Université Lille1
Sciences et Technologies

POLYTECH LILLE

Projet de fin d'étude
Département Informatique Microélectronique Automatique

Lighting contactless

Auteur :
Benjamin Lafit

Tuteur d'entreprise :
M. Mickael Coronado
Tuteur école :
M. Alexandre Boé

Année 2014/2015

Remerciements

J'aimerai remercier Polytech Lille de m'avoir permis de faire ce projet en corrélation avec la société INODESIGN. J'aimerai aussi remercier M.Coronado, gérant d'INODESIGN, pour sa patience ainsi que pour le suivi et les conseils qu'il peut m'apporter. Je voudrais aussi remercier les trois autres ingénieurs de la société, M.Guiraud, M.Deschodt et M.Valenti pour l'aide qu'il peuvent m'apporter sur ce projet. Enfin je remercierai mon tuteur d'école, M.Boe pour son suivi ainsi que pour ses conseils.

Table des matières

Table des figures

Introduction

Durant la troisième année mon cursus ingénieur spécialité Informatique Microélectronique et Automatique, j'ai effectué un projet de fin d'étude [1] en corrélation avec la société INODESIGN. C'est un bureau de recherche et développement en électronique spécialisée dans le traitement du signal et le traitement d'information massivement parallèle, le plus souvent basé sur des FPGA [2]. Actuellement en contrat étudiant en sein de cette même société, j'ai pu bosser sur des projets en lien avec mon PFE, je m'appuierai donc aussi sur le travail effectué durant mon contrat.

Mon projet porte sur le domaine des objets connectés, il s'agit d'un interrupteur sans contact, pouvant, de plus, communiquer au moyen d'une liaison Bluetooth Low energy. Celui-ci a pour but de remplacer les traditionnels interrupteurs au sein de nos maisons.

Dans ce présent rapport je vous détaillerai le cahier des charges, le fonctionnement global du système ainsi que les différentes solutions retenues. Dans une seconde partie, je décrirais les différentes parties de la carte électronique réalisée ainsi que les charactéristiques spécifiques au routage d'une telle carte. Pour finir, je vous expliquerais le fonctionnement du module BLE [3] dans sa globalité puis le fonctionnement son programme ainsi que celui de l'application android de test.

1. PFE
2. Field-programmable gate array, le FPGA est un circuit logique programmable
3. Bluetooth Low Energy

1 Présentation du projet et des solutions technologiques

1.1 Cahier des charges

Le But de ce projet est de piloter l'éclairage d'une habitation, le produit devra donc s'alimenter en 230V (P+N) et rentrer dans une boite électrique encastrable standard NF [4] (environ 65mm*65mm). Le dispositif pourra aussi être caché dans le mur (placo) pour disparaître.

Il doit permettre à l'utilisateur de piloter via des gestes de la main l'éclairage (base Microchip MGC3130) :

— mouvement de la main du haut vers le bas pour éteindre
— mouvement de la main du bas vers le haut pour allumer
— rotation de la main pour augmenter ou diminuer la luminosité

De plus, un contrôle par liaison bluetooth au moyen d'un télépone android devra être possible, le développement d'une application Android sera donc nécessaire.

1.2 Solutions choisies

1.2.1 Liaison Bluetooth

Pour la partie Bluetooth, le choix s'est porté sur du BLE [5] afin de réduire au maximum la consommation énergétique du système. La puce choisie est le CC2541 de Texas Instrument pour plusieurs raisons, tout d'abord celle-ci possède toutes les entrées sorties nécessaires ainsi que les protocoles de communications nécessaire à la réalisation du projet (I2C), sa consommation est moindre (environ $0.5\mu A$ en veille). De plus c'est une puce qui est utilisée dans d'autres projets d'INODESIGN, il est donc normal de porter son choix dessus si elle correspond tant d'un point de vue technique que tarifaire.

Afin de gérer la communication bluetooth, nous ajouterons une led verte et une rouge afin de pouvoir voir l'état de la connexion bluetooth. De plus, nous placerons un bouton poussoir permettant de sortir la puce du mode veille et de réactiver le bluetooth afin de pouvoir se connecter avec notre smartphone.

1.2.2 Détection de mouvement

Pour la détection de mouvement le choix s'est porté sur une puce très récente et la première du genre. Il s'agit du MGC3130 de Microchip, qui est le premier contrôleur (au monde) de geste et de suivi dans un environnement en trois dimensions utilisant un champ électrique(E-Field). Celui-ci permet, une fois connecté à 5 ou 6 électrodes de générer un champs électrique et de capter et suivre les mouvements jusqu'à

4. Norme Française
5. Bluetooth Low Energy

environ 15cm puis de transmettre les données via une connexion SPI ou I2C. Cette même connexion sera aussi utilisée dans le but de le configurer.

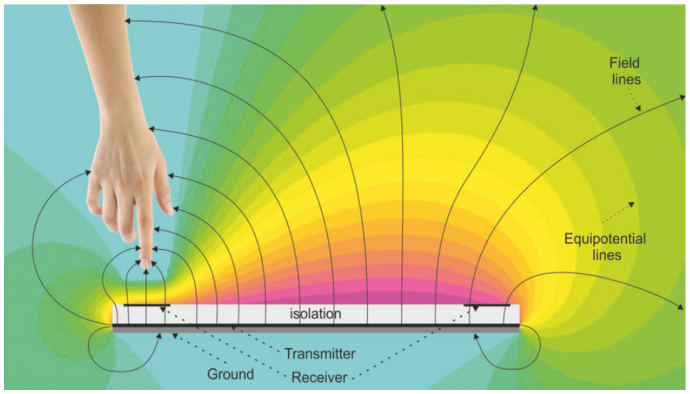


FIGURE 1 – Champ électrique

Dans la figure ci-dessus, nous pouvons voir le champs électrique géré, qui lorsqu'il est perturbé permet de connaitre la position du doigt. Cela permet aussi de détecter des mouvements particuliers comme une rotation et des mouvements de haut en bas ou gauche à droite.Le MGC3130 nous renvoi directement une position x,y,z ou un mouvement prédéfinis (exemple : mouvement de gauche à droite).

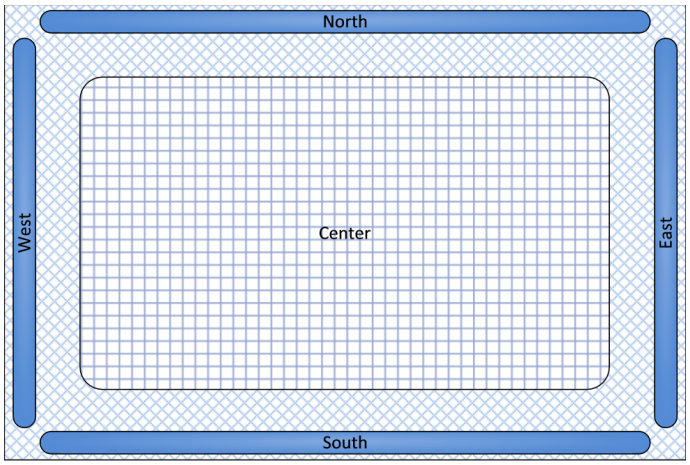


FIGURE 2 – Électrodes

Dans la figure ci-dessus on peut voir un exemple de disposition des électrodes, on peut voir 5 électrodes RX qui sont celles qui vont récupérer l'information. Nous avons aussi une sixième électrode (TX) qui couvre entièrement la surface et permet de générer le champ électrique.

1.2.3 Pilotage de l'éclairage

Le pilotage de l'éclairage n'est pas une chose aisée, dans la mesure où pour faire un produit commercial, il faut couvrir un maximum de public. Or, le système de gradation "standard" reposant sur l'utilisation de triac (explications en partie 2),

ne permet que de le faire sur des ampoules incandescentes. Ces ampoules, ne sont, à l'heure actuelle, presque plus commercialisées, les ampoules basse consommation (fluo-compactes) ainsi que les ampoules led tendent à les remplacer définitivement. Après recherches, je n'ai pas trouvé le moyen de faire de la gradation sur des ampoules basse consommation, en revanche il existe des ampoules fluocompactes compatibles avec les variateurs standards, ce qui permet donc de rester sur ce type de variateur. En ce qui concerne les ampoules LED, il faut faire une régulation en courant commandée, pour faire varier son intensité lumineuse. En raison de la place que prends le circuit, je ne peux faire une carte pouvant piloter au choix des ampoules à leds et des ampoules à incandescence, notre choix s'est donc porté sur le gradateur standard pour le premier prototype (celui visé par le PFE).

