

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 Hardware	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

1 Introduction

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

1.1 Cahier des charges

Voir Cahier des charges en annexe

1.2 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.

1.3 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.

2 Design

2.1 Boitier

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

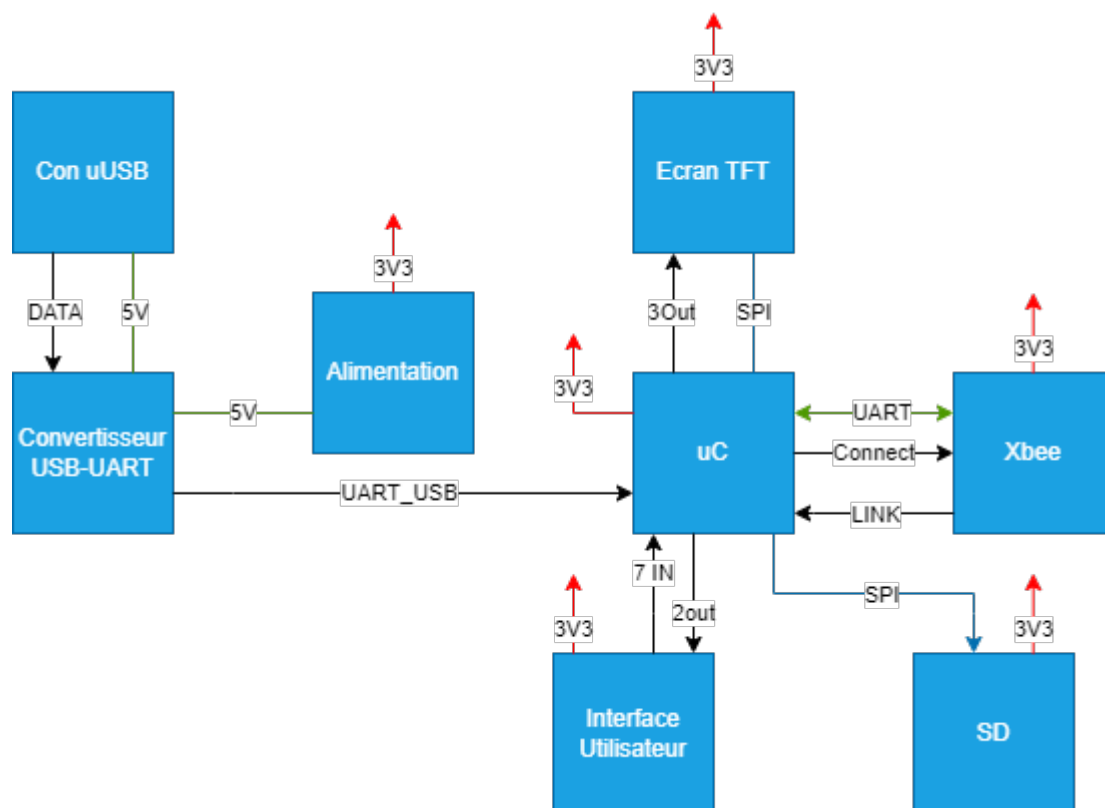
En plus de cela j'ai constaté que l'antenne du Xbee était près des plans de masse ce qui est déconseillé, comme je n'aurais pas la place de le déplacer avec la taille des boîtiers existant je devrais prendre un boîtier plus grand ce qui me permettrait lorsque j'aurais besoin de changer de composant de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

2.2 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

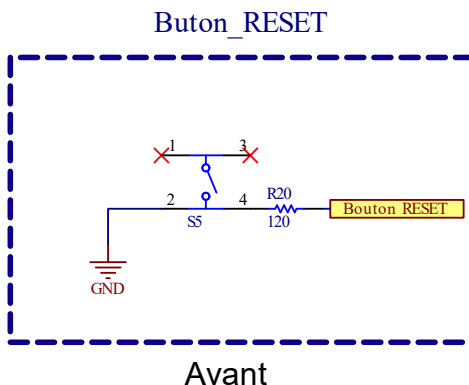
2.2.1 Schéma bloc



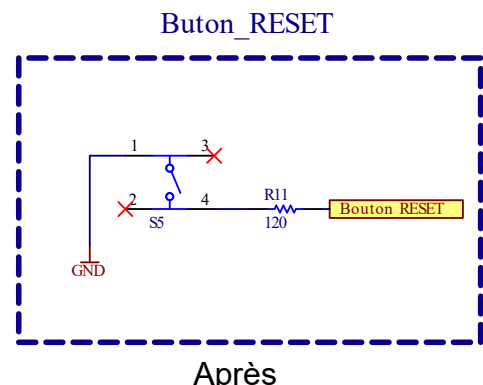
2.2.2 Bouton Reset

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 composants connectés au reset ne pouvaient 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

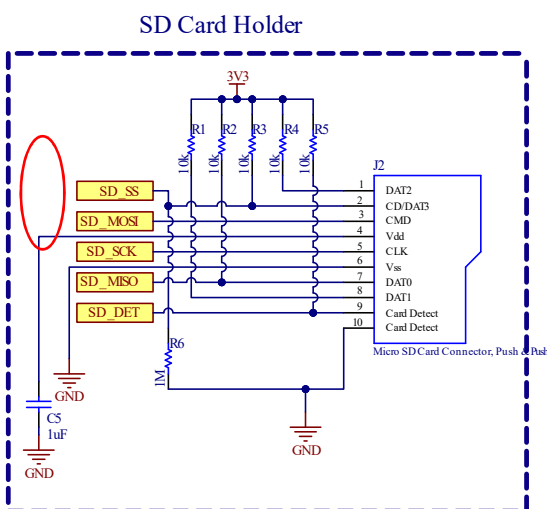


Après

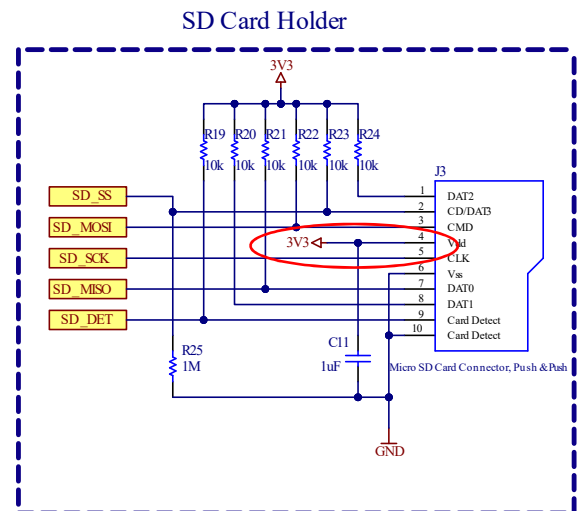
2.2.3 Alimentation carte SD

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

2.2.4 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.

2.2.5 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.

