Rapport projet

**Ecole supérieure**

Électronique

Système de ticketing pour questions d’étudiants

Projet n°1811C

**Réalisé par :**

Dos Santos Mario

**Date :**

Début du diplôme : 22 Août 2022

Fin du diplôme : 27 Septembre 2022

**Table des matières :**

Système de ticketing pour questions d’étudiants 1

Projet n°1811C 1

1 Introduction 5

1.1 Cahier des charges 5

1.2 Analyse de l’existant 5

1.3 Amélioration 5

2 Design 6

2.1 Boitier 6

2.2 Carte Master 6

2.2.1 Schéma bloc 6

2.2.2 Bouton Reset 7

2.2.3 Alimentation carte SD 7

2.2.4 Pull Up 7

2.2.5 Régulateur 3V3 8

2.2.6 Xbee 8

2.2.7 Communication USB 8

2.2.8 Ecran TFT 9

2.3 Carte Slaves 9

2.3.1 Schéma bloc 9

2.3.2 Erreur de résistance 10

2.4 Choix boitier 10

2.4.1 Boitier Master 11

2.4.2 Boitier Slaves 11

3 Routage 12

3.1 Taille PCB 13

3.1.1 PCB Master 13

3.1.2 PCB Slave 13

3.2 Plan de masse 13

3.3 Stitching 14

3.4 Vue 3D 14

3.4.1 PCB Master 14

3.4.2 PCB Slave 15

4 Mise en service 16

4.1 Mesure alimentation 16

4.2 Correction Hadware 16

4.3 Vérification de la version précédente 16

5 Firmware/Software 16

5.1 Master 16

5.1.1 Configuration 16

5.1.1.1 System Clock 16

5.1.1.2 UART 17

5.1.1.3 SPI 17

5.1.1.4 Carte SD 18

5.1.1.5 Timer 18

5.1.1.5.1 Timer 1 18

5.1.1.5.2 Timer 2 19

5.1.2 Structogramme/diagramme 20

5.1.2.1 Pulling 20

5.2 Slave 21

5.2.1 Configuration 21

5.2.1.1 System Clock 21

5.2.1.2 UART 21

5.2.1.3 Timer 22

5.2.1.3.1 Timer 1 22

5.2.1.3.2 Timer 2 23

5.2.2 Structogramme/diagramme 24

5.3 Application C# 25

5.3.1 Interface 25

5.3.2 Etat 26

5.4 Trames 27

5.4.1 Commande 27

5.4.2 Broadcast 27

5.4.3 Donnée 27

6 Test et mesure 28

6.1 Ecran TFT 28

6.2 Driver TFT 29

6.3 Xbee 29

6.3.1 Code 29

6.3.2 Temps de communication 30

7 Planification 31

8 Etat d’avancement 31

9 Amélioration 32

10 Conclusion 32

11 Annexe 33

11.1 Cahier des charges 33

11.2 Planification 33

11.3 Procès-Verbaux 33

11.4 Journal de travail 33

11.5 Schéma électrique complet 33

11.6 Liste de pièce 33

11.7 Listing Code 33

11.8 Mode d’emploi 33

# Introduction

Je dois reprendre un projet déjà existant et le corriger/modifier selon le cahier des charges

## Cahier des charges

Voir Cahier des charges en annexe

## Analyse de l’existant

J’ai essayé de faire fonctionner le projet précédent mais sans succès

Mais il est dans un état avancer. De ce j’ai pu comprendre le projet fonctionnait lorsque le module RF (projet 1623) n’était pas implémenter sur les cartes.

D’après le rapport il n’y a pas d’erreur conséquent mais lorsque j’ai analyser les cartes j’ai pu constater que l’antenne du module RF de la carte master se trouvait en dessous d’un plan de masse cela fera partie des modifications que je devrais faire.

## Amélioration

Mon but sera :

* De corriger les erreurs hardware.
* D’améliorer le software.
* De rajouter une communication USB qui permettra de communiquer avec les ordinateurs.
* De rajouter le software du projet 2126 qui permet de directement aller chercher le nom de la personne connecter à l’ordinateur pour ne plus passer par une carte SB et un carte Slave personnel à chaque élève.
* De créer un programme Windows avec les mêmes fonctionnalités que le LCD.
* De faire une mise en boitier qui ne nécessitera pas l’utilisation d’une imprimante 3D.

# Design

## Boitier

Dans le cahier des charges il est indiqué que je dois changer le boitier pour un modèle commercial, celui précédemment utilisé avait été créée par un étudiant à l’aide d’une imprimante 3D.

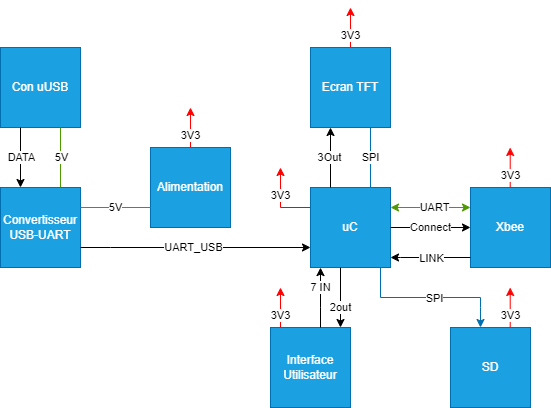
En plus de cela j’ai constaté que l’antenne du Xbee était près des plans de masse ce qui est déconseiller, comme je n’aurais pas la place de le déplacer avec la taille des boitiers existant je devrais prendre un boitier plus grand ce qui me permettrait lorsque j’aurais besoin de changer de composant de ne pas me soucier de leurs tailles.

## Carte Master

La première étape a été de corriger les erreurs de mon prédécesseur :

* Faire fonctionner le bouton reset
* Rajout de l’alimentation pour la carte SD
* Rajout de la pull up du MOSI de la carte SD

### Schéma bloc



### Bouton Reset

Le reset était toujours connecté au GND sachant que le reset est actif à l’état bas. La conséquence de cette erreur était que la carte ne pouvait pas fonctionner car tous les composant connecter au reset ne pouvait pas fonctionner.

Pour cette erreur il suffit de changer l’une des 2 pistes et la brancher sur la patte 1 ou 3.

J’ai décidé de déplacer la connexion du GND car lors du routage ça ne fera pas de grande différence grâce au plan de masse.

Avant Après

### Alimentation carte SD

Il y a eu un oubli de l’alimentation(VDD/3V3) de la carte SD

J’ai simplement mis le symbole du 3V3 et réarranger les pistes.

Avant Après

### Pull Up

Lorsque j’ai regardé la carte j’ai constaté qu’il y a eu un rajout d’une résistance. Il était directement branché sur le 3V3 et un vias du MOSI.

De ce que j’ai pu comprendre on utilise des pullups seulement si les pattes du master sont en open-collecteur.

Après discussion avec M.moreno on a décidé de laisser les footprint des résistances et dans le cas où on en aurait besoin on les brasera.

