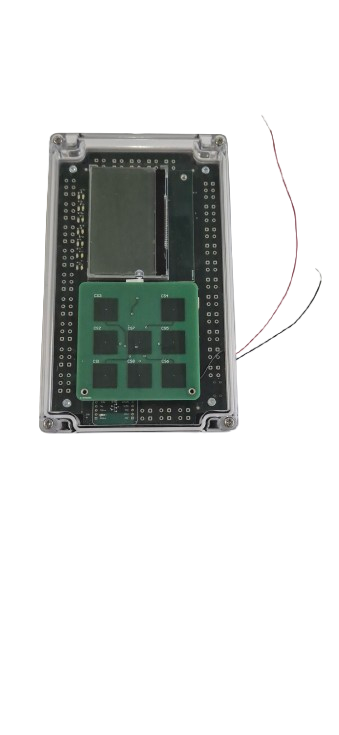
Travail de semestre

**Ecole supérieure**

Électronique

**Réalisé par : A l’attention de :**

Aymeric Clauzel Moreno Juan José

Castoldi Serge

Schubert Edouard

**Dates :**

Début du travail : 19.08.2024

Fin :

# Table des matières :

1 Table des matières : 3

2 Pré-étude : 5

2.1 Introduction : 5

2.2 Cahier des charges : 5

2.3 Schéma général du système : 5

2.4 Schéma bloc hardware : 7

2.5 Choix technologiques : 7

2.6 Interaction avec le monde : 10

2.7 Évaluation des couts : 13

2.8 Planning : 14

2.9 Conclusion & perspectives : 15

2.9.1 Retour sur la présentation: 16

3 Design 17

3.1 Microcontrôleur 17

3.2 Entrée spéciales : 17

3.3 Bluetooth 19

3.4 Sortie relais 19

3.5 Entrées analogique 19

3.6 Protection 20

3.7 Switch de configuratio4n 20

3.8 PCB TOUCH\_CAPA 20

3.9 Réalisation 21

3.9.1 Routage du PCB : 21

3.9.1.1 Points critiques ou particuliers 26

3.9.2 Montage/ Mise en service 29

3.9.3 Mise en service 31

3.9.4 Programmation logicielle 33

3.9.5 Tests et mesures : 39

3.9.6 Modifications hardware : 40

3.9.6.1 Raisons 40

3.9.7 Intégration mécanique 43

3.9.8 Problèmes rencontrés restant sans solutions 44

3.10 Analyse de durabilité 45

3.10.1 Aspects environnementaux 45

3.10.2 Aspects sociaux 45

3.10.3 Aspects économiques 45

3.11 Conclusion 45

3.11.1 Bilan global 45

3.11.2 Compétences acquises 46

3.11.3 Améliorations possibles / suite du projet 46

3.11.4 Mode d'emploi du système 46

3.12 Annexes 46

3.12.1 Cahier des charges initial 46

3.12.2 Fichiers de CAO (schémas, PCB, implantation) 46

3.12.3 Liste des pièces et coûts réels 46

3.12.4 Listing des différentes apps 46

3.12.5 Planning initial vs réel 46

3.12.6 Journal de travail - git commit 46

# Pré-étude :

## Introduction :

Au cours de cette deuxième année à l’ETML-ES, un projet de semestre nous est attribué. Certains proviennent de clients extérieurs tels que DES sécurité.

Dans le cadre de ce projet il nous est d’abord demander de faire le cahier des charge.

Cette phase de pré-étude permet pour ma part de clarifier le cahier des charge transmis. Pour ma part 2 cahier des charges m’ont été transmis l’un par M. Schubert, l’autre part M. Moreno, ce qui ne simplifie pas la chose.

Cette pré-étude a été faite avec peu de compréhension des domaine « détection incendie » et des normes en découlant n’y ayant jamais été confronté. Le document évoluera après le rendus et le rapport pour devenir le rapports final. Elle sert de « proposition » au client et évolue au fils des discussions.

## Cahier des charges :

Voir les annexes physique : « dossier rouge » & « DES SPB 7 Test Sprinkler 2124 »

Cahier des charges final, voir annexe : « 2414-DES\_SPB\_7\_TestSprinkler-CDC »

## Schéma général du système :

Une image contenant texte, carte de visite, Police, capture d’écran

Description générée automatiquement

Le système à réaliser se présente en deux différent PCB « TouchCapa » et « SPB7 » dans la figure de gauche.

Le PCB « 4IN/2Out » de la marque HoneyWell est fournis par M. Schubert. Et doit venir se monter directement sur le « SPB7 ». Quant au « TouchCapa » il doit se trouver directement contre le couvercle du boitier.

Figure 1 Principe

Une image contenant texte, capture d’écran, diagramme, Police

Description générée automatiquement

Le PCB « TouchCapa » auras pour but de rester très « vide » avec le strict nécessaire. La contrainte mécanique permet d’assurer la détection l’ors de l’appuis.

Figure 2 touch capa

Une image contenant diagramme, Rectangle, capture d’écran, carré

Description générée automatiquement

Figure 3 SPB7

Une image contenant capture d’écran, vert, conception

Description générée automatiquement

Pour visualiser l’encombrement dans le boitier une des première chose importante est de déjà ajouter des composants aux fichier d’assemblage du boitier transmis par M. Moreno.

Figure 4 une idée du Touchcapa

Une image contenant Ordinateur tablette, capture d’écran, texte, gadget

Description générée automatiquement

Figure 5idée d ela place prise

## Schéma bloc hardware :

Une image contenant capture d’écran, Rectangle, carré, conception

Description générée automatiquement

Figure 6 schéma bloc

## Choix technologiques :

Avant de parler des choix, commençons par les choses imposées et les choix en découlant. Tout d’abord le système se doit pouvoir intégrer un module existant de chez Honeywell, le « esserbus alarm transponder, 4 IN/2 OUT ». Ce module a été fournis avec les documents disponible lors de l’achat.

Une image contenant Composant électronique, circuit, Ingénierie électronique, Appareils électroniques

Description générée automatiquement

Figure 7 4In/2Out board

Le boitier doit avoir un couvercle transparent. Et par chance a été fournis par M.Moreno :

Une image contenant évier

Description générée automatiquement

Ce boitier de chez Bud Industries fait 200mm x 119,99 mm pour 75,13mm de haut, en PC (polycarbonates) plastic. Avec une épaisseur de 3,05mm, et un couvercle transparent. De plus le boitier possède des inserts pours montage et le statut actif.

