

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'aimerais remercier Polytech Lille pour me permettre de faire ce projet en corrélation avec la société INODESIGN. J'aimerais 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

Introduction	4
1 Présentation du projet et des solutions technologiques	5
1.1 Cahier des charges	5
1.2 Solutions choisies	5
1.2.1 Liaison Bluetooth	5
1.2.2 Détection de mouvement	5
1.2.3 Pilotage de l'éclairage	6
2 Bluetooth Low Energy	8
2.1 Profils	8
2.1.1 GAP	9
2.1.2 GATT	9
2.2 Protocole Bluetooth Low Energy	10
2.3 Programmation du CC2541	11
2.3.1 Différentes parties d'un programme sur CC2541	11
3 Conception de la partie électronique	14
3.1 Fonctionnement global	14
3.2 Partie tension secteur	14
3.3 Partie basse tension	15
3.4 Carte de démonstration	15
Conclusion/prévisions	16

Table des figures

1	Champ électrique	6
2	Électrodes	6
3	Incandescente	7
4	Fluocompacte	7
5	Led	7
6	BLE stack	10
7	Schéma bloc CC2541	12
8	Structure d'un programme	13
9	Triac	14

Introduction

Durant la troisième année mon cursus ingénieur spécialité Informatique Micro-électronique et Automatique, je suis en train d'effectuer un projet de fin d'étude¹ 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 FPGA². 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 ainsi que les différentes solutions retenues. Dans une seconde partie, je vous expliquerai le fonctionnement du module BLE³ ainsi que la façon de le programmer. Pour finir, je décrirais les différentes parties de la carte électronique que je devrais réaliser afin de réaliser le système complet.

-
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 220V (P+N) et rentrer dans une boîte électrique encastrable standard NF⁴. 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

L'utilisateur pourra récupérer l'information de consommation électrique sur son smartphone et pourra aussi allumer/éteindre ou faire une gradation tout cela en liaison Bluetooth Low Energy (Bluetooth 4.0).

1.2 Solutions choisies

1.2.1 Liaison Bluetooth

Pour la partie Bluetooth, le choix s'est porté sur du BLE⁵ 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 RGB 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

4. Norme Française

5. Bluetooth Low Energy

de générer un champs électrique et de capter et suivre les mouvements jusqu'à 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.

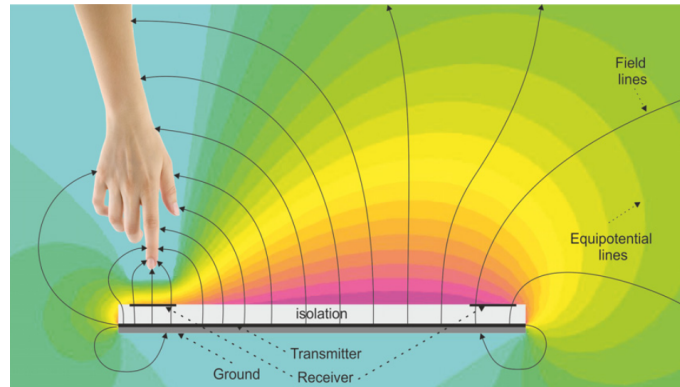


FIGURE 1 – Champ électrique

Dans la figure ci-dessus, nous pouvons voir le champs électrique géré, qui lorsqu'il est perturbé permet de connaître 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).