### Régulateur 3V3

J’ai dû changer le régulateur 3V3 car celui qui y était implémenter était en rupture de stock dans les différents fournisseurs

Dans ce cas j’ai préféré prendre le régulateur MAX1793 qui est fréquemment utilisé à l’etml-es et qui en plus possède du stock (à l’ETML-ES et chez les fournisseurs). Le composant est plus imposant que l’ancien utilisé mais comme je n’ai plus à me soucier de la place je peux me permettre de le prendre

En plus de cela il peut transmettre plus de courant ,1A au lieu de 250mA.

### Xbee

Lors de la version A il a été décider d’utiliser le Xbee créer par l’école (projet n° 1623) mais le projet n’a jamais fonctionné avec ce Xbee.

Il a donc été décider que l’on allait continuer de l’utilisé en espérant que le problème ne me fasse pas perdre trop de temps et si cela ne fonctionne toujours pas nous utiliseront un autre Xbee (XB24CAUIT-001) qui a déjà servi dans plusieurs projets.

Heureusement les pins et la distance entre les pattes sont identique pour les 2 Xbee (alimentation + communication uart).

### Communication USB

Pour cette version nous allons ajouter une communication USB qui nous permettra de récupérer le nom de l’élève sans l’utilisation de la carte SD et des adresses des Xbee.

Je vais utiliser le contrôleur de pont USB-UART du projet 2126 (CY7C64225) car il y en a en stock et qu’il a déjà été utiliser.



Dans le datasheet il est précisé que si le microcontrôleur et le contrôleur USB-UART sont alimenter en 3V3 il faut mettre un pont diviseur entre le VBUS du connecteur USB et la patte VBUS de l’IC.



### Ecran TFT

Je n’arrive pas à trouver le modèle de l’écran.

J’ai dû chercher dans les écrans d’adafruit les modèles les plus proche de celui sur la carte.

Les seules informations que je possède sont :

* Le nombre de pins : 40
* La disposition de ces pins
* La taille de l’écran 2.8’’
* Le driver : ILI9341

J’ai pris le modèle 2090 car il était plus récent que le 1770 et que la seule différence que j’ai pu observer était que le premier avait un touchscreen capacitif et le second un résistif.

Comme la seule différence se trouver dans le touchscreen, que je n’utilise pas,,je pense qu’il n’y aura pas de problème lors de la programmation.

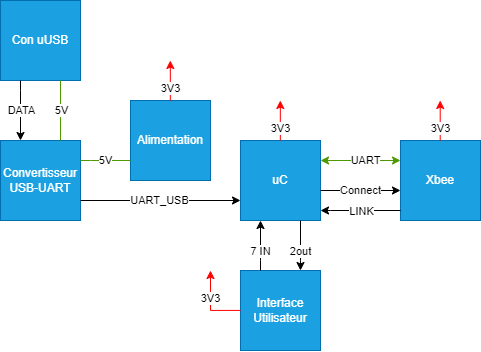
## Carte Slaves

Nous refaisons les mêmes corrections et ajoute que pour la carte Master sauf pour la partie SD (correction alim et ajout pull up).

Donc :

* Correction du bouton reset
* Changement du régulateur 3V3
* Ajout de la communication USB

### Schéma bloc



### Erreur de résistance

Lors de la review de schéma, M.moreno à remarquer qu’une des résistance utiliser pour les leds étaient 10 fois supérieur aux autres.

Comme j’ai repris un schéma déjà existant et qu’il n’y avait pas d’erreur de signaler sur les leds, je n’y avait pas fait intention.

J’ai pu aussi remarquer que les résistances permettent de soutirer jusqu’à 20mA alors que les I/O du microcontrôleur ne permettent que 15mA

Donc j’ai décidé de tout recalculer.

Dans les sites de fournisseur (Digikey et Mouser) il était indiqué que le courant forward était de 2mA alors que dans le datasheet il est indiqué à 25mA.

J’ai donc décider de suivre les indiquation du datasheet mais avec le courant max des I/O

## Choix boitier

J’ai eu pas mal de peine à trouver un boitier pour la carte commande car je voulais garder un boitier semblable à celui qui avait été fait par le précèdent diplômant mais qui est un peu plus large car je dois mettre le Xbee dans un bord sans rien autour (plan de masse, composant, piste).

J’ai été voir chez Hammond manufacturing car ils ont un grand nombre de boitier et qu’ils ont un système de filtrage qui est facilement utilisable.