Figure 8 PN-1334-C

Les connecteurs :

Une demande est que les connecteur fils-à carte ne sois pas à vis (terminal block). Vis-à-vis de cette demande, j’ai eu l’occasion de confirmer le type connecteur, grâce à une visite de Schubert. les connecteurs utilisés :



Le choix s’est prononcé pour la simplicité d’utilisation, un tournevis permettant d’actionner le ressort. Ce connecteur accepte des fils de 10 à 28 AWG, ce qui correspond au 0,8mm demandés pour les câbles U72

Figure 9 691413720002B

Fixation mécanique :

Entre les cartes des spacer SMD avec filetage M3 serons surement utilisées, cars si bien montés, ils apportent l’avantage de pouvoir géré l’espacement avec d’autre pcb via des entretoises ou autres solutions choisies plus tard (le modèle final reste à définir):



Figure 10 9774020360R

Pour les TouchCapa :

l’IC imposé est le AT42QT2120, de chez Atmel, il permet jusqu’à 9 touches + une Wheel/slider, ce qui semble sympa à tester. Le nombre de touche étant laissé à définir mon envie est d’ajouter au moins un slider si la place le permet. La figure ci-dessous, montre que le boitier ne sera pas un problème pour la détection des touches ou du slider.  
 Une image contenant texte, Police, capture d’écran, blanc

Description générée automatiquement

Figure 11AT42QT2120

Le choix des connecteurs carte à carte :

Servant à relier les PCB « TouchCapa » & « MainBoard », a été orienté par M. Moreno, mon choix s’est précisé, pour la facilité d’accessibilité à déclencher l’étrier depuis le dessus. Les choix du connecteur :



Validés par M. Schubert ces connecteurs de chez Wurth, 10 position pour câble plats de 3mm de haut pour 17mm de long

Figure 12 68611014422

Logger de data :

Un socle pour carte microSD a été choisi car il me semblait judicieux, dans l’idée ou les données pourraient être lues sur différents systèmes style laptop smartphone tablette.

Les protections :

A propos des protection des entrées le choix n’as pas encore été fait entre diodes zener ou TVS. L’idée d’optocoupleur reste en suspens pour les entrées digitales mais dans l’idée permettant d’appliquer un rapport directement aux tensions lues, mais impliquerais des une baisse de réactivité du système, et pour des raisons de couts.

Régulation :

L’alimentation du module Esser est de 10-28V, l’utilisation d’un buck semble propice puisque que la demande de M.Schubert est que le systèmes puisse fonctionner entre 10-42V. Le choix n’a pas encore été fait.

La rtc :

La carte utilisée pour les TP de l’école seras prise comme exemple, car elle est fonctionnelle et que les composant sont disponibles. A voir pour trouver une RTC en SPI, ce qui permettrais de réduire le nombre de pins.

Des light pipes :

Pour améliorer la diffusion l’utilisation de light pipe est validée par M. Schubert, le modèle précis n’est pas encore défini.

Le microcontrôleur :

Dans la hâte le même microcontrôleur me semble être le plus adéquat, il permet l’utilisation du Protocol USB, possède un AD, et la vitesse nécessaire afin d’avoir l’occasion de séparer les différentes fonctionnalités par port, le choix demande à être précisé. Le courant max que peux fournir le microcontrôleur est de 200mA, pour piloter les led une estimation faite est ~5mA par LED ce qui ferais un peux moins de 100mA au total. Le format 100 pin étant disponible, ne connaissant pas de microcontrôleur avec autant d’entrée ou sortie sans l’utilisation d’io extander, c’est sur cette solution que je me dirige dépassant les 64 pins avant d’avoir fini cette pré-étude.

Relais :

Des relais dans le format de ceux du module 4in/2out ont été choisi, avec une tension de commande de 3,3V, les relais piloterons une tensions de 10V avec un courant max de 2A. Ils font 20x10x11mm ce qui semble adéquat vu le nombre devant être à disposition. Des relais de secours avec un format plus petit ont été choisi mais un courant de commutation de 1A max, ceux-ci mesurent 10,6X5,3X6,5mm.

Le moyens de connecter le module Esser reste encore à trouver, la fenêtre est quasiment certaine pour laisser l’espacement des composants, mais le moyens de fixation qui assure une connexion optimale reste encore à trouver

## Interaction avec le monde :

Les entrées :

Les TouchCapa :

Une des consigne qui m’a été donné étais d’utiliser le projet « » de M.Cohello, de le mettre en service et l’utiliser pour tester la détection aux travers d’une paroi de boitier « «  de « «  d’épaisseur. Des mesures ont été faite grâce à l’aide de M. Castoldi, qui ont permis de valider l’utilisation de ce microcontrôleur et des touches capacitives conseillées. Voir résultats

en **annexe**

Pour les PCB, il me fallait d’abords connaitre les dimensions pour faire une estimation. Pour le « TouchCapa », une touche devant faire la taille d’un doigt (~10mm de diamètre), selon la disposition plus haut un pcb en figure XX, une estimation de la taille serais 60mm x 70 mm

Concernant le SPB7, je vais tenter de faire le plus grand possible pour avoir le maximum d’espace pour les connecteurs et relais quitte à réduire la taille au fur et à mesure du design (pour le SPB7).

Figure 13 1051620001

Une carte SD doit être disponible pour « logger » les données et permettre l’analyse postérieure, le choix du connecteur a déjà été fait, le plus petit possible pour laisser la place aux header et aux relais :   
 Une image contenant texte, capture d’écran

Description générée automatiquement

Figure 14 RN4678 Click

Le module Bluetooth :

L’école étant familière à l’utilisation des module MikroE, je n’ai pas cherché d’autre alternatives face aux arguments, tels que la simplicité à changer de protocole de communication avec des adaptation software. Le module, il mesure 43x25,4mm :



Concernant la signalisation sonore :

il est demandé d’ajouter un buzzer. Dans un boitier hermétiquement fermés, le son ne dois surement pas bien se propager, j’ai donc décider d’ajouter une proposition/alternative, ajouter un transducteur ultrason pour tenter de faire résonner le boitier à une fréquence audible. Après avoir fait un test à la maison sans aop, avec juste le transduceur directement connecter sur une entrée TTL j’ai réalisé que dans un boitier ou non c’est le transduceur lui-même qui émet le sons, malgré qu’il soit audible dans l’état c’est inutilisable. Peut-être avec un actionneur piezzo. Dommage de ne pas avoir pris plus de temps pour faire plus de différents tests. Aucun composant n’a encore été choisi.

Les switch de configuration :

Encore une fois grâce un M. Moreno, un choix a été fait malheureusement j’ai préféré chercher une option un peu moins cher en restant dans l’idée d’utiliser des commutateur au lieu de cavalier, pour faciliter l’utilisation et la configuration lors de l’installation du système. :



Figure 15 SLW-913535-2A-SMT

Les sorties :

Moniteur graphique :

En discutant avec M. Schubert j’ai compris que l’afficheur servait à « nommée » les Signaux aux entrées des 5 entrées libres. Les plus grand LCD « standards » utilisé à l’ES n’ont que 4 lignes, nous avons donc décider de partir sur un LCD graphique. Lequel a été choisi dans la foulée : Une image contenant texte, horloge, nombre, Police

Description générée automatiquement

Figure 16 NHD-C12864LZ-FSW-FBW-3V3

Les voyants :

Ils n’ont pas encore été choisi mais le but est de trouver des voyant qui consomment environs 5 mA chacun. Au nombre de 17 de couleurs différentes. 2 Led de vie ont été ajoutée, celle de la partie alimentation qui est « bête » et celle de vie du microcontrôleur (Controller par ce dernier)

* 5 Free\_IN, orange
* 1 Vanne IN orange
* 1 rouge in presostat
* 7 blanche save
* 1 verte LIFE\_SYS indication
* 1 BT bleue
* 1 verte ALIM vie

## Évaluation des couts :



## Planning :



## Conclusion & perspectives :

Beaucoup de temps est nécessaires à la mise en place rien que du cahier des charges de ce projet, qui semble être un projets plutôt complet, mes estimations lors de la planification sont à revoir. De plus il semble compliquer d’arriver à une solution fiable et fonctionnel du premier coup en si peu de temps prenant en considération le temps nécessaire à la compréhension du sujet, ainsi qu’aux choix à faire restant.