FIGURE 3 – Incandescente

FIGURE 4 – Fluocompacte

2 Conception de la partie électronique

2.1 Fonctionnement global

La carte électronique a un fonctionnement global relativement simple, elle est alimentée via le secteur (230V alternatif) et a un bornier de sortie sur lequel on vient connecter notre ampoule. Celle-ci contient le nécessaire pour contrôler la luminosité de la pièce ainsi que tout ce qu'il faut pour la détection de mouvement et la communication Bluetooth Low Energy. Lorsque l'utilisateur passe sa main devant, le mouvement est capté par le MGC3130 qui envoi l'information au microcontrôleur contenu dans le CC2541 via une communication I2C. Le microcontroleur agis ensuite en conséquence afin d'allumer, d'éteindre ou faire varier l'intensité lumineuse en agissant sur le montage de gradation.

On peut diviser cette carte en 4 parties :

— Filtrage secteur
— Zero crossing
— Gradateur
— Basse tension 3.3V

Cette carte a été réalisé au moyen du logiciel de CAO [6] Eagle.

2.2 Filtrage secteur

Notre montage étant relié au secteur il est impératif d'avoir un filtrage afin que notre montage ne perturbe pas le réseau EDF et ne soit pas non plus perturbé par d'éventuels problèmes provenant du réseau. De plus l'utilisation d'un triac est source de parasites divers car il aura pour rôle de "découper" la tension secteur. Notre filtre devra filtrer les courants parasites des modes différentiels et commun.

Qu'est ce que mode commun ?

Le mode commun c'est lorsque les fils d'alimentation sont connectés ensemble à leurs extrémités, le courant est le même dans chaque fils et sont de même sens.

Qu'est ce que le mode différentiel ?

Le mode différentiel c'est lorsque les courants circulant dans les fils d'alimentations sont de valeur égales mais de sens opposés.

L'alimentation secteur délivre un courant différentiel qui n'amène que peu de parasites. Le but va essentiellement être de filtrer les courant conduit par notre montages, ceux-ci pouvant être des deux modes.

Afin de filtrer les courants parasites de mode différentiel, j'utiliserai un condensateur de type X2(C21,C22,C24) qui sera connecté entre la phase et le neutre, j'y ajouterai une résistance en parallèle(R18) afin de décharger la capacité plus rapidement. Le but de ce filtre est d'éliminer les courants différentiels de haute fréquence (»50Hz).

6. Conception Assistée par Ordinateur

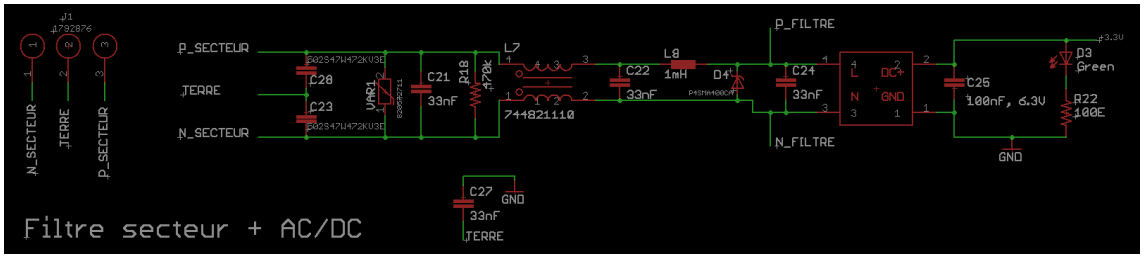


FIGURE 5 – Schéma du filtre secteur

Pour filtrer les courants parasites de mode commun à hautes fréquence, j'utiliserai plusieurs composants, tout d'abord des capacités de type Y(C23 et C28) qui permettent à hautes fréquence d'amener les courants parasites vers la terre, ils sont toujours disposés par deux (un entre phase et terre et l'autre entre neutre et terre). De plus, une inductance de mode commun sera utilisée, celle-ci étant spécialement conçu pour pour le filtrage de courants communs.

Une fois que les parasites des deux modes ont été filtré, il faut aussi ajouter des composants de protection du circuit à savoir la varistance (VAR1) protégeant contre les grosses surtensions ainsi que la diode transil protégeant contres les derniers pics de tension résiduelles.

En sortie de ce filtre nous placerons notre convertisseur AC/DC afin d'obtenir une tension de +3.3v continue servant ainsi à alimenter notre partie basse tension.

Le filtre secteur est la partie prenant le plus de place sur la carte du à la tension élevée qu'il doit supporter, en effet les composants conçus pour la tension secteur sont plus gros et plus coûteux, la difficulté a été de trouver les composants les plus petits possible et de préférence en CMS.

2.3 Zero crossing

Cette partie va nous permettre de générer des impulsions à chaque passage par 0 de la tension secteur. Ces impulsions vont nous permettre de détecter les passages par 0, chose indispensable pour la commande de notre triac. En effet, le fonctionnement du triac est tel que lorsqu'on le déclenche, il laisse passer le courant jusqu'au prochain passage par 0. Il faudra donc réenclencher le triac après chaque passage par zéro c'est pourquoi ce montage est indispensable.

Ce montage est basé sur un optocoupleur bidirectionnel et des résistances. L'utilisation d'un optocoupleur est obligatoire afin d'isoler galvaniquement la tension secteur de la basse tension. L'optocoupleur choisis est bidirectionnel puisque la tension est alternative et que l'on veut récupérer tout les passages par zéro. Les résistances placées avant la partie émettrice de l'optocoupleur permettent de fixer le courant afin qu'il ne soit pas trop élevé mais permette tout de même le déclenchement du phototransistor. Un courant d'environ 2mA étant convenable nous avons donc choisis les résistances en conséquences. Le fait de diviser la résistances en plusieurs permet une meilleure dissipation de chaleur et permet donc de réduire la taille de celles-ci. J'ai pu tester cette partie du montage et voir que lorsque l'on augmente la résistance, la

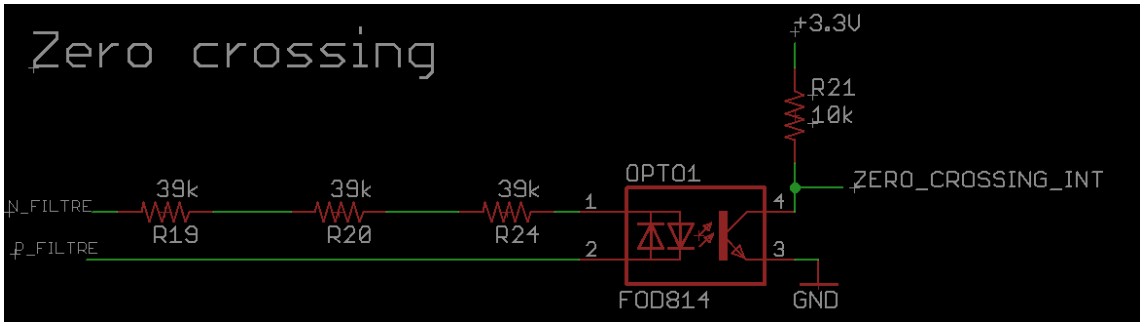


FIGURE 6 – Zero crossing

largeur d'impulsion (front montant) est plus élevée.

Ce schéma fonctionne de la façon suivante, lorsqu'il y a de la tension le transistor conduit et l'interruption zero_crossing_int est à l'état bas puis lors du passage par zéro, le transistor arrête de conduire et un front montant est généré et le niveau haut est conservé jusqu'à reconduction du transistor. Ce front montant déclenchera une interruption signalant au microcontrôleur un passage par zéro.

Ci-dessous, un relevé avec deux résistances de $56k\Omega$.

FIGURE 7 – Passages par zéro du secteur

Nous avons donc bien une impulsion toute les 10ms correspondant au passages par zéro d'une tension sinusoïdale de 50Hz. Ici la tension maximale de l'impulsion est à 12V mais sur notre montage elle sera de 3.3V.

2.4 Gradateur

La partie gradateur est basé sur un montage à base de triac, le triac ayant la propriété de conduire dans les deux sens, il est particulièrement bien adapté aux ten-

sions alternatives. Afin de pouvoir le déclencher de façon sécurisé, nous allons utiliser un optotriac pour avoir une isolation galvanique, la partie émettrice étant reliée à un IO du microcontrôleur(P2_0) qui lorsqu'elle sera a niveau haut déclenchera la conduction. Lorsque la conduction sera déclenchée, le triac conduira jusqu'au prochain passage par 0. La résistance R25 et le condensateur C26 forme un filtre snubber dont le principal intérêt est d'amortir les oscillations du à la mise en conduction du triac, ici il n'est pas obligatoire puisque le triac utilisé est indiqué snubberless mais il est quand même prévu à des fins de tests.

