

+

**Ecole supérieure**  
ÉlectroniqueETML-ES  
Salle R110**Diplôme - 2407**  
**Commande Ouverture Frigo****Réalisé par :**

Antonio Do Carmo

**A l'attention de:**Philippe Bovey  
Lionel Yersin  
Steve Jacot-Guillarmod**Dates :**Début : 05 Août 2024  
Fin : 9 Septembre 2024

## Table des matières

1	Introduction .....	4
1.1	Contexte .....	4
1.2	But du projet .....	4
1.3	Organisation .....	4
2	Conception .....	5
2.1	Schéma bloc hardware .....	5
2.2	Serrure électromagnétique .....	6
2.3	Wi-Fi .....	6
2.3.1	ESP32-C3-WROOM-02-N4 .....	7
2.3.2	Connecteur de programmation .....	8
2.4	Ethernet .....	8
2.4.1	ENC28J60-I/SO .....	8
2.4.2	CRJ011-ML3-TH .....	9
2.5	RFID .....	9
2.5.1	Chilli UART B1 .....	9
2.6	Affichage .....	10
2.6.1	LCD .....	10
2.6.2	LEDs batterie .....	10
2.6.3	LEDs .....	10
2.7	Alimentation .....	11
2.7.1	Chargeur de batterie .....	11
2.7.2	USB-C .....	12
2.7.3	Batterie .....	13
2.7.4	Lecture batterie .....	13
2.7.5	DC-DC 12V .....	14
2.7.6	DC-DC 3.3V .....	15
2.7.7	PIC32MX130F064D-I/PT .....	15
2.7.8	I/O Expander .....	16
2.7.9	I/O supplémentaires .....	16
3	Hardware .....	17
3.1	Description générale .....	17
3.2	Répartition des blocs .....	17
3.3	Vues des couches du PCB .....	18
3.4	Vues 3D du PCB .....	19
3.5	Détails de conception .....	20
3.5.1	Chargeur de batterie .....	20
3.5.2	ESP32 .....	20
3.6	Largeur des pistes DC-DC 12V .....	20
4	Modifications .....	21
4.1	DC-DC 12V .....	21
5	Boîtier .....	22
6	Software .....	23
6.1	Configuration du clock externe .....	23
6.2	Timer 1 .....	23
6.3	Interruption du Timer 1 .....	24
6.4	Timer 2 .....	24
6.5	OC1 .....	24
6.6	Bus USART 1 .....	25
6.7	Bus USART 2 .....	25
6.8	Marche à suivre configuration ESP32 .....	25
6.9	App_Tasks .....	28
6.10	LedControl .....	29

---

6.11	ReadBattery.....	30
6.12	DoorOpening .....	31
7	Mesures.....	32
7.1	PWM gâche .....	32
7.2	UART RFID .....	32
7.2.1	Activation du polling.....	32
7.2.2	UID .....	32
7.3	I2C I/O expander .....	33
8	Problèmes rencontrés .....	34
8.1	Hardware .....	34
8.1.1	Connecteurs RFID et gâche.....	34
8.1.2	Support d'accumulateurs .....	34
8.2	Fréquence des badges .....	34
8.3	Software .....	34
8.3.1	Lecture de l'UID du badge .....	34
8.3.2	Communication Wi-Fi.....	34
8.3.3	Ouverture de la gâche .....	34
9	État actuel du projet .....	34
9.1.1	Tâches effectuées .....	34
9.1.2	Tâches restantes .....	35
10	Analyse de durabilité.....	35
10.1	Social.....	35
10.2	Économique.....	35
10.3	Environnement.....	35
11	Apports du travail de diplôme.....	35
12	Conclusion.....	36
Annexe A	Cahier des charges.....	37
Annexe B	Planification .....	37
Annexe C	Journal de travail .....	37
Annexe D	Procès-verbaux.....	37
Annexe E	Schémas électriques .....	37
Annexe F	Fichiers de fabrication.....	37
Annexe G	Software .....	37
Annexe H	Schéma de mesure.....	37
Annexe I	Feuille de modifications .....	37
Annexe J	Mode d'emploi .....	37
Annexe K	Boîtier plans d'assemblage .....	37

## 1 Introduction

### 1.1 Contexte

Ce projet est réalisé dans le cadre de mes études à l'ETML-ES au sein de la filière génie électrique, spécialisation électronique. Il s'agit du travail de fin de diplôme permettant de valider les connaissances acquises durant ces deux ans de formation. La durée de celui-ci est de 5 semaines et conduira à l'obtention de mon diplôme de technicien en électronique. Le client et maître de diplôme est M.Philippe Bovey et les experts mandatés par l'école sont M.Lionel Yersin et M.Steve Jacot-Guillarmod.

### 1.2 But du projet

Le but de ce travail de diplôme est de réaliser une commande d'ouverture de frigo, à l'aide d'un badge RFID, qui pilotera une gâche magnétique. Le système devra pouvoir communiquer avec une base de données (non incluse dans ce diplôme), qui informera si le badge lu, correspond à un membre de l'association Fixme ; si oui, l'ouverture du frigo sera autorisée ; pour cela il faudra implémenter une communication soit à l'aide d'un module ESP323 (Wi-Fi), soit à l'aide d'une communication Ethernet.



Figure 1 Illustration du principe de fonctionnement

Le système électronique sera équipé d'un petit affichage permettant de visualiser l'ID du badge RFID, voir par la suite, d'afficher d'autres messages (indication) comme le nom du membre de l'association, autorisation, etc... Un système de leds, sera mis en place pour savoir si le frigo est ouvert / fermé, lecture de la carte RFID (lecture – permission), indication lumineuse si batterie trop faible.

Il sera également nécessaire d'étudier si le système électronique peut être alimenté par accumulateurs (commande de la gâche, communication, affichage, ...), avec la contrainte suivante : que le système devrait être alimenté au minimum pendant une semaine de manière automne (alimentation par batteries). Que cela soit possible ou non, le système électronique devra pouvoir se mettre en veille, et être réveillé par la lecture du badge RFID.

### 1.3 Organisation

Le projet a été planifié en différentes phases et une documentation quotidienne des activités a été réalisée dans le journal de travail.

Une réunion hebdomadaire était organisée avec le maître de diplôme M.Bovey afin d'examiner l'évolution du projet. Un procès-verbal a été dressé à chaque séance (Annexes B et C)

## 2 Conception

### 2.1 Schéma bloc hardware

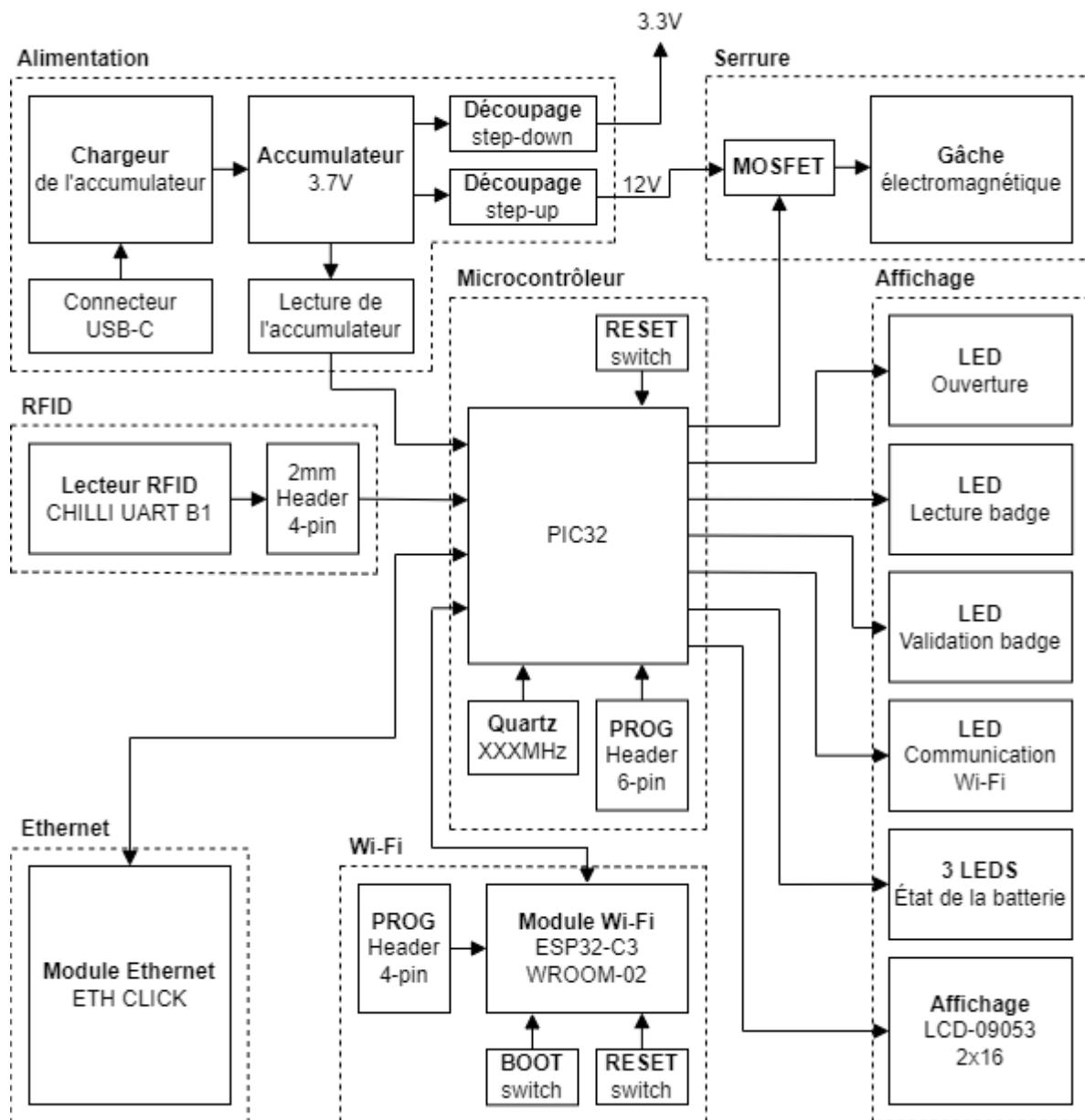


Figure 2 Schéma bloc

## 2.2 Serrure électromagnétique

Pour le choix de la gâche, j'ai tout d'abord commencé par étudier les contacts magnétiques conseillés dans le cahier des charges du travail de diplôme. Cependant je suis vite arrivé à la conclusion que ceux-ci n'étaient pas viables pour deux raisons : leur grande consommation en courant et leur besoin constant d'être alimentés. En effet les contacts magnétiques de ce genre consomment constamment des courants aux alentours de 200 [mA] afin de maintenir un champ magnétique permettant de garder le frigo fermé. Il est donc impossible de les alimenter durant une semaine (comme demandé dans le cahier des charges) avec une batterie Li-Ion classique. L'alimenter via le réseau électrique n'améliorerait pas la situation car il suffirait de débrancher la prise pour accéder au frigo.

$$\text{Durée de vie de la batterie [h]} = \frac{\text{Capacité de la batterie [mAh]}}{\text{Courant dans la charge [mA]}} = \frac{3350}{200} = 16h45$$

Équation 1 Durée de vie de la batterie avec une gâche magnétique

Après plusieurs recherches je suis tombé sur un modèle de gâche nécessitant un pic de courant uniquement lors de l'ouverture de la porte. Cela réglerait donc les problèmes cités précédemment.

Tableau comparatif des gâches		
Nom	Gâche électromécanique	Gâche magnétique
Fabricant	OEM	Assa Abloy
Fournisseur	Ampul	OPO Oeschger
Illustration		
Prix	8.89 chf	36 chf
Taille	80x59x13 mm	55x10x10 mm
Tension d'alimentation	12V	30V
Consommation en courant	1.4A à l'ouverture	250mA constant
Avantages	Consommation à l'ouverture	Petite taille
Inconvénients	Grand courant à l'ouverture	Consommation constante

Tableau 1 Tableau comparatif des gâches

## 2.3 Wi-Fi

Le module Wi-Fi que j'ai sélectionné est le ESP32-C3-WROOM-02-N4 en raison de son utilisation au sein de l'ES pour certains projets. Cela permet de pouvoir bénéficier de l'expérience de mes prédecesseurs concernant ce composant. La documentation présente sur internet est également abondante ce qui me facilitera certainement la tâche. Tous les composants passifs adjacents sont dimensionnés selon le datasheet de celui-ci.