2.2.6 Xbee

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

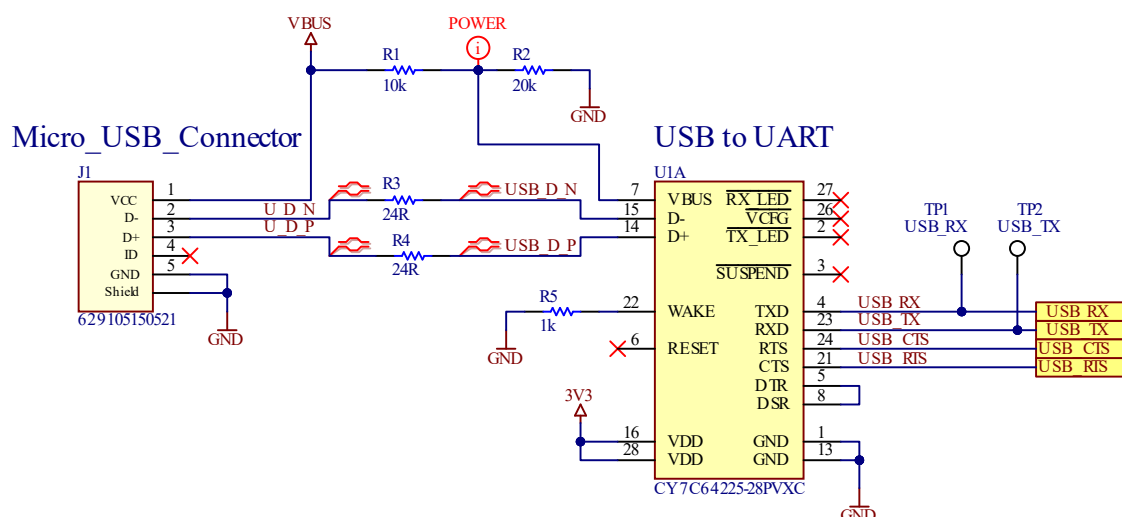
Il a donc été décidé que l'on allait continuer de l'utiliser en espérant que le problème ne me fasse pas perdre trop de temps et si cela ne fonctionne toujours pas nous utiliseront un autre Xbee (XB24CAUIT-001) qui a déjà servi dans plusieurs projets.

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

2.2.7 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 utilisé.



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

If both CY7C64225 and the microcontroller (MCU) are operating at 3.3 V, connect a divider circuit to provide 3.3 V to VBUS pin of CY7C64225 from VBUS pin of USB port.

$$U_{pont} = \frac{R2}{R2 + R1} * V_{BUS} = \frac{20k}{20k + 10k} * 5 = 3,33V$$

2.2.8 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 trouve dans le touchscreen, que je n'utilise pas, je pense qu'il n'y aura pas de problème lors de la programmation.

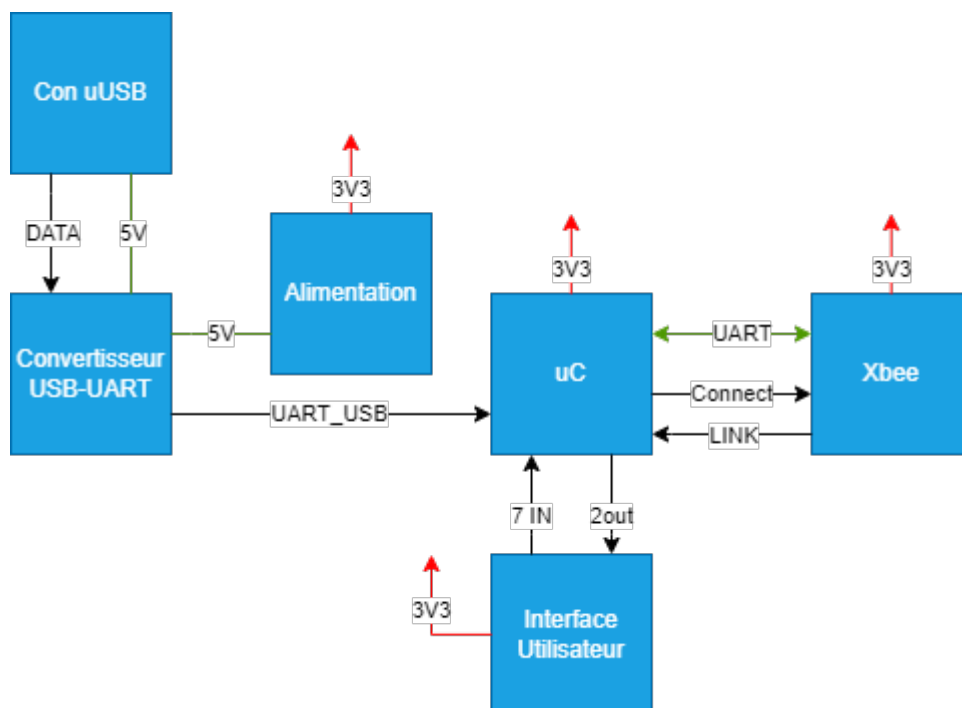
2.3 Carte Slaves

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

Donc :

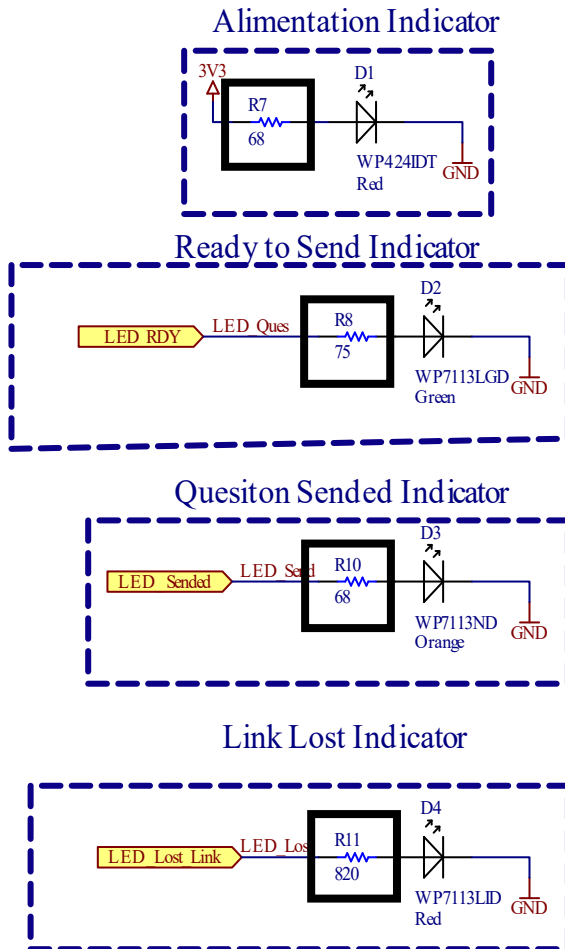
- Correction du bouton reset
- Changement du régulateur 3V3
- Ajout de la communication USB

2.3.1 Schéma bloc



2.3.2 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.

$$R_{alim} = \frac{VCC - V_{led_{red}}}{I_{led_{red}}} = \frac{3.3 - 2}{20m} = 65 \rightarrow 68 E24$$

$$R_{ques} = \frac{VCC - V_{led_{green}}}{I_{led_{green}}} = \frac{3.3 - 1.9}{15m} = 93.3 \rightarrow 100 E24$$

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 indication du datasheet mais avec le courant max des I/O

$$R_{Send} = \frac{VCC - V_{led_{orange}}}{I_{led_{orange}}} = \frac{3.3 - 2.05}{15m} = 83.3 \rightarrow 91 E24$$

$$R_{lost_link} = \frac{VCC - V_{led_{red}}}{I_{led_{red}}} = \frac{3.3 - 1.7}{15m} = 106.6 \rightarrow 110 E24$$

2.4 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 précédent diplômé 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.