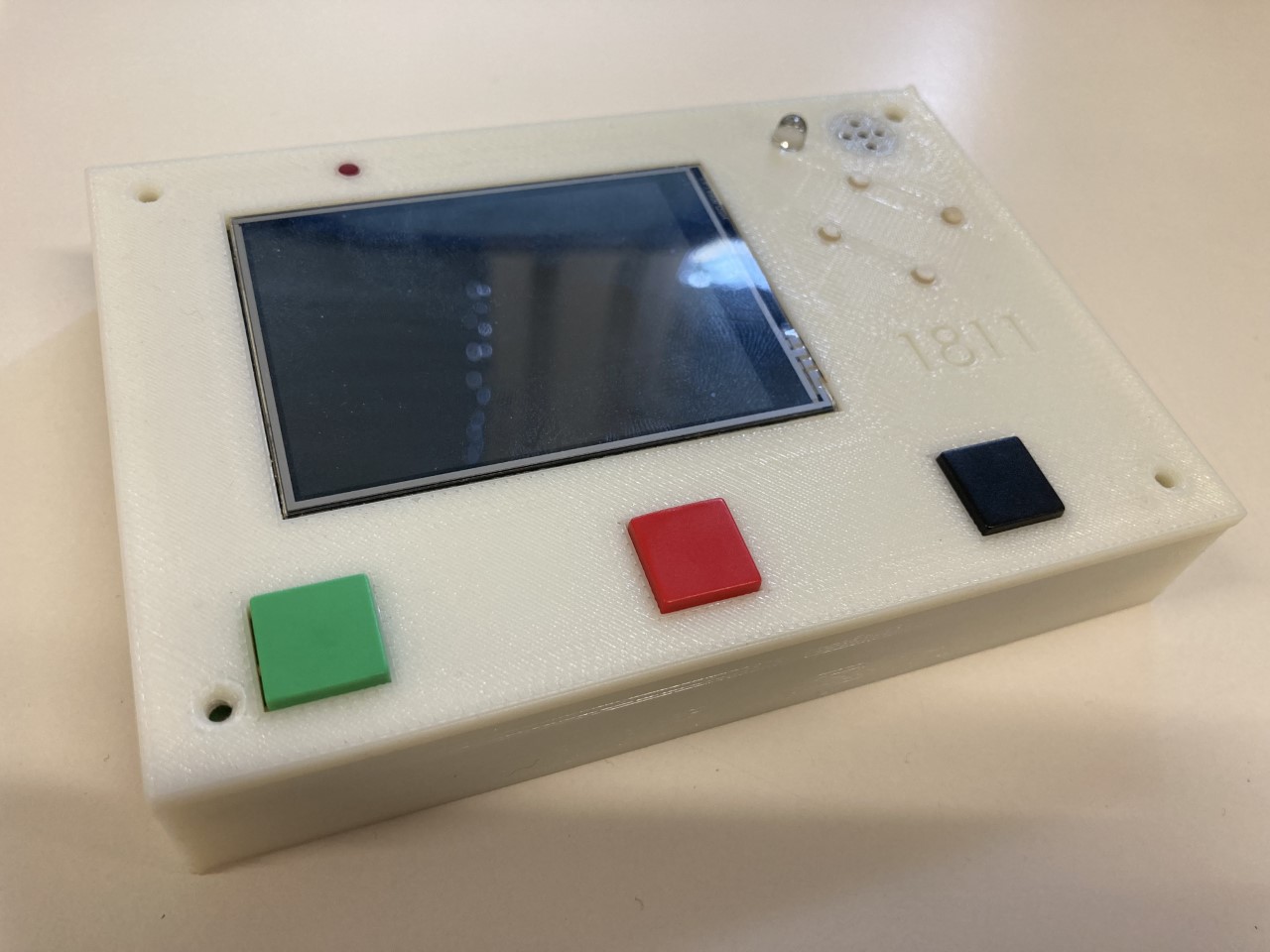
### Boitier Master

Le boitier de la version A faisait ~131 mm x 91 mm x 23 mm

Pour mes recherches j’ai cherché les boitiers de taille 135 mm x 110 mm x 20 mm (Longueur X Largeur X Hauteur) et qui étaient en plastique.

J’ai paramétrer la hauteur à 20 mm car je pensais trouver des boitiers avec un minimum de 15mm de hauteur en interne du boitier et j’espérais trouver des boitiers assez proches de cette taille. Mais sur Hammond ils affichent les boitiers plus grand que ce que le filtre donc sa éliminait quand même les boitiers trop petits en longueur x largeur.

Si la taille est plus grande je devrais trouver un moyen avec des colonnettes pour avoir un résultat similaire à ce qui a été fait.



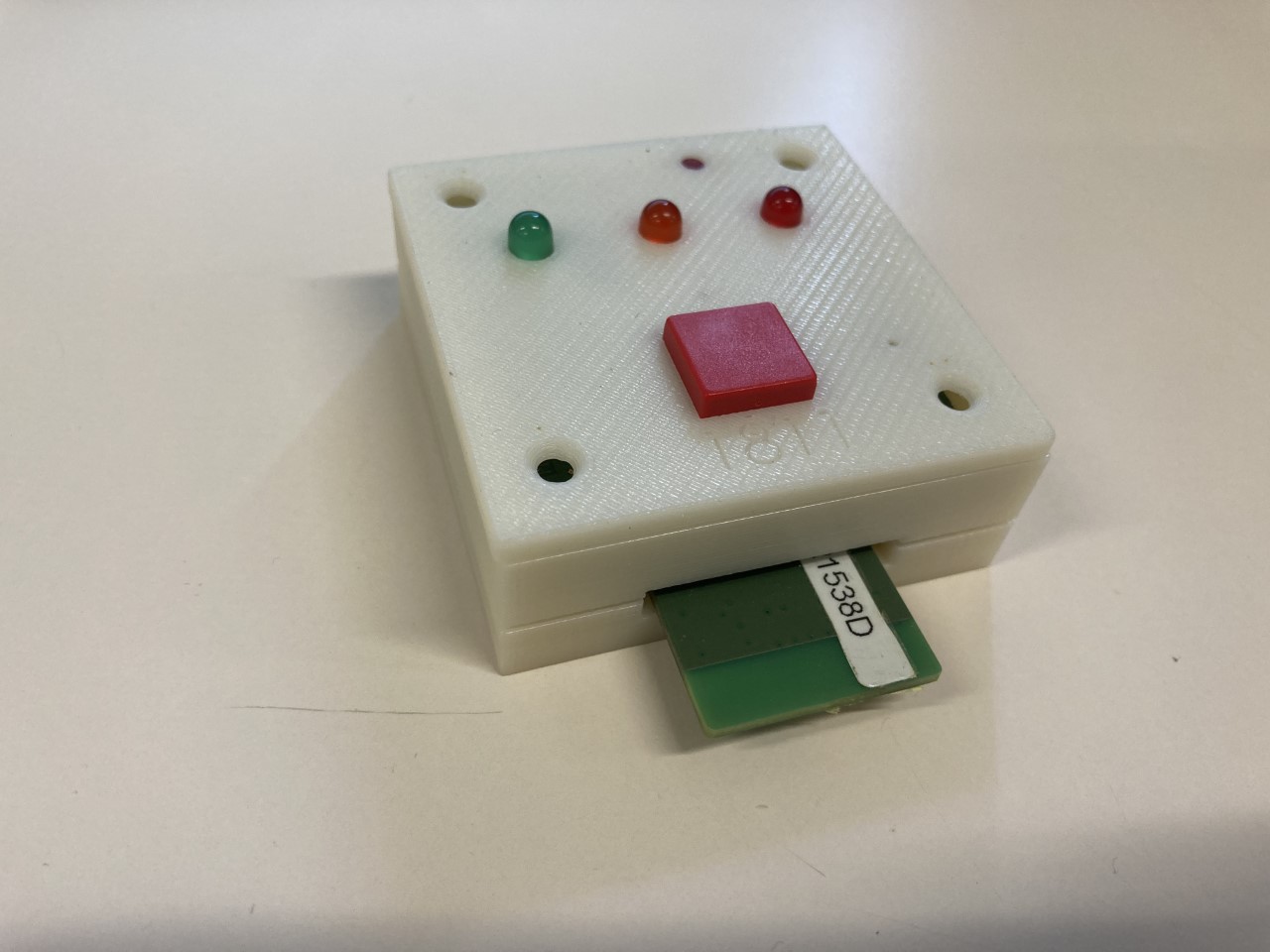
Boitier Master version A

Je vais utiliser le boitier 1555HL2GY qui fait 180 mm x 119 mm x 46 mm. C’est le plus petit boitier en hauteur avec au moins la largeur et longueur égal ou plus grands de ce que j’avais prévu

### Boitier Slaves

Le boitier des cartes slaves fut beaucoup plus simple à trouver.

Je cherchais un boitier de ~70 mm x 55 mm x 20mm et j’ai pu trouver rapidement un boitier de 84 mm x 56 mm x 23 mm.



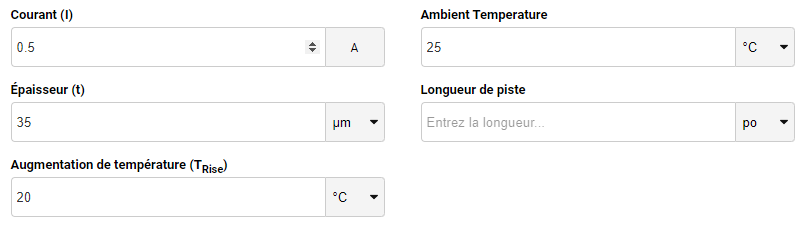
Boitier Slave version A

# Routage

Pour le routage j’utilise la version 21.6.1 d’Altium.

J’ai calculer la largeur des pistes minimum pour être sur qu’il n’y ait pas de problème

Pour ça j’ai utilisé le calculateur de Digikey :





J’ai décidé de faire mon routage avec des pistes de 0.254mm, hors alimentation, et des pistes de 0.5mm pour les piste d’alimentation.