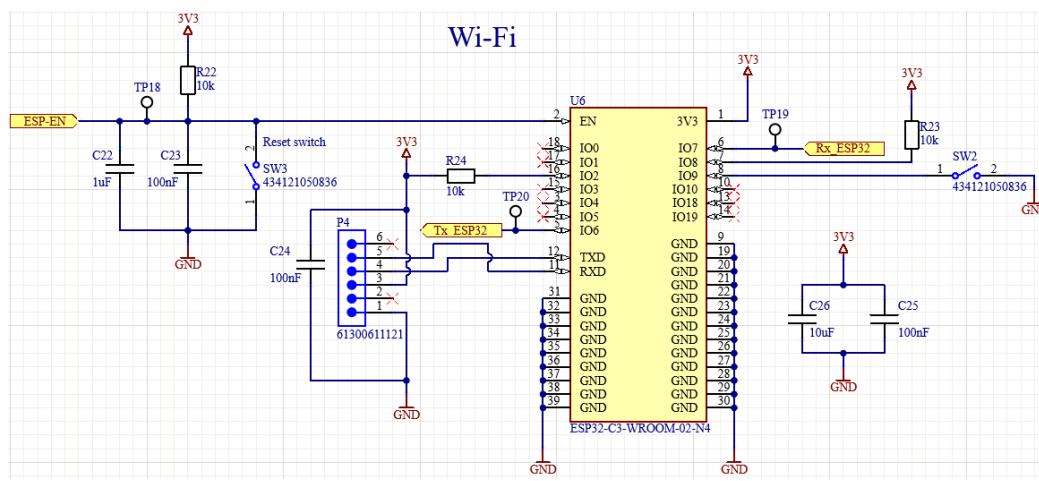


Figure 3 Schéma WiFi

### 2.3.1 ESP32-C3-WROOM-02-N4

Le composant ESP32-C3-WROOM-02-N4 est un module conçu spécialement pour la communication via Wi-Fi. Il supporte une bande passante de 20[MHz] à 40[MHz] dans la bande Wi-Fi 2.4[GHz].

ESP32-C3-WROOM-02-N4	
Fabricant	Espressif
Description	Module Wi-Fi
Tension d'alimentation	3 ~ 3.6 [V]
Courant de réception	82 ~ 84 [mA]
Courant de transmission	280 ~ 345 [mA]
Programmation	Via bus UART



Tableau 2 Caractéristiques ESP32-C3-WROOM-02-N4

#### 2.3.1.1 Pin EN

La pin 2 « EN » est la pin d'enable du composant. Un niveau haut active l'ESP32 tandis qu'un niveau bas le désactive. J'ai donc relié cette pin au microcontrôleur afin de pouvoir gérer l'activation du composant via le firmware pour éviter que celui-ci ne consomme trop de courant lorsqu'il n'est pas utilisé. J'ai également ajouté un bouton de redémarrage inspiré du projet 2312 de l'ETML-ES afin de pouvoir réinitialiser le module manuellement si nécessaire.

#### 2.3.1.2 Pins IO2, IO8 et IO9

Les pins IO2, IO8 et IO9 sont les « strapping pins ». Elles permettent d'activer le « Boot download mode » au démarrage du module afin de pouvoir le programmer. J'ai connecté des résistances de pull-up aux pins IO2 et IO8 afin de les maintenir à l'état haut. La pin IO9 possède une pull-up interne, un switch à la masse permettra donc de la commuter à l'état bas comme indiqué dans le datasheet.

Booting Mode <sup>1</sup>			
Pin	Default	SPI Boot	Download Boot
GPIO2	N/A	1	1
GPIO8	N/A	Don't care	1
GPIO9	Internal weak pull-up	1	0

Figure 4 Booting mode ESP32

#### 2.3.1.3 Pins IO6 et IO7

Les pins IO6 et IO7 sont les pins RX et TX de l'UART1 intégré dans l'ESP32. Ce bus de données communiquera avec le microcontrôleur afin d'envoyer et recevoir des données via Wi-Fi.

Function of Connection	ESP32-C3 Board Pins	Other Device Pins
Download/Log output <sup>1</sup>	<b>UART0</b> <ul style="list-style-type: none"> <li>• GPIO20 (RX)</li> <li>• GPIO21 (TX)</li> </ul>	<b>PC</b> <ul style="list-style-type: none"> <li>• TX</li> <li>• RX</li> </ul>
AT command/response <sup>2</sup>	<b>UART1</b> <ul style="list-style-type: none"> <li>• GPIO6 (RX)</li> <li>• GPIO7 (TX)</li> <li>• GPIO5 (CTS)</li> <li>• GPIO4 (RTS)</li> </ul>	<b>USB to serial converter</b> <ul style="list-style-type: none"> <li>• TX</li> <li>• RX</li> <li>• RTS</li> <li>• CTS</li> </ul>

Figure 5 UART ESP32

#### 2.3.1.4 Pins TXD et RXD

Les pins TXD et RXD sont les pins de programmation du module Wi-Fi. Celui-ci se programme via le protocole UART.

### 2.3.2 Connecteur de programmation

Le câble m'ayant été fourni pour la programmation du module est le [TTL-232R-3V3](#) du fabricant FTDI. J'ai donc ajouté des pins berg mâle avec le pinout correspondant à celui-ci pour la programmation du module.

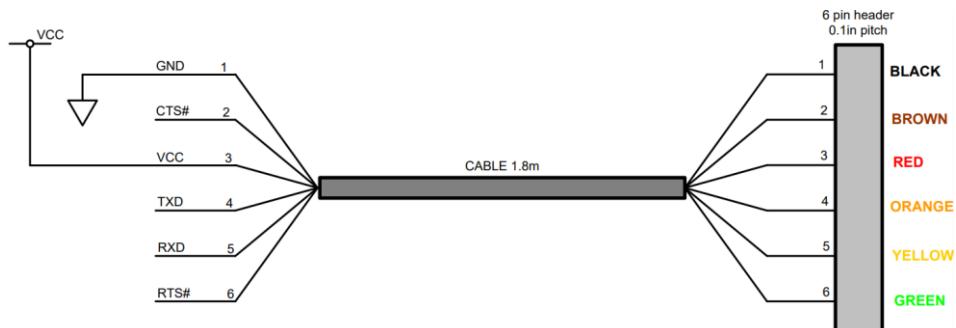


Figure 6 Câblage TTL-232R-3V3

### 2.4 Ethernet

Le bloc Ethernet est inspiré du module [ETH Click](#) de Mikroe. Le [schéma électrique](#) de celui-ci étant disponible sur le site du fabricant, j'ai repris le système permettant la communication Ethernet en supprimant les éléments superflus.

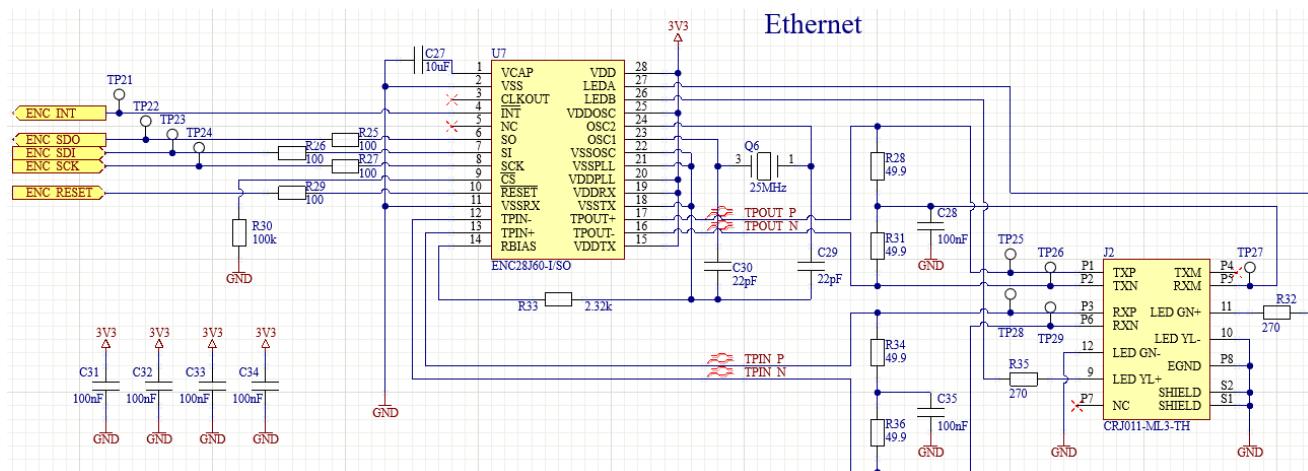


Figure 7 Schéma Ethernet

#### 2.4.1 ENC28J60-I/SO

Le composant ENC28J60 est un contrôleur Ethernet autonome possédant une interface SPI afin de dialoguer avec le microcontrôleur. L'interface possède une vitesse d'horloge pouvant aller jusqu'à 20 [MHz] et le composant intègre un rejet automatique des paquets de données erronés.

ENC28J60-I/SO	
Fabricant	Microchip
Description	Contrôleur Ethernet
Tension d'alimentation	3.1 ~ 3.6 [V]
Courant de fonctionnement	160 ~ 180 [mA]
Courant en stand-by	1.2 ~ 2 [mA]
Protocole	SPI

Tableau 3 Caractéristiques ENC28J60-I/SO

##### 2.4.1.1 Pin INT

La pin INT du contrôleur ENC28J60 est une pin d'interruption indiquant si une trame a été reçue ou non via le port Ethernet. J'ai donc connecté celle-ci directement à une pin I/O du microcontrôleur afin de pouvoir la décoder.

#### 2.4.1.2 Pins SO, SI et SCK

Ces pins correspondent au protocole SPI utilisé par le composant pour communiquer avec le microcontrôleur. J'ai placé des résistances de  $100[\Omega]$  à l'entrée de celles-ci comme conseillé dans le datasheet du fabricant.

#### 2.4.1.3 Pins CS et RESET

La pin CS pour « Chip select » est la pin permettant de sélectionner le slave avec lequel le microcontrôleur souhaite communiquer. Dans ce cas précis, je l'ai relié à une pull-down afin que le CS soit constamment à l'état haut car il n'y aura qu'un slave sur le bus de communication.

La pin reset quant à elle est reliée directement au microcontrôleur afin de pouvoir allumer ou éteindre le composant via le firmware.

#### 2.4.1.4 Pins TPIN-, TPIN+, TPOUT- et TPOUT+

Ces pins sont les paires différentielles par lesquelles transitent les informations brutes envoyées et reçues via l'Ethernet. Le datasheet du composant conseille de faire les pistes des paires différentielles très courtes et avec une impédance de  $100[\Omega]$  afin de garantir l'intégrité des paquets. Des résistances  $49.9[\Omega]$  et des condensateurs  $100[nF]$  sont placés en sortie de ceux-ci selon les spécifications du datasheet.

#### 2.4.1.5 Pin Rbias

Le datasheet indique qu'il faut impérativement connecter une résistance de  $2.32[k\Omega]$  à la pin Rbias car cette résistance influerait directement l'amplitude du signal de sortir TPOUT+/. Elle devra également être placée au plus près possible de ENC28J60.

#### 2.4.1.6 Pins OSC1 et OSC2

Il est dit dans le datasheet que le composant est conçu pour fonctionner à une fréquence de  $25[MHz]$ . J'ai donc placé un quartz de cette valeur aux bornes des pins OSC1 et OSC2 comme indiqué dans le document.

### 2.4.2 CRJ011-ML3-TH

Le port RJ45 que j'ai choisi pour la ligne Ethernet est le CRJ011-ML3-TH. Celui-ci est doté d'un blindage et de deux LEDs jaune et vert. Il est conçu selon la norme Ethernet 10/100BASE-T ce qui correspond également à la norme du contrôleur Ethernet.

CRJ011-ML3-TH	
Fabricant	CUI Devices
Description	Connecteur RJ45
Blindage	Blindé
Couleur LED	Jaune - Vert
Ethernet	10/100BASE-T



Tableau 4 Caractéristiques CRJ011-ML3-TH

## 2.5 RFID

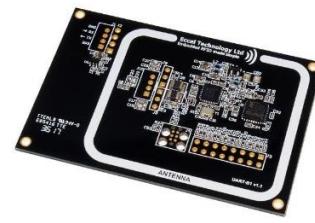
Pour le bloc RFID, j'ai commencé par effectuer des tests sur la carte Fixme fournie par M.Bovey à l'aide du logiciel Tag Reader et d'un des lecteurs RFID de l'ES. J'en ai conclu que la fréquence de fonctionnement de celle-ci est de  $13.56 [MHz]$ . Le module RFID a donc été choisi en conséquence.

#### 2.5.1 Chilli UART B1

Mon choix s'est porté sur le module CHILLI\_UART\_B1. Ce module possède l'avantage d'être facile à placer sur le couvercle du boîtier, ce qui facilitera grandement les choses. Il a également déjà été utilisé pour un projet au sein de l'ES je devrais donc facilement trouver des librairies.

<u>Chilli UART B1</u>	
Fabricant	Eccel Technology Limited
Description	Module RFID
Tension d'alimentation	3.1 ~ 3.8 [V]
Courant à la lecture	120 [mA]
Fréquence	13.56 [MHz]
Protocole	UART

Tableau 5 Caractéristiques Chilli UART B1



## 2.6 Affichage

Le cahier des charges m'impose les indications visuelles à placer sur mon système. La première contrainte est un écran LCD permettant d'afficher le nom de la personne possédant le badge et potentiellement d'autres informations. La deuxième est de placer des leds indiquant l'ouverture du frigo, la lecture du badge, l'état de la batterie, etc...