2.4.1 Boitier Master

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

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

J'ai paramétré la hauteur à 20 mm car je pensais trouver des boîtiers avec un minimum de 15mm de hauteur en interne du boîtier et j'espérais trouver des boîtiers assez proches de cette taille. Mais sur Hammond ils affichent les boîtiers plus grand que ce que le filtre donc ça éliminait quand même les boîtiers 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.



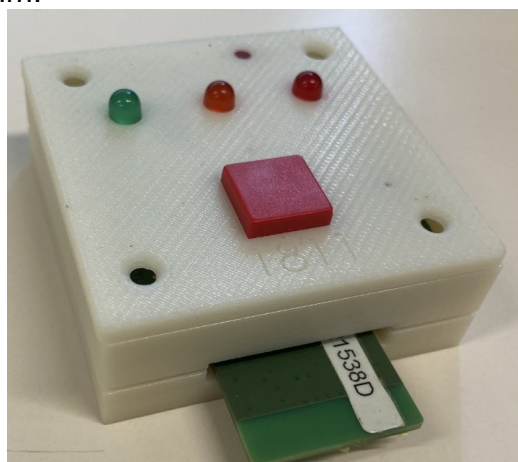
Boîtier Master version A

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

2.4.2 Boîtier Slaves

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

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



Boîtier Slave version A

3 Routage

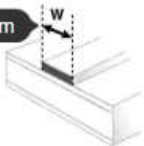
Pour le routage j'utilise la version 21.6.1 d'Altium.

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 :

Courant (I)	<input type="text" value="0.5"/> A	Ambient Temperature	<input type="text" value="25"/> °C
Épaisseur (t)	<input type="text" value="35"/> µm	Longueur de piste	<input type="text" value="Entrez la longueur..."/> po
Augmentation de température (T_{Rise})	<input type="text" value="20"/> °C		

Largeur de piste minimum

0.07581804597 mm



Couches externes à l'air libre

Largeur de piste requise (W)

mm

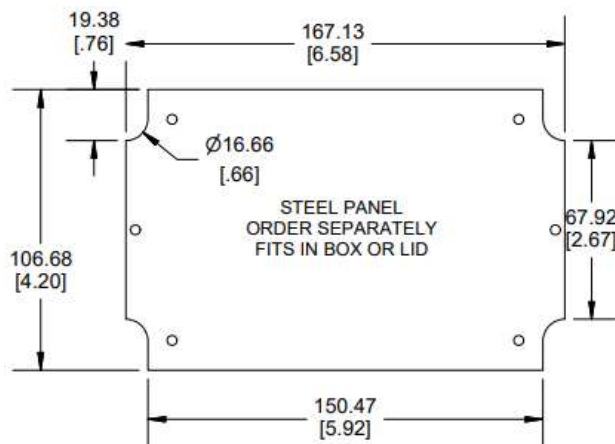
J'ai décidé de faire mon routage avec des pistes de 0.254mm, hors alimentation, et des pistes de 0.5mm pour les piste d'alimentation.

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

3.1 Taille PCB

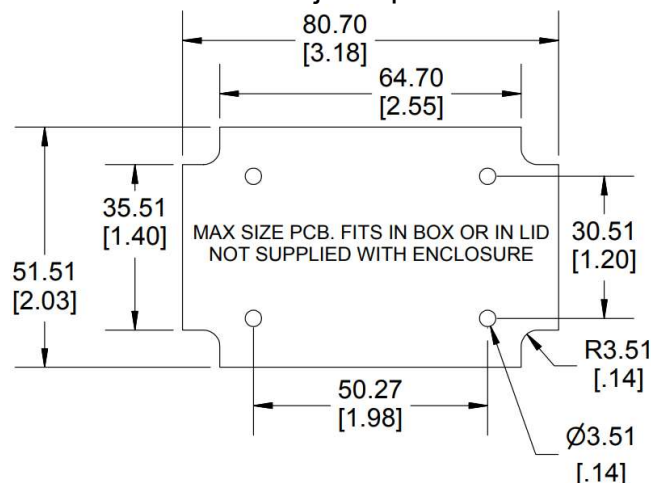
3.1.1 PCB Master

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.



3.1.2 PCB Slave

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



3.2 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.

3.3 Stitching

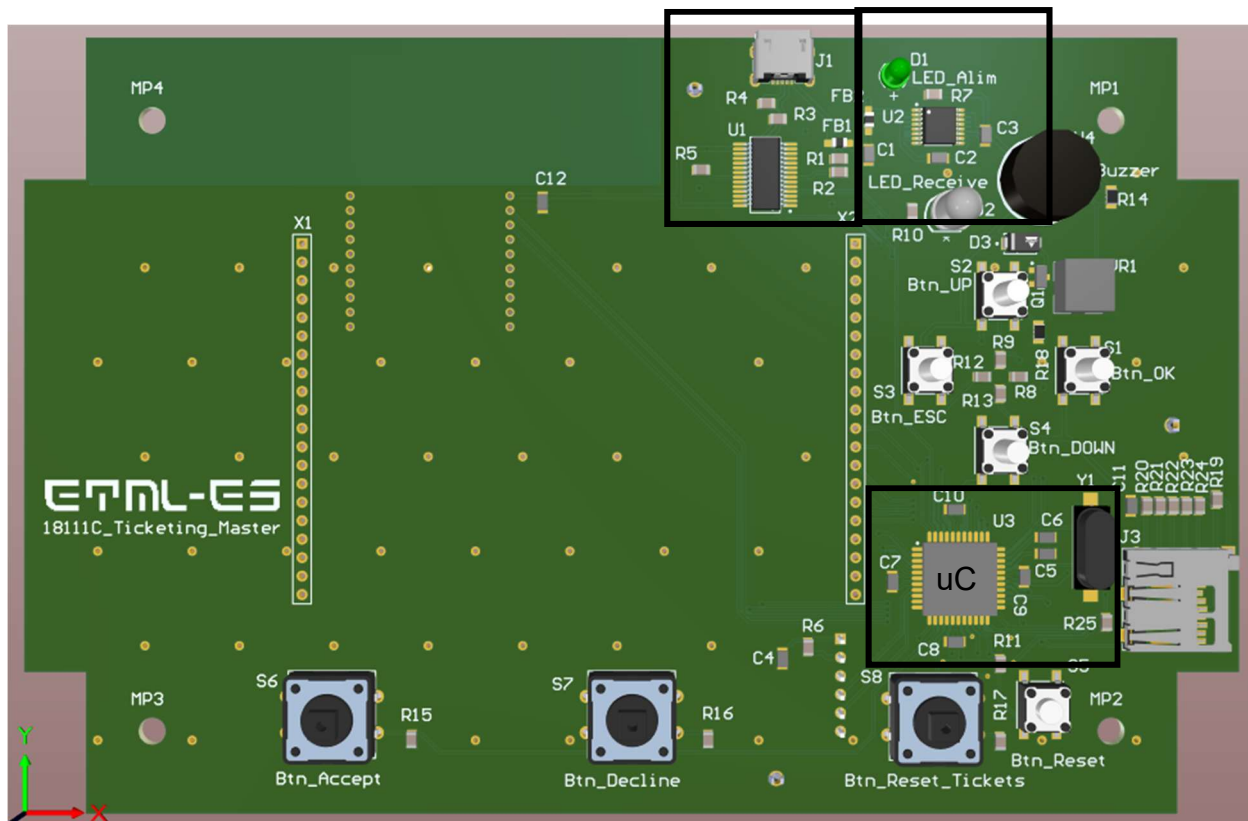
Le stitching permet d'avoir un plan de masse plus homogène sur tout le PCB