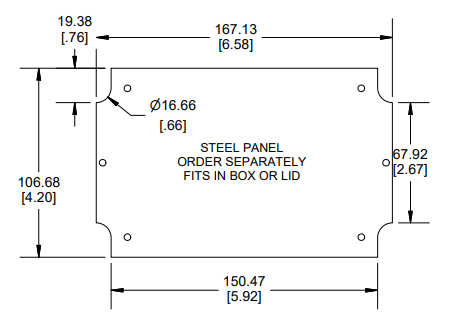
Les contraintes de routage ont été réglée pour correspondre à un PCB de classe 6C sur Eurocircuit.

## Taille PCB

### PCB Master

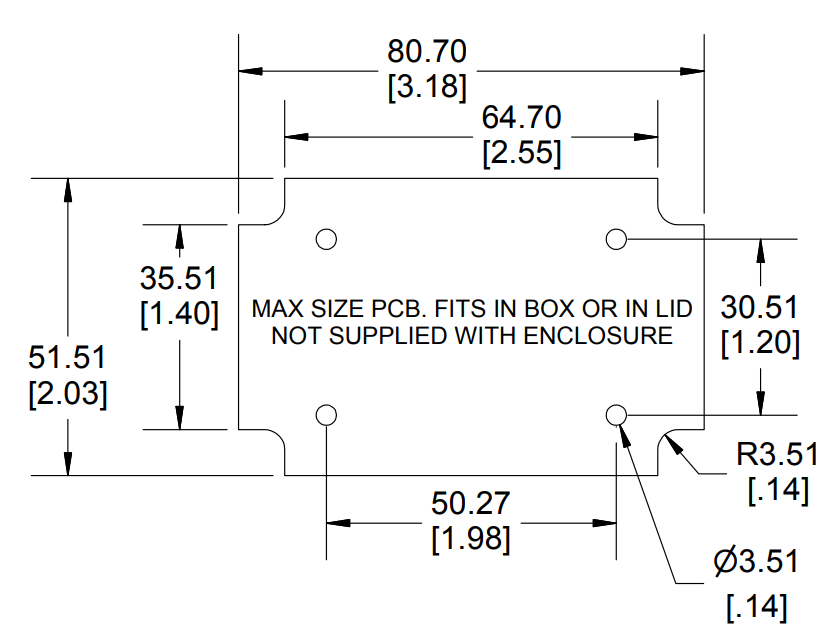
Pour la taille du PCB Master j’ai repris les dimensions données pour une plaque en acier.

Il est surtout important de bien placer les trous pour les vis.



### PCB Slave

Pour la taille du PCB j’ai repris les dimensions données dans le datasheets du boitier



## Plan de masse

Je mets un plan de masse au top et au bottom pour permettre une meilleure distribution du GND et réduire les perturbations (bruits et interférence).

Dans mon cas je dois faire attention à ne pas mettre de plan de masse sous le module Xbee car il possède une antenne, le plan de masse le perturberait si on le met en dessous.

## Stitching

Le stitching permet d’avoir un plan de masse plus homogène sur tout le PCB

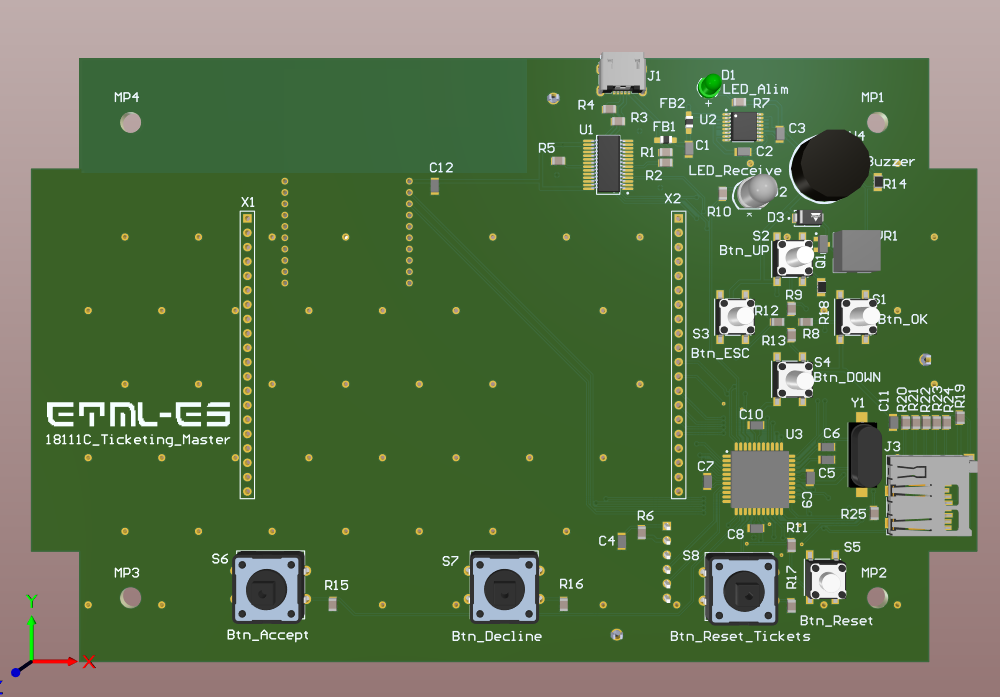
## Vue 3D

### PCB Master

Régulateur 5V to 3V3

Convertisseur

USB-UART



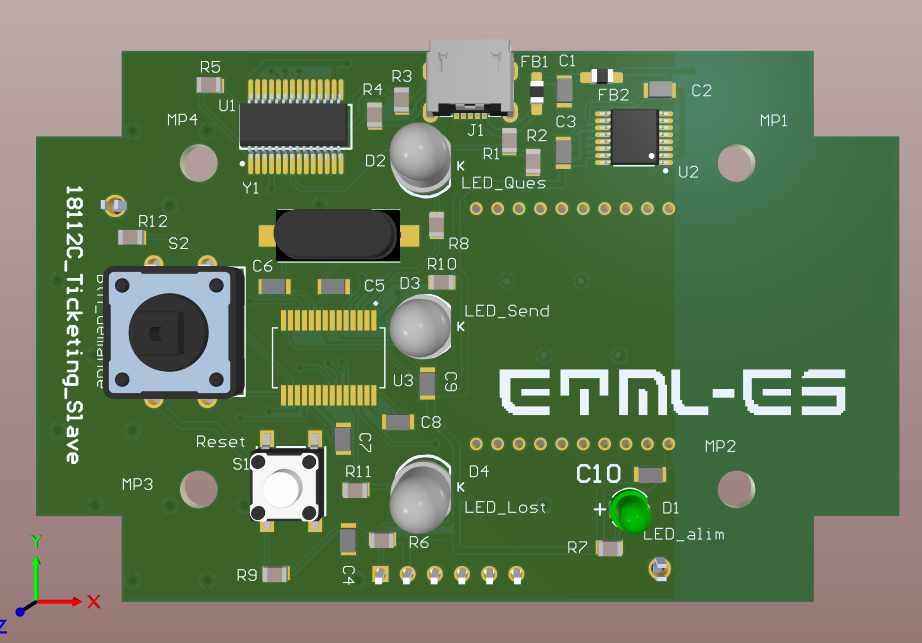
uC

### PCB Slave

Régulateur 5V to 3V3

Convertisseur

USB-UART



uC

# Mise en service

## Mesure alimentation

J’ai mesuré les alimentations 3,3V des 3 cartes (1 Master et 2 slave).