### 2.6.1 LCD

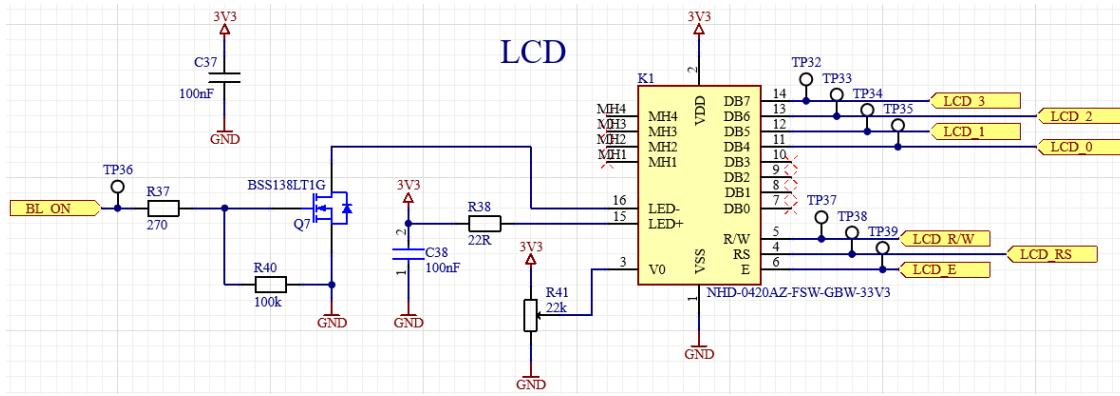


Figure 8 Schéma LCD

J'ai donc sélectionné le [NHD-0420AZ-FSW-GBW-33V3](#) en raison de sa compatibilité avec les librairies de l'ES et également car il a déjà été utilisé dans le cadre de précédents projets.

### 2.6.2 LEDs batterie

Lors d'une séance avec M.Bovey il a été décidé que l'indication lumineuse de l'état de la batterie devrait être une LED RGB s'allumant en rouge, orange et vert en fonction du niveau de charge. J'ai donc sélectionné la LED XZCMEDGCB110W car elle possède l'avantage d'être une LED à très basse consommation (2[mA]), ce qui évitera de décharger la batterie trop rapidement. Un système avec des MOSFET a également été mis en place afin de commander les LEDs.

### 2.6.3 LEDs

J'ai sélectionné une LED RGB pour l'affichage de la batterie, une LED RG pour la validation du badge et trois LEDs rouges pour les indications du Wi-Fi, de la lecture du badge et de l'ouverture de la porte.

$$R = \frac{3.3 [V] - V_{LED} [V]}{I_{LED} [A]}$$

Équation 2 Calcul des résistances de l'affichage

Référence	<a href="#">XZCMEDGCB110W</a>			<a href="#">150080SG54050</a>		<a href="#">APTD2012LSURCK</a>
Couleur	Rouge	Vert	Bleu	Rouge	Vert	Rouge
V <sub>LED</sub> [V]	1.8	2.65	2.65	2	3.3	1.75
I <sub>LED</sub> [mA]	0.01	0.05	0.05	20	20	2
R [Ω]	680	330	330	56	10	820

Tableau 6 Calcul des résistances de l'affichage

## 2.7 Alimentation

Le système doit dans l'idéal pouvoir être entièrement alimenté par un accumulateur et les tensions subsidiaires au montage doivent être réalisées pas des alimentations à découpage DC-DC. J'ai donc sélectionné deux batteries 3400[mAh] possédant une protection contre les surtensions afin de fournir une tension de 3.7 [V] au circuit. Le reste du système sera conçu de sorte que cela soit suffisant pour maintenir l'appareil sous tension durant au moins une semaine.

Le bloc d'alimentation se composera donc de cinq parties distinctes ; l'accumulateur, son système de recharge, un connecteur USB-C, le système de lecture du niveau de batterie et un step-down.

### 2.7.1 Chargeur de batterie

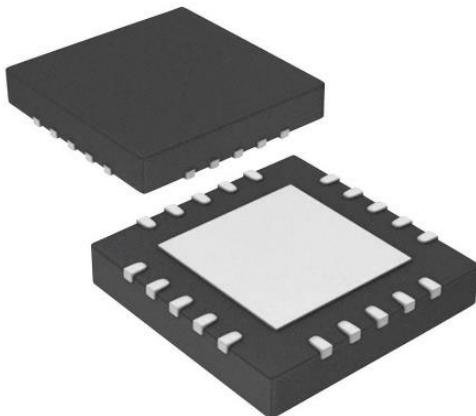


Figure 9 MCP73871T-2CCI/ML

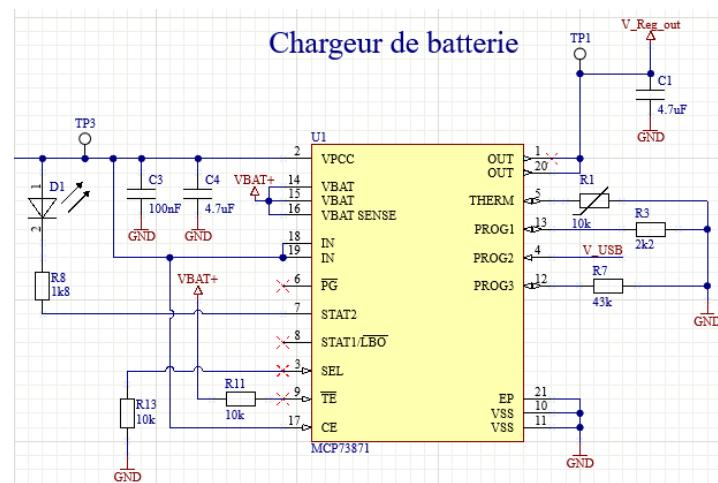


Figure 10 Schéma chargeur de batterie

Le composant choisi pour cette application est le [MCP73871T-2CCI/ML](#). Il s'agit d'un chargeur dédié permettant une recharge autonome des accumulateurs Lithium-Ion. Il possède également l'avantage d'avoir déjà été utilisé dans le cadre des projets au sein de l'ES. Le datasheet précise qu'il est nécessaire de dimensionner les résistances  $R_{PROG1}$  et  $R_{PROG3}$  (toutes les deux des pull-down).

#### 2.7.1.1 Pin PROG1

Selon le datasheet du composant, la pin PROG1 permet de régler le courant de charge constant de l'accumulateur. Il est précisé dans le datasheet que le courant de charge typique est de 450[mA]. J'ai donc calculé la valeur de résistance en me basant sur cette formule :

$$R_{PROG1} = \frac{1000[V]}{I_{reg}[A]} = \frac{1000}{0.45} = 2.22[k\Omega] \rightarrow 2.2[k\Omega]$$

Équation 3 Calcul  $R_{PROG}$

La résistance  $R_{PROG1}$  vaudra donc 2.2[kΩ].

#### 2.7.1.2 Pin PROG2

La pin PROG2 quant à elle permet de déterminer la limite de courant qu'il est possible de tirer sur l'USB-C durant la recharge de la batterie. Le datasheet mentionne qu'il est possible de mettre la pin à un niveau de tension « haut » afin de limiter ce courant à 500[mA]. Je l'ai donc directement relié à la tension présente sur la batterie (V\_USB).

#### 2.7.1.3 Pin PROG3

La dernière pin de programmation PROG3 permet d'établir le seuil de tension à partir duquel l'accumulateur est considéré comme rechargé. Le courant de charge typique est le même qu'au point 2.2.1.1 J'ai donc calculé la valeur de résistance en me basant sur cette formule :

$$I_{TERM} = I_{ChargeTypique}[A] \cdot Ratio_{TERM}[-] = 0.45 \cdot 0.05 = 22,5[mA]$$

Équation 4 Calcul  $I_{TERM}$

$$R_{PROG3} = \frac{1000[V]}{I_{TERM}[A]} = \frac{1000}{22.5 \cdot 10^{-3}} = 44.4[k\Omega] \rightarrow 43[k\Omega]$$

*Équation 5 Calcul R<sub>PROG</sub>*

#### 2.7.1.4 Sortie

Un condensateur de 4.7[nF] a été placé sur les pins de sortie 1 et 20 (OUT), afin de prévenir les potentielles chutes de tension. La valeur du condensateur est conseillée dans le datasheet du MCP à la page 2.

#### 2.7.1.5 Pin THERM

Une résistance pull-down de 10[kΩ] a été placée sur la pin 3 (SEL) comme cela est conseillé dans le datasheet du MCP à la page 2.

#### 2.7.1.6 Pins SEL et TE

La pin SEL permet de sélectionner le type d'entrée. Un état bas signifie un port USB et un état haut un adaptateur AC-DC. Je mets donc la patte à l'état bas.

#### 2.7.1.7 Pin STAT2

J'ai connecté une LED sur la pin STAT2 exclusivement car cette pin permet à elle seule d'indiquer si la batterie est en charge ou si elle est pleinement chargée.

CHARGE CYCLE STATE	STAT1	STAT2	$\overline{PG}$
Shutdown ( $V_{DD} = V_{BAT}$ )	High-Z	High-Z	High-Z
Shutdown ( $V_{DD} = IN$ )	High-Z	High-Z	L
Shutdown ( $CE = L$ )	High-Z	High-Z	L
Preconditioning	L	High-Z	L
Constant Current	L	High-Z	L
Constant Voltage	L	High-Z	L
Charge Complete - Standby	High-Z	L	L
Temperature Fault	L	L	L
Timer Fault	L	L	L
Low Battery Output	L	High-Z	High-Z
No Battery Present	High-Z	High-Z	L
No Input Power Present	High-Z	High-Z	High-Z

Figure 11 Pin STAT2 MCP73871T-2CCI/ML

Nous observons sur ce tableau que lorsque la charge est en cours la LED sera allumée et qu'elle s'éteindra lorsque la charge est finie

#### 2.7.2 USB-C

J'ai choisi le connecteur USB-C [USB4125-GF-A-0190](#) afin d'obtenir une tension de 5[V] en entrée du chargeur de batterie. Ce type d'USB est la norme pour les nouveaux appareils électroniques et est facilement trouvable, ce qui simplifiera la tâche à l'utilisateur.

Des résistances pull-down de 5,1[kΩ] comme le précise la norme de l'USB-C. Si ces résistances ne sont pas présentes il peut y avoir des dysfonctionnements avec certains câbles.



Figure 12 USB4125-GF-A-0190

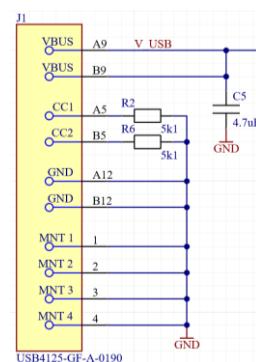


Figure 13 Schéma USB-C

### 2.7.3 Batterie

Pour la batterie, j'ai choisi d'utiliser les accumulateurs Ecell 18650 ECE1 car l'ETML-ES en possédait déjà et car ils sont équipés d'un circuit de protection contre les courts-circuits. J'ai donc estimé avec les composants consommant le plus dans mon système, l'autonomie du circuit.

Composant	Bloc	Utilité	Consommation max à l'ouverture [mA]	Consommation max en stand-by [mA]
MCP73871	Alimentation	Recharge de l'accumulateur	3.75	3.75
Chilli UART-B1	RFID	Lecture du badge RFID	120	1.85
ESP32-C3-WROOM-02-N4	Wi-Fi	Transmission/réception Wi-Fi	280	15
Gâche	Serrure	Ouverture de la porte	1400	0
APTD2012LSURCK	Affichage	LEDs d'indication	14	6
NHD-0420AZ-FSW-GBW-33V3	Affichage	LCD	2.5	0
PIC32MX130F064D	Microcontrôleur	Microcontrôleur	300	11
TOTAL			2120.25	37.6
Heures estimées avec batterie 3400 mAh				90.43

Tableau 7 Consommation de batterie

Sur le tableau ci-dessus nous constatons que le nombre d'heures estimées est d'environ 90 heures. Afin que le circuit soit autonome durant une semaine comme demandé dans le cahier des charges, je décide d'en mettre deux sur le PCB. Le support de batterie que j'ai choisi est le 1049 de Keystone Electronics.

### 2.7.4 Lecture batterie

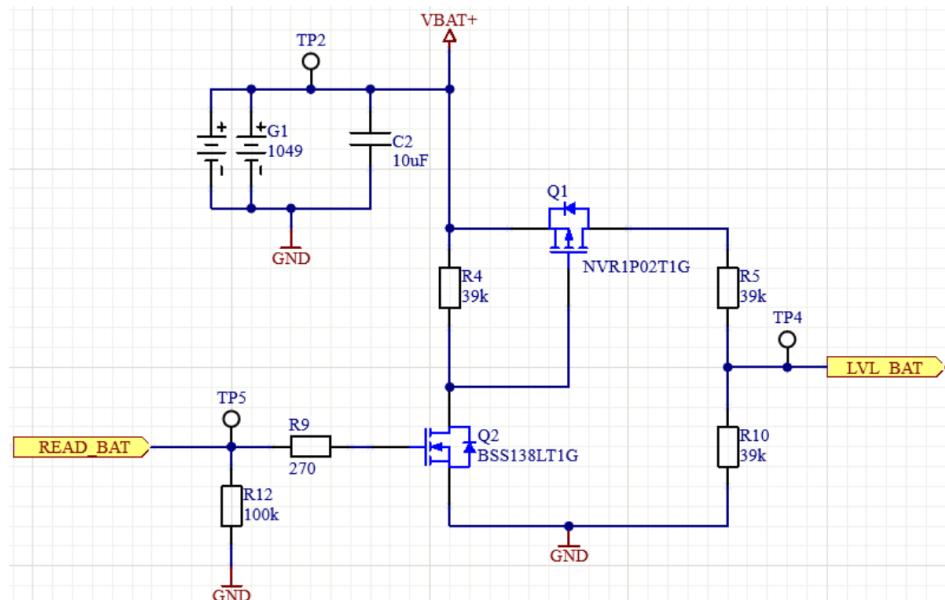


Figure 14 Schéma lecture de batterie

Ce système de transistors MOSFET permet de lire facilement la tension aux bornes de la batterie sans consommer de courant lorsque la tension n'est pas lue. Le pont diviseur est actionné par le signal READ\_BAT sortant du microcontrôleur. La tension est divisée par deux afin que celle-ci n'excède pas la tension maximale admissible sur les entrées du microcontrôleur.