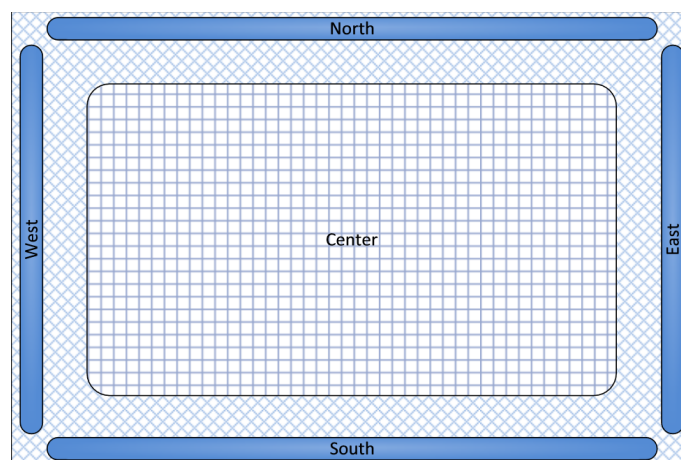


FIGURE 2 – Électrodes

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 aisé, 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 3), ne permet que de le faire sur des ampoules incandescentes. Ces ampoules, 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 pour le moment 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 suffit d'une régulation en courant commandée, pour faire varier son intensité lumineuse. Le but de notre système sera donc de pouvoir, si possible, avoir le choix entre les ampoules incandescentes/Fluocompactes compatibles et les ampoules Led. Dans un premier temps, je ferais un prototype permettant de commander une ampoule incandescente.



FIGURE 3 – Incandescente



FIGURE 4 – Fluocompacte



FIGURE 5 – Led

2 Bluetooth Low Energy

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 SIG⁶ 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 NFC⁷ qui ne permet une communication qu'à quelques centimètres et obligeant ainsi à être près de l'interrupteur.

2.1 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.

6. Special Interest Group

7. 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.

2.1.1 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.

2.1.2 GATT

La terminologie du protocole GATT est celle utilisée par tous les autres profils, elle se définit 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 : donnée 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.

2.2 Protocole Bluetooth Low Energy

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

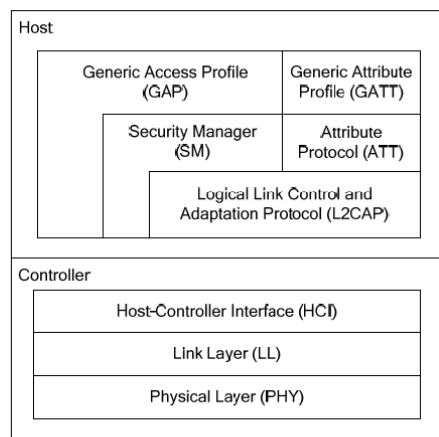


Figure 1: BLE Protocol Stack

FIGURE 6 – 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.

2.3 Programmation du CC2541

Au cours de mes horaires de travail j'ai pu développer sur le module CC2541 pour un projet de capteur UV connecté (en cours de développement) ainsi que de créer une application Android de démonstration. Ce qui me permet de pouvoir d'avoir pris en main la programmation Android avec utilisation du BLE ainsi que d'expliquer globalement comment fonctionne un programme sur la puce BLE.

Cette puce CC2541 intègre un microcontrôleur 8051 et la programmation se fait en C à l'aide du logiciel IAR Embedded Workbench 8051. Le diagramme bloc ci dessous permet d'identifier les différents modules ainsi que les différentes entrées sorties disponibles.

2.3.1 Différentes parties d'un programme sur CC2541

Le programme peut être divisé ainsi :

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 :