Je me retrouve avec des alimentations d’environ 3,36V stable.

## Correction Hadware

Je n’ai pas eu besoin de faire de correction hardware.

## Vérification de la version précédente

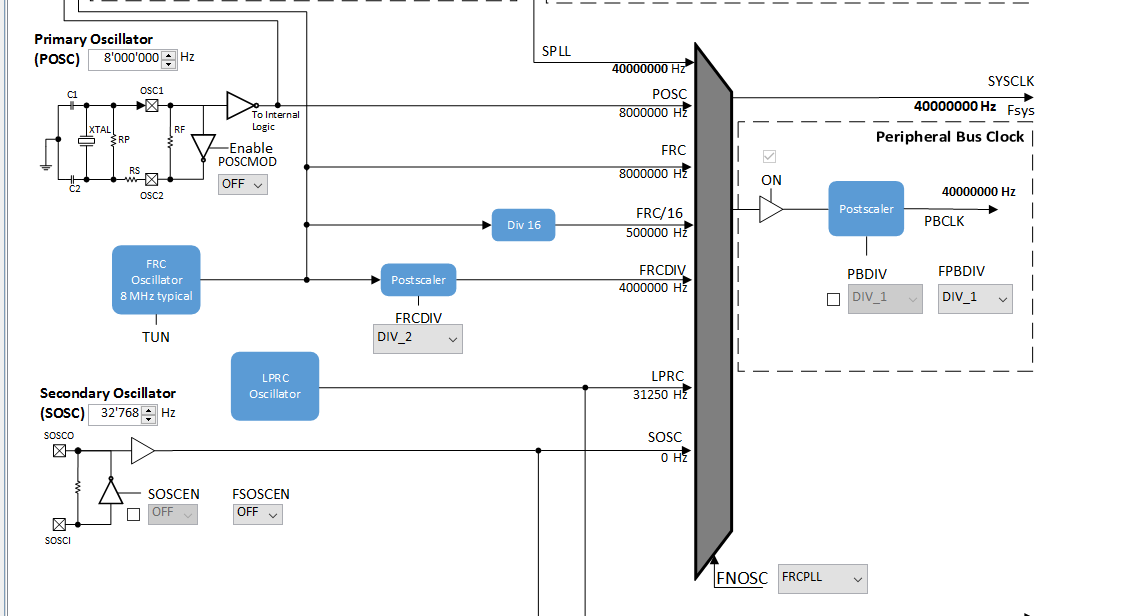
En programmant ma carte avec l’ancien programme j’ai pu observer que je n’arrivais juste à allumer le backlight du LCD de la carte Master.

# Firmware/Software

## Master

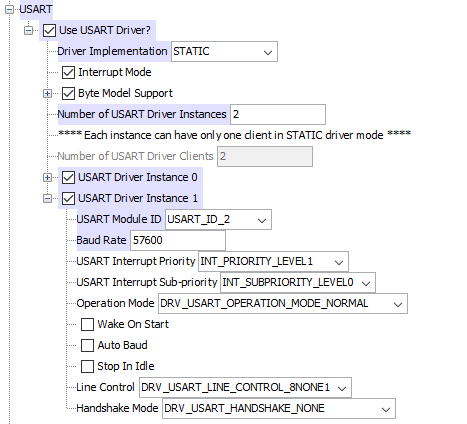
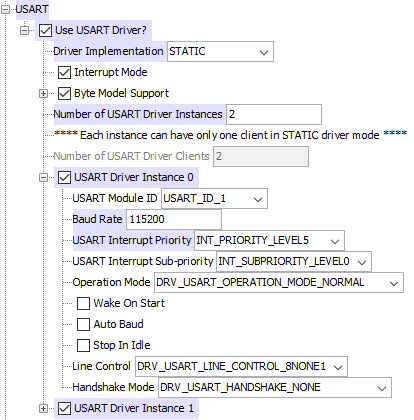
### Configuration

#### System Clock



J’ai décidé de laissé le sysclk à 40MHz qui avait été utilisé dans la version A

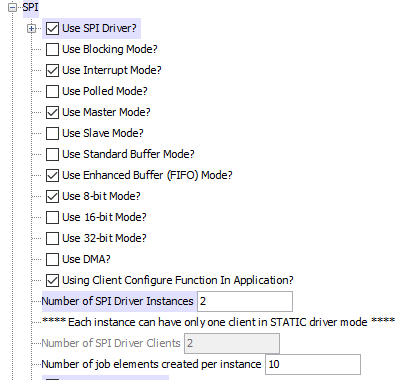
#### UART

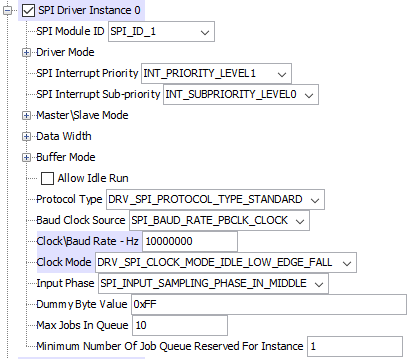
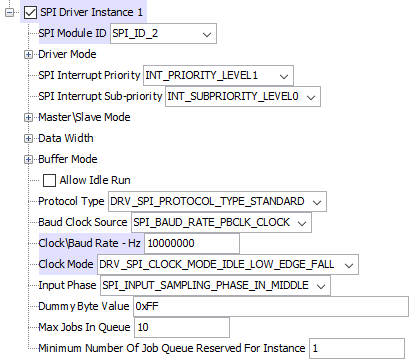


UART Xbee UART USB

Pour le Xbee j’ai repris les paramètre utilisé dans le projet 1623 (20200214\_Module\_Xbee\_06\_SCA) qui a été modifier par M.Castoldi.

#### SPI



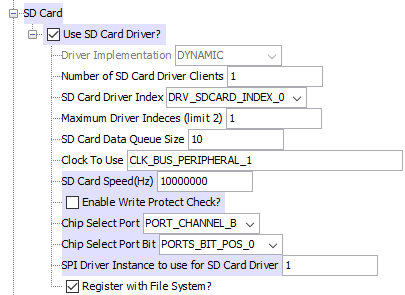
 

SPI écran TFT SPI Carte SD

J’ai laissé les paramètres déjà utilisé précédemment.

Dans mon projet l’écran et la carte SD ne sont pas essentiel mais je vais quand même essayer de les faire fonctionner.

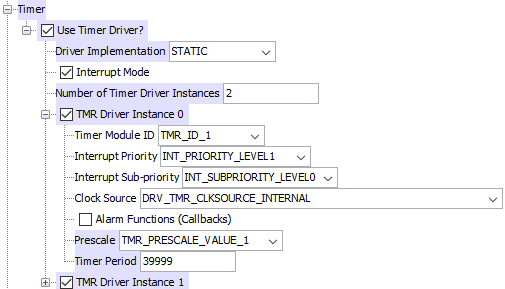
#### Carte SD



Les paramètre de la carte SD avait déjà été implémenter.