## 2.7.5 DC-DC 12V

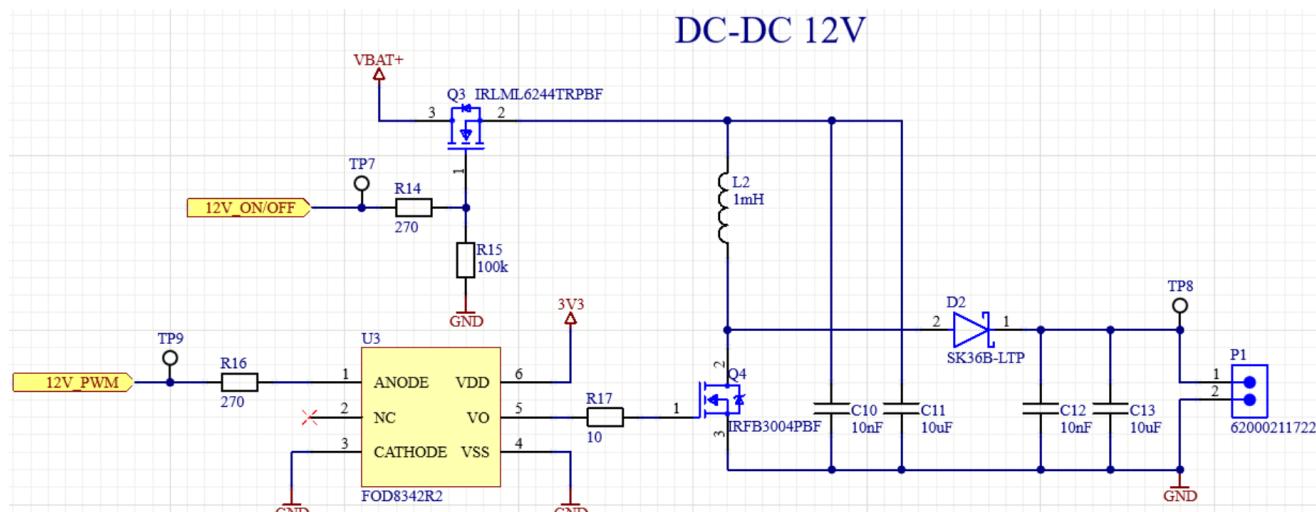


Figure 15 Schéma DC-DC 12V

L'alimentation DC-DC 12[V] pour la commande de la gâche a été dimensionnée selon la théorie du cours d'EIND. J'ai commencé par définir la fréquence de fonctionnement de l'alimentation à découpage qui est généralement la fréquence de fonctionnement du microcontrôleur divisée par 256 :

$$f = \frac{f_{CPU}}{256} = \frac{8 \cdot 10^6 [Hz]}{256} = 31250 [Hz]$$

*Équation 6 Calcul fréquence DC-DC 12V*

Puis je calcule le rapport cyclique de fonctionnement du DC-DC. Je calcule cela avec une tension d'entrée de 4.2[V] ce qui correspond à la tension de la batterie lorsqu'elle est totalement chargée :

$$D = \frac{-V_{in}}{V_{out}} + 1 = \frac{-4.2 [V]}{12 [V]} + 1 = 0.65 [-]$$

*Équation 7 Calcul rapport cyclique*

Maintenant que je connais le rapport cyclique je peux calculer l'inductance de mon système. Je définis l'ondulation du courant à 10% du courant de sortie, ce qui me donne un  $\Delta I_L$  de 80[mA] :

$$L = \frac{(V_{out} - V_{in})(1 - D)}{\Delta I_L \cdot f} = \frac{(12 - 4.2)(1 - 0.65)}{0.08 \cdot 31250} = 1.092[mH] \cong 1[mH]$$

*Équation 8 Calcul inductance*

Je revérifie l'ondulation du courant en prenant en compte l'arrondi de l'inductance :

$$\Delta I_L = \frac{(V_{out} - V_{in})(1 - D)}{L \cdot f} = \frac{(12 - 4.2)(1 - 0.65)}{1 \cdot 10^{-3} \cdot 31250} = 87.36[mA]$$

*Équation 9 Calcul  $\Delta I_L$*

Enfin, je calcule le pic de courant dans l'inductance afin de pouvoir choisir celle-ci en conséquence :

$$I_L = \frac{I_{out}}{(1 - D)} + \frac{\Delta I_L}{2} = \frac{800 \cdot 10^{-3}}{(1 - 0.65)} + \frac{87.36 \cdot 10^{-3}}{2} = 2.33[A]$$

*Équation 10 Calcul  $I_L$*

### 2.7.5.1 Driver de gate

Le transistor Q4 commutant très vite afin de fournir la tension nécessaire à la gâche, il a fallu ajouter un driver de gate afin de s'assurer que le système fonctionne correctement. J'ai donc choisi le FOD8342 qui est idéal pour ce genre d'applications.

## 2.7.6 DC-DC 3.3V

Ce bloc est l'alimentation principale du système. Il permet de transformer la tension de sortie du chargeur d'accumulateur en tension régulée 3.3[V].

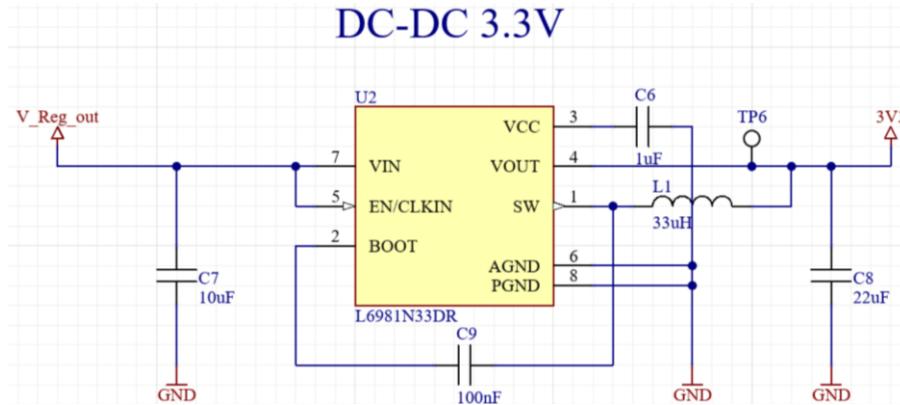


Figure 16 Schéma DC-DC 3.3V

### 2.7.6.1 L6981N33DR

Le [L6981N33DR](#) est un régulateur DC-DC « buck » à découpage avec une tension de sortie fixée à 3.3[V]. Son courant de sortie peut monter jusqu'à 1.5[A] ce qui est amplement suffisant pour ce système, étant donné que la gâche consommera directement sur la batterie et non pas sur le régulateur.

### 2.7.6.2 Composants passifs

Les valeurs des composants sont directement reprises du datasheet :

Symbol	Value	Description
C <sub>IN</sub>	10 µF	Input capacitor
C <sub>VCC</sub>	1 µF	VCC bypass capacitor
C <sub>BOOT</sub>	100 nF	Bootstrap capacitor
C <sub>OUT</sub>	22 µF	Output capacitor
R <sub>1</sub>	400 kΩ	VOUT divider upper resistor
R <sub>2</sub>	82 kΩ	VOUT divider lower resistor
L	33 µH	Output inductor

Figure 17 Extrait du datasheet L6981N33DR

## 2.7.7 PIC32MX130F064D-I/PT

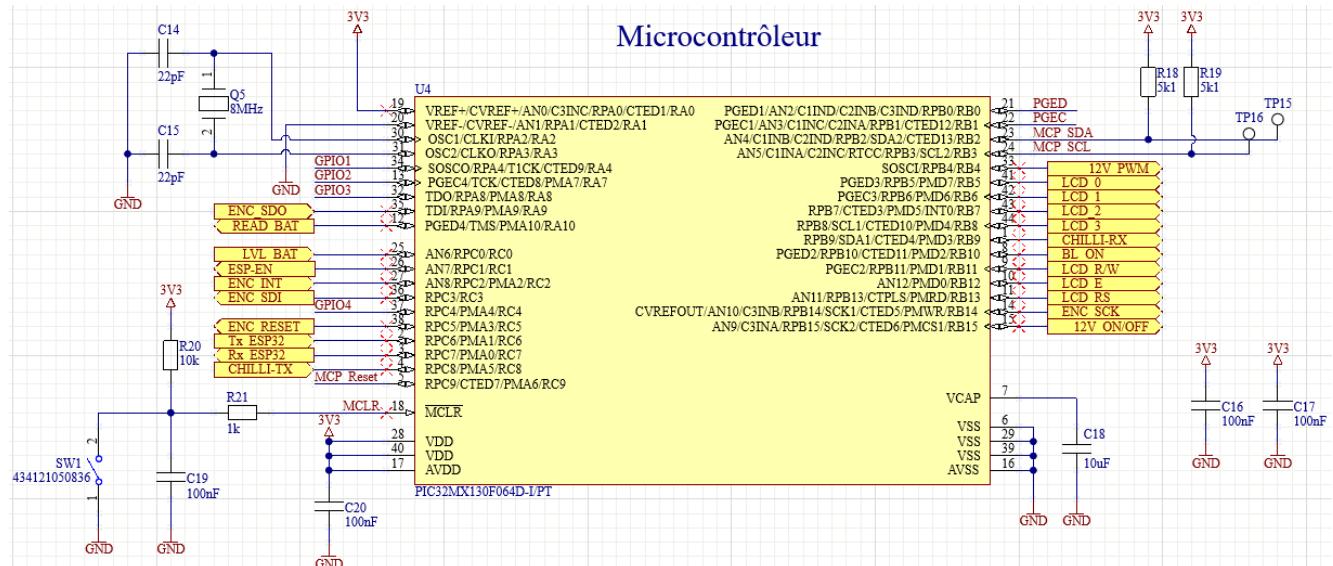


Figure 18 Schéma microcontrôleur

Le microcontrôleur que j'ai choisi est le [PIC32MX130F064D-I/PT](#). Il possède 44 pattes et des pins configurables pour chaque bus nécessaire dans le système (UART, SPI, ...). J'ai également ajouté un quartz externe à celui-ci afin d'être plus précis lors des transmissions de données.

## 2.7.8 I/O Expander

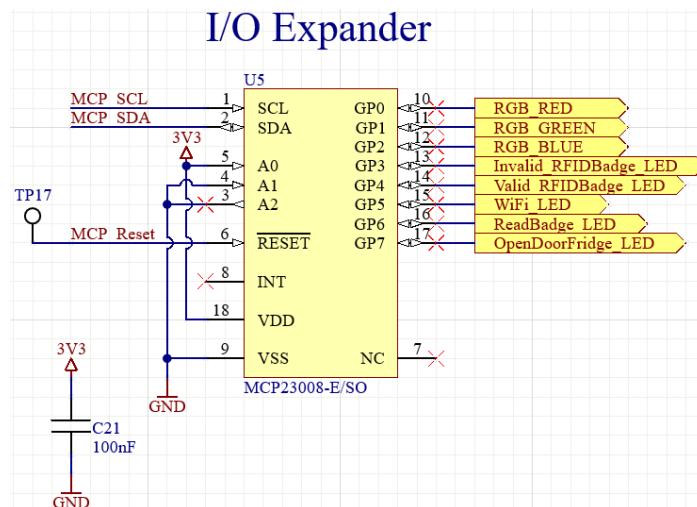


Figure 19 Schéma I/O Expander

Malheureusement le nombre de pattes du microcontrôleur n'était pas suffisant pour contenir tous les signaux nécessaires. J'ai donc ajouté l'I/O expander MCP23008-E/SO au schéma.

MCP23008-E/SO	
Fabricant	Microchip
Description	I/O Expander
Tension d'alimentation	1.8 ~ 5.5 [V]
Nombre d'I/O supplémentaires	8
Tension de sortie critique	V <sub>DD</sub> - 0.7 [V]
Protocole	UART

Tableau 8 Caractéristiques MCP23008-E/SO



Celui-ci me permettra d'activer ou non les LEDs d'affichage du point 2.6.3 via un bus I2C.

## 2.7.9 I/O supplémentaires

Le cahier des charges précisait que des I/O supplémentaires devaient être prévus pour de possibles futurs ajouts. J'ai donc ajouté un connecteur berg mâle à 6 positions contenant 4 sorties GPIO du microcontrôleur ainsi que le VCC et le GND à cet effet.