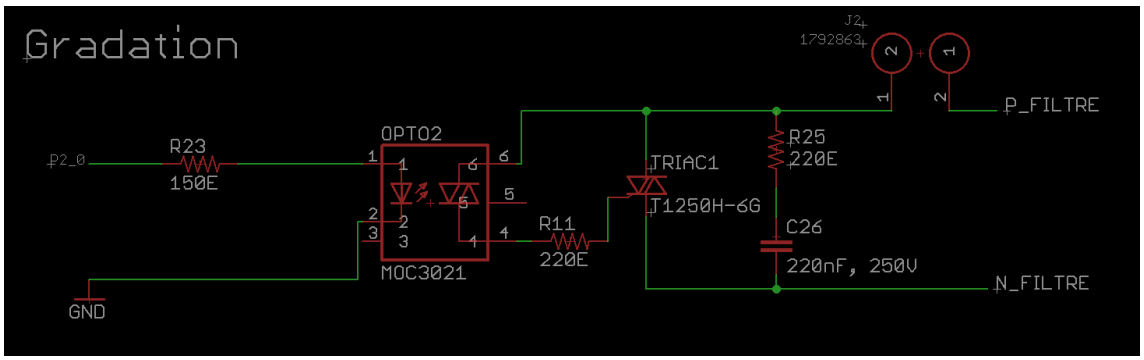


FIGURE 8 – Gradateur

Pour de faire de la variation de lumière (charge résistive), le but est de ne laisser passer qu'une partie de la sinusoïde, pour cela lorsqu'il y aura un passage par zéro le microcontrôleur attendra un certains temps (<10ms) avant de déclencher le triac. Ce temps dépendra de la luminosité voulue, plus le retard sera grand et moins l'ampoule s'illuminera. Le schéma ci-dessous montre l'allure de la tension aux bornes de l'ampoule, on peut voir le retard t avant la mise en conduction.

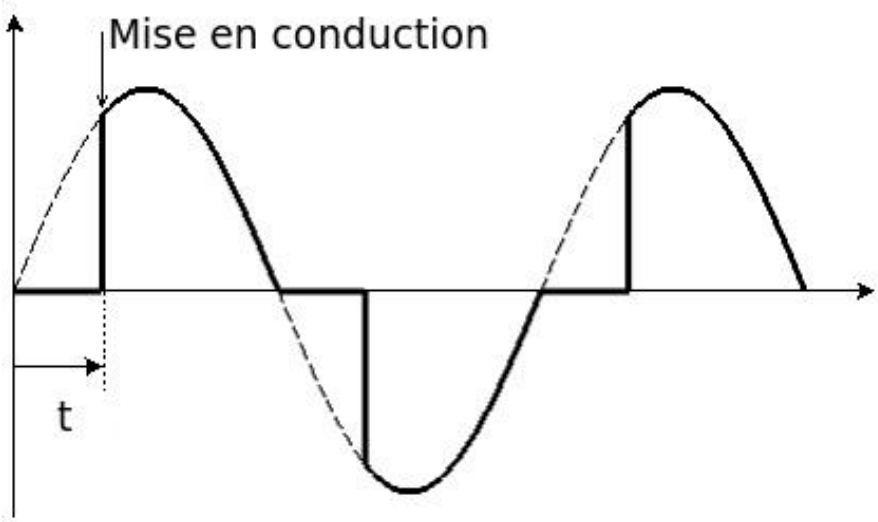


FIGURE 9 – Tension aux bornes de l'ampoule

2.5 Basse tension 3.3V

Cette partie est alimentée à l'aide du transformateur AC/DC, le CC2541 est le composant central puisque c'est lui le microcontrôleur. Celui-ci est relié au travers de résistances à deux leds de notifications (une rouge et une verte), ainsi qu'à un bouton poussoir connecté sur une interruption externe afin de pouvoir faire entrer et sortir la puce du mode veille. Il est aussi connecté au MGC3130 à travers d'une liaison I2C, le CC2541 étant le maître et le MGC3130 l'esclave.

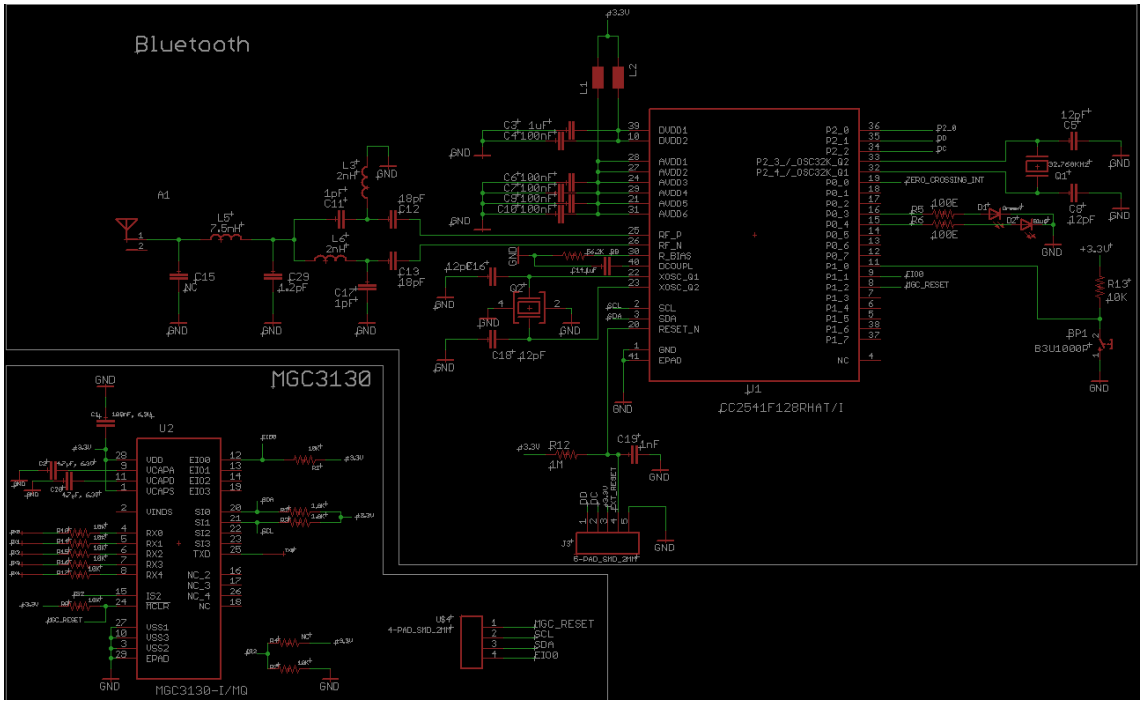


FIGURE 10 – Schéma basse tension

2.6 Routage

Le routage de cette carte n'est pas simple à effectuer car plusieurs contraintes se sont présentées, tout d'abord l'isolation entre toutes les pistes liées au 230V et n'étant du même signal, doivent être séparées d'au moins 3mm. De même que les pistes basses tensions(électrodes comprises) doivent être éloignées d'au minimum 3mm des pistes liées au 230V. L'autre contrainte est la taille de la carte qui ne doit pas excéder 65mm*65mm afin de pouvoir rentrer dans un boitier encastrable d'interrupteur. Ces deux contraintes m'ont données beaucoup de difficultés au routage qui fut très long à réaliser. J'ai dû effectuer quelques changements sur le choix de certains composants utilisés pour le 230V, les condensateurs choisis n'étaient pas en CMS car beaucoup plus cher mais leur taille était un véritable handicap, de plus avoir des trous dans la carte avec du 230V n'est pas sécurisant sachant que tous les composants (hormis les leds de notifications et le boutons poussoir) sont du côté bottom afin d'avoir les électrodes libres. Il n'existe pas de convertisseur AC/DC en CMS j'ai donc du

mettre du traversant mais signaler, sur la sérigraphie, les endroits à ne pas toucher. Idem sur le bottom où tout les endroits où il y a la tension secteurs sont hachurés.

J'ai donc router cette carte sur 4 couches afin d'avoir la place de faire correctement passer les pistes. Les couches sont utilisées comme suit :

1. Top : sur cette couche il y aura les leds de notifications, celle indiquant l'alimention en 3.3V ainsi que le bouton poussoir. C'est aussi sur cette couche que ce trouverons les électrodes de réception (Rx). Sur l'annexe 3.5 page 25, vous pourrez voir la couche top en rouge ainsi que le placement des composants en blanc et sa sérigraphie.
2. Inner layer 1 : sur cette couche se trouve l'électrode de transmission Tx. Voir annexe 3.5 page 26
3. Inner layer 2 : sur cette couche se trouve un plan de masse pour la partie 3.3V située en haut à droite ainsi qu'un plan de terre pour la partie secteur. De plus, elle a été utilisée afin de faire passer les pistes ne pouvant pas être routée sur la couche bottom. Voir annexe 3.5 page 27
4. Bottom : sur cette couche, nous trouverons tout le reste des composants, la partie 3.3V et la partie secteur étant nettement séparées. Sur l'annexe 3.5 page 28 pourrez voir la couche bottom ainsi que sa sérigraphie et son placement de composants.