En effet la méthode de détection d’interruption de ligne reste encore à définir avec la méthode pull-up/dow + zener.

Notamment les presses étoupes.

La manière dont les résistances 10k pour fermer les boucles / contact SPB\_IN, seront mis en place (manuelle/automatique).

Les protections des entrées ne sont pas encore clairement définies, ainsi que les protections des relais (Schottky ou pas ?), Sans penser aux sur multiples oublis de ma parts lors de cette pré-étude. la pré-étude n’étant pas terminée il est prioritaire de finir les test nécessaires et validé les choix faits jusqu’ici, pour aller de l’avant.

perspectives :

Difficile de se projeter, alors autant se concentrer sur ce qui reste à portée.

L’esthétique globale seras une chose importante, la possibilité d’un stickers pour la face avant doit être proposée. Je m’y suis surement pris un peu trop à la cool pour le départ de ce projet, il reste des tonnes de travail à abattre et peu de temps pour le faire. Je crains de ne pas réussir à fournir de résultat.

### Retour sur la présentation:

Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

Description générée automatiquement

À la suite de ces retours :

Des recherches ont été faites, le projet doit-il suivre les normes de détection incendie. N’ayant pas accès aux normes ni extraits (EN54), la vue d’ensemble manquera toujours.

Concernant la solution USB, le choix d’un FTD231 est fait, car j’ai choisi de ne pas changer les habitudes de l’ES, d’utiliser un chip qui simplifie la vie.

Depuis la présentation, je suis revenu sur mon choix de ne pas utiliser d’optocoupleur, car au fils des recherches j’ai trouvé des optocoupleurs linéaires utiliser dans les boucle de courant.

Et finalement après conseils de Mr Moreno, je suis revenu encore une fois sur ce choix.

La RTC sera alimenter par l’entrée du système travers du régulateur avec de grande capacités de découplage.

# Design

## Microcontrôleur

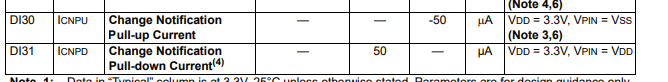
Suite à la présentation, le conseil était d’utiliser le même microcontrôleur du starter kit utiliser pour la formation en deuxième année disponible dans le projet n°11020E.

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

L’avantage est que j’ai pu récupérer pas mal de partie de schéma supposément juste. Car fournis par nos enseignants. Comme la RTC, la carte SD, les différents oscillateur, l’interface UART, le module MIKROE, la partie de reset et debug,

## Entrée spéciales :

Pour la manière de lire les contact secs, j’ai d’abord voulu utiliser les pull-ups et pull-downs du microcontrôleur. En effectuant des recherches sur les valeurs dans le datasheet du microcontrôleur (figure du dessous), j’ai pu trouver cette information, dans un forum, quelqu’un expliquait qu’il fallait plus les vois comme des sources de courant. J’ai pensé plus sûr, d’utiliser un autre moyens pour être sûr de fournir suffisamment de courant. 

J’ai d’abord penser récuperer la source de courant de servant à alimenter la backlight du projet n°11020, étant une source de courant 20mA, ajouter un suiveur et une resistance de 500Ω reviendrais à au même que la lecture des capteur 4-20mA, en recalculant la résistance en parallèle de la jonction du transitor pour le courant voulus. Suivit d’un suiveur, et d’une résistance.

Une image contenant texte, diagramme, ligne, nombre

Description générée automatiquement

Voyant le nombre de composants, et le nombre d’entrée à lire, j’ai changé d’avis pour simplifier le tout, utiliser simplement 3 résistance de 100 Ω.

Une image contenant diagramme, texte, ligne, capture d’écran

Description générée automatiquement

Le cas critique étant le moment ou la ligne est coupée, uniquement la résistance de « pull-up » limitera le courant. Le courant admissible par les entrée du microcontrôleur choisit est d’environ 15mA maximum d’après le tableau ci-dessous.

Une image contenant texte, nombre, Police, capture d’écran

Description générée automatiquement

J’ai pensé à garder l’aop et ajouter un gain de 0.5 par exemple, pour finalement décider de choisir une résistance de 330Ω

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| res | ligne coupée | | contact ouvert | | contact fermer | |
|  | U | I | U | I | U | I |
| 100 | 3,3 | 0,033 | 1,98 | 0,00792 | 1,65 | 0,0066 |
| 330 | 3,3 | 0,01 | 1,1 | 0,002222222 | 1,65 | 0,003333 |

Un des avantage est que les entrée avec nécessité du dispositif EOL-R sont « compatible » avec la lecture des contact sec, la différence est la nécessité des entrées analogiques permettant de lires les différents niveaux de tension. Alors que pour la lecture des contact sec une pull-up suffit. Le courant étant « contrôler », malheureusement est-ce que 10mA suffirons ?

## Bluetooth

Une des demande qui m’a été faite est de trouvé une alternative pour le module bluetooth. Celle que j’ai trouvée semble adéquate mais au vue des pin à dispositions sur le microcontrôleur une discussions est nécessaires. Car dans l’idée, il me fallait ajouter les empreintes des 2 modules pour plus tard, choisir lequel monter lors de l’assemblage. Une solution serait de ne pas mettre l’empreinte du module mikroE, uniquement celui du module. Ou ajouter un I/O extander pour piloter les leds de signalisation, et laisser de l’espace directement sur le microcontrôleur

Une image contenant texte, diagramme, Plan, Dessin technique

Description générée automatiquement

## Sortie relais

Pour piloter les relais, un MOS N, avec une résistance série, et une pull-down, pour figer l’état à 0V. et une diode de roue libre 1N4001, car je ne pense pas qu’une Schottky soit nécessaires.

Une image contenant texte, diagramme, ligne, Parallèle

Description générée automatiquement

## Entrées analogique

Le but du montage permet la lecture d’un capteur 0-10V entre 0 et 3,3V c’est un simple pont diviseur 1/3. Lorsque le commutateur est enclenché la résistance de 500Ω mise en parallèle fixe la tension sur le pont à 10V . Ce qui permet de lire la tension une fois encore entre 0 et 3V.

Une image contenant texte, diagramme, nombre, ligne

Description générée automatiquement

## Protection

Les diodes TVS choisies sont celle à disposition dans le stock de l’ES qui nous à été conseiller d’utiliser pour ces projets. Elles ont été placées sur les entrée analogiques, les entrées destinées à d’autres SPB, et sur les entrée « Spéciales » « SPE\_IN ».

Une image contenant diagramme, ligne, croquis, Dessin technique

Description générée automatiquement

## Switch de configuratio4n

Pour la redirection des signaux sur la sortie de dérangements une série de switch directement reliés à des pull-up / pull-down.

Une image contenant texte, nombre, capture d’écran, Police

Description générée automatiquement

## PCB TOUCH\_CAPA

Pour le pcb touch capca, il reste simple avec comme exemple le projet n°2123 Démonstrateur de panneau Touch. Les résistances ont été sélectionnées à la valeur minimale recommandée par le datasheet pour apporter le plus de sensibilité possible.