3.4 Vue 3D

3.4.1 PCB Master

Convertisseur
USB-UART

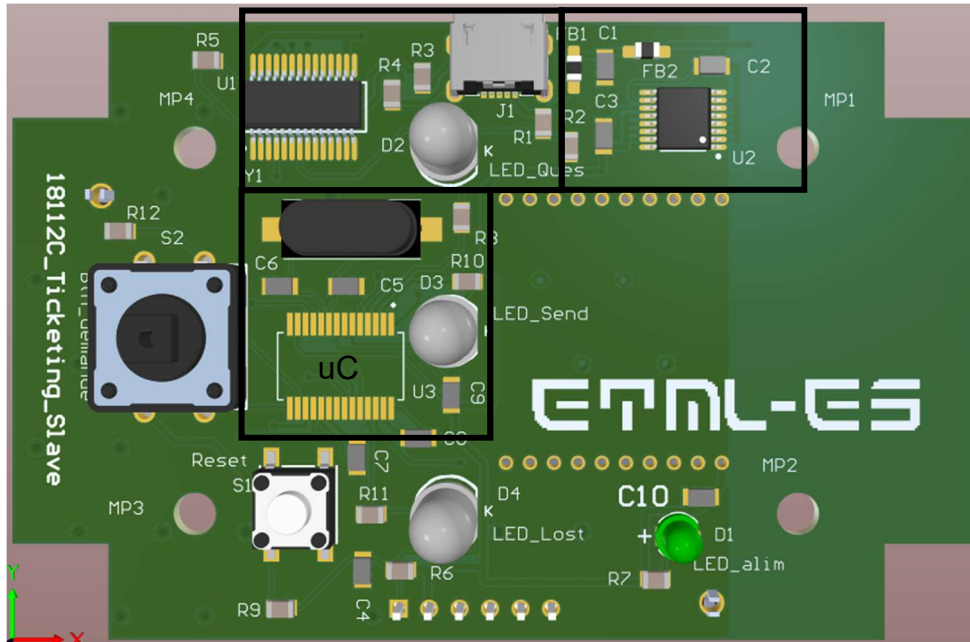
Régulateur
5V to 3V3



3.4.2 PCB Slave

Convertisseur
USB-UART

Régulateur
5V to 3V3



5.1.1.2 UART

UART Xbee

UART USB

Pour le Xbee j'ai repris les paramètres utilisés dans le projet 1623 (20200214_Module_Xbee_06_SCA) qui a été modifié par M.Castoldi.

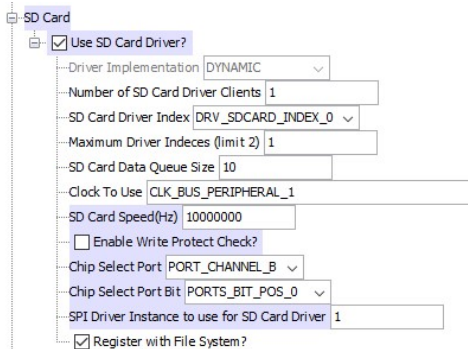
5.1.1.3 SPI

SPI écran TFT

SPI Carte SD

J'ai laissé les paramètres déjà utilisés précédemment. Dans mon projet l'écran et la carte SD ne sont pas essentiels mais je vais quand même essayer de les faire fonctionner.

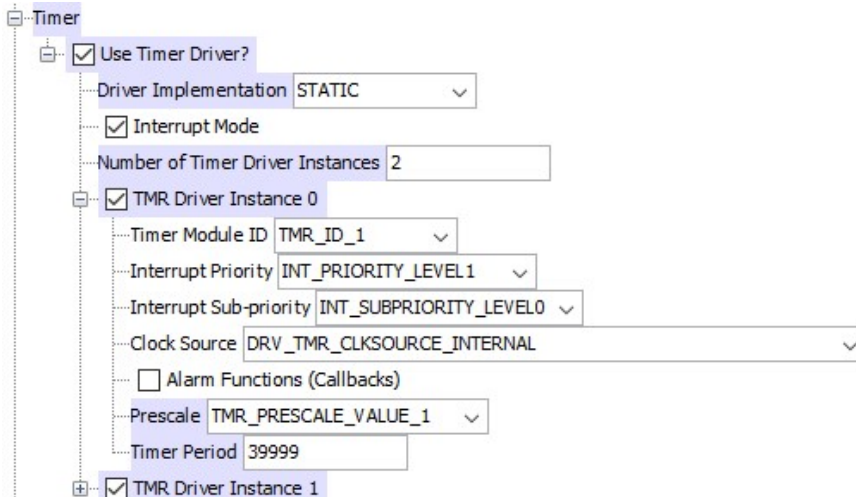
5.1.1.4 Carte SD



Les paramètre de la carte SD avait déjà été implémenter.

5.1.1.5 Timer

5.1.1.5.1 Timer 1



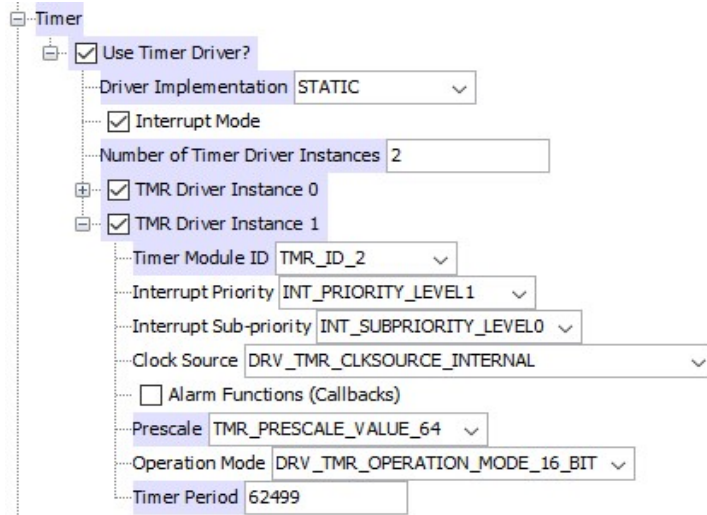
Ce timer est utilisé lors d'une réception de ticket. Il fera vibrer le buzzer à la fréquence donnée.