Afin d'assurer le bon fonctionnement du système la carte est fabriquée de la façon suivante :

— les couche top et bottom auront une épaisseur de cuivre de $18\mu m$
— les couches internes auront une épaisseur de cuivre de $35\mu m$
— l'isolation entre la couche 1 (top) et 2 (innerLayer1) sera de 0.9mm
— l'isolation entre la couche 2 et 3 (innerLayer2) sera de 0.45mm
— l'isolation entre la couche 3 et 4 (bottom) sera de 0.54mm
— l'épaisseur total sera donc de 1.996mm

3 Bluetooth Low Energy et android

Le bluetooth low energy aussi appelé Bluetooth LE ou encore Bluetooth smart est une nouvelle version du bluetooth (V4.0) visant à réduire considérablement la consommation d'énergie par rapport au bluetooth "classique" (V1,V2,V3) tout en gardant un débit équivalent(environ 1 Mbit/s). Il a été introduit par Nokia sous le nom de Wibree en 2006 et est devenu un nouveau standard bluetooth en 2010. Il est essentiellement utilisé dans le monde des objets connectés comme des bracelets permettant de suivre la fréquence cardiaque ou des montres. Le bluetooth est basé sur des profils qui ont été définies par la Bluetooth SIG [7] qui est une organisation qui gère tout les standards liés à la technologie bluetooth.

Le bluetooth smart a introduit de nouveaux profils et certains qui étaient utilisés par le bluetooth "classique" ne sont plus supportés, comme celui lié au protocole de communication série ou encore l'AD2P (permettant la distribution audio) dans le but d'économiser de l'énergie. En effet, l'AD2P, par exemple, est un protocole où il y a un flux de données permanent, or le but du BLE est d'économiser le maximum d'énergie et de communiquer qu'à des moments ponctuels.

Le bluetooth Low Energy est la technologie la plus adaptée à notre projet, puisqu'elle permet une communication très faible consommation avec un rayon d'action d'environ 10m. Contrairement à d'autres technologies sans contact comme le NFC [8] qui ne permet une communication qu'à quelques centimètres et obligeant ainsi à être près de l'interrupteur.

3.1 Profils

Un profil donne des spécifications sur comment doit fonctionner l'appareil dans une application particulière. Il existe plusieurs profils définis par la Bluetooth SIG, les principaux sont :

Generic Access Profile (GAP)
Il définit les procédures génériques de recherche d'appareils, de connexion et de sécurité, c'est le profil de base dont tous les autres héritent.

Generic Attribute Profile (GATT)
Il est utilisé par tous les appareils utilisant le BLE, il donne une interface de programmation de profils. Tous les profils utilisent la même terminologie, la même architecture.

Profils liés à la santé

HTP : pour les dispositifs de mesure de température médicale.

GLP : pour les moniteurs de glucose dans le sang.

BLP : pour la mesure de pression artérielle.

Profils liés au sport

HRP : pour les dispositifs de mesure de fréquence cardiaque.

7. Special Interest Group
8. Near Field Communication

CSCP : pour les capteurs de cadence et de vitesse d'une roue de vélo.

RSCP : pour la mesure de cadence et de vitesse d'un coureur.

LNP : pour la localisation et la navigation.

Autres profils : la liste ci dessus n'est pas exhaustive, il existe encore d'autres profils définis par la bluetooth SIG.

3.1.1 GAP

Le profil GAP permet de faire fonctionner l'appareil BLE dans 4 rôles différents :

Broadcaster : le module envoi des données sans initiation de connexion. Exemple : un capteur de température.

Central : l'appareil scanne la réception de données et peu initier une connexion, il agit en tant que maître. Exemple : un smartphone.

Peripheral : le module agit comme un périphérique nécessitant une connexion afin d'opérer en tant qu'esclave. Exemple : une montre.

Observer : le module scanne la réception de données mais ne peut initier une connexion . Exemple : un affichage de température.

De plus, le BLE permet de combiner des rôles :

Peripheral et Observer : le périphérique scanne la réception de données sans connexion mais peut aussi fonctionner en mode connecté en tant qu'esclave.

Broadcaster et peripheral : le périphérique peut envoyer des données qui ne nécessitent aucune connexion mais également fonctionner en mode connecté en tant qu'esclave.

Central et Broadcaster : le périphérique peut envoyer des données qui ne nécessitent aucune connexion mais peut aussi initier une connexion en tant que maître.

3.1.2 GATT

La terminologie du protocole GATT est celle utilisée par tous les autres profils, elle se définie comme suit :

Client : périphérique qui peut envoyer des commandes GATT et recevoir des réponses. Exemple : un smartphone.

Server : périphérique qui peut recevoir des commandes GATT et envoyer des réponses. Exemple : un capteur de fréquence cardiaque.

Characteristic : donné transférée entre le client et le serveur, elles sont accessibles grâce à leur UUID ou leur "handle"(obligatoire dans le cas d'une écriture).

Service : ensemble de "characteristics" se relatant à une fonction particulière.

Descriptor : informations supplémentaires sur une "characteristic", comme l'unité de la mesure envoyée.

Identifier : “Descriptor”, “service” et “characteristic” sont désignés comme des attributs et identifiés par des UUIDs. La bluetooth SIG a réservé des gammes d’UUIDs pour les attributs standards.

Ce protocole donne accès à des commandes particulières afin que le client puisse découvrir les informations du serveur(server).

1. Découvertes des UUIDs des services primaires
2. Trouver un service au moyen de son UUID
3. Trouver les services secondaire grâce au premier service
4. Découvrir toutes les “characteristics” d’un service
5. Trouver la “characteristic” associé à un UUID
6. Lire les “descriptors” pour une “characteristic”

De plus, les “characteristics” peuvent être lues ou écrites, la lecture se fait dans le sens serveur vers client et l’écriture dans le sens inverse. Le client peut aussi demander au serveur d’être notifié d’un changement sur une “characteristic” particulière comme un changement de température. Dans ce cas là, dès que le serveur aura un changement de la valeur de la “characteristic” demandée, il notifiera le client de l’arrivée d’une nouvelle valeur.

3.2 Protocole Bluetooth Low Energy

Le protocole BLE forme la pile (stack) ci-dessous.

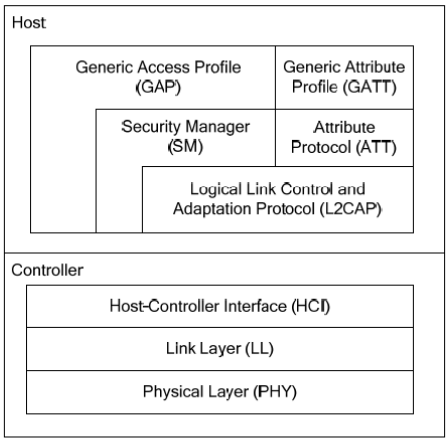


Figure 1: BLE Protocol Stack

FIGURE 11 – BLE stack

Lorsque l’on veut développer une application, nous devons obligatoirement configurer les couches de hauts niveaux : GAP et GATT. Les couches de niveau inférieures, permettent des utilisations particulières que je n’ai pas encore abordé.

3.3 Programmation du CC2541

Cette puce CC2541 intègre un microcontrôleur 8051 et la programmation se fait en C à l'aide du logiciel IAR Embedded Workbench 8051. Le flashage de la puce se fait, quant à lui, au moyen de la sonde CC debugger (programmation ISP [9]). Le diagramme bloc ci dessous permet d'identifier les différents modules ainsi que les différentes entrées sorties disponibles.