HAL⁸ : 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

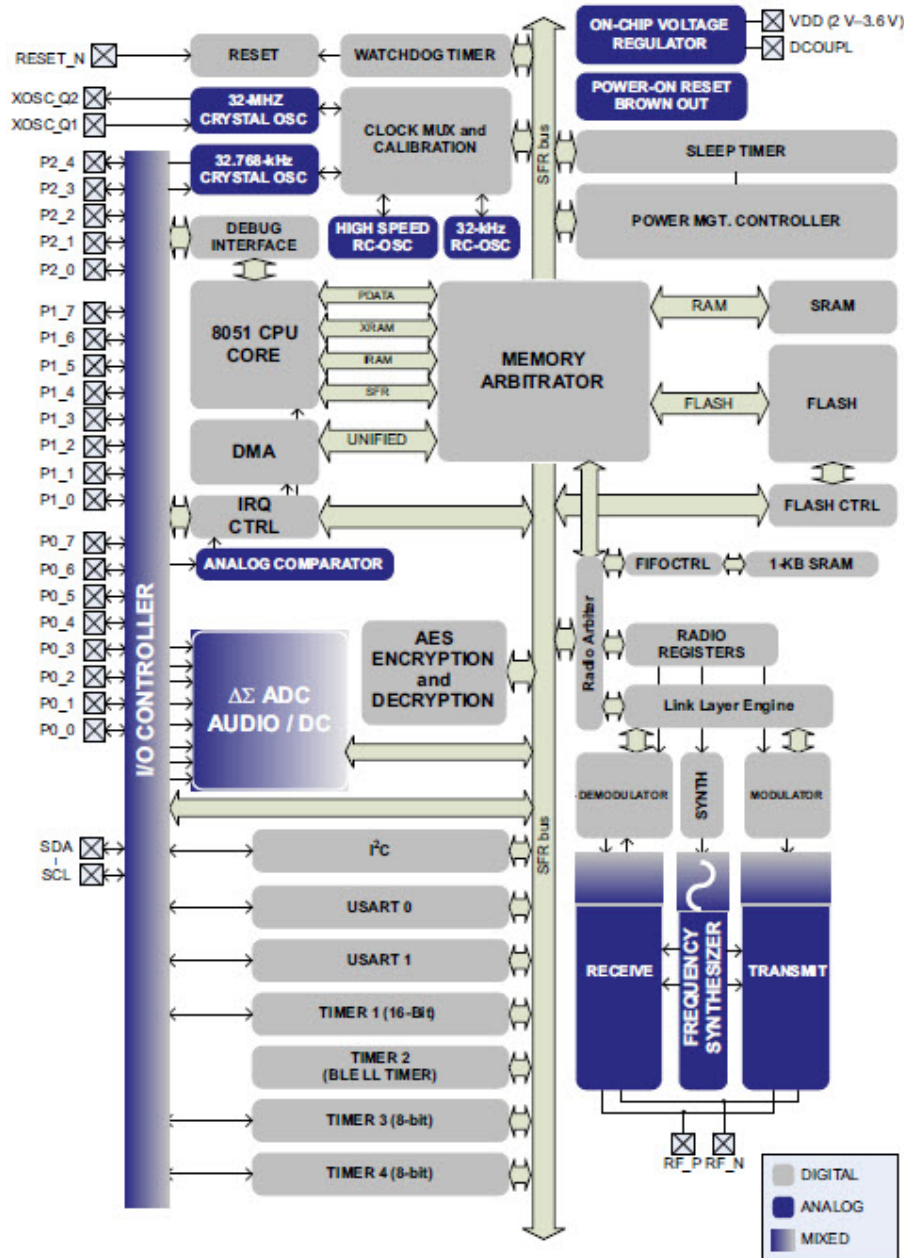


FIGURE 7 – Schéma bloc CC2541

PWM⁹... Il faut donc l'adapter à notre carte électronique.

Application : c'est dans cette couche que l'on retrouvera le comportement de notre application. C'est ici que je devrais gérer la communication I2C (grâce à l'HAL) avec le MGC3130, le contrôle de la lumière, de la LED de notification (si nécessaire) ainsi que la gestion du profils de communication bluetooth(mode de connexion, déconnexion...). C'est la couche "haute" de la programmation.

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

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

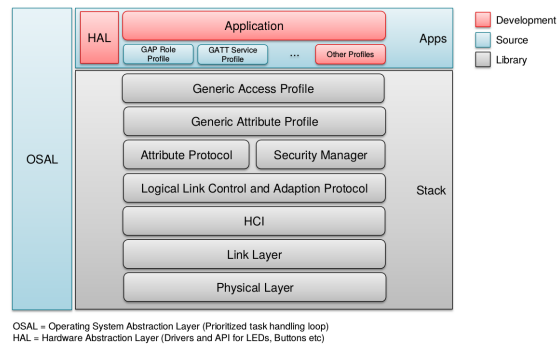


FIGURE 8 – Structure d'un programme

les besoin du projet je devrais en créer un afin de gérer la transmission d'informations entre le module et le client (smartphone).

3 Conception de la partie électronique

3.1 Fonctionnement global

La carte électronique finale aura un fonctionnement globale relativement simple, elle sera alimentée par le secteur et se connectera exactement comme un interrupteur standard. Celle-ci contiendra 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 passera sa main devant, le mouvement sera capté par le MGC3130 qui enverra l'information au microcontrôleur contenu dans le CC2541 via une communication I2C. Le microcontrôleur agira ensuite en conséquence afin de d'allumer, d'éteindre ou faire varier l'intensité lumineuse en agissant sur le montage de gradation.