#### Timer

##### Timer 1

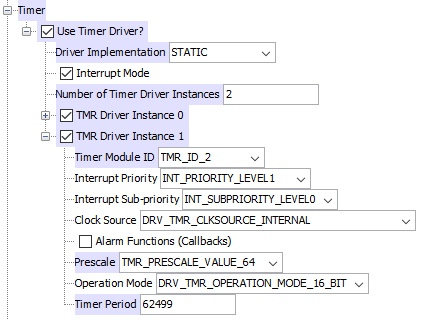


Ce timer est utilisé lors d’une réception de ticket. Il fera vibrer le buzzer à la fréquence donnée.

Le temps du timer est de :

Ce timer était déjà implémenter dans la version A.

##### Timer 2



Ce timer sera utilisé pour envoyer un broadcast et le pulling toute les 2s

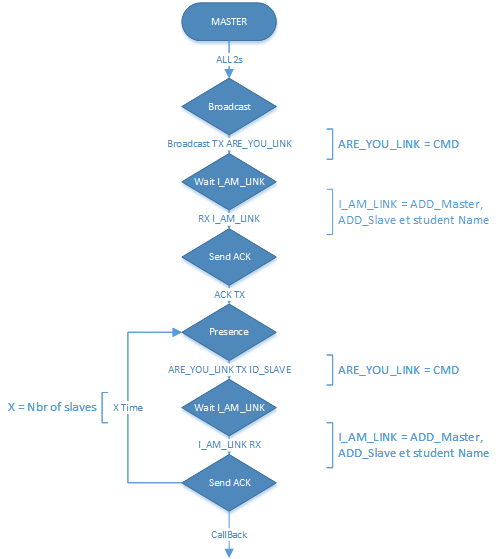
La période cible sera de 100ms. Je ne mets pas le timer à 2s pour ne pas utiliser un timer 32bits.

Le clock du microcontrôleur(SYSCLK) est de 40Mhz

La valeur du timer doit être de :

### Structogramme/diagramme

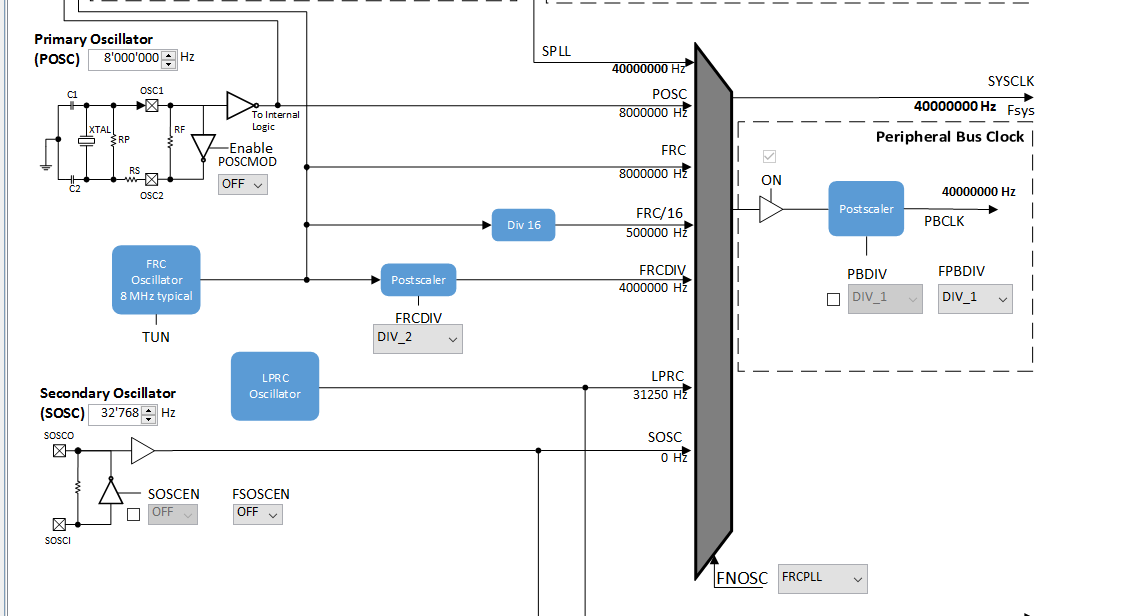
#### Pulling



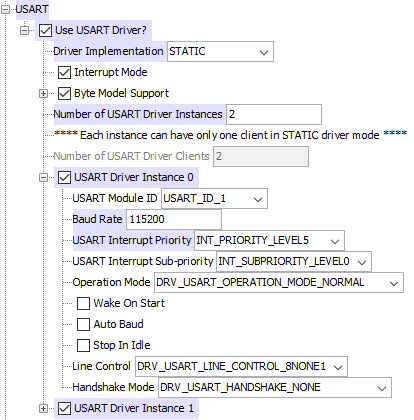
## Slave

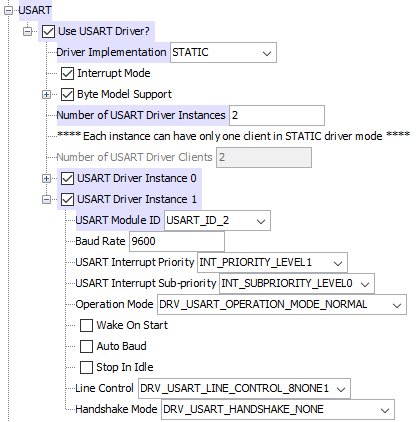
### Configuration

#### System Clock



#### UART





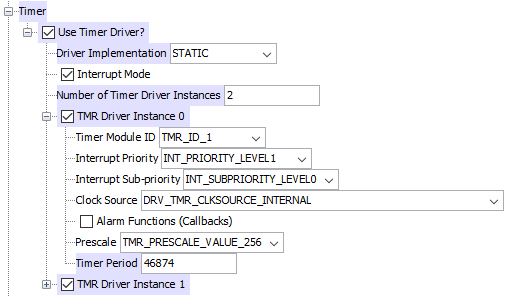
UART Xbee UART USB Slave

J’utilise les même paramètre d’uart du Xbee que le programme maitre

Le programme Slave communiquera en USB avec une application windows (2126\_AffichageMatriciel) via Uart donc j’ai décidé de reprendre les paramètres de l’UART

#### Timer

##### Timer 1

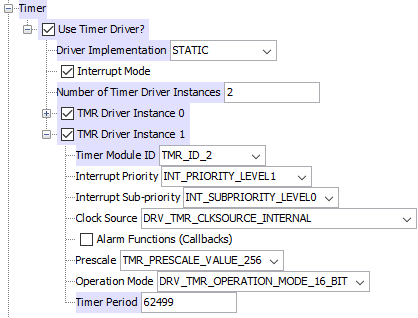


Ce timer est utilisé lorsque la carte Slave n’est pas link au Master. Il fera clignoter la led rouge à la fréquence donnée.

Le temps du timer est de :

Ce timer était déjà implémenter dans la version A.