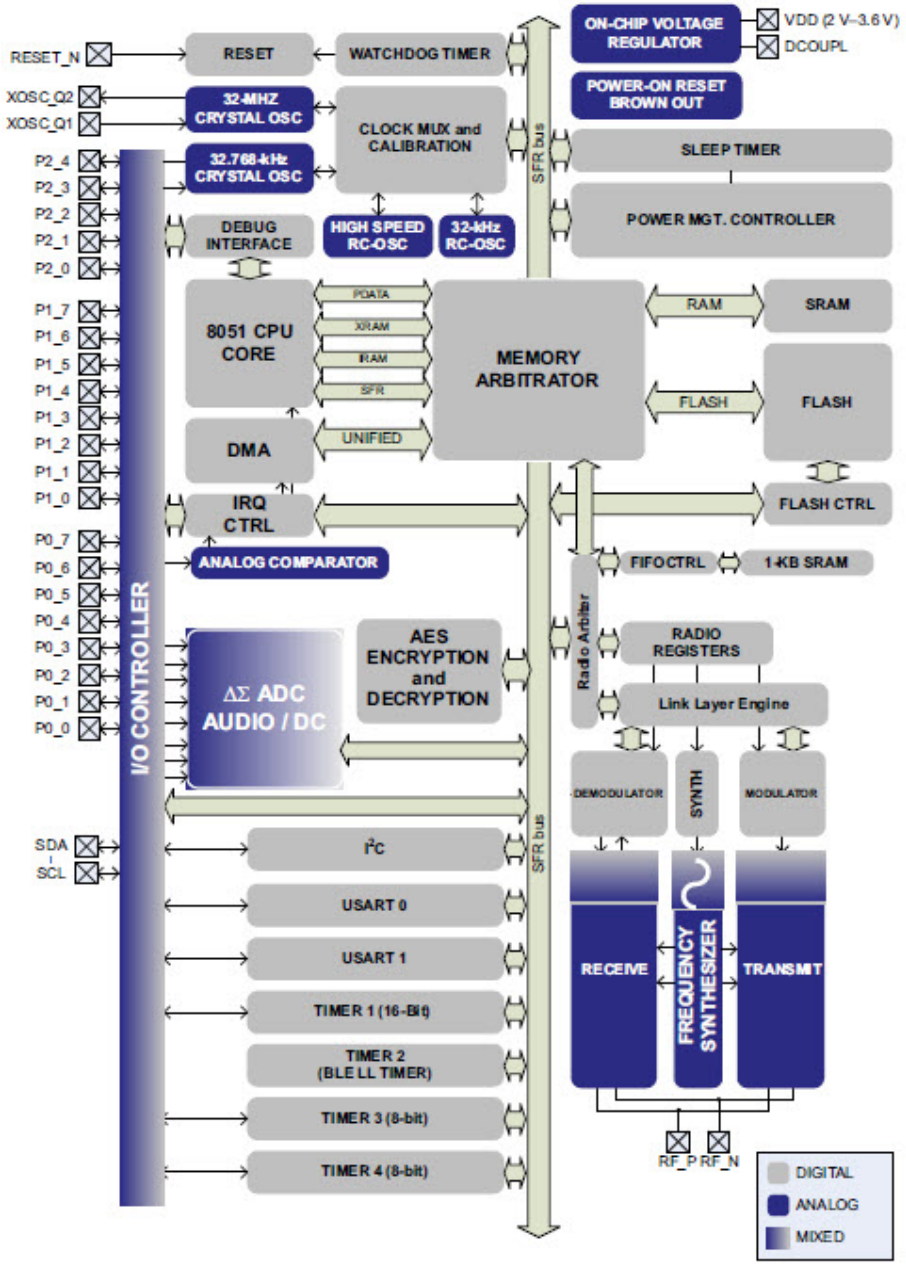


FIGURE 12 – Schéma bloc CC2541

9. In-System-Programmable Flash

3.3.1 Différentes parties d'un programme sur CC2541

Le programme peut être divisé ainsi :

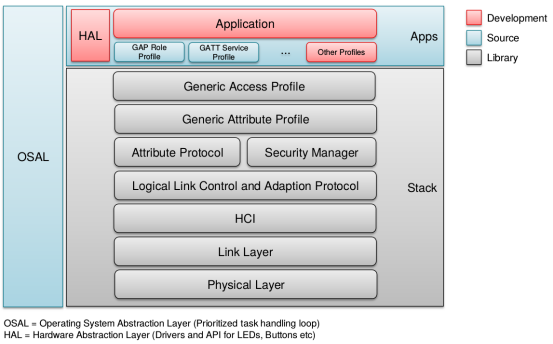


FIGURE 13 – Structure d'un programme

On peut ainsi remarquer 3 grands ensembles :

Operating System Abstraction Layer (OSAL)

C'est la couche de plus haut niveau, c'est elle qui va permettre d'exécuter les différents événements avec un ordre de priorité défini. Celui d'ID 0 ayant la plus haute priorité et 11 la plus basse.

ID	Task
0	Link Layer
1	HAL
2	HCI
3	OSAL Callback Timer
4	L2CAP
5	GAP
6	GATT
7	SM
8	GAP Role
9	GAP Bond Manager
10	GATT Server
11	Application

Stack

C'est la pile complète du protocole Bluetooth Low Energy développée par Texas Instrument.

Apps

C'est dans cet ensemble que se trouve les fichiers que l'on doit modifier pour créer notre application. On y trouve les couches :

HAL [10] **:** c'est dans cette couche que l'on code nos interfaces pour gérer facilement la partie hardware tel que les ADC, l'UART, l'I2C, les LEDs, les PWM [11]... Il faut donc l'adapter à notre carte électronique.

11. Pulse Width Modulation : modulation par largeur d'impulsion

Application : c'est dans cette couche que l'on retrouvera le comportement de notre application. C'est la couche "haute" de la programmation.

Other profile : dans cette couche nous coderons nos propres profils.

3.3.2 Mon programme

La carte étant en cours de production à l'heure où je rédige ce rapport, je n'ai pas pu tester mon programme sur la carte finale. J'ai donc utilisé une carte ayant déjà été fabriquée au sein d'INODESIGN et comprenant un CC2541, 5 leds et un capteur UV (non utile ici). Je n'ai donc pu que simuler certains comportements de la carte finale.

Gestion des connexions et mode de fonctionnement

Le module fonctionnera en mode "peripheral" ainsi il attendra que le téléphone instancie la connexion et lui envoi les informations. Afin de pouvoir se connecter dessus, il faudra faire passer la puce en mode "advertising", pour cela il suffit d'appuyer un bouton poussoir qui est relié à une interruption. La led du milieu s'allumera alors (et ce à chaque fois que nous serons en mode "advertising"), lorsque le téléphone se sera connecté cette led s'éteindra et la rouge s'allumera signalant ainsi que la connexion a bien été effectué et s'éteindra dès la réception de valeur pour la gradation. Si l'on se deconnecte, la puce repasse automatiquement en mode "avertising". Un second appuie sur le bouton permettra alors de remettre la puce en mode veille.

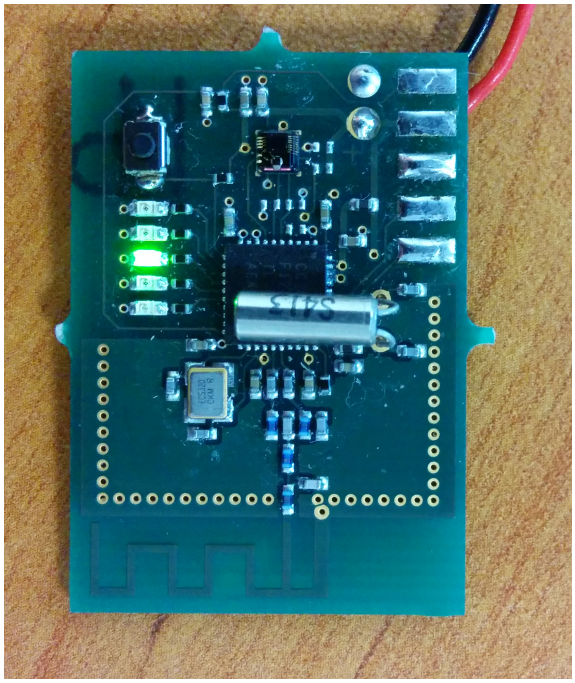


FIGURE 14 – Mode "advertising", attente d'une connexion

Récupération des informations de gradation par bluetooth

Pour cette partie j'ai créé un profil afin de pouvoir échanger des informations entre le téléphone et le microcontrôleur. Ce profil ne possède qu'un seul service.

Ce service a pour nom LightControl et possède comme UUID 0x90 + l'attribut standard donnée par la bluetooth SIG (xxxxxxxx-0000-1000-8000-00805F9B34FB) soit 0090-0000-1000-8000-00805F9B34FB.

Il possède donc 2 "characteristics" :

1. GestEN_CHAR : permet d'activer ou non la reconnaissance de mouvement
   - UUID : 0x91 + attribut standard
   - Handle : 0x25
   - Permission : R/W (lecture et écriture)

   Si on écrit :
   - 0x00 : on désactive la reconnaissance de mouvement(MGC3130 en deep mode)
   - 0x01 : on active la reconnaissance de mouvement (MGC3130 en self wake up mode)
2. DimVal_CHAR : en écrivant dessus une valeur entre 0 et 255, cela permet de choisir l'intensité lumineuse. 0 correspondant à une lumière éteinte et 255 à une lumière complétement allumée.
   - UUID : 0x92 + attribut standard
   - Handle : 0x28
   - Permission : R/W (lecture et écriture)

L'écriture sur une characteristic déclenche obligatoirement une fonction callback dans le programme principal permettant ainsi d'agir tout de suite en fonction de ce qui a été demandé : appel de la fonction de gradation ou d'activation/désactivation de la reconnaissance de mouvements.