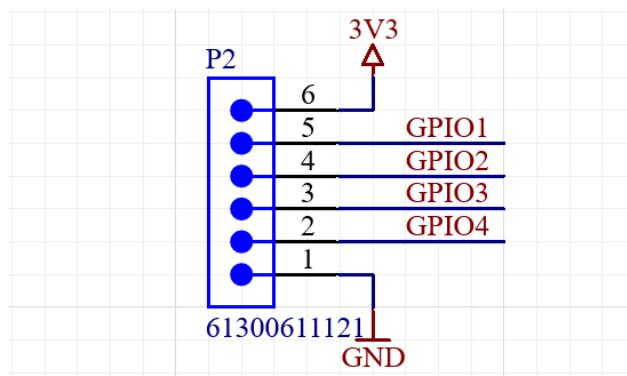


Figure 20 Schéma I/O supplémentaires

### 3 Hardware

#### 3.1 Description générale

Les dimensions du circuit sont de 140x110[mm] et celui-ci comporte des trous de passage M3 à ses extrémités afin de pouvoir se fixer facilement au boîtier. Les pistes classiques sont épaisses de 0.2[mm] et les pistes d'alimentation de 0.5[mm]. Le circuit contient un plan de masse sur les deux faces, reliés par un via stitching et il correspond à la classe de fabrication 6C d'Europcircuits. Tous les composants sont placés sur la couche TOP et les connecteurs pour l'USB-C, la gâche et l'Ethernet sont disposés sur le circuit afin d'être facilement accessibles via des ouvertures sur le boîtier. Le module RFID et le LCD ont été déportés sur le couvercle du boîtier, des connecteurs ont donc été prévus à cet effet (K1 et P5). Des points de test ont également été disposés sur tout le PCB afin de pouvoir facilement effectuer des mesures.

#### 3.2 Répartition des blocs

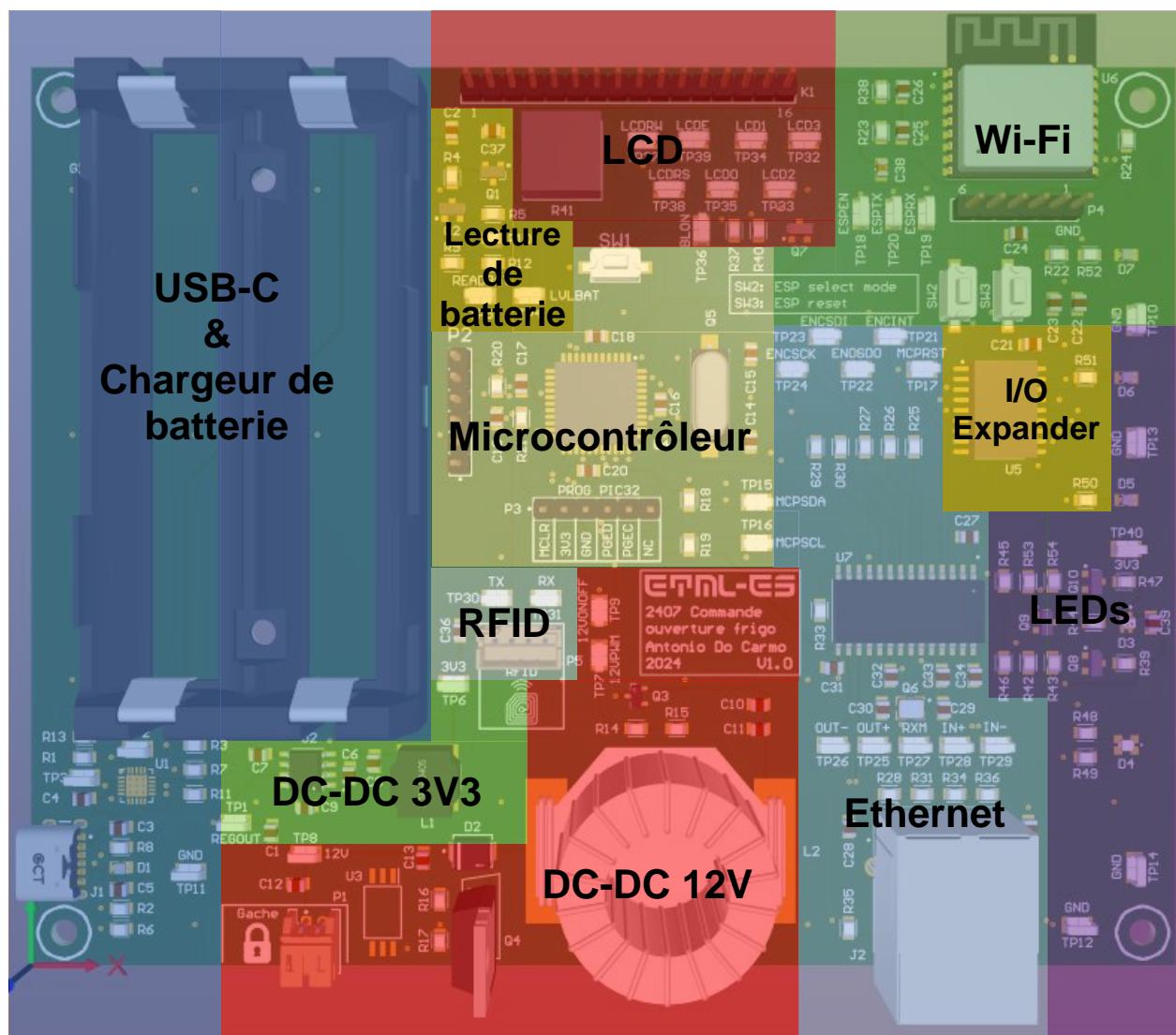


Figure 21 Répartition des blocs hardware

Les composants ont été disposés sur le circuit de façon à former des blocs distincts comme démontré sur la figure ci-dessus.