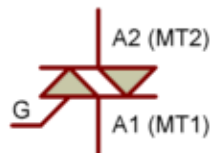
On peut diviser cette carte en 2 grandes parties, la partie relatant de la gestion de la tension secteur et de la gradation, et la partie relatant à la basse tension.

Cette carte sera réalisée au moyen du logiciel de CAO¹⁰ Eagle.

3.2 Partie tension secteur

Dans cette partie nous trouverons un transformateur AC/DC afin d'obtenir une tension continue de 3.3V dans le but d'alimenter le système basse tension (cc2541, MGC3130, led...). J'ai choisis le Vigortronix VTX-214-001-103 qui à l'avantage de ne pas être très encombrant et de pouvoir être soudé sur une carte, de plus il permet de délivrer une puissance d'1 watt, ce qui sera suffisant pour notre système.

L'autre partie qui sera connectée sur la tension secteur sera le montage de gradation. Ce montage se basera sur l'utilisation d'un triac. Un triac est un composant qui permet de conduire le courant dans les deux sens, c'est cela qui va nous permettre de l'utiliser sur secteur. On peut l'apparenter à deux thyristors montés tête-bêche, il possède aussi 2 électrodes principales A1 et A2 (Anode 1 et Anode 2) et une électrode de commande G qui permet de déclencher la conduction du courant entre A1 et A2.



A1 = Anode 1 (MT1 = Main Terminal 1)
 A2 = Anode 2 (MT2 = Main Terminal 2)
 G = Gachette

FIGURE 9 – Triac

Quelques explications sur le fonctionnement du triac :

¹⁰. Conception Assistée par Ordinateur

Il devient passant (s'amorce) lorsque la tension entre A1 et A2 dépasse la tension d'amorçage, cette tension est fortement réduite si on applique une tension sur la gâchette. le triac reste passant tant que le courant entre les anodes A1 et A2 reste au dessus d'une certaine limite (ne dépend plus de la gâchette, la tension d'EDF étant de 50 hz, ce phénomène se déroulera 100 fois par seconde.

Le but va être de retarder la conduction du triac afin de ne pas laisser toute la sinusoïde passer et ainsi diminuer la luminosité. Je n'ai pas encore fais le schéma électrique de commande du triac, il sera fait prochainement.

Cette partie risque de nous poser des problèmes de passage de prototype à produit commerciale en raison des normes régissant sur les périphériques se branchant sur le secteur. Ceci sera étudié prochainement.

3.3 Partie basse tension

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

3.4 Carte de démonstration

J'ai développé une carte de démonstration basé sur un microcontrôleur PIC16F1713, un MGC3130 ainsi que des leds afin de pouvoir rapidement tester le contrôle du MGC3130. Cette carte a été réalisée sur 4 couches afin d'avoir un plan de masse et un plan de Tx (génération du champ électrique, transmission) le moins perturbé possible. Les couches sont décomposés ainsi (en accord avec les préconisations données par la datasheet du MGC3130, annexe 3.4) :

1. Top : sur cette couche il n'y aura que les leds et une sérigraphie
2. Inner layer 1 : cette couche n'est pas utilisée
3. Inner layer 2 : couche Tx
4. Bottom : sur cette couche, nous trouverons le reste des composants, à savoir le microcontrôleur, le MGC3130, le régulateur 3.3V, les différentes résistances, capacités, les connecteurs (alimentation, programmation, I2C) ainsi qu'un plan de masse.

Le schéma électrique est donné en annexe 3.4 page 17, les 4 couches du PCB sont données en annexes 3.4 3.4 3.4 3.4.

Conclusion/Prévisions

Pour conclure ce rapport intermédiaire, j'ai pour le moment pu me familiariser avec les différentes solutions qui seront utilisées dans le projet final (BLE, programmation Android, MGC3130, Eagle). J'éprouve encore quelques difficultés à mettre en fonction la carte de démonstration afin de pouvoir faire des tests ainsi que de pouvoir valider le fonctionnement de ce système.