Gestion de la gradation

La gestion de la gradation est faite de la façon suivante, lorsque la lumière sera allumée et qu'une valeur de luminosité sera demandée, nous configurons le Timer 1 avec une valeur de comptage maximum dependant de DimVal_CHAR et nous activons l'interruption externe du zéro crossing. Plus la valeur sera faible et plus le timer devra compter dans la limite de 10ms (demi période de 50Hz). Eteindre la lumière revenant à ne plus faire conduire le triac, le timer pourra être désactivé afin d'économiser de l'energie tout comme l'interruption du zéro crossing. Idem dans le cas d'une demande de puissance maximale où l'on pourra laisser la gachette du triac à l'état haut.

Lorsque ceci est fait, lorsque nous détecterons un passage par 0, nous activerons le timer de sorte qu'il génère une interruption interne lorsqu'il arrive à la valeur de comptage. Cette interruption sera le moment d'activer le triac et de desactiver le timer qui sera donc réactivé au prochain passage par 0.

De cette manière, nous sommes capable de ne laisser passer qu'une partie de la sinusoide et de faire de la gradation de lumière.

Cette partie, n'est pour le moment que théorique, n'ayant pas encore la carte finale je n'ai pu faire de tests pratiques.

Gestion de la reconnaissance de mouvement

Cette partie de programation sera basée sur la communication I2C, la puce MGC3130 communique par le biais de messages de la GestIC library. C'est cette bibliothèque qui nous permet de configurer la puce ou de recevoir des informations quant aux gestes détectés. Cette bibliothèque est obligatoire pour la reconnaissance de gestes mais elle doit être flashée sur le MGC3130, en effet la puce n'est livrée qu'avec un bootloader et nous devons donc lui flasher notre propre version GestIC configurée par rapport à notre carte.

Je me suis rendu compte de ceci lorsqu'au moyen d'un montage avec un Arduino j'ai récupéré la première trame envoyée par la puce. Celle-ci donnant des informations de firmware/hardware sur la puce.

La trame reçu est la suivante : 84 00 00 83 00 63 80 fa 03 64 00 00 00 00 ....

En l'analysant d'après la datasheet (GestIC library) :

Les 4 premiers octets forment le header.
- 0x00 : le flag est à 0
- 0x00 : le message a été transmis avec le numéro de séquence 0
- 0x83 : l'identifiant du message est 0x83, c'est bien un message "Fw_version_info"

Les octets suivant forment le payload
- 0x00 : c'est le FwValid, il indique 0x00 c'est à dire qu'il n'a pas de librairie GestIC valide d'installée dans la puce. Si une librairie GestIC était disponible nous aurions 0xAA.
- 0x63 0x80 : c'est la version du hardware qui est donc 99.128
- 0xfa : l'adresse de début des paramètres est 250 * 128 = 32000
- 0x03 0x64 : library loader version : 100.3

Afin de m'assurer de cette information et dans le but de connaître les différentes solutions pour remédier à ce problème j'ai contacté une personne de la société Microchip. Pour moi la seule solution était d'acheter le kit de développement afin d'obtenir le hillstar I2C bridge permettant de connecter la puce au logiciel Aurea et de pouvoir la flasher. Ce kit coutant près de 160euros, j'ai préféré attendre la réponse de Microchip pour être sur qu'il n'y avait pas d'autre solutions moins onéreuses. Après près d'un mois (entre relance et changement de contact), j'ai eu pour réponse ce que je pensais, à savoir qu'il fallait flasher la bibiliothèque dans la puce, je n'ai donc malheureusement pas pu commander le kit en temps et en heure. C'est à cause de ce point que je n'ai pu faire de test sur la carte de démonstration.

3.4 Application android

La programmation Android s'est faite en java [12] au moyen de l'ADT (Android Development Tool) fourni par google. Celui-ci est basé sur le logiciel "Eclipse", bien connu dans le monde de la programmation.
Fonctionnement de l'application :

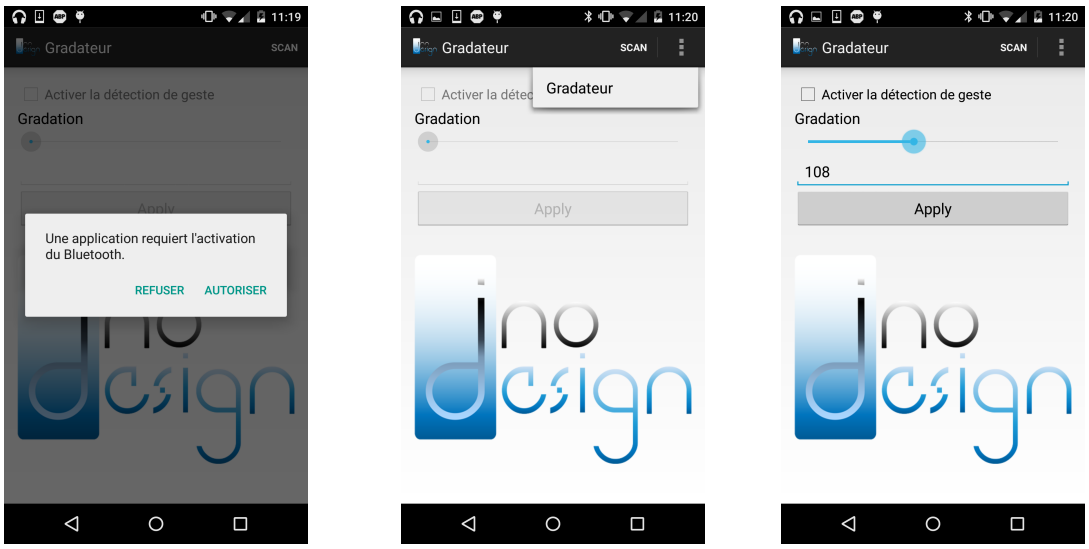
1. Au lancement, une boite de dialogue s'ouvre si le bluetooth est désactivé. Celle-ci permet d'activer le bluetooth.
2. Scan des périphériques BLE aux alentours (seulement ceux nommés "Gradateur"), grâce au bouton scan en haut à droite.
3. Avec le boutons d'option (les trois petits points) situé à côté du bouton "scan", on choisit le périphérique sur lequel se connecter. Juste après la connexion, le programme va lire la valeur de GestEn_CHAR afin de checker ou non la checkbox gérant la reconnaissance de geste.
4. Les différents boutons seront valides et le resteront le temps que la connexion est valide. On peut alors activer ou non la reconnaissance de gestes ainsi que faire varier la lumière au moyen de la bar ou en rentrant directement une valeur puis en validant par "apply".

Descriptif des classes utilisées :

MainActivity : classe principale qui gère toute les interactions avec l'interface graphique et qui met en place les callbacks sur les méthodes liées à la communication bluetooth low energy.

LightControl : classe avec les méthodes gérant l'activation/désactivation de la reconnaissance de geste ainsi que la l'envoi des valeurs pour la variation de lumière mais aussi les différent UUIDs des services et "characteristics".

UI : L'interface graphique est gérée par le fichier activity_main.xml



12. langage de programmation orienté objet

Afin de tester le fonctionnement de ce code, j'ai utilisé la carte avec les leds, celles-ci forment un bargraph dont le nombre de led allumé augmente en fonction de la luminosité demandée(une led par pas de 50).

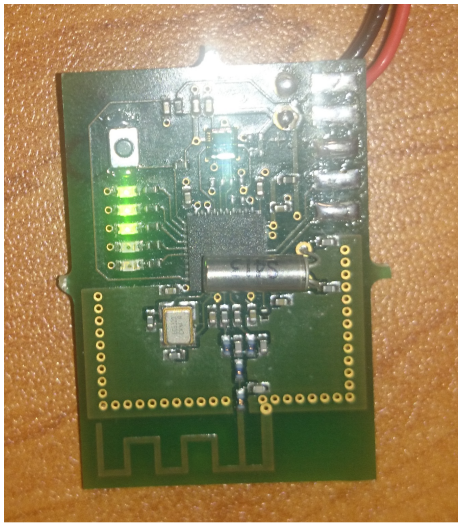


FIGURE 15 – Luminosité entre 100 et 150

3.5 Amélioration envisagées

Plusieurs fonctionnalités pourront être rajoutées afin d'égayer le système :

— récupérer les temps d'allumage de la lumière afin de pouvoir en faire des statistiques que l'on afficherait sur le téléphone

— une amélioration indispensable pour pouvoir utiliser plusieurs gradateurs serait de pouvoir changer le nom afin qu'ils ne s'appellent pas tous "gradateur".