Une image contenant texte, diagramme, Plan, ligne

Description générée automatiquement

## Réalisation

### Routage du PCB :

|  |
| --- |
| Figure 17 emcombrement des différents pcb dans le boitier |
| J’ai commencé importer le boitier sur la couche Mec Sketch afin de crée les contours de la carte en laissant 1mm de jeu. Placer les trous de fixation puis les autres PCB avec eux aussi leurs trous de fixation.  Ensuite le placement des composants, à été fait par fonctionnalité puis en essayant de faire au mieux pour toutes les spécificités de chaque bloc. Exemple : les antennes et connecteurs en bords de carte. Puis l’alimentation à été placer dans un coin pour plus de facilité. Ensuite le Microcontrôleur au cœur etc.. |

|  |
| --- |
| Une image contenant capture d’écran, Rectangle, art, Caractère coloré  Le contenu généré par l’IA peut être incorrect.  Figure 18 Vue de la face avant |
| Voici le Top layer du PCB, on peut y distinguer les différents bloc ou parties du schéma comme les ConfigsSwitchs, les registres à décalages, les tableaux de transistor, et bien sur les connecteurs tout le tour excepté dans le bord droite ne bas pour laisser les modules en bords de carte histoire de favoriser l’émission/réception. |

|  |
| --- |
| Figure 19Vue de la face arrière |
| On peut facilement suivre la piste +24V, allant alimenter le module 4In/2out. Pour les couches internes elles n’ont rien de particulier. La disposition des plans est faite de manière à retrouver un plan de GND aux faces de contact à l’air et enfouir un plan de 3,3V. De telle manière j’imagine que le découplage capacitif entre VCC et GND seras plus grand et donc meilleur. Dans l’idée avoir les plans de GND en extérieur permettrais d’avois une meilleure immunité aux bruits pensant aux CEM. Sans oublier l’aisance du routage des pistes VCC et GND depuis des plans (non-interrompus). |

|  |  |
| --- | --- |
| TopLayer1 | GND |
| TopLayer2 | GND |
| BottomLayer1 | VCC3,3V |
| BottomLayer2 | GND |

|  |  |
| --- | --- |
|  | Concernant le Pcb Touch Capa, ils n’y a pas grand-chose à dire |
|  | Mis à part qu’il ne doit **pas** y avoir de plan de masse sur le PCB |
|  |  |
|  |  |
|  |  |
|  | Concernant le PCB Screen holder c’est le seul à ne pas nécessiter de modifications |

|  |  |
| --- | --- |
| Rendus 3D |  |
| Figure 20 Face avant | Figure 21 face arrière |

#### Points critiques ou particuliers

|  |  |
| --- | --- |
| Une image contenant texte, diagramme, capture d’écran, nombre  Le contenu généré par l’IA peut être incorrect.  Figure 22 Contexte Registres à décalage | Une image contenant capture d’écran, Caractère coloré, ligne, motif  Le contenu généré par l’IA peut être incorrect.  Figure 23 Routage Registres à décalage |
| Lors du routage des registres à décalage pour les leds le choix de passer en PCB 4 couches à été fait. Comme le montre la figure XX la fenêtre laissée pour faire passer le module 4In/2Out d’Esser et les connecteurs laisse peu d’espace malgré un PCB 4 couches il est difficile de laisser passer le plan de GND entre chaque piste. | |

|  |  |
| --- | --- |
| Une image contenant capture d’écran, Caractère coloré, motif, art  Le contenu généré par l’IA peut être incorrect.  Figure 24 routage paire diff | Une image contenant texte, capture d’écran, Police, logiciel  Le contenu généré par l’IA peut être incorrect.  Figure 25 profils d'impédance |
| Les profils d’impédances du projet ont été paramétré, un pour 50 et un pour 90 , sur la figure XX le connecteur USB à ét router avec une imdépance caractéristique de 90 . | |

|  |
| --- |
| Une image contenant Caractère coloré, capture d’écran, circuit, art  Le contenu généré par l’IA peut être incorrect.  Figure 26 Marge restante pour éspacer les composant ~1cm |
| La facilité de routage apporter par le placement par fonctionnalité, rend le PCB très peuplé sur certains endroits, mais on peut voir qu’il reste de la marge pour laisser respirer tout ce monde. |

|  |
| --- |
| Une image contenant capture d’écran, Logiciel multimédia, Logiciel de graphisme, Caractère coloré  Le contenu généré par l’IA peut être incorrect.  Figure 27 Utilisation des guides |
| L’utilisation des guides à été découverte un peu tard, mais elle a permis de snapper dessus pour aligner les vias. |

|  |  |
| --- | --- |
| Figure 28 Piste d'alimentation | Figure 29 propriété de la piste |
| Les pistes d’alimentations ont été adaptée pour supporter jusqu’à ~3A | |

Étant à couts de temps et en manque de place, en accord avec M. Moreno, la sérigraphie à été retirée. Cependant, concernant le manque de point de test malgré les demandes à plusieurs reprises, est une grave erreur de ma part et qui m’a porté préjudice pour tout le reste du projet autant pour la mise en service que pour valider le fonctionnement et d’autant plus pour le justifier durant la rédaction du rapport.

### Montage/ Mise en service

Avant toute choses les tests mécaniques ont été la priorité :

* Montage des « papillons » et test si la carte Esser fit
* Test boitier (si le PCB rentre)

J’ai voulu adopter une stratégie, qui consiste simplement à monter certains composant par fonctionnalité pour les tester au fur et à mesure, de cette manière la rédaction de ce rapport, la mise en service, les tests mesures ainsi que le software auraient pu évoluer avec l’intégration progressive des différentes parties. Pour entrer dans les conventions de ce rapport je vais donc « séparer » le montage mise en service et résultats.

J’ai d’abords commencer à monter le MCu avec une plaque chauffante, accompagné du nécessaire pour le faire fonctionner, oscillateur condensateurs etc.

|  |  |
| --- | --- |
| Figure 30 Microcontrolleur condensateur et oscillateur | Figure 31Switch de reset et connecteur de programmation |

Ensuite, que ça serve de motivation la stratégie à été revue pour commencer par les fonctionnalités les plus simples. J’ai décidé de monter le buzzer ce qui était un des principaux composant pouvant générer le plus de bruits sur la carte. Comme le montre le datasheet (p.3) une des plus forts niveaux sonores se trouve aux alentours des 4Khz, qui correspondrais environ à 76dB

|  |  |
| --- | --- |
| Une image contenant ligne, Tracé, diagramme, texte  Le contenu généré par l’IA peut être incorrect.  Figure 32 caractéristique du buzzer | Le faire à ce moment permet de tester les sources de parasites prioritaires |

Ensuite les configs switchs

|  |  |
| --- | --- |
| Figure 33 switch de configuration | Permet de valider la lecture des entrées et l’utilisation pour la suite des tests de fonctionnalités |

Après viennent les TVS, et les interrupteurs de configuration pour les capteurs analogiques.

|  |  |
| --- | --- |
| Figure 34 les TVS | Permet de tester les entrée analogiques |

La RTC

|  |  |
| --- | --- |
|  | Permet de tester la communication vias I2C |

Les relays et les Ic’s de commande de ces derniers

|  |  |
| --- | --- |
|  | Permet de tester les relais et la commande |

Connecteur USB et interface FTD230

|  |  |
| --- | --- |
|  | L’objectif de le monter à ce moment était que durant le dev de l’app de test, la communication série aurais pu être utilisée pour le debug « en live » |

Pour finir par la carte SD, et les connecteur carte à carte et carte à fil

L’ordre est important car il permet de monter la face principale au complet pour ensuite passer au bottom. Ce qui j’imagine serais l’idéal pour un PCB tand chargé, monter le plus possible au four et finir par les composant traversant et les composant au bottom à la fin. Les connecteurs de sortie ne sont pas montés et l’alimentation se fait vie alimentation de laboratoire lors de tous les tests. Le montage a été fait principalement à la plaque chauffante malheureusement le nombre de retouche rend le résultat désagréable à première vue.