J'ai aussi pu faire beaucoup de recherches dans le but de choisir les solutions à adopter pour la partie gradation mais celles-ci restent encore sujettes à modifications (notamment entre la phase de prototypage et celle de produit final).

Je focalise donc mes efforts sur la mise en fonction de la carte de démonstration car c'est une partie délicate du projet du à la complexité de prise en main du MGC3130. Le but étant d'avoir cette carte fonctionnelle le plus rapidement possible. En parallèle je continue d'établir le schéma électrique sous Eagle.

C'est un projet très intéressant qui nécessite des connaissances très variées et qui, je pense, m'apportera beaucoup, tant sur le plan technique que sur la gestion de projet.

Annexes

Recommandation PCB 4 couches

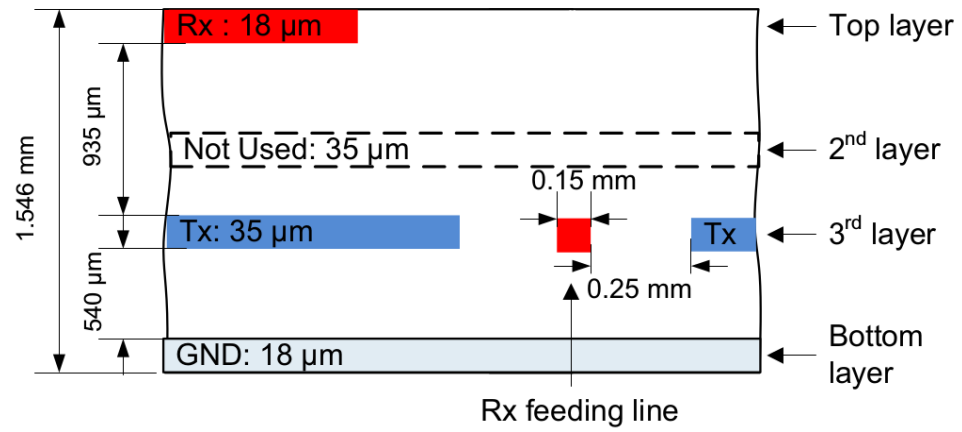
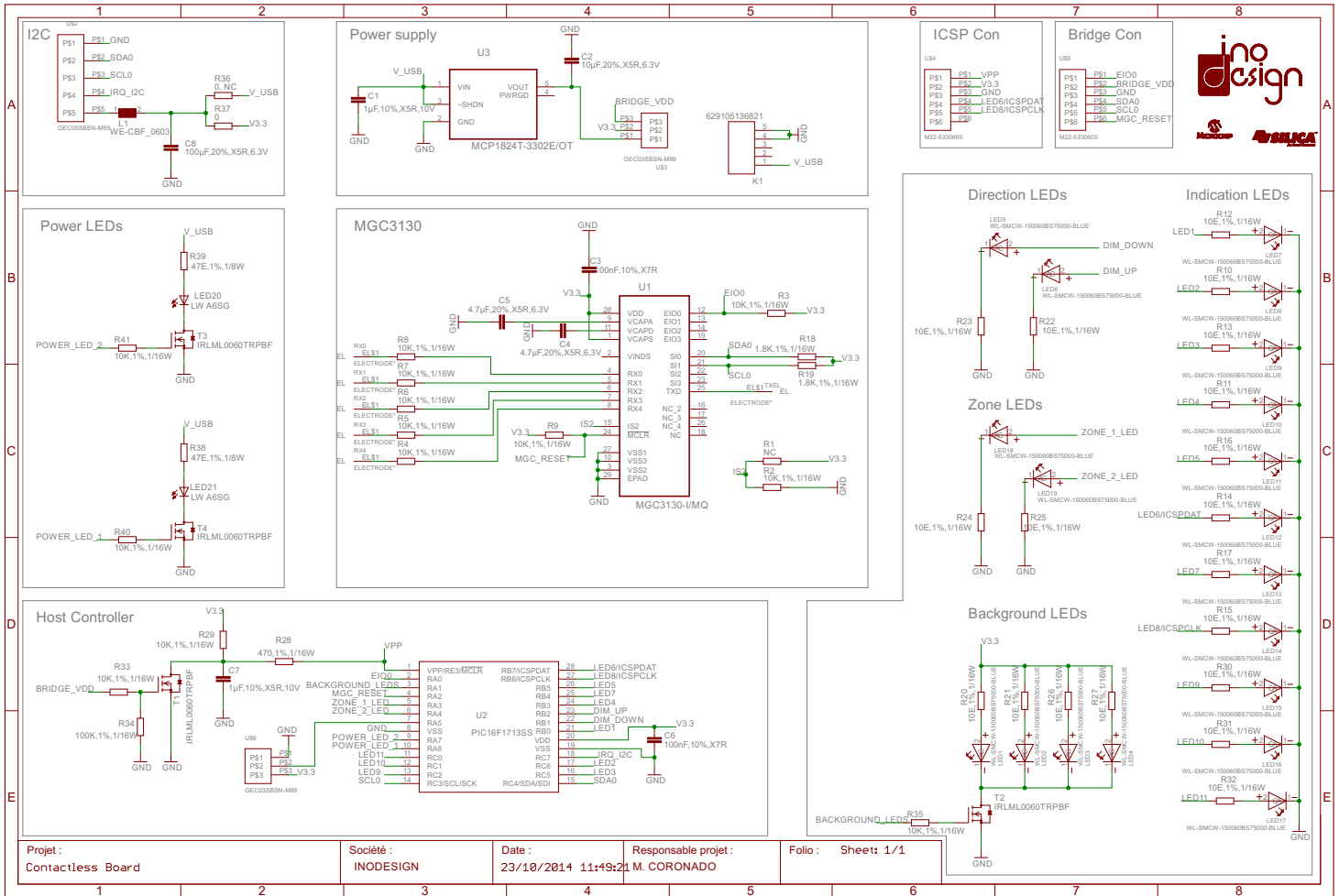
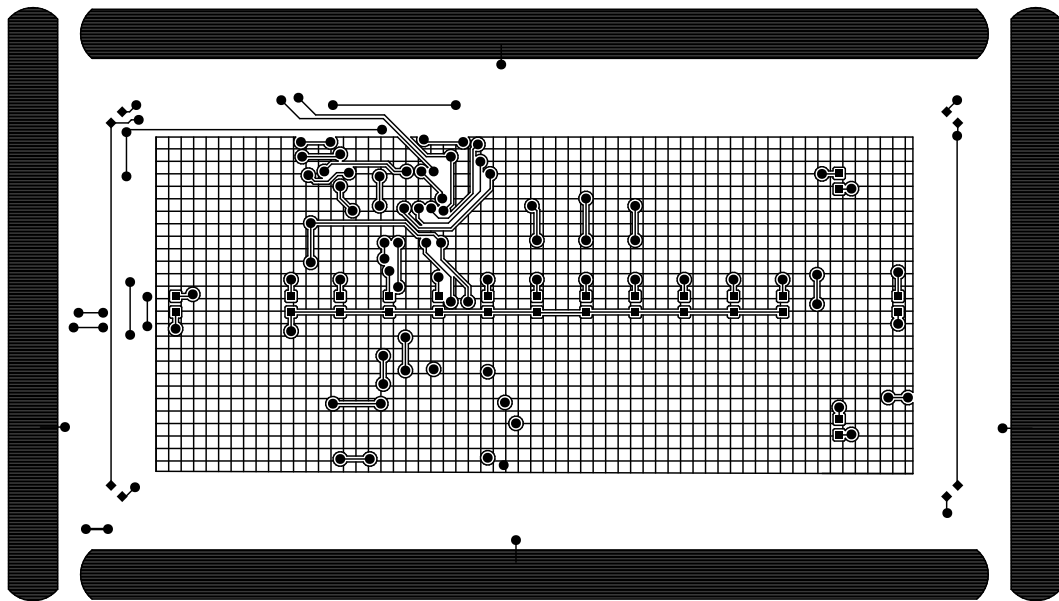