### 3.3 Vues des couches du PCB

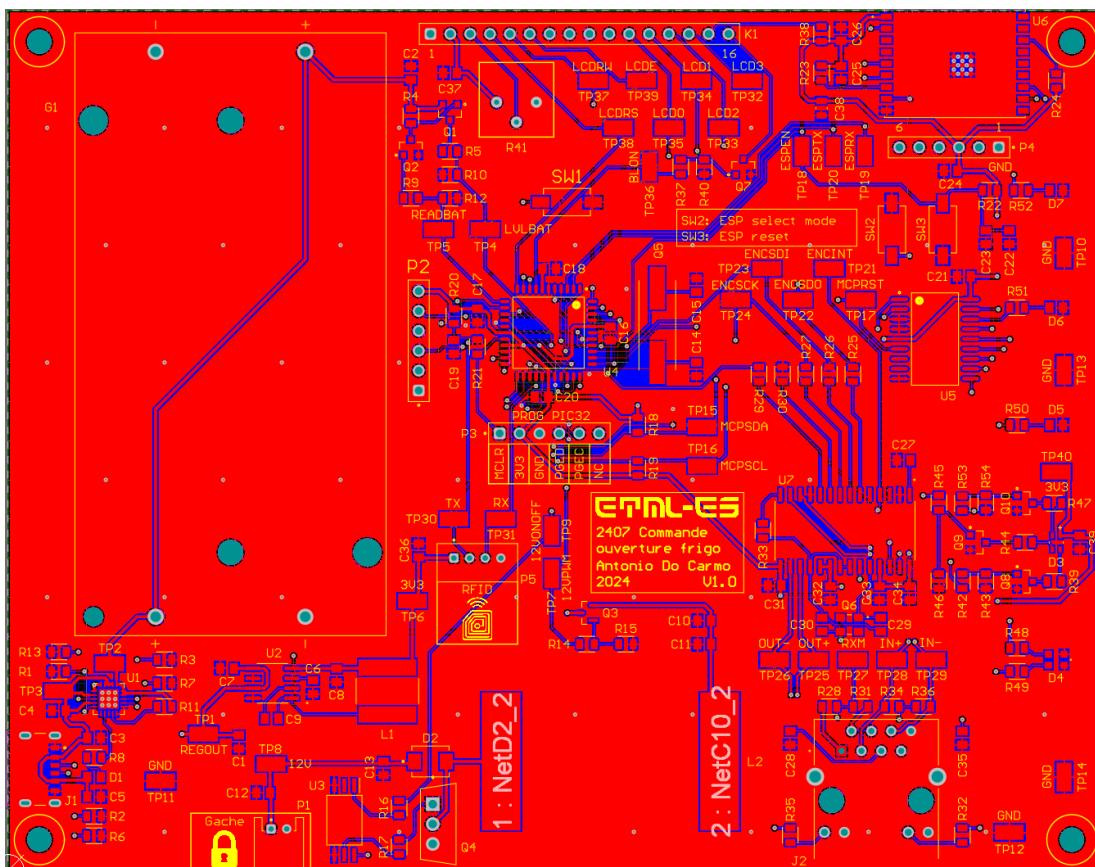


Figure 22 Vue TOP du PCB

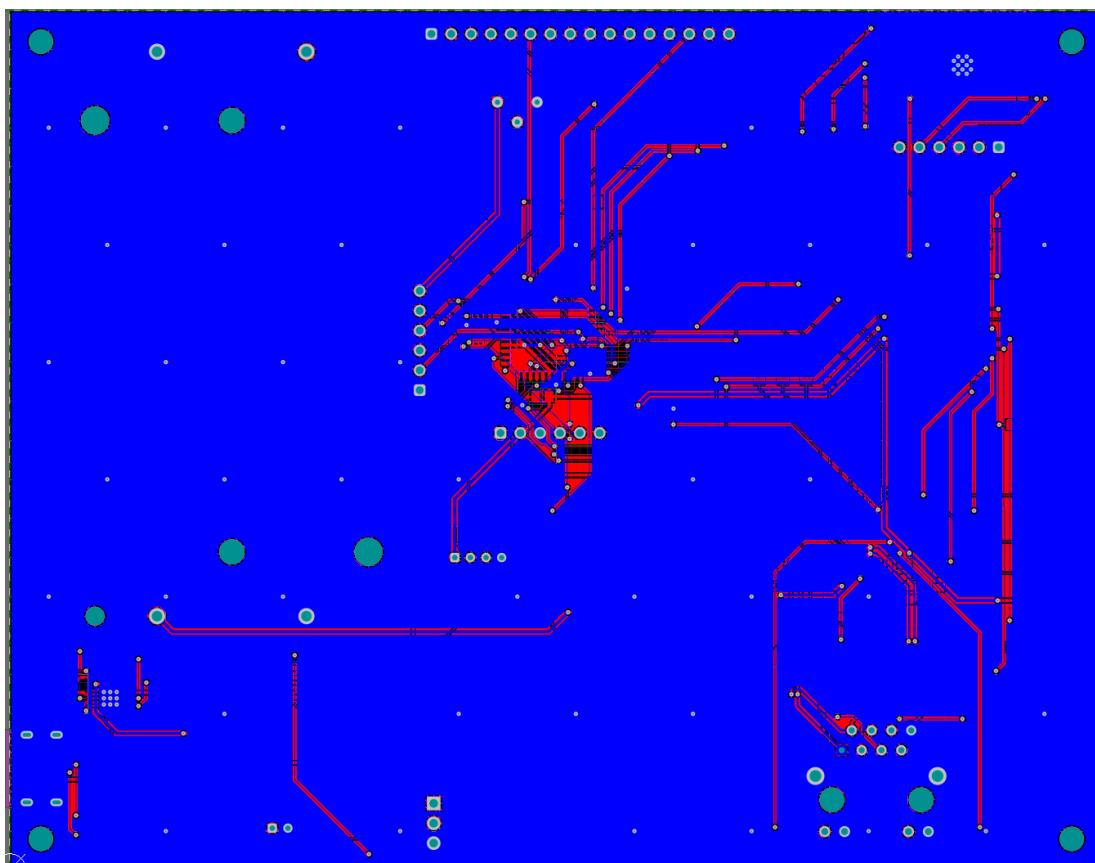


Figure 23 Vue BOTTOM du PCB

### 3.4 Vues 3D du PCB

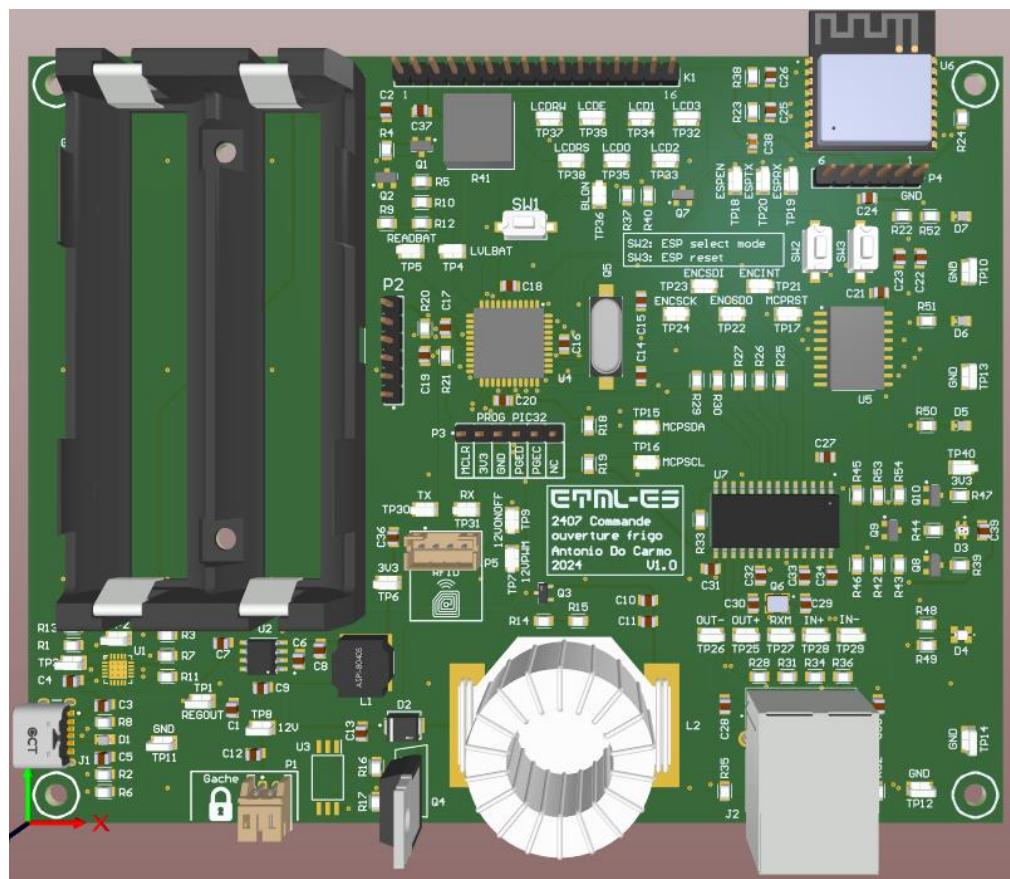


Figure 24 Vue 3D du PCB de dessus

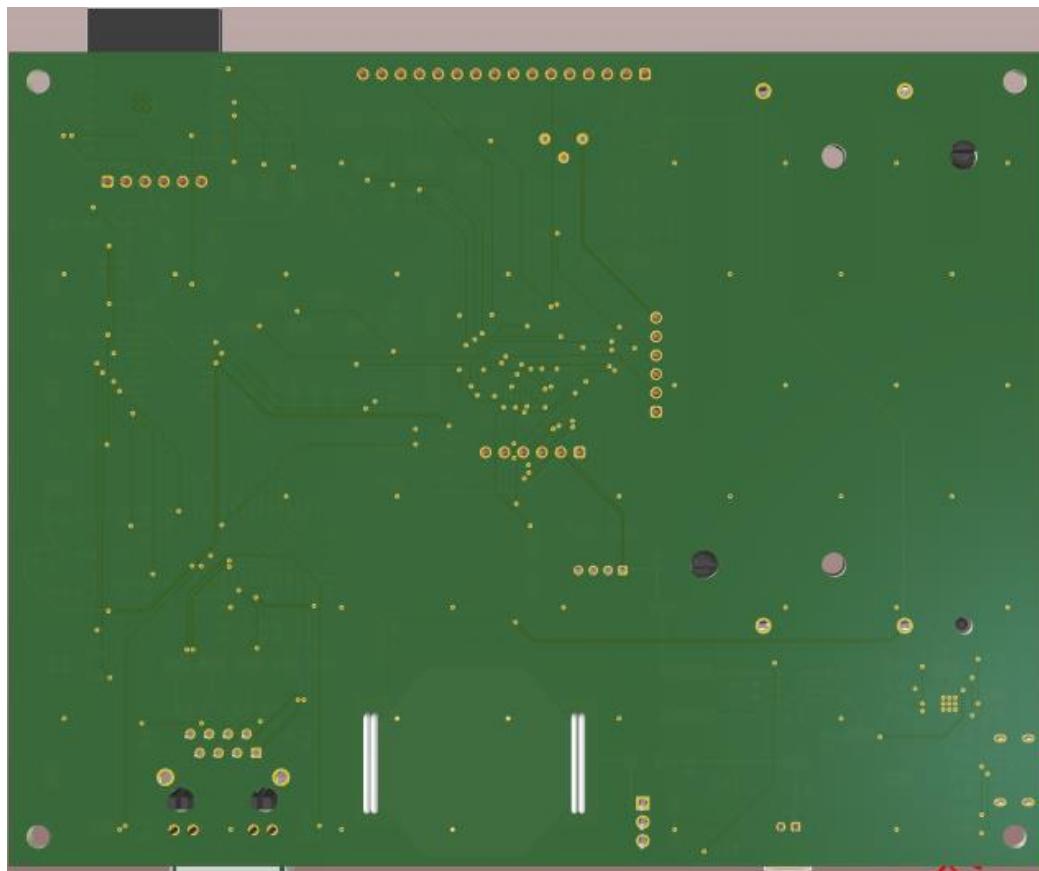


Figure 25 Vue 3D du PCB de dessous

### 3.5 Détails de conception

#### 3.5.1 Chargeur de batterie

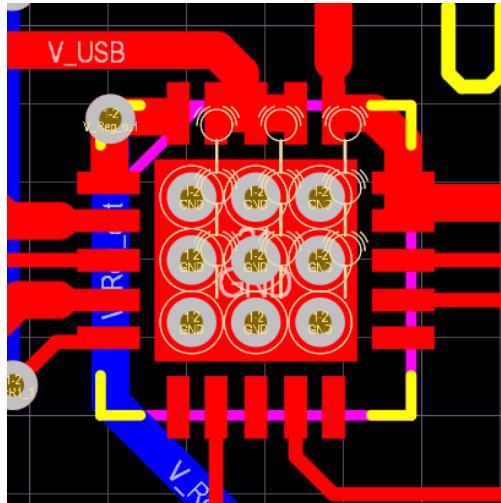


Figure 26 MCP73871 footprint

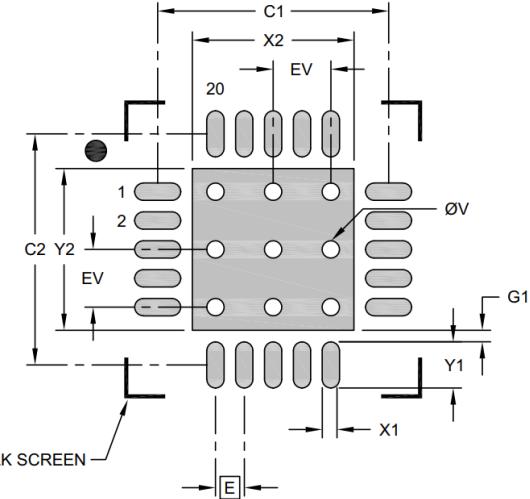


Figure 27 MCP73871 recommandations du datasheet

Lors de la conception de mon PCB, j'ai dû apporter une attention particulière au pad thermique du MCP73871. En effet le datasheet de celui-ci précisait qu'il fallait placer 9 vias sur celui-ci afin que le composant dissipe correctement la chaleur. J'ai donc pris le soin d'ajouter les vias.

#### 3.5.2 ESP32

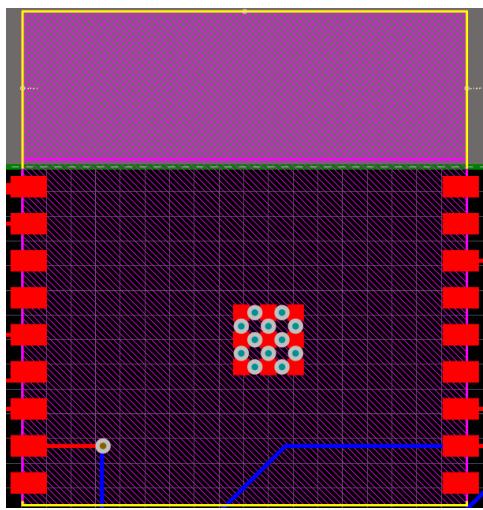


Figure 28 ESP32 footprint

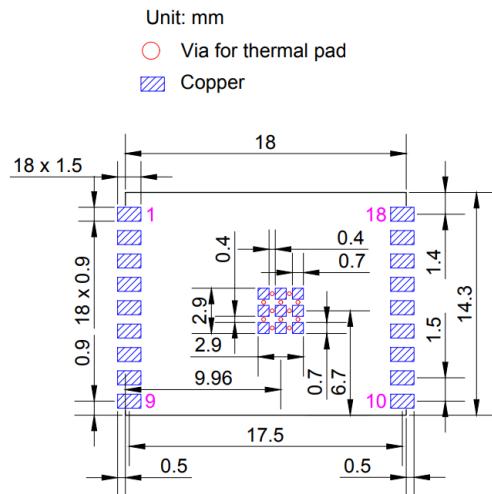


Figure 29 ESP32 recommandations du datasheet

Le module Wi-Fi ESP32-C3-WROOM-02-N4 possède également des pads thermiques. J'ai apporté une attention particulière à ce que des vias soient présents sur le pad afin de ne pas porter préjudice à la communication Wi-Fi.

### 3.6 Largeur des pistes DC-DC 12V

Le bloc DC-DC 12V étant prévu pour générer un courant de sortie de 800 [mA], j'ai dû faire attention à ce que les pistes du PCB puissent supporter un tel courant. D'après des tests effectués sur la gâche, celle-ci prend environ 1,4 [s] à s'ouvrir dans ces conditions. Je me serai donc du logiciel de calcul Saturn afin de calculer le courant max pouvant être supporté par les pistes de 0,5 [mm] durant ce temps. Je constate ainsi que cette largeur de piste est amplement suffisante.

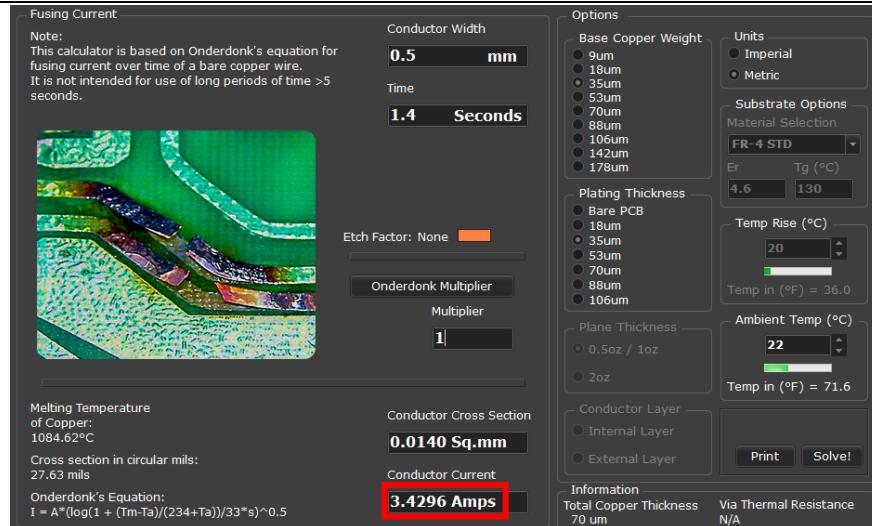


Figure 30 Calcul largeur des pistes DC-DC 12V

## 4 Modifications

### 4.1 DC-DC 12V

Après la commande du PCB je me suis rendu compte d'une erreur dans la conception du bloc step-up DC-DC 12[V]. En effet, le système d'activation du step-up ne peut pas fonctionner en raison de la tension de threshold. Le datasheet du MOSFET IRLML6244TRPBF indique que la tension sur la grille doit être plus élevée que la tension à la source de 0.9[V]. Ce qui n'est pas le cas dans le montage effectué de la figure 29. La tension sur la grille du MOS sera toujours plus élevée que la tension de la grille en raison de la nature de la batterie Li-ion (plage de tension d'environ 3.5 à 4.2[V]). Afin de remédier à cela je décide d'effectuer une modification sur le circuit.

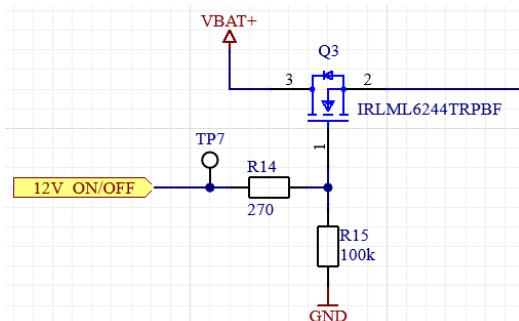


Figure 31

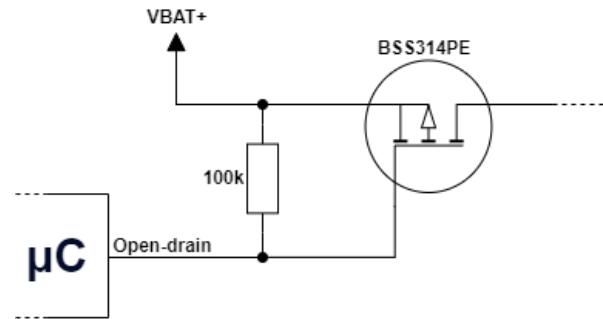


Figure 32

Je vais changer de transistor pour le BSS314PE qui est un MOSFET à canal P. Pour que cela soit fonctionnel il a également fallu supprimer R14 et R15 et ajouter une résistance de 100[kΩ] entre la grille et la source (voir figure 30) et configurer la sortie du microcontrôleur en mode open-drain. Ainsi lorsque la sortie du μC est flottante, la grille du transistor est tirée à la tension VBAT+, ce qui permet de s'assurer que le MOS ne conduit pas.



Figure 33 Photo de la modification

## 5 Boîtier

Le boîtier a été conçu sur le programme Solidworks 2023. Ses dimensions sont de 135x160x53[mm]. Trois ouvertures sont disposées sur les faces de la base du boîtier afin de permettre à l'utilisateur de brancher facilement l'USB-C, le câble Ethernet et le connecteur de la gâche.