Conclusion

Pour conclure, ce projet, malgrès que non terminé à l'heure d'aujourd'hui, m'a beaucoup appris sur la conception d'une carte électronique, j'ai pu utiliser des composants que je n'avais encore jamais utilisé. De plus les contraintes liées au 230V ainsi qu'à la dimension de la carte ont fais de cette carte un véritable petit défi pour arriver à faire un routage correct. J'ai aussi beaucoup appris quant à la programmation du CC2541 ainsi que de la programmation android même si celle-ci ne me sert qu'à faire une application de test.

J'attends maintenant de recevoir la carte afin de pouvoir tester son bon fonctionnement, j'espère avoir bientôt un kit de développement Hillstar afin de pouvoir tester la reconnaissance de geste qui reste, à l'heure actuelle, le point le moins avancé du projet.

Durant ce projet, j'ai pu mettre en application plusieurs compétences que ce soit d'un point de vue électronique mais aussi informatique, le fait de partir de zéro m'a permis de voir les différentes étapes quant à la conception d'un projet. J'ai aussi pu me rendre compte que la conception d'une carte électronique necessite beaucoup de temps, de recherche et d'expérience afin de ne pas perdre de temps.

Annexes

Couche Top

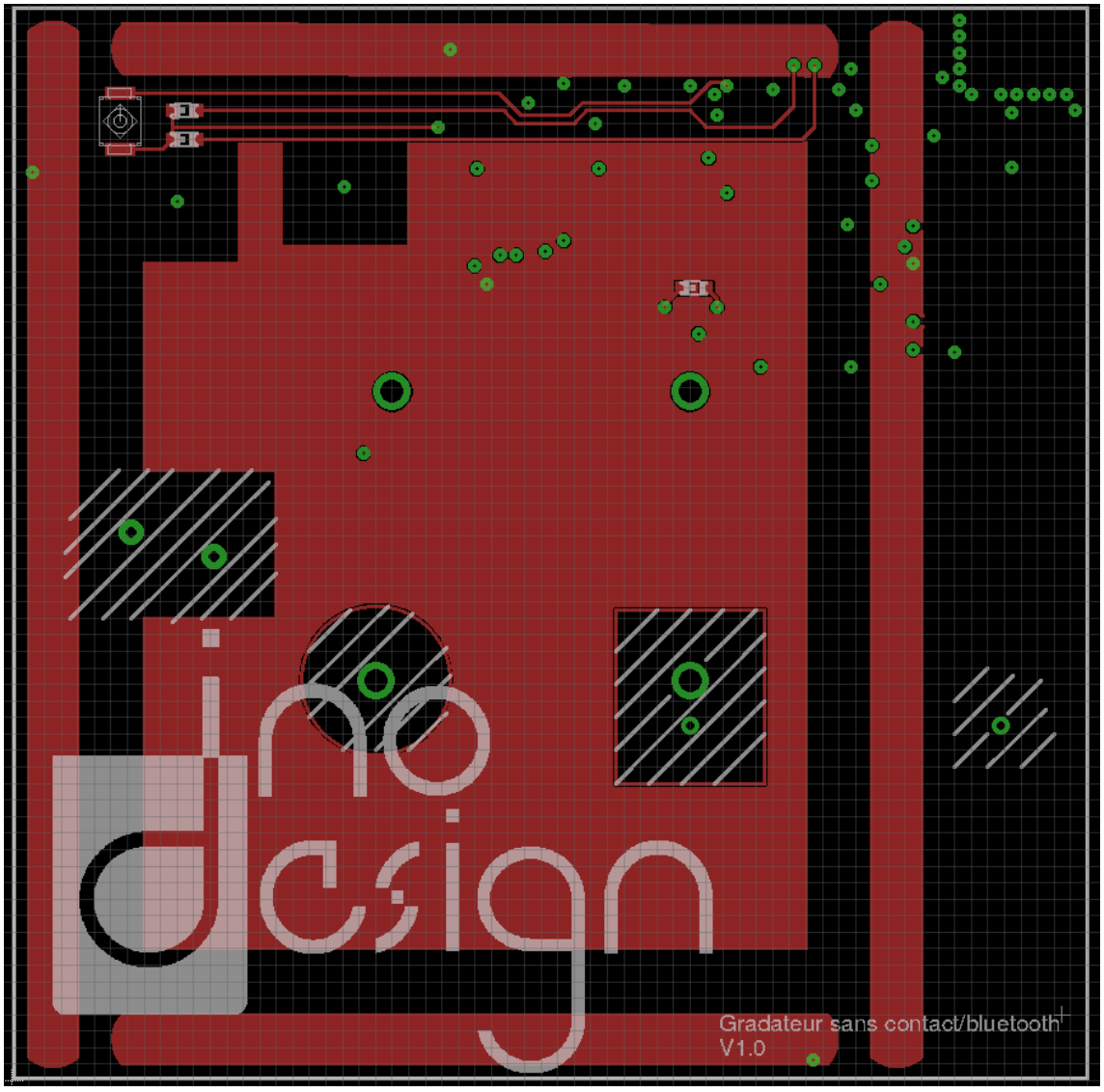


FIGURE 16 – Top + sérigraphie + placement composants

Couche 2

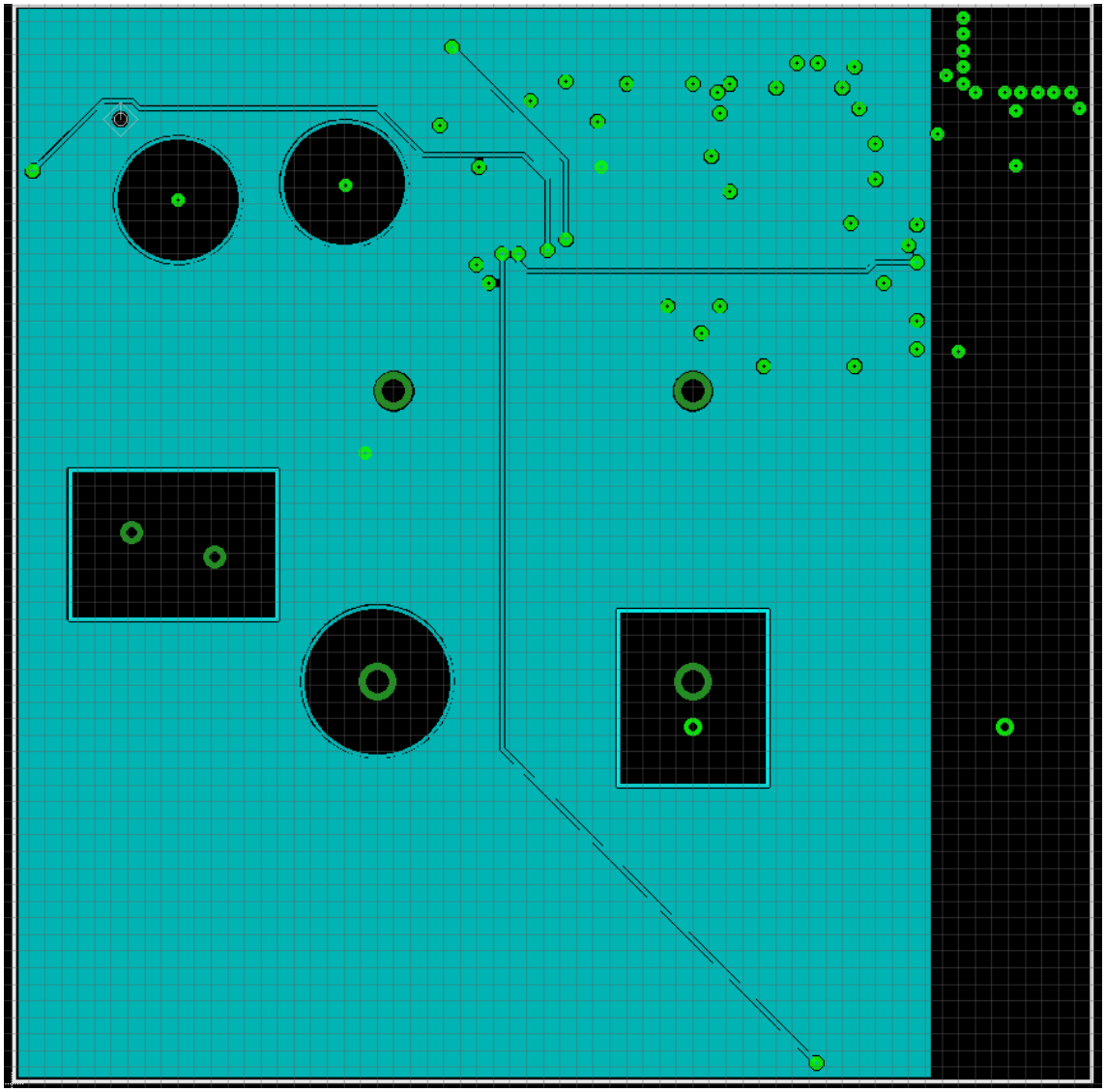


FIGURE 17 – Inner Layer 1

Couche 3

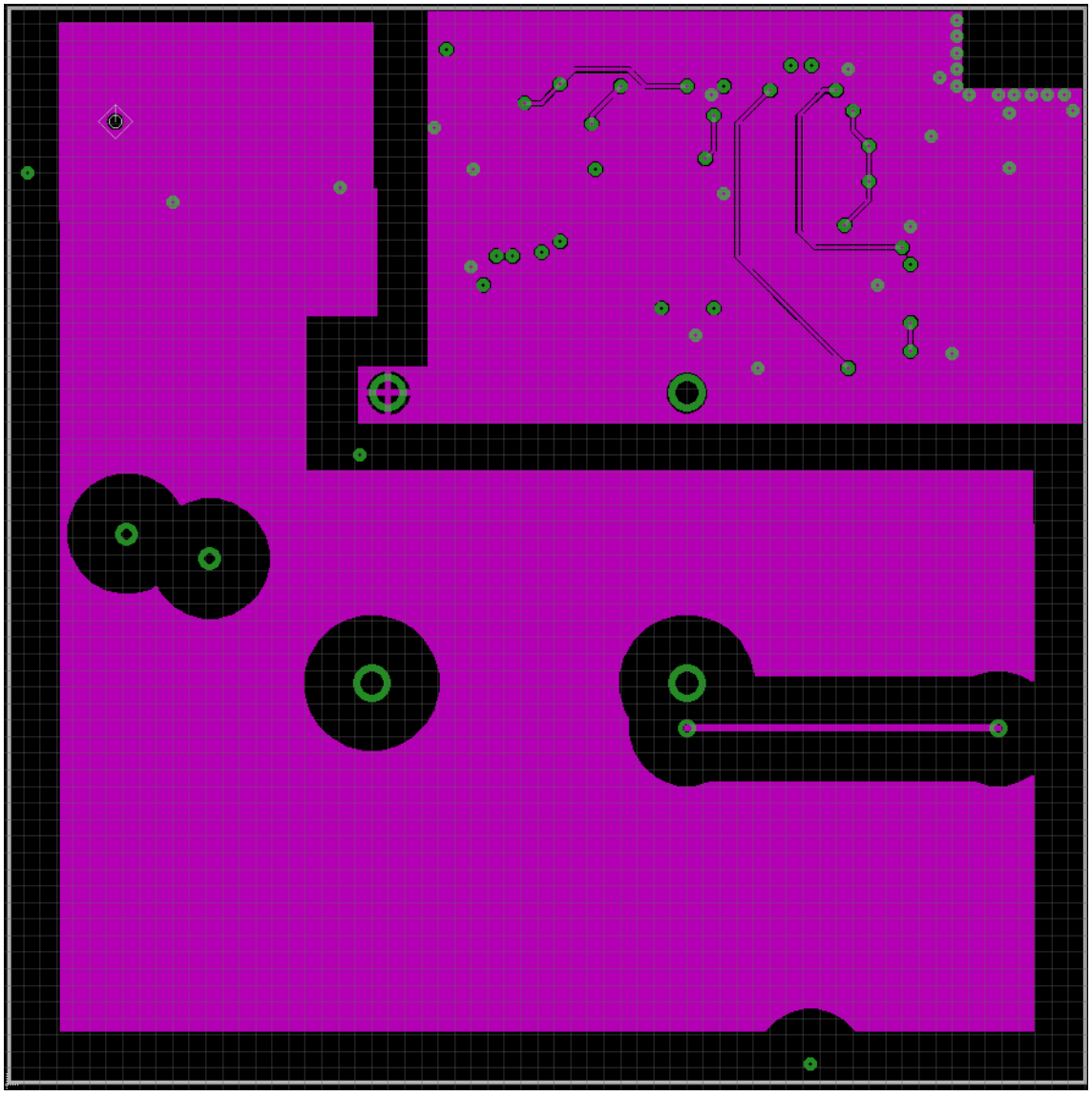


FIGURE 18 – Inner Layer 2

Couche bottom

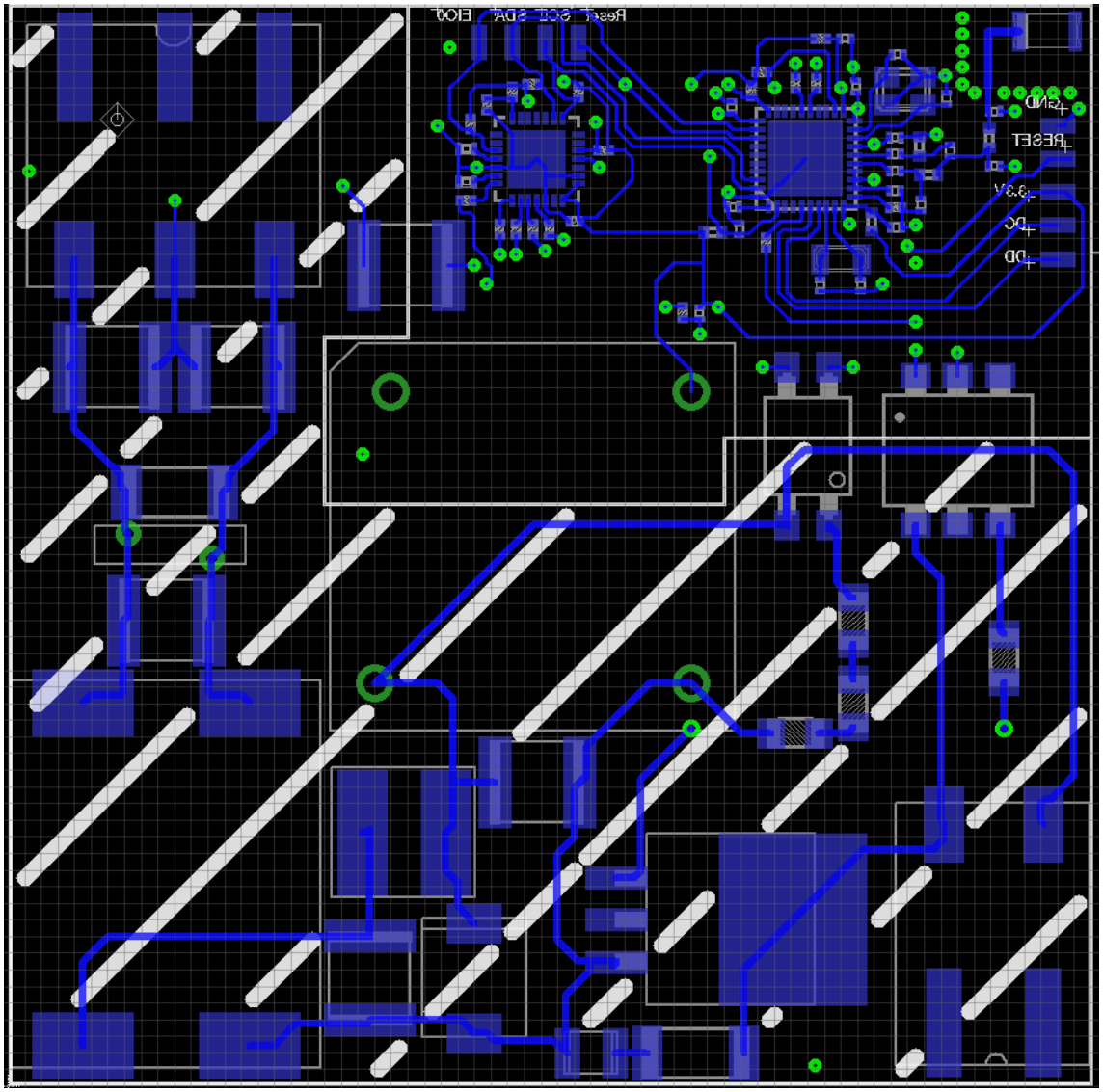


FIGURE 19 – Bottom + sérigraphie + placement composants