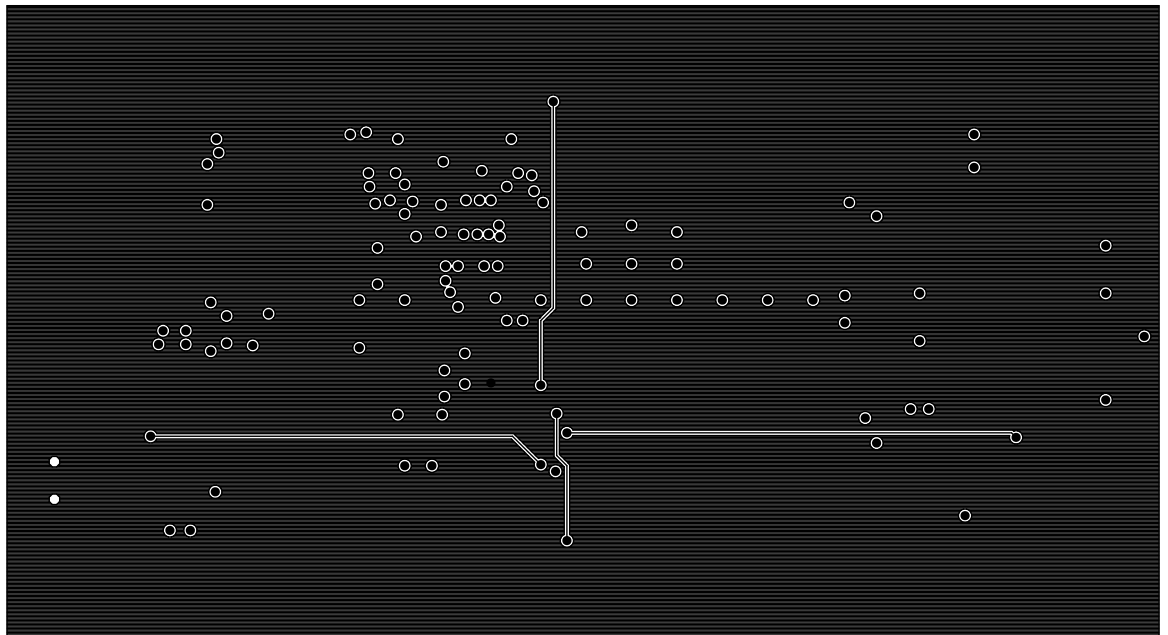
Schéma électrique



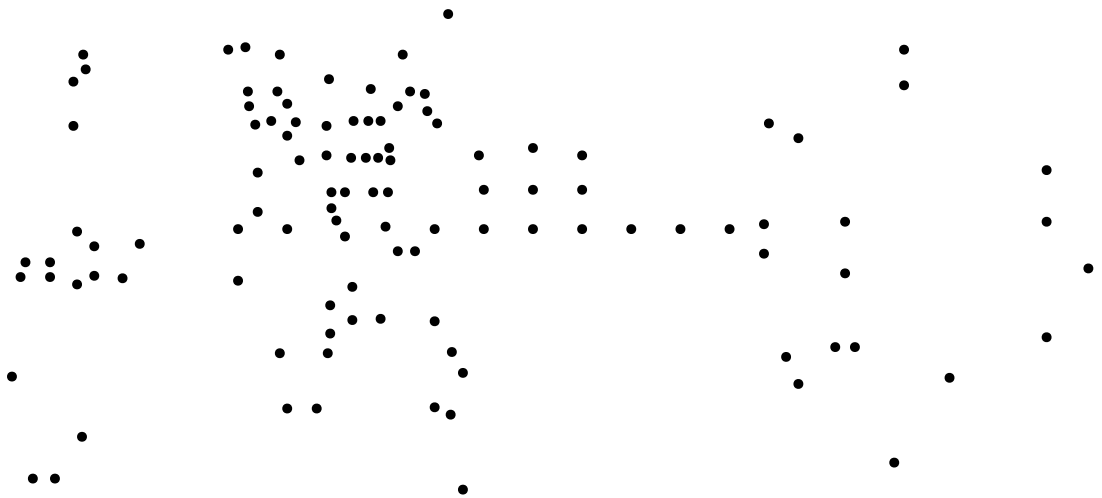
Couche Top



Couche 2



Couche 3



Couche bottom