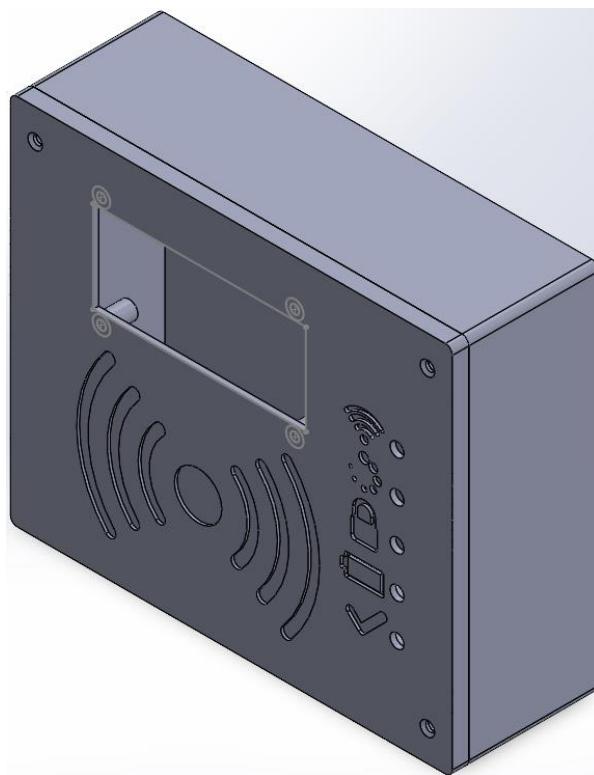


Figure 34 Vue 3D du boîtier.

Le couvercle du boîtier quant à lui possède une ouverture de 77x33[mm] afin de pouvoir emboîter l'affichage LCD. Quatre trous de fixation sont disposés autour de cette ouverture à cet effet. Cinq trous sont également présents sur le côté droit du couvercle de façon à pouvoir y glisser des guides de lumière pour les LEDs présentes sur le PCB. Des indications sont présentes à côté de ces trous afin d'indiquer respectivement la communication Wi-Fi, la lecture du badge, l'ouverture de la gâche, le niveau de la batterie et la validation du badge. Un logo RFID est également présent au centre afin d'indiquer à l'utilisateur où le badge doit être placé. Les plans mécaniques du boîtier sont présents en Annexe K.



Figure 35 Photographie du boîtier

## 6 Software

### 6.1 Configuration du clock externe

Afin de cadencer mon microcontrôleur, j'ai implémenté un quartz externe de 8 [MHz] dans mon circuit. J'ai donc dû modifier le « Clock Diagram » sur le configurateur graphique Harmony pour indiquer au microcontrôleur l'existence du quartz externe. Pour ce faire, j'ai dû ajuster la valeur du « Primary Oscillator » à la fréquence de mon quartz et sélectionner « XT » sur le menu déroulant :

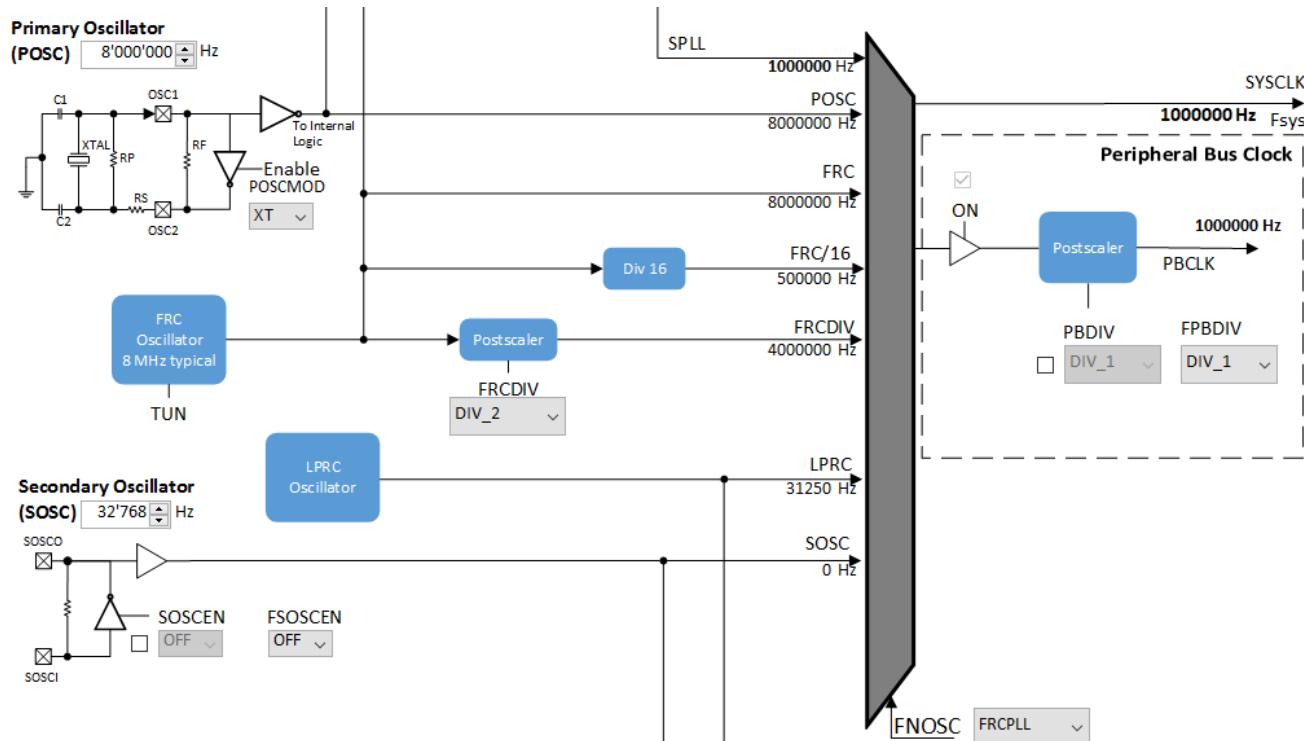


Figure 36 Configuration graphique du SYSCLK

### 6.2 Timer 1

La principale utilité du timer 1 dans mon programme est de cadencer l'exécution de l'application. J'ai donc défini la période de l'exécution à 100 millisecondes. Voici comment j'ai effectué mes calculs :

$$Tick_{timer1} = \frac{T_{timer1}}{T_{Oscillateur}} = \frac{100 \cdot 10^{-3}}{125 \cdot 10^{-9}} = 800'000 \text{ tick}$$

La capacité maximale du timer est de 65'536 tick (16 bits), je dimensionne donc un prescaler de 256 afin d'obtenir un nombre de tick inférieur à cette valeur :

$$T_{final\,timer1} = \frac{T_{timer1}}{\text{prescaler}} = \frac{800'000}{256} = 3125 \text{ tick}$$

Je soustrais 1 à cette valeur afin de prendre en compte la valeur 0 du compteur, j'arrive donc à une valeur finale de 3124 tick :

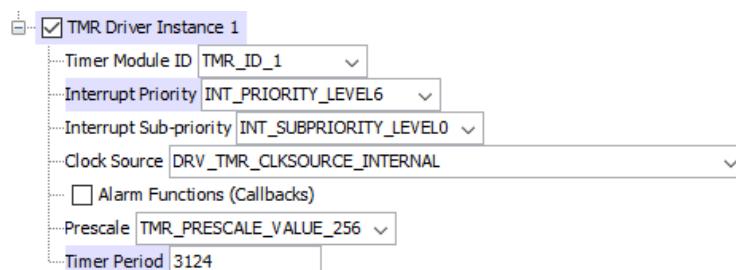


Figure 37 Configuration du timer 1

## 6.3 Interruption du Timer 1

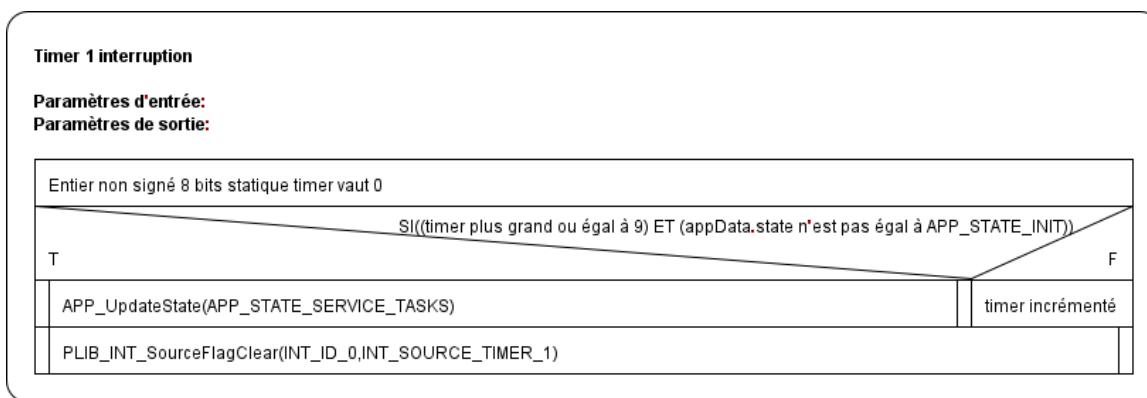


Figure 38 Structogramme timer 1

## 6.4 Timer 2

Le timer 2 est le timer permettant de déterminer la fréquence de l'OC générant le PWM pour l'ouverture de la gâche. Lors du point 2.7.5, il a été déterminé que la fréquence de ce PWM devait être de 31'250 [Hz]. Voici donc comment j'ai dimensionné le timer :

$$T_{timer2} = \frac{1}{f_{timer2}} = \frac{1}{31'250} = 32[\mu\text{s}]$$

$$Tick_{timer2} = \frac{T_{timer2}}{T_{Oscillateur}} = \frac{32 \cdot 10^{-6}}{125 \cdot 10^{-9}} = 256 \text{ tick}$$

Cette valeur étant inférieure à 65'536, il n'est pas nécessaire d'établir un prescaler pour ce timer.

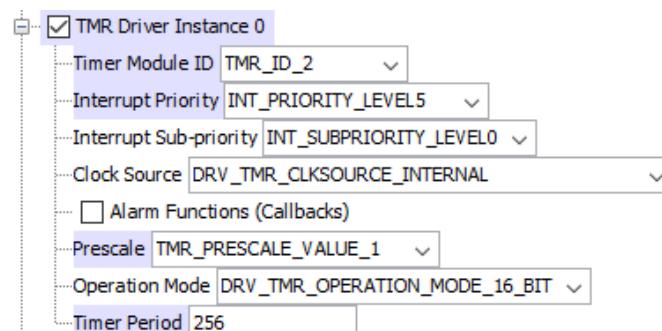


Figure 39 Configuration du timer 2

## 6.5 OC1

L'OC1 a été configurée afin de générer un signal PWM basé sur le timer 2. L'OC pulse width est donc réglée au même nombre de tick que celui-ci et le mode sélectionné est le mode « OC\_COMPARE\_PWM\_MODE\_WITHOUT\_FAULT\_PROTECTION » afin de pouvoir faire varier le rapport cyclique dans le code sans que la période ne change

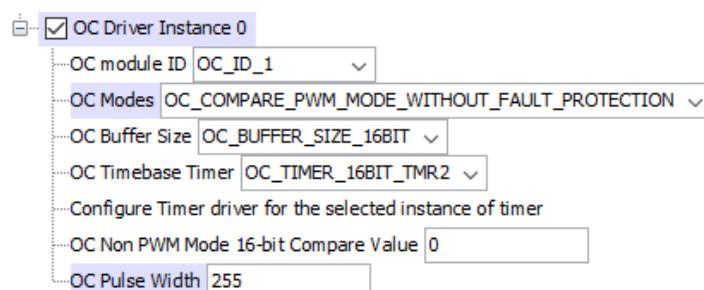


Figure 40 Configuration de l'OC1

## 6.6 Bus USART 1

Le bus USART1 est le bus utilisé afin de communiquer avec le module Wi-Fi ESP32C3WROOM02. Je l'ai configuré ainsi afin de correspondre aux indications dans le datasheet de celui-ci :

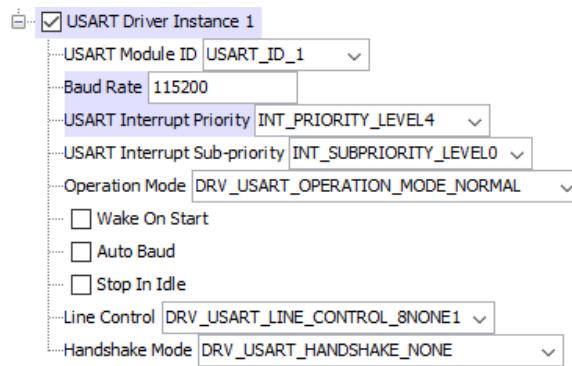


Figure 41 Configuration USART1

## 6.7 Bus USART 2

Le bus USART2 est le bus utilisé afin de communiquer avec le module RFID Chilli UART B1. Je l'ai configuré ainsi afin de correspondre aux indications dans le datasheet de celui-ci :

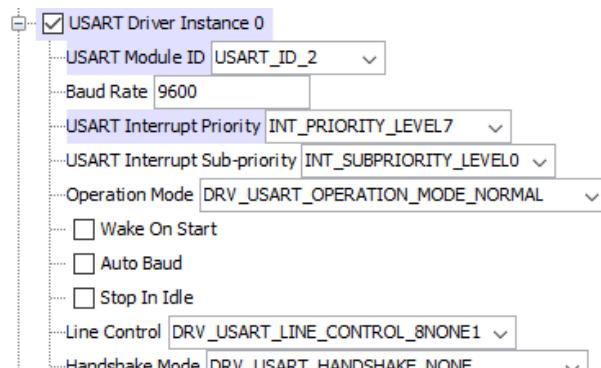


Figure 42 Configuration USART2

## 6.8 Marche à suivre configuration ESP32

Tout d'abord il est absolument nécessaire de télécharger le [firmware](#) de l'ESP32C3WROOM02 dans sa version la plus récente.