##### Timer 2



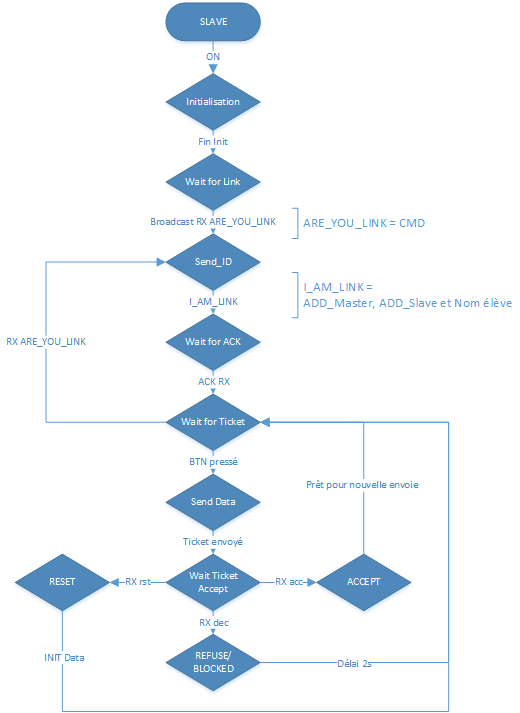
Ce timer sera utilisé pour les 20s de blockage de la carte slave

La période cible sera de 400ms. Je ne mets pas le timer à 20s pour ne pas utiliser un timer 32bits.

Le clock du microcontrôleur(SYSCLK) est de 40Mhz

La valeur du timer doit être de :

### Structogramme/diagramme



## Application C#

### Interface

Accepter la demande

Mise à jour des paramètres du port et ouverture du port

Liste de nom d’élève mis par ordre d’arriver

Nom de l’application / N° de projet

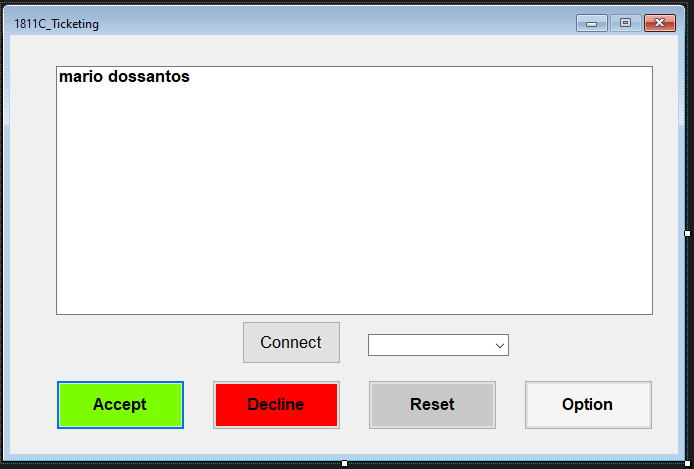
Refuser la demande

Reset toutes demande déjà accepter et ceux en attente

Paramétrage du serial port

Choix du port COM

Paramètre de la fenêtre



Interface de contrôle

Icone de l’application

Nom de l’application / N° de projet

Paramètre de la fenêtre

.

Transfert sur le site internet de l’etml ES pour voir la liste d’élève des 2 classes SLO

Confirmer les paramètres

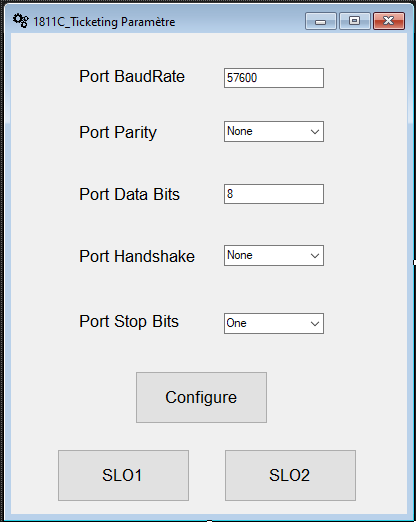
Choix du Nombre de stop bits

Choix du Nombre de Byte

Choix de la parité

Choix du Handshake

Choix du BaudRate



Interface paramètre

J’affiche les valeurs de paramétrage par default du serial port

Dans la deuxième interface le bouton **Configure** permet de confirmer que les bonnes valeurs ont été inscrit. Et le bouton **Connect** permet de fermer le port COM, de le reprogrammer avec les paramètre inscrit dans l’interface précèdent et de rouvrir le port COM.

### Etat

Par manque de temps je n’ai pu que faire l’interface de l’application.

J’essayerais de faire fonctionner une partie pour les portes ouvertes et si possible complètement pour la présentation.

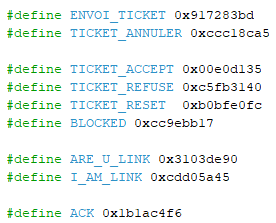
## Trames

Chaque élément est en 32bits (8 bytes) sauf à une exception qui le nom de l’élève

### Commande

Dans la version A des valeurs ont été donnée pour chaque commande.

Je n’ai aucune idée si ces valeurs ont été définie par hasard ou non. Je les ai laissées pour l’instant car je n’ai pas eu de problème avec mais si par hasard une adresse de xbee correspondrait à une de ces valeurs je ne peux pas garantir le fonctionnement de la carte



### Broadcast

Broadcast

 ADD Master

 ARE\_YOU\_LINK

Broadcast = 0xFFFFFFFF

ADD Master = récupération de l’adresse du Xbee

ARE\_YOU\_LINK = adresse prédéfini(0x3103de90)

### Donnée

Adresse expéditeur

 Adresse destinataire

 Data

Adresse expéditeur = adresse de 32 bit qui peut soit être celui du master ou celui du slave

Adresse destinataire = adresse de 32 bit qui peut soit être celui du master ou celui du slave

Data = nom d’un élève ou command

Nom de l’élève → prénom de l’élève + première lettre du nom de famille converti en hexadécimal

Commande → Envoie ticket, accepter, refuser, …

Les valeurs sont prédéfinies

# Test et mesure

## Ecran TFT

Lorsque j’ai voulu tester l’écran avec le programme de la version A, j’ai pu rapidement voir que je n’arrivais pas à afficher les informations voulu mais seulement à allumer le backlight.

J’ai pu confirmer que le modèle précédemment utiliser était le adafruit 1770 et non le 2090 car c’était un modèle qui avait déjà été utiliser dans des projets passer.

Comme le drive du TFT est le même sur les 2 modèles je me suis dit qu’il fonctionnerait de la même manière.

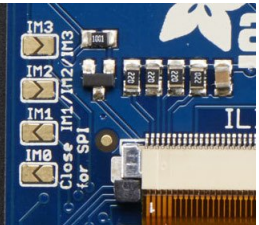
Mais lorsque je l’ai programmé j’ai constaté que seul le backlight fonctionnait. J’ai revérifier que je n’avais pas modifier une partie du code mais je n’ai trouvé aucune différence.

J’ai ensuite essayer de reprendre le code du driver donnée par adafruit sur github mais là aussi dans succès.

Après plusieurs essaye non concluant j’ai demandé à M.Castoldi, qui connaissait déjà un peu l’écran de la version A, s’il avait déjà eu ce genre de problème auparavant. Suite à cette discussion il m’a proposé d’utiliser le même écran de la version A qui n’était pas baser sur un autre projet (1706x\_Spectrometre) pour vérifier si le problème se trouvait dans mon code ou mon écran.