Les connecteurs de bordure de carte ne sont pas montés car dans le cas ou des modification aurait dû être fait ils auraient gêné le passage lors du rework

### Mise en service

Pour un peu de contexte deux étapes principales peuvent décomposer la première mise en service

* Step1 : valider fonctionnement du MCU et alimentations
* Step2 : ajouter des fonctionnalités une à une

|  |  |
| --- | --- |
| STEP1 : Pour tester le fonctionnement Microcontroleur l’utilisation d’une alimentation de laboratoire réglée à 3,3VDC max 50mA directement sur P1.2 (voir ci-dessuos) puis essayer de rentrer simplement en debug. | |
| Une image contenant texte, diagramme, capture d’écran, ligne  Le contenu généré par l’IA peut être incorrect.  Figure 35 alimentation | Une image contenant capture d’écran, motif, diagramme, Caractère coloré  Le contenu généré par l’IA peut être incorrect.  Figure 36 P1 pin alimentation |
| Le deuxième faire commuter la sortie qui pilote la LED verte (nommée lifeled) nécessite une validation visuelle | |
| Figure 37 Connectetur d'entrée et protecttion | |
| Ensuite c’est l’alimentation qui à été testée toujours à l’aide de l’alimentation de laboratoire cette fois -ci en +24VDC directement sur en amont du fusible F1.1 figure XXX. Tout d’abord seule (jumper P1 non monté), puis avec P1 monté, une mesure de niveau en avec l’oscilloscope en DC et une pour la propreté du signal en AC. | |
|  | |
| Figure 38Alim 3v3 | Figure 39Alim 3v3 AC |
| Dans la mesure figure 38 on vois bien du 3,3VDC | Et grâce au mesure en ac, on retrouve 1khz d’ondulation pour une amplitude de moins de 20mV |
| STEP2 : La première fonctionnalité tester est le buzzer, ce qui permis de valider le fonctionnement d’un timer une validation sonore suffit pour valider le fonctionnement. Mais aussi de valider l’état de l’alimentation à gauche lorsque que l’alimentation de labo est connectée directement sur le jumper P1 en +3V3, voir figure XX à droite lorsque l’alimentation de laboratoire alimente le PCB au travers de l’alimentation en +24V. | |
| Figure 40 Alim labo buzz | Figure 41 aLim trace buzz |
| On peut voir à gauche sans l’alimentation que par-dessus les pic due au comutation du MOS/Buzzer un autre signal surement directement le 50Hz mais au vue de l’amplitude ~50mV.. | En passant par l’alimentation « Traco » on vois déjà une amélioration dans la qualité de l’alimentation. Avec toujour les résidus de commutation. |
| La seconde, les entrée digitales (les switchs de configuration servant à la redirection des entrées sur la sortie de dérangement) pour validation du fonctionnement ils ont servi à activer ou désactiver le buzzer | |
| Ensuite les entrée analogiques destinée aux capteurs analogiques et détection d’interruption de ligne pour valider la lecture de l’adc une tension à été apliquée avec l’alimentation de laboratoire en +1V (masse commune) puis +2V puis 10V pour vérifier l’utilisation normale. Le calcul utiliser pour valider  (sur une entrée des capteur 🡪 1/3 de la tension au connecteur) | |
| Les registre à décalages pour piloter les leds, dans l’idée de pouvoir commencer construire la logique | |
| Les commandes des relais et relais pour valider que la reproduction des entrées pourra être faite une fois la logique construite | |
| La RTC pour le « event logger », permettant ainsi de tester la première communication d’un bus. | |
| « Dans le dérouler réel des évènements c’est après être resté bloquer sur la configuration de la RTC que j’ai été aider par M. Castoldi (mauvaise compréhension du datasheet), après que la RTC fonctionne enfin et compte, voulant mettre me mettre un coup de stress pour aller monter le connecteur de carte SD, le PCB a cédé à la pression et n’a pas supporter de recevoir les 24V d’alimentation directement sur une diode TVS… A partir de là le cauchemar ne faisait que commencer. Des courts-circuits entre les alims, pour déterminer d’où venais le problème, j’ai retiré composant par composant jusqu’à ne plus rien avoir mis à part le microcontrôleur. Après l’avoir changé, j’ai recommencé la même procédure. » | |
| Normalement pour suivre la logique la détection des touches capacitives aurait dû être la fonctionnalité suivante, mais c’est finalement l’écriture / lecture de la carte SD qui à été la fonctionnalité suivante. | |
| C’était donc le tour de la communication sérial | |
| Ensuite les Touches capacitives ont été testée | |
| L’écran est la dernière fonctionnalité tester au moment de la rédaction | |
| //test solution pour uart | |
| // test BT | |