Le temps du timer est de :

$$Temps = \frac{(Période[tick] + 1) * prescaler}{SYSCLK} = \frac{(39999 + 1) * 1}{40 * 10^6} - 1 = 1 \text{ ms}$$

Ce timer était déjà implémenter dans la version A.

5.1.1.5.2 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

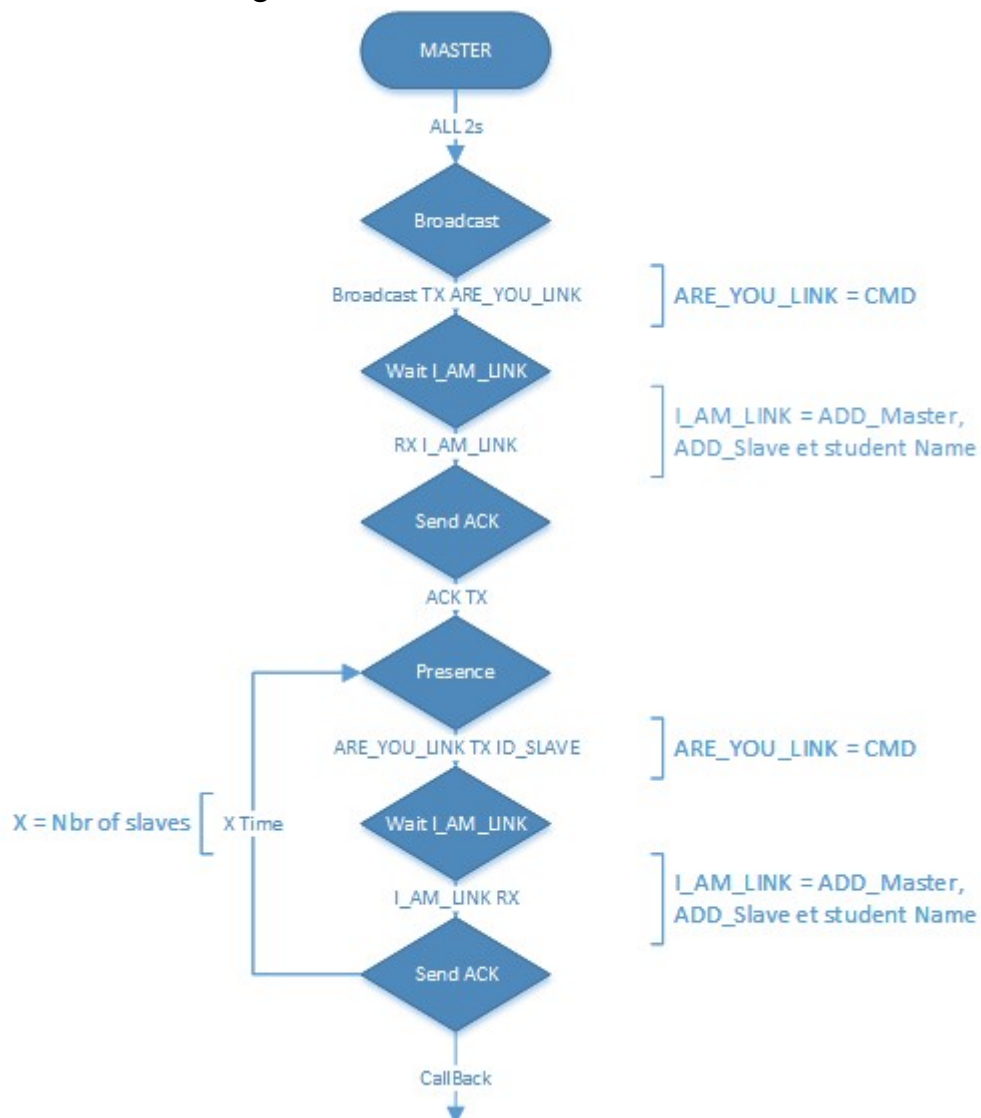
$$Prescaler_{min} = \frac{T * SYSCLK}{2^{16}} = \frac{100 * 10^{-3} * 40 * 10^6}{65536} = 61.03 \rightarrow 64$$

La valeur du timer doit être de :

$$Période[tick] = \frac{T * SYSCLK}{prescaler} - 1 = \frac{100 * 10^{-3} * 40 * 10^6}{64} - 1 = 62499$$