### ESP32-C3-MINI-1 Series %

- v3.3.0.0 [ESP32-C3-MINI-1-AT-V3.3.0.0.zip](#) (Recommended)

Figure 43 Téléchargement du firmware ESP32

Il faudra également télécharger le [programme](#) permettant de flasher la mémoire du microcontrôleur :

Flash Download Tools		Expand all +	Download selected		
<input type="checkbox"/>	Title	Platform	Version	Release Date ▾	Download
<input type="checkbox"/>	Flash Download Tools	Windows PC	V3.9.7	2024.06.07	

Figure 44 Téléchargement flash tool

Puis, à l'aide du logiciel Putty, réaliser cette série d'actions :

- Ouvrir le gestionnaire de périphériques de l'ordinateur
- Connecter l'interface au port USB du PC et récupérer son nom de port
- Ouvrir Putty
- Sélectionner la communication série et configurer les options selon l'image ci-dessous (le numéro de port COM doit être le même que celui qui a été relevé sur le gestionnaire de périphériques)

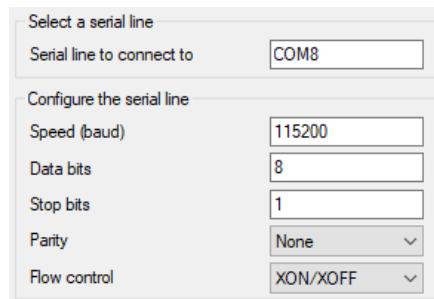


Figure 45 Configuration Putty

- Appuyer sur « Open »
- Alimenter le PCB et appuyer simultanément sur SW2 et SW3 afin de démarrer le « Boot download mode »
- Le message suivant doit s'afficher sur Putty :

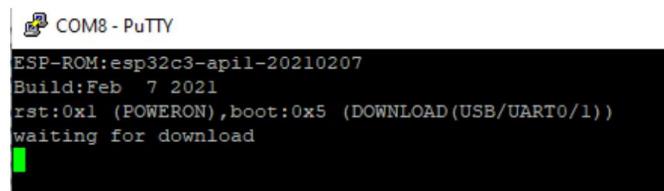


Figure 46 Boot download mode Putty

- Fermer Putty

Le microcontrôleur est maintenant en « Boot download mode ». Il reste à présent à flasher sa mémoire :

- Ouvrir le flash download tool téléchargé précédemment
- Sélectionner la configuration suivante :

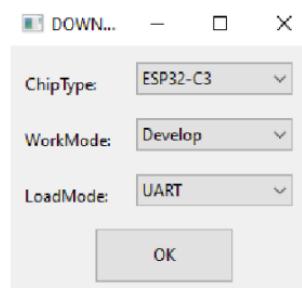


Figure 47 Démarrage flash download tool

Puis dans l'ordre :

1. Cocher la case de la première ligne, sélectionner le fichier « factory\_MINI-1.bin » puis mettre la valeur 0x00 dans la colonne de droite
2. Cocher « DoNotChgBin
3. Sélectionner le port com utilisé sur Putty et régler le baudrate à 115'200 [Bd]
4. Cliquer sur le bouton « Start » afin de démarrer la programmation

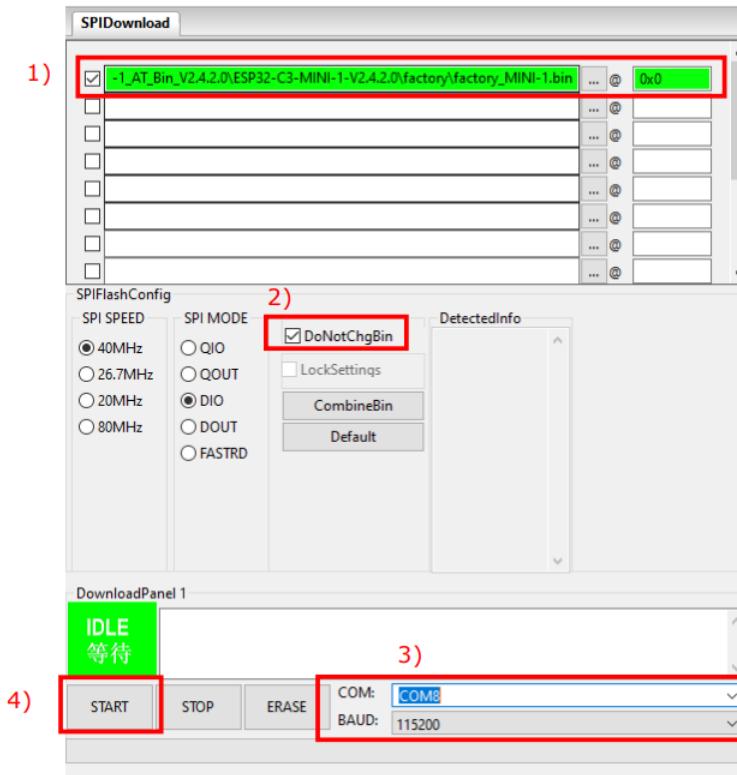


Figure 48 Configuration du flash tool

- Une fois le module programmé, fermer le programme et ouvrir Putty
- Régler les paramètres de la communication série comme indiqué précédemment et ouvrir la communication
- Appuyer simultanément sur SW2 et SW3
- Si l'ESP32 a été correctement programmé, le message suivant doit s'afficher :

```

COM8 - Putty
ESP-ROM:esp32c3-april-20210207
Build:Feb 7 2021
rst:0xl (POWERON),boot:0xc (SPI_FAST_FLASH_BOOT)
SFNIP:0xee
mode:DIO, clock div:2
load:0x3fcde100,len:0x18c8
load:0x403ce000,len:0x8d4
load:0x403d0000,len:0x2d6c
entry:0x403ce000
I (31) boot: ESP-IDF qa-test-v4.3.3-20220423 2nd stage bootloader
I (31) boot: compile time 07:45:53
I (32) boot: chip revision: 3
I (35) boot.esp32c3: SPI Speed      : 40MHz
I (39) boot.esp32c3: SPI Mode       : DIO
I (44) boot.esp32c3: SPI Flash Size : 4MB
I (49) boot: Enabling RNG early entropy source...
I (54) boot: Partition Table:
I (58) boot: # Label           Usage     Type ST Offset  Length
I (65) boot: 0 oadata          OTA data  01 00 0000d000 00002000
I (73) boot: 1 phy_init        RF data   01 01 0000f000 00001000
I (80) boot: 2 nvs             WiFi data 01 02 00010000 0000e000
I (87) boot: 3 at_customize    unknown   40 00 0001e000 00042000
I (95) boot: 4 ota_0           OTA app   00 10 00060000 001d0000
I (102) boot: 5 ota_1           OTA app   00 11 00230000 001d0000
I (110) boot: End of partition table
I (114) boot: No factory image, trying OTA 0
I (119) esp_image: segment 0: paddr=00060020 vaddr=3c140020 size=2a620h (173600)
map
I (166) esp_image: segment 1: paddr=0008a648 vaddr=3fc91200 size=03c6ch ( 15468)
load
I (170) esp_image: segment 2: paddr=0008e2bc vaddr=40380000 size=01d5ch ( 7516)
load
I (173) esp_image: segment 3: paddr=00090020 vaddr=42000020 size=133ab0h (126020)
map
I (455) esp_image: segment 4: paddr=001c3ad8 vaddr=40381d5c size=0f3c4h ( 62404)
load
I (471) esp_image: segment 5: paddr=001d2ea4 vaddr=50000000 size=00014h (   20)
load
I (471) esp_image: segment 6: paddr=001d2ec0 vaddr=50000018 size=00010h (   16)
load
I (482) boot: Loaded app from partition at offset 0x60000
I (524) boot: Set actual ota_seq=1 in oadata[0]
I (524) boot: Disabling RNG early entropy source...
module_name:MINI-1
max_tx_power=78,ret=0
2.4.2

```

Figure 49 Confirmation de la programmation sur Putty

## 6.9 App \_Tasks

La machine d'état du code fonctionne selon le modèle de base du fichier App.c. Le premier état par lequel la machine passe est APP\_STATE\_INIT afin d'initialiser les différents périphériques et les I/O du microcontrôleur.

Lorsque cet état a fini de s'exécuter je mets la machine d'état sur APP\_STATE\_WAIT. Cet état ne fait rien à part passer le processeur en mode IDLE, ce qui permet de diminuer la consommation du système.

Enfin l'état principal de la machine d'état est APP\_STATE\_SERVICE\_TASKS qui est le mode qui exécute les différentes actions nécessaires au fonctionnement du système.

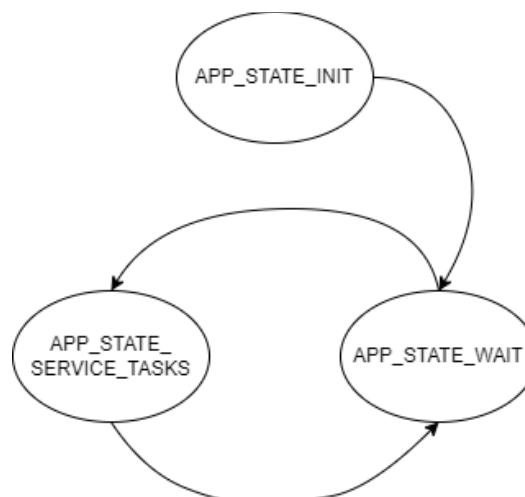


Figure 50 Diagramme d'état

Le structogramme ci-dessous permet d'observer dans le détail le code implémenté :

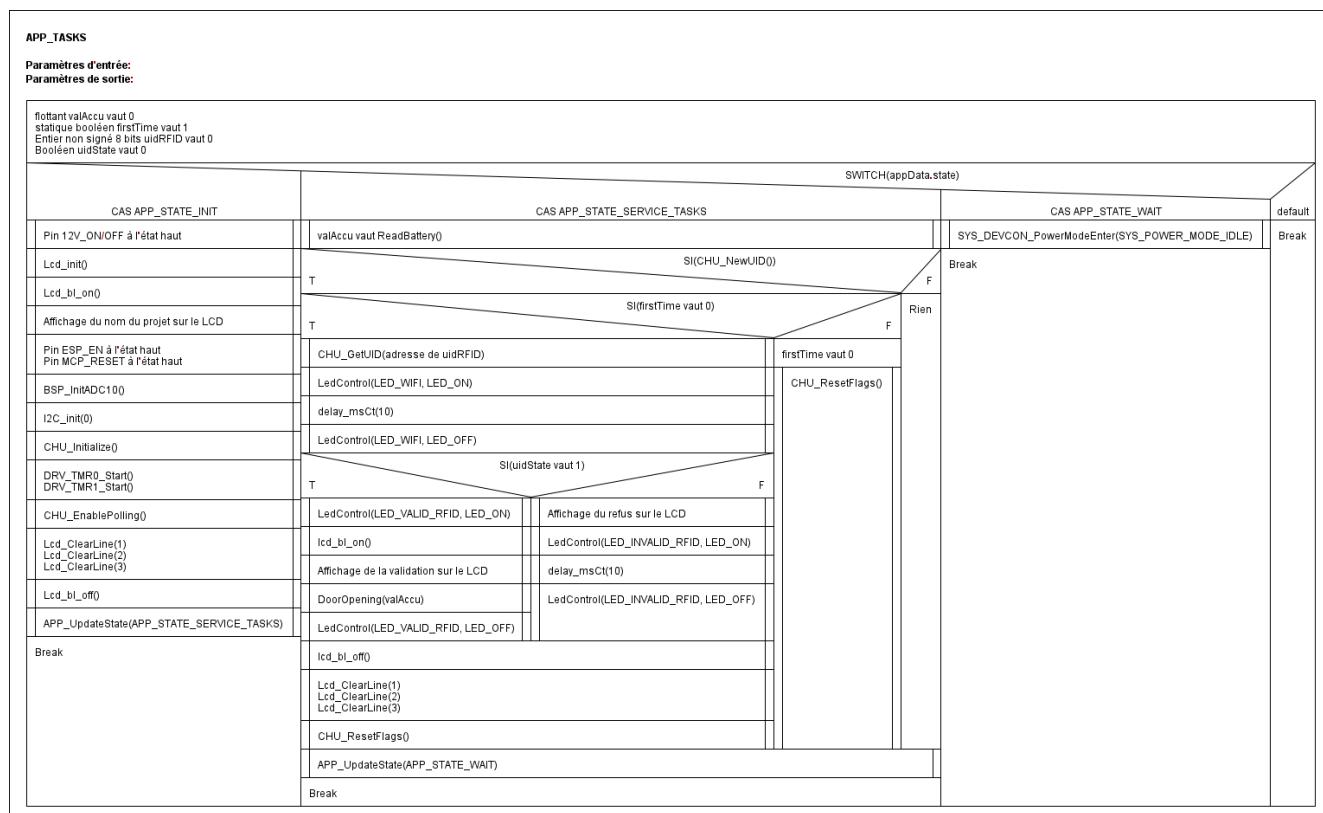


Figure 51 Structogramme APP\_TASKS

**APP\_STATE\_INIT :**

- La pin « 12V\_ON/OFF » du système d'ouverture de la gâche passe à l'état haut afin de bloquer la consommation du courant sur la gâche
- Le LCD est initialisé, sa backlight est allumée et les informations du projet sont affichées.
- La pin « ESP\_EN » est passée