Les // signalent que ces fonctionnalités n’ont pas été implémentée ni testée

### Programmation logicielle

|  |  |
| --- | --- |
| Une première app à été faite, elle a permis d’ajouter fonctionnalités les unes après les autres, cependant après un certain temps l’encombrement à rendus le tout difficile à manipuler. Et est venue l’idée de faire de simples variables globales testée avant chaque chose pour savoir si l’exécution devais se faire. Et donc permettre de tester uniquement l’ADC, ou la lecture des entrées. Pour ensuite avoir à changer un flag pour additionner une fonctionnalité à tester ou pas. Jusqu’au moment de tester la communication serial via l’USB-C monté sur la carte, pour tester j’ai voulu ajouter des logs de débug qui aurais permis de ne pas placer de point d’arrêt de Watch et avoir une autre méthode pour débugger les fonctionnalités à venir. | |
| Description de l’app « 2414-DES\_SP7\_TestSprinkler » et liens avec la méthode de mise en service : | |
| * Création d’un projet basic Mplab * Configuration Harmony * Ajout d’un état dans la machine d’état de l’app correspondant ex :APP\_STATE\_READ\_ADC * Valide le fonctionnement avant de monter la suite | |
| Des flow charts sont disponibles dans : « C:\microchip\harmony\v2\_06\apps\MINF\Projet\2414-DES\_SP7\_TestSprinkler \Flowchart »  Une représentation visuelle des appels de fonction est disponible avec une donc en format html dans « C:\microchip\harmony\v2\_06\apps\MINF\Projet\2414-DES\_SP7\_TestSprinkler\html »  Et une save des config utilisée « C:\microchip\harmony\v2\_06\apps\MINF\Projet\2414-DES\_SP7\_TestSprinkler\firmware\src\system\_config\default\backup.mhc » | |
| Après tous ces échec la décision de remettre de l’ordre s’est imposée, j’ai donc déconstruit en différentes application chaque fonctionnalité par exemple pour la lecture des touches capacitives, une application dédiée en mode state machine, de même pour l’écrans. Après ces deux applications fonctionnelles était venus le temps de les merge. Et ainsi de suite vint le projet SPB7\_Assembly | |
| La méthodologie utilisée « mergeApp » :   * Création de l’app dédié * Refractor des noms pour éviter les conflits d’app * Merge (copier les fichier sources) * Appel de appXXX dans SYS Tak et SYS INIT (dans main.c) * Ajout de la gestion du mode « sale » (signifiant un changement à traiter) (taskctrl) * Implémenter la tache dans les autres app * Ajout de la gestion EventBus (permet la transmission de donnée entre les applications) si nécessaire | |
| Le projet 2414-DES\_SP7\_TestSprinkler\_bootlaoder.X, est un test non concluant de bootloader, sans communication série compliqué… | |
| Le projet SPB7\_ADC\_IN est l’application dédiée à la lecture des entrées de configuration et de l’adc pour les config de l’ADC voir XXXX | |
| Le projet SPB7\_MCP est une tentative de récriture du drivers MCP\_SM (pour la rtc) utilisant la librairie « non bloquante » trouvées dans le support de cours. Actuellement une tentative de séquence a été faite mais je bloque au niveau du passage des donnée d’init. Précédemment faite avec un pointeur sur un buffer dans une boucle for, il faudrait faire en sorte de préparer le buffer, peut être en dur, et utiliser un compteur -). L’objectif de base de cette app était dans le cas où la lecture des différents composant sur le bus I2C aurais poser un problème. Mais après est venue l’idée de l’app taskctrl et eventbus. | |
| Le projet SPB7\_Screen est l’application dédiée à l’écran, il n’y a pas grand-chose sitôt fonctionnelle elle à été merge à SPB7\_TouchCapa, pour crée SPB7\_Assembly | |
| Le projet SPB7\_SREG\_LED contient une tentative de driver faite en mode state machine | |
| L’état actuel certaines app sont ne sont pas terminés : - SPB7\_MCP  D’autre fonctionnent déjà ensemble, mais ne sont pas terminées :   * App\_eventbus.c //gestion des données entre apps * app\_taskctrl.c //gestion des taches * appDisp.c //gestion de l’écran * appReg.c // gestion du registre à décalage des leds * appTouch.c // gestion des touches capacitives   Sur le commit le plus de la branche « VS » actuellement appInput(le projet SPB7\_ADC\_IN ) est entrains d’être implémentée  Avant, il faudrait implémenter un descripteur pour les touches capacitives dans l’assembly car la gestion des touches est mal traitée.  Après avoir implémenter cette app il reste toujours en utilisant la méthodologie « mergeApp » à crée une aplication dédiée pour la RTC ceci impliquant un choix à faire soit finir de réécrire le driver gestMCP\_sm et ensuite de même pour le touchCapa qu’il faudra modifier dans l’assembly ou tenter le tout pour le tout avec le driver bloquant et la gestion de tache.  Ensuite crée une application dédiée pour le buzzer  Petit problème se pose c’est qu’actuellement le 5ème timer serai utilisé pour la gestion du buzzer  N’en laissant plus aucun dans ces cas envisager core timer ou réutiliser et réadapter APP\_SetTMR0\_Frequency.  Crée une app dédiée pour gestion Esser board surement nécessitant un timer de plus  Ensuite une app dédiée BT surement nécessitant un timer de plus  Tenter le bootloader | |
| App\_eventbus : | |
| Figure 42 Flow-chart eventbus app | |
| Le but de cette app est simplement d’être appelé par les événements intervenants nécessitant des actions à effectuer. Il suffit d’y ajouter une librairie d’appeler/crée la fonction de handle dans la machine d’état. | |
| Déclarer en externe ce sont des variables globales initialisée active mais pas occupée, l’objectif est que dans le service task de chaque app un test du flag isActive soit fait avant de pouvoir exécuter le reste. Dans le cas où la tâche est active elle s’exécute et dans le cas ou elle prend tu temp elle le signal avec le flag isDirty | |
| Pour ne pas faire le détail de toute les app le principe est qu’elle soit faire en state machine non bloquante (sauf appTouch.c) | |
| Configuration actuelle dans harmony  : | |
| Figure 43 Pinning Microcontrolleur | |
|  | L’ADC est configurer en mode SCAN de manière à avoir une configuration rapide qui valide simplement la lecture. Lire les 14 entrée analogique d’un coup. De plus le mode auto sample étant activer le temp nécessaire à la lecture des 14 échantillons. 1 echantillon est mesurer et convertis-en 12 TAD clk, De plus l’auto sample est activée. |
|  | Le spi est générer comme demander par le driver de LCD avec une fréquence de 100kBauds, le rafraîchissement de l’écran dure environs 7ms |
|  |  |
|  | L’utilisation des timer est décrite plus bas, le timer un est réglé avec une période de 10ms  Le timer 2 à 50ms est utiliser pour mettre l’appToucCapa en état actif permettant la détection d’un appui assé fréquemment  Le timer 3 est réglé avec une période de 50ms lui aussi mais il est utilisé sans interruption car il cadence les conversions de l’adc  Le timer 4 est réglé à une fréquence de 10ms pour la gestion du registre à décalage  Le timer 5 est réglé lui aussi à une fréquence de 10ms et est utiliser pour le LCD. |
| Figure 44 config Harmony |  |
|  |  |
|  |  |
| Et une save des config utilisée « « C:\microchip\harmony\v2\_06\apps\MINF\Projet\SPB7\_Assembly\firmware\src\system\_config\default\backup.mhc » | |
| Une représentation visuelle des appels de fonction est disponible avec une donc en format html dans « C:\microchip\harmony\v2\_06\apps\MINF\Projet\SPB7\_Assembly\firmware\src\html\index.html » | |
| Après une discussion avec M.Castoldi, la méthodologie utiliser n’est pas bonne, le conseil et de repartir d’un projet vierge et ajouter les différentes app directement dans Harmony, avant d’implémenter quoi que ce soit de nouveau.  Une fois le nouveau projet crée une des principaux choses à implémenter est un descripteur pour les touches de tel manière à ce que la gestion soit simplifiée. | |
| Exemple de la documentation générée | |