5.1.2 Structogramme/diagramme

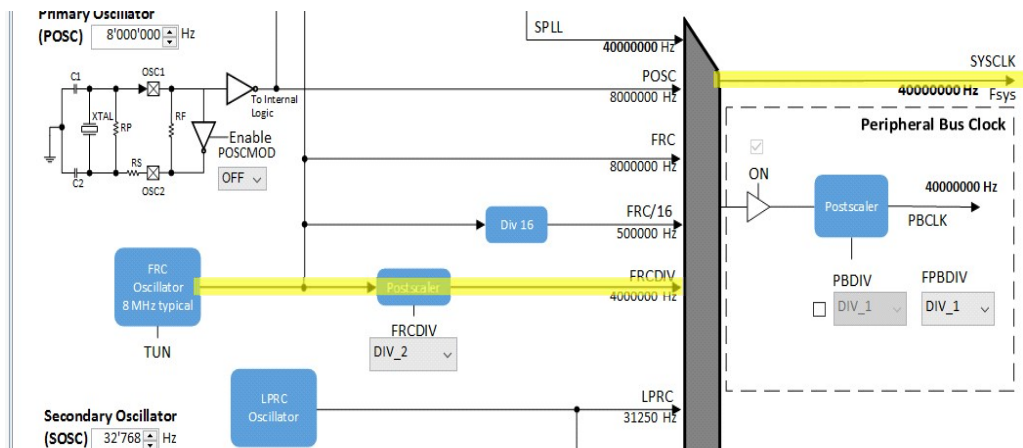
5.1.2.1 Pulling



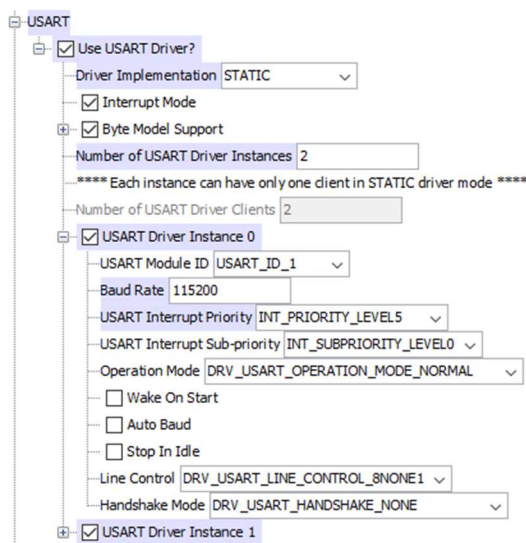
5.2 Slave

5.2.1 Configuration

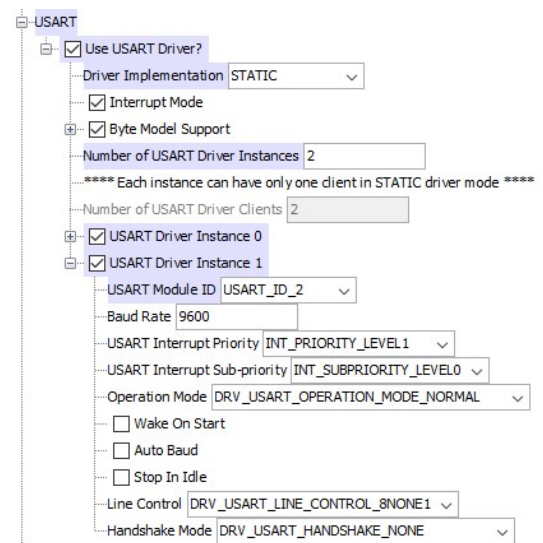
5.2.1.1 System Clock



5.2.1.2 UART



UART Xbee



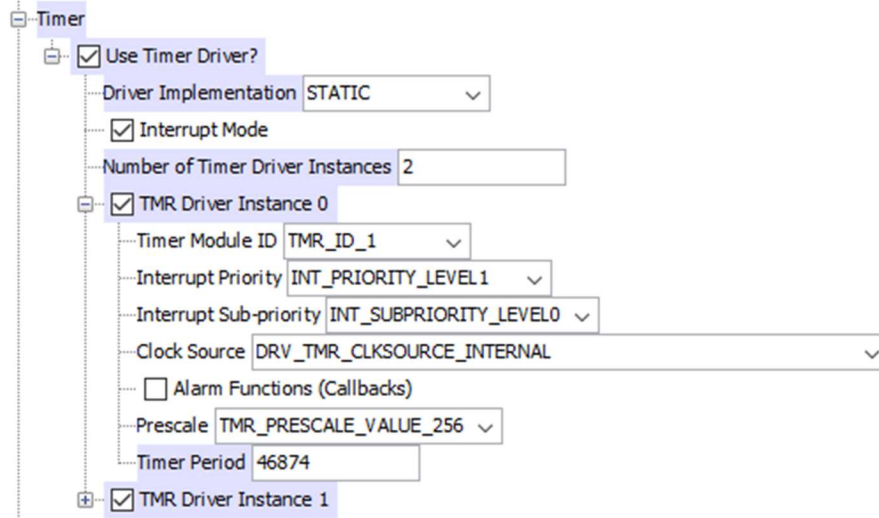
UART USB Slave

J'utilise les mêmes paramètres d'uart du Xbee que le programme maître

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

5.2.1.3 Timer

5.2.1.3.1 Timer 1



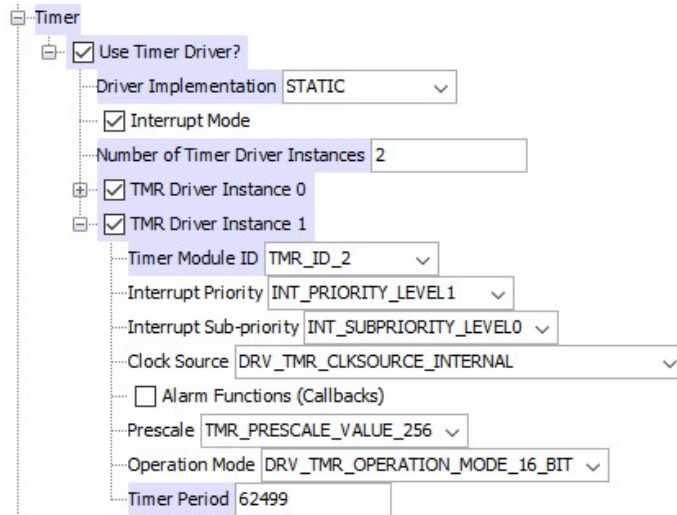
Ce timer est utilisé lorsque la carte Slave n'est pas link au Master. Il fera clignoter la led rouge à la fréquence donnée.

Le temps du timer est de :

$$Temps = \frac{(Période[tick] + 1) * prescaler}{SYSCLK} = \frac{(46874 + 1) * 256}{40 * 10^6} - 1 = 300 ms$$

Ce timer était déjà implémenter dans la version A.

5.2.1.3.2 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.

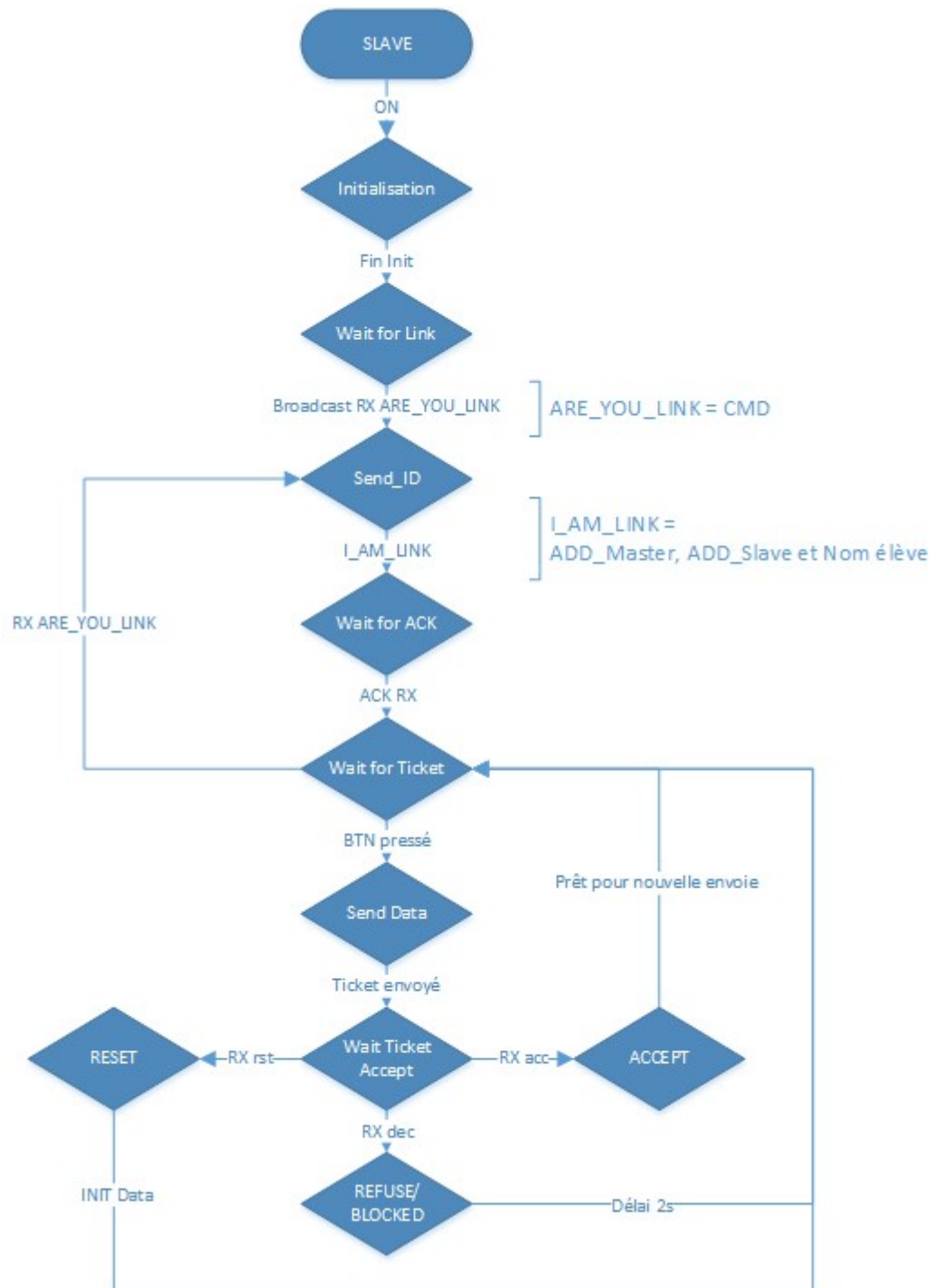
Le clock du microcontrôleur(SYSCLK) est de 40Mhz

