de couleurs différentes (Mouser)
* Smartphone sous Android
* Aimants (Farnell)
* 3 Résistances CMS 330 Ohms (Mouser)

Concernant les outils de développement, nous utiliserons Android Studio pour développer notre application mobile principalement en Java. La programmation du RFduino se fera avec l’IDE Arduino. Le service web hébergé sur la Raspberry se fera en PHP et on utilisera MySql pour implanter la base de données. Pour la conception de la carte PCB, nous avons choisi le logiciel Altium Designer.

## Calendrier prévisionnel

Suite à l’établissement du cahier des charges résumant le matériel nécessaire et les objectifs à atteindre, nous pouvons alors dresser une liste de tâches que nous aurons à effectuer :

* Configurer la Raspberry :
* Installer et configurer le serveur
* Créer le diagramme UML de notre future base de données
* Créer la base sur le serveur
* Configurer la connexion Wi-Fi
* Implémenter le service web pour gérer nos requêtes SQL
* Monter et programmer le RFduino :
* Réaliser une carte PCB pour connecter les différents modules au RFduino (accéléromètre, LEDs, buzzer).
* Récupérer les consignes du smartphone grâce au Bluetooth
* Programmer le comportement du RFduino en fonction des consignes
* Envoi des alertes possibles (porte du frigo ouverte)
* Construire le boitier final
* Implémenter l'application mobile Android :
* Une première activité permettant de se connecter au RFduino et à la Raspberry
* Scanner une date et un code barre
* Ajout/suppression d'un aliment dans la base de données
* Envoi de consignes au RFduino
* Alerter l’utilisateur de différents problèmes avec des notifications
* Rendre l'interface utilisateur ergonomique (lister les aliments, envoyer vers une page de recettes avec l'utilisation des aliments bientôt périmés...)

Après s’être donné une idée de ce qui devait être accompli, nous pouvons alors faire des prévisions sur la répartition de notre temps de travail sur toute la durée du projet, soit un total de 240 heures pour un binôme :

* Partie RFduino : 60h
* Partie Raspberry : 60h
* Partie développement mobile : 120h

# Déroulement du projet

## Le serveur Raspberry

## La conception de l’OptaliBox

L’OptaliBox n’est rien d’autre qu’un module complémentaire à l’application mobile qui vient se fixer sur le frigo auquel l’application est dédiée. Ce système est donc composé de notre RFduino contenant le programme permettant d’interagir avec les différents composants soudés sur notre carte PCB, qui vient se fixer directement sur le contrôleur. On rajoute également le shield pour pouvoir alimenter le RFduino avec deux piles AAA.

Pour que ce système puisse être fonctionnel, il fallait commencer par créer notre carte PCB pour pouvoir par la suite tester notre programme sur RFduino. Une fois que cela était fait, nous avons pu attaquer la conception de notre boitier qui contiendrait tout le système.

### Conception de la carte PCB

Notre RFduino doit communiquer avec les différents composants électroniques. Afin que cela soit plus esthétique pour la modélisation finale, nous allons créer une carte PCB pour rassembler tous ces composants dessus.

Avant de représenter notre circuit général, nous allons commencer par créer une librairie rassemblant tous les composants que nous utiliserons par la suite, en particulier nos composants CMS (accéléromètre et résistances). Pour ce qui est des LEDs et du buzzer, qui sont traversants, nous pourrons utiliser des headers classiques.

Il a donc fallu créer le schematic ainsi que l'empreinte de nos composants CMS. Voici un exemple avec notre accéléromètre :

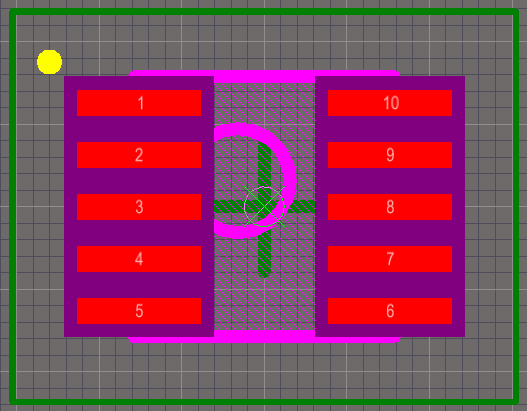
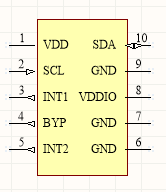


Figure 2: Empreinte PCB de l'accéléromètre

Figure 1 : Schematic de l'accéléromètre

Après avoir fait ceci avec tous les composants qui nécessitent ce traitement particulier, nous pouvons maintenant construire notre schematic du montage entier et ensuite le fichier PCB afin de pouvoir graver notre circuit.

### Programmation du RFduino

### Réalisation du boitier

## L’application mobile Optali

# Analyses personnelles

## Difficultés rencontrés

## Améliorations possibles