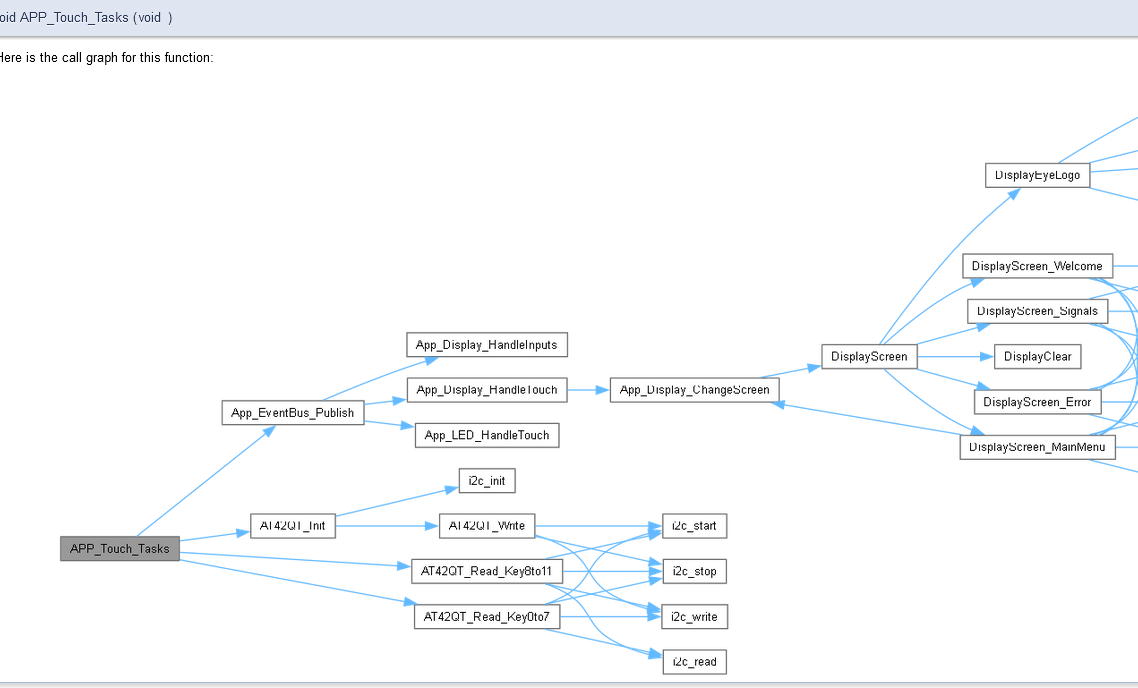


Figure 45 exemple de la doc disponible

### Tests et mesures :

|  |  |
| --- | --- |
| Les tests importants que j’aurais aimé faire est la consommation maximum quand tout consomme c’est-à-dire tous les relais toute les leds etc mais actuellement le soft, et le hardware ne le permet pas. | |
| Actuellement la lecture de tension est validée sur les 14 entrée mais aucune logique n’a été testée avec, mis à part la lecture des capteur 4/20mA et 0-10V | |
| De plus, au niveau des protections (inversion sur l’alimentation et fuse) n’ont pas été testée. | |
| Mesure SPI LCD |  |
| Figure 46Communication avec LCD | Figure 47 clocks du registre à décalage |
| Communication SPI avec le LCD, on remarque le bruit sur les signaux | Résultat du driver pour les registres à décalage, les signaux mesurer sont les clocks. |
| Figure 48 MCP configuration | Figure 49 MCP configuration |
| RTC init\_1, on peux voir que le mcp79411 renvois un ACK | RTC init\_2, on peux voir que le mcp79411 renvois un ACK |
| Figure 50 mcp lecture des registres de temps | Figure 51 datasheet mcp page 27  En réalité une erreur d’interprétation de ma part, de ce que j’avais compris bit 3 à 1 pour utiliser un oscillateur extern, mais non. Après quoi la lecture était fonctionellle. |
| Lors de la demande de lecture des resgistres d’heure la RTC ne renvoyais rien |  |

|  |  |
| --- | --- |
|  |  |
| Mesure SD cartd |  |
| Figure 52 les 80 coup de clock pour passer en mode SPI | Figure 53 aucune réponse de la SD |
| Malgré la lecture du document d’application note pour la communication avec une carte SD (Implementing File I/O Functions Using Microchip’s Memory Disk Drive File System Library) | Je n’ai pas réussi à comprendre pourquoi la lecture ne s’effectuai pas correctement. |
| Une mesure pour vérifier si la capa utilisée pour maintenir en vie la RTC à été faite. En plaçant une sonde directement sur la capa et en éteignant l’alimentation. Le niveau passe directement à 0v ce qui après coup semble « logique » au vue des consommateurs router sur la même piste (les condensateur de maintiens sont mis en évidence) | Figure 54 RTC |

### Modifications hardware :

#### Raisons

|  |  |
| --- | --- |
| Buzzer | Composant actuellement brasé à 45°, besoins de correction du **footprint.** |
| Figure 55 buzzer monter à 45 | Raison : Pas/Pas assé de vérification/rigueur lors du design |
| Registre a décalage    Figure 56 Modif CLK et CLEAR | Router à la main, modification du **symbole** nécessaire, et router les pistes manquante /SRCLK : la clock pour le shifting. SRCLR : le clear du registre et Qh’ pour la mise en série des 2 registres  Raison : Pas/Pas assé de vérification/rigueur lors du design |
| Après modifications complètement terminée :    Figure 57 Détail modifsSreg |  |
| Relais et tableau de resistance | **Positionnement** relais – SD besoin de correction malgré la hauteur des relais, faire passer le réseau de résistance ailleurs… de même pour le connecteur de carte SD, elle dépasse du socle…  Raison : Pas/Pas assez de vérification/rigueur lors du design |
| Figure 58 eurreur d eplacement |  |
|  |  |
| Connectors mirroring | Modification **schéma** connecteur/changer la liste de pièce pour le câble plat avec les contact opposés |
|  | Raisons : les connecteurs souples à l’ES ont les contacts du même cotés. Et pas/pas assez de vérification/rigueur lors du design |
| Transistor de puissance d’entrée « anti-inversion de polarité | Le footprint SOT-89 atypique n’est pas forcément disponible chez les revendeurs préférentiels de L’ETML de plus pas forcément des plus simples à monter à la main |
| Relais alimentations | **Schéma** Les relais choisit sont polarisés… |
| Figure 59 Modifs pour relais | L’alimentation de l’inductance étant polarisée il faudrait faire la modification sur tous les relais, ce que je n’ai pas fait. La modification d’un seul permet la validation du fonctionnement. |
| Connecteur USB-C | **Schéma** le connecteur choisit est un connecteur uniquement pour l’alimentation |
|  |  |
| Registre à décalage | Au lieu de rooter n’importe comment utiliser l’entrée utilisée normalement à cet effet pin9 QH’. Modification **Schéma** |
|  |  |
| PCB TouchCapa |  |
| Figure 60 Modification touchCapa | **Schéma** Les lignes SDA et SCL sont inversée entre les PCB |
| Figure 61 Pin configuration | Raison : éviter de devoir faire une inversion bit à bit sur la moitié de la valeur de commande du registre à décalage. |
| Capa pour la RTC | Logique fausse revoir le **schéma** |
|  |  |
| Transitor array |  |
| Figure 62 Invesion | **Schéma.** La logique de base était juste mais lors de la revue de schéma avec M.Moreno nous avons fait une modification car il semblait que la « COM » (sur le symbole) devais être connectée à la masse d’après lui |

### Intégration mécanique

|  |  |
| --- | --- |
| Figure 63 boitier avec PCB's dedans | La sensibilité des touches capacitives a été adaptée pour qu’un appui soit détecté au travers du boitier. Pour se faire, la valeur du seuil de détection a été divisée par 2, puis après un nouvel essai, si les touches n’étaient toujours pas détectées, la valeur de threshold était à nouveau divisée par 2 |
| Figure 64 assemblage des pcbs | L’assemblage les 3 pcb entre eux pourrait être ajuster, surtout au niveau hauteur écrans et touches capacitives. Pour que les deux pcb soient collées au couvercle du boitier  Détails entretoises :   |  |  | | --- | --- | | To (PCB) |  | | Screen\_Holder | 5mm brasée | | Touch capa | 14mm brasée | | Boitier | 38mm visée | |
|  | Les presses étoupes (non commandées) et autre modification du boitier n’ont pas été faites |