Après avoir tester mon code avec l’écran du spectromètre j’ai constaté que l’écran s’allumait. Donc j’ai un problème avec mon écran.

Dans différent forum j’ai pu lire qu’il y avait souvent ce genre de problème (backlight qui s’allume mais pas l’écran) et le problème souvent signaler était le connecteur/pcb flexible qui était mal brancher. Après avoir débraser mon écran j’ai constaté que ce n’était pas un problème de connexion. Mais cela m’a permis de voir que je n’avais pas braser les jumpers pour communiquer en SPI.



Après avoir rébraser mon écran et reprogrammer, j’ai constaté qu’il n’y avait pas de changement.

Lorsque je cherchais le problème j’ai pu voir que mon TFT était mal plaquer sur la carte dans j’ai réchauffer les brasures. Après cela j’ai pu constater que l’écran s’allumais.

Au début mon écran ne fonctionnait pas car j’essayais de communiquer en SPI et lui mode 8bits et ensuite je pense que j’avais soit une brasure froide soit un court-circuit que je n’avais pas vu.

## Driver TFT

Il y a plusieurs fonctions qui fonctionne entre les 2 LCD mais aussi certaine qui ne fonctionne pas ou presque pas.

Fonctionnel :

* tft\_setTextColor : qui permet de changer la couleur du texte
* tft\_setCursor : permet de modifier l’emplacement du curseur
* tft\_writeString : permet d’afficher un texte
* tft\_drawFastHLine : permet de dessiner une ligne

Pas totalement fonctionnel :

* tft\_setTextSize : permet de changer la taille du texte. On est obligé de mettre le texte en taille 1 sinon il empêchera toute écriture.

Pas fonctionnel :

* tft\_fillRect : permet de dessiner un rectangle
* tft\_fillScreen : permet de remplir l’écran d’une couleur

Ces 2 fonction ne font juste rien. N’empêche pas l’écriture de texte.

## Xbee

### Code

Il m’était impossible d’envoyer un message par Xbee. En lisant le rapport précédant (Rapport projet 1811) j’ai pu lire la phrase suivante :



Donc cela me confirme que les Xbee arrive bel et bien à communiquer entre elle.

Durant 2 jours j’ai essayé de les faire fonctionner mais sans succès.

J’ai demandé de l’aide à M.Castoldi qui s’en est servi et qui fonctionnait très bien.

Après discussion avec lui nous avons conclu que le problème ne venait pas de ma carte ou de mon code mais du code du Xbee qui s’après M.Castoldi était un peu bancal.

J’ai donc reprogrammer mes carte Xbee avec le firmware de M.Castoldi (20200214\_Module\_Xbee\_06\_SCA) du projet 1623

### Temps de communication

P1

Trigger : CH1

Mode : Normal

Protocol : UART

UART trigger : Start bit

CH1

XBee\_TX

MASTER

CH3

XBee\_RX

SLAVE



On peut voir sur l’oscillogramme que le temps que le xbee met à transmettre les données est d’environ 11.37ms.

J’ai fait d’autres mesures est j’ai pu constater que le temps de transmission variait entre 10 et 12 ms.

MASTER

XBEE

XBEE

SLAVE

Xbee

TX

Xbee

RX

Schéma d’envoi d’une trame

# Planification

Je savais que je ne pouvais pas passer directement du design de mon PCB au montage mais j’ai oublié de l’afficher dans ma planification.

Le montage de ma carte été faite plus tard que prévu car il y avait les 5 jours de fabrication/livraison au lieu des 3 jours qui était prévu.

J’ai passé plus de temps que prévu sur mon programme car le code de j’ai récupérer de l’ancienne version était truffé d’erreur qui n’était pas préciser dans le rapport et qu’il y avait des parties qui n’était même pas fait. En plus j’ai eu énormément de problème avec le Xbee et l’écran TFT.

Je n’ai pas pu avancer sur la partie mécanique car j’ai pris trop de temps sur la programmation.

# Etat d’avancement

|  |  |
| --- | --- |
| Affichage | NOK |
| Réception Nom élève | OK |
| Communication Xbee | OK |
| Envoie de ticket | OK |
| Gestion des tickets | En cours |
| Gestion de la liste de question | En cours |
| Application C# gestion des donnée | NOK |
| Communication C# - Master | NOK |
| Boitier Master | NOK |
| Boitier Slave | NOK |

Je vais essayer de faire fonctionner les communications entre Slave et Master et de les afficher sur l’application C# pour les portes ouverts du 30.09.

Et je vais essayer de faire fonctionner l’application C# (gestion de la liste) pour ma présentation du 07.10.

L’affichage n’est pas fonctionnel et je ne pense pas m’attarder dessus durant la prochaine semaine.

Le boitier Slave est presque fini d’être dessiner (sur solidworks) il manquera plus qu’à faire le plan de perçage avec les cotations et de le percer le boitier qui a déjà été réceptionner

Le boitier Master est presque fini d’être dessiner (sur solidworks) il manquera de dessiner l’ouverture pour le TFT et la carte SD. Ensuite il faudra faire le plan de perçage avec les cotations et de le percer le boitier qui a déjà été réceptionner

# Amélioration

Je pense qu’il serait utile d’enlever l’écran TFT et la carte SD si nous ne les utilisons pas. Cela permettra de réduire la taille de la carte, les coûts de fabrication seront réduit et cela nous permettrait de prendre un boitier plus petit qui sont souvent moins cher.

Finaliser le Code. Je vais essayer d’avancer le plus possible sur le code jusqu’à la présentation.

# Conclusion

Ce projet avait l’air assez simple sur le papier mais certain problème firmware et la reprise d’un projet existant mon rapidement fait penser le contraire.

La partie hardware c’est bien dérouler et j’ai pu commencer la programmation C# et les dessins mécanique en attentant la livraison de mes cartes.

La partie firmware était beaucoup plus compliqué car j’ai dû pas mal modifier le code à cause de la carte SD et de l’écran TFT qui ne fonctionnait pas comme je le souhaitait.

Ce n’est pas la première fois que je reprends un projet pour le continuer/corriger mais à chaque fois je me rends compte se serait plus simple de commencer depuis le début. Dans mon cas la partie hardware était assez facile.

La version A n’avait pas beaucoup de problèmes, même si j’en ai trouvé certain qui n’était indiqué nulle part. La partie firmware était beaucoup plus compliqué car le code n’était pas commenté et que certaine partie du code, qui était écrit dans le rapport comme fait, ne l’était pas.

J’ai beaucoup appris de ce projet je suis un peu déçu du résultat actuel mais je vais continuer à avancer dessus jusqu’aux présentation. J’ai pu constater le résultat que peut avoir un rapport auxquelles ils manquent des informations important surtout au niveau des composants et de même pour un code qui manquent de commentaire.

Lausanne, le 27.09.2022

Dos Santos Mario

# Annexe

## Cahier des charges

## Planification

## Procès-Verbaux

## Journal de travail

## Schéma électrique complet

## Liste de pièce

## Listing Code

## Mode d’emploi