$$Prescaler_{min} = \frac{T * SYSCLK}{2^{16}} = \frac{400 * 10^{-3} * 40 * 10^6}{65536} = 244.14 \rightarrow 256$$

La valeur du timer doit être de :

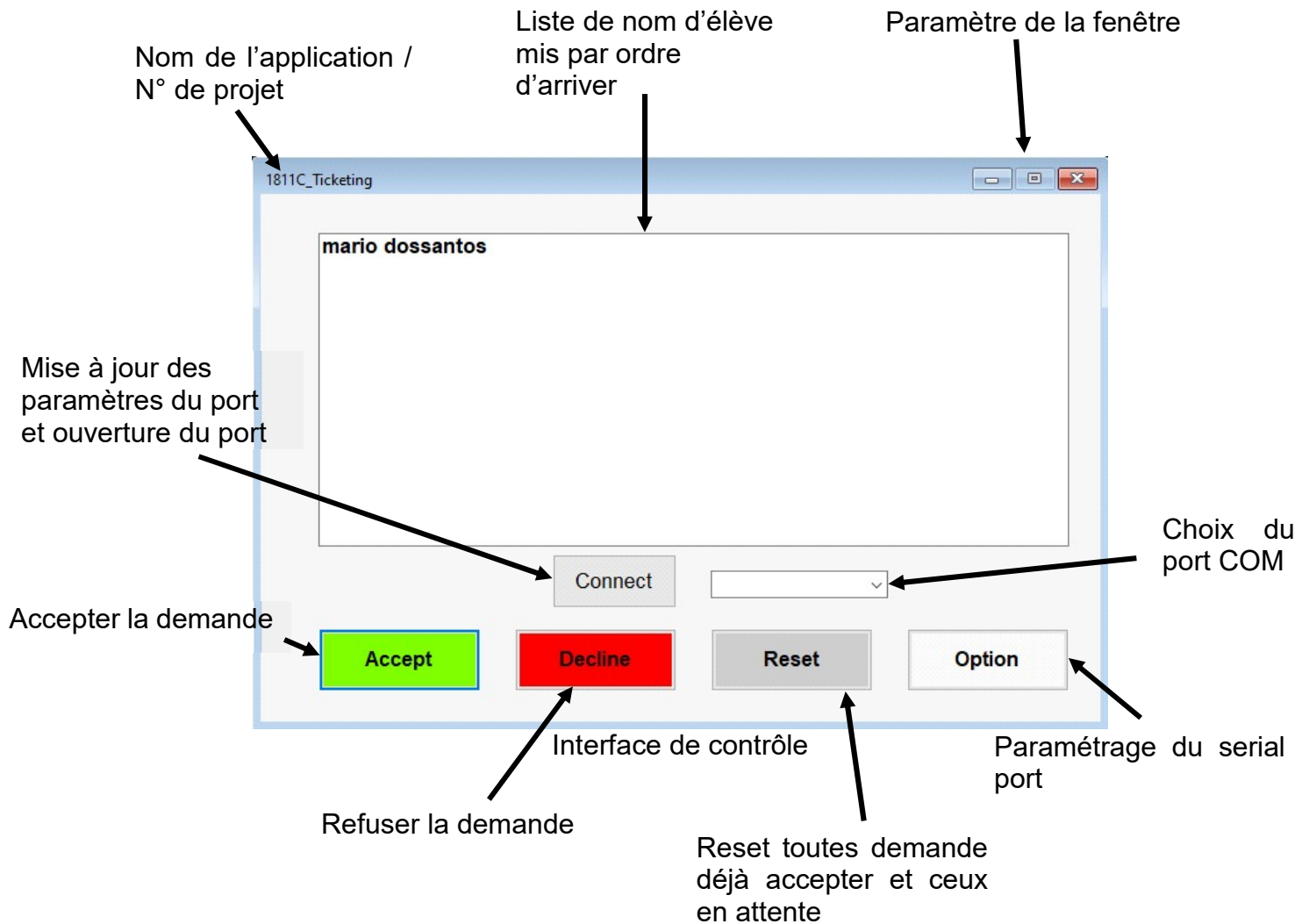
$$Période[tick] = \frac{T * SYSCLK}{prescaler} - 1 = \frac{400 * 10^{-3} * 40 * 10^6}{256} - 1 = 62499$$

5.2.2 Structogramme/diagramme



5.3 Application C#

5.3.1 Interface



Icone de l'application
Nom de l'application /
N° de projet

Paramètre de la fenêtre

1811C_Ticketing Paramètre

Port BaudRate 57600

Port Parity None

Port Data Bits 8

Port Handshake None

Port Stop Bits One

Configure

SLO1 SLO2

Choix du BaudRate

Choix de la parité

Choix du Nombre de Byte

Choix du Handshake

Choix du Nombre de stop bits

Confirmer les paramètres

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

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 précédent et de rouvrir le port COM.

5.3.2 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.

5.4 Trames

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

5.4.1 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

```
#define ENVOI_TICKET 0x917283bd
#define TICKET_ANNULER 0xcccl8ca5

#define TICKET_ACCEPT 0x00e0dl35
#define TICKET_REFUSE 0xc5fb3l40
#define TICKET_RESET 0xb0bfe0fc
#define BLOCKED 0xcc9ebb17

#define ARE_U_LINK 0x3103de90
#define I_AM_LINK 0xcdd05a45

#define ACK 0x1blac4f6
```

5.4.2 Broadcast

Broadcast	ADD Master	ARE_YOU_LINK
-----------	------------	--------------

Broadcast = 0xFFFFFFFF

ADD Master = récupération de l'adresse du Xbee

ARE_YOU_LINK = adresse prédéfini(0x3103de90)

5.4.3 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

6 Test et mesure

6.1 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.

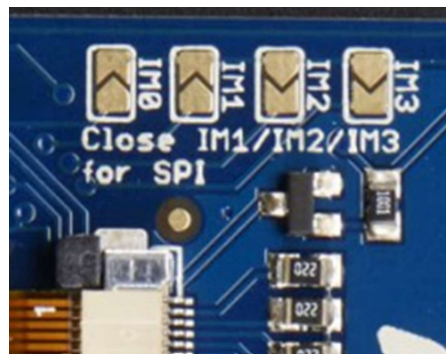
Mais lorsque je l'ai programmé j'ai constaté que seul le backlight fonctionnait. J'ai revérifier que je n'avais pas modifier une partie du code mais je n'ai trouvé aucune différence.

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.

6.2 Driver TFT

Il y a plusieurs fonctions qui fonctionnent entre les 2 LCD mais aussi certaines qui ne fonctionnent 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 fonctions ne font juste rien. N'empêche pas l'écriture de texte.

6.3 Xbee

6.3.1 Code

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

Le module RF est beugé sans comprendre pourquoi, car avant son implémentation sur mes cartes les modules communiquaient bien.

Donc cela me confirme que les Xbee arrivent bel et bien à communiquer entre elles.

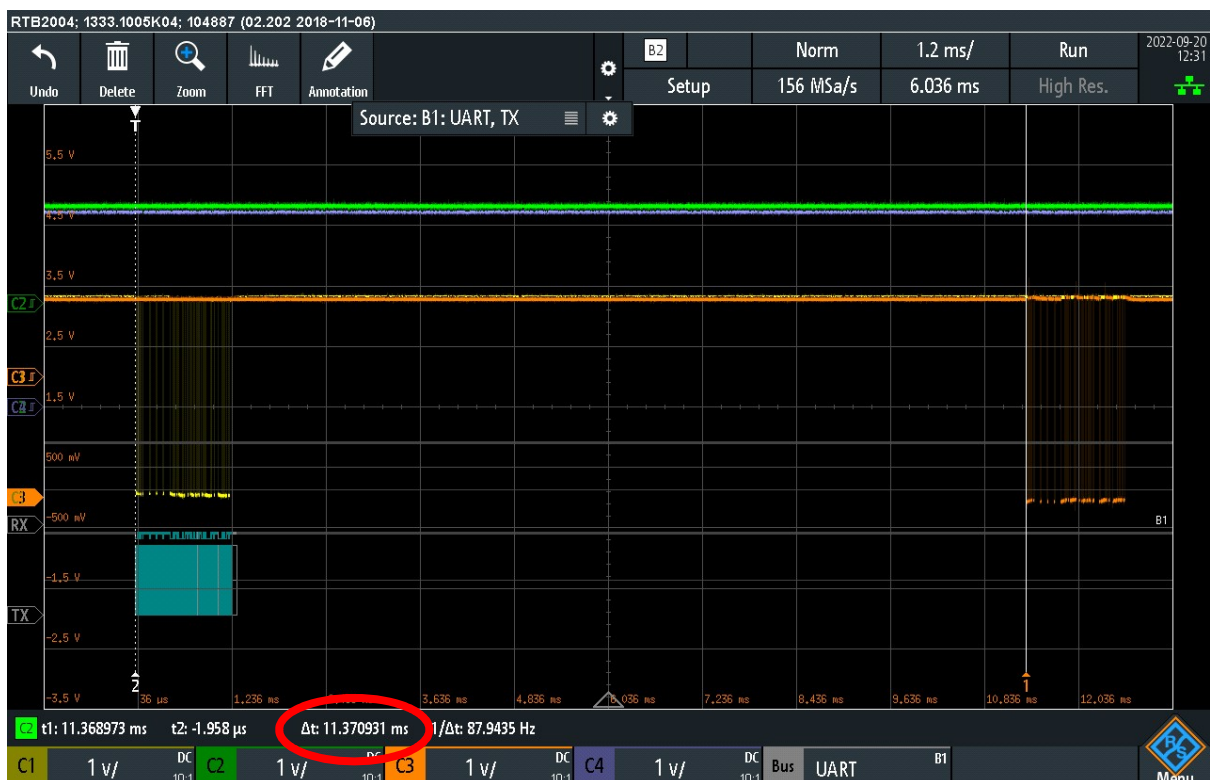
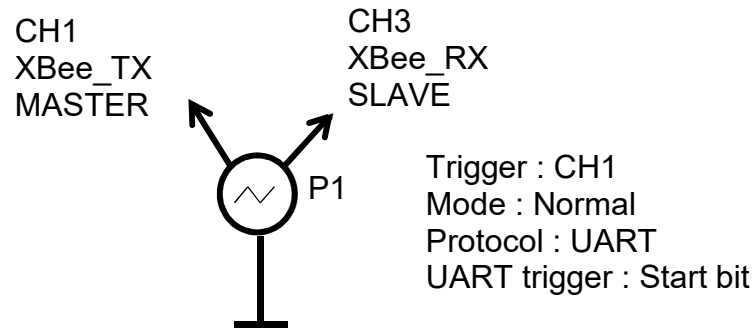
Durant 2 jours j'ai essayé de les faire fonctionner mais sans succès.

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 reprogrammé mes cartes Xbee avec le firmware de M.Castoldi (20200214_Module_Xbee_06_SCA) du projet 1623

6.3.2 Temps de communication



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.



Schéma d'envoi d'une trame

7 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.

8 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

9 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 réduits et cela nous permettrait de prendre un boîtier 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.

10 Conclusion

Ce projet avait l'air assez simple sur le papier mais certains problèmes de firmware et la reprise d'un projet existant m'ont rapidement fait penser le contraire.

La partie hardware c'est bien déroulé et j'ai pu commencer la programmation C# et les dessins mécaniques en attendant la livraison de mes cartes.

La partie firmware était beaucoup plus compliquée 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 souhaitais.

Ce n'est pas la première fois que je reprends un projet pour le continuer/corriger mais à chaque fois je me rends compte qu'il 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 certains qui n'étaient indiqués nulle part. La partie firmware était beaucoup plus compliquée car le code n'était pas commenté et que certaines parties du code, qui étaient écrites dans le rapport comme faites, ne l'étaient pas.

J'ai beaucoup appris de ce projet je suis un peu déçu du résultat actuel mais je vais continuer à avancer dessus jusqu'à la présentation. J'ai pu constater le résultat que peut avoir un rapport auxquelles il manque des informations importantes surtout au niveau des composants et de même pour un code qui manque de commentaires.

Lausanne, le 27.09.2022

Dos Santos Mario

11 Annexe

11.1 Cahier des charges

11.2 Planification

11.3 Procès-Verbaux

11.4 Journal de travail

11.5 Schéma électrique complet

18111C_Ticketing_Schéma_électrique_Master

18112C_Ticketing_Schéma_électrique_Slave

11.6 Liste de pièce

11.7 Listing Code

18111C_Ticketing_Listing_Code_Master

18112C_Ticketing_Listing_Code_Slave

1811C_Ticketing_Listing_Code_Application

11.8 Mode d'emploi

11.9 Datasheets