### Problèmes rencontrés restant sans solutions

|  |  |
| --- | --- |
| Le buzzer pas suffisamment fort | |
| Figure 65 Pinning USB-C | Figure 66 pinning du conneteur choisit |
| Concernant le connecteur USB-C et la communication sériel, elle est impossible dans l’état le choix du connecteur USB est faux come montrer sur le pinning en page 3 de la datasheet, les signaux à disposition sont « CC1 » et « CC2 » et comme décris dans la description de pins ce sont des pins de configuration et non de données. | |
|  | |
| Carte Esser 4in /2out | |
| J’ai tenté d’alimenté la carte fournie comme mentionner dans la doc (cahier des charges dossier rouge) mais aucun voyant ne s’allume clairement, l’intégration de cette dernière n’a jamais été tester. De plus la gestion de la terre a été oubliée lors du design. | |
|  | |
| La lecture de la carte SD | |
| Malgré plusieurs tentatives d’utilisation de la carte SD pour écriture s’est soldée par un échec, L’erreur se trouve au niveau du Système de fichier (d’après le débug) le lecteur n’est jamais monté malgré le formatage en FAT 16 et FAT 32 avec des unités d’allocations différentes rien n’y fait. Peut-être un problème lié au hardware à investiguer | |
|  | |
| Modification du boitier, n’ayant pas choisi de presse étoupes lors du design, j’ai tenté de réutiliser ceux du projet transmis, malheureusement l’ES ne dispose pas de foret de diamètre adéquat pour percer le boitier. Un trou à été fait pour laisser passer l’alimentation lors des tests d’intégration. | |
| La led de test placé au bottom est censée être verte et pas rouge. | |

## Analyse de durabilité

Plusieurs fois j’ai laissé le montage alimenter intentionnellement pour voir si quelque chose allait se passer mais non

### Aspects environnementaux

Actuellement la consommation tourne aux alentours des 300mA, malgré le choix de led 5mA.

Le choix des relais n’est pas le plus optimal car ils ont besoin d’être alimenter lors du changement d’état et de maintenait l’alimentation. Il existe d’autre type de relais qui serais plus judicieux d’utiliser.

### Aspects sociaux

Les erreurs sont un bon sujet de moquerie, plus sérieusement socialement à part améliorer les relations entre employé et employeur, dans l’idée de simplifier l’installation avec la modification du type de connecteur

### Aspects économiques

J’ai tenté de régénérer les BOMS de tous les PCB pour faire une comparaison avec les prévisions des couts, malheureusement, l’assignations des suppliers/fornisseur à disparue surement due à la copie du projet lors de l’aide apportée par M. Moreno, elle sera refaite d’ici la présentation pour donner une comparaison et les couts réel pendant cette dernière.

Les choix des composant se sont principalement concentrer vers des solutions Würth pour éviter de perdre du temps sur la recherche du prix le plus bas / fournisseur meilleur marché. Pour l’avenir le choix de composant moins cher peut toujours être fait avec un travail de réduction des coups

## Conclusion

### Bilan global

La structure et la rigueur étaient principalement absente, ce qui amène à un projet pas terminer avec encore du travail, autant niveau hardware que soft, il aurait fallu réfléchir avant à comment construire le programme. Car l’utilisation des différents timer est mal faite et demande à être reréfléchies car dans la première application à chaque nouvelle fonctionnalité un timer lui était « dédier », le fait d’avoir gardé cette structure pour l’app « assembly » était une erreur.

La logique aurait voulu qu’une fois le hardware développer tester et fonctionnel l’implémentation du module 4in/2out aurais dû être faite, actuellement mis à part l’encombrement rien n’a été tester.

La méthodologie utilisée de monter programmer tester mesurer dégguger aurait été un atout si j’avais à l’avance établis une structure de logiciel claire et méthodique. Aux lieux de me faire gagner du temps j’ai eu l’occasion de rester bloquer 4 jours sur une RTC pour un souci de compréhension de datasheet. Des gardes fous lors du design plus rigoureux aurait mis en valeur la méthode utilisée.

Après le feu d’artifice et remonter le microcontrôleur est ce qu’un excès de sécurité ne m’as pas fait trop limiter le courant et penser que des cout circuit était toujours présents ce qui m’as amené à tout redébraser et commande pour rien ? …

Malheureusement après la restructuration du projet pour le «Assembly », l’état actuel est qu’il reste à implémenter appInput, et y implémenter le taskcontrol et eventBus, ainsi qu’implémenter les fonction de handling dans les app nécessaires. je n’ai pas eu le temps d’implémenter la détection d’un changement sur les valeurs d’ADC, totalement ni d’implémenter totalement le handling des entrée de configuration, mais une version commencée existe le repo <https://github.com/Sweedy3960/MINF> dans la branche « VS » commit [73a5ee6](https://github.com/Sweedy3960/MINF/commit/73a5ee6e5d22e55c2fc72c96a135159146b618c2).

Une version utilisée pour la rédaction du rapport et présentation sur le même dépôt dans la branche « RapportVersion »

Globalement ce projet m’aura montré à quel point je manque de méthodologie rigueur et structure dans ma manière de travailler.

Concernant l’USB-C grâce à M. Castoldi au sujet de l’USB-C j’ai pu aussi débloquer un projet perso.

### Compétences acquises

« Git branch revert » Avec toute les applications et version différente est devenus essentiel l’utilisation et l’apprentissage des commande un peu plus avancée de git

« Merging multiple app/ Task programming » Bien que déjà expérimentée en TP, la création d’une app en partant de 0 en essayant de réfléchir machine d’état était une première.

« Footprint paranoïa » dorénavant j’irais vérifier avec plus d’attention les numéross des pates avec les fonctions liées.

### Améliorations possibles / suite du projet

Il faut reclarifier la structure de l’app, réorganiser l’utilisation des timers, faire les corrections schéma et pcb, corriger les logiques fausse comme la capa pour la RTC par exemple, aussi au niveau du code des corrections sont à faire comme la gestion des touches. Mais aussi interfacer le module Essert avec le PCB principal, monter et tester le bluetooth. Tester le bootloader

### Mode d'emploi du système

Pour l’état actuel mettre du +3V3 sur le fil rouge et la masse sur le fil noir si le code est flashé une séquence de bienvenus apparais tapoter les touches capacitives affiche « n’importe quoi » l’état de la variable utilisée pour stocker l’état des touchCapa est utilisée pour l’écrant lcd pour simuler une quittance sur les leds aussi.

## Annexes

### Cahier des charges initial

### Fichiers de CAO (schémas, PCB, implantation)

### Liste des pièces et coûts réels

### Listing des différentes apps(sur github)

Voir <https://github.com/Sweedy3960/2414_SPB7_DOC> et <https://github.com/Sweedy3960/MINF/tree/RapportVersion>

### Planning initial vs réel

### Journal de travail - git commit

Lausanne, le 09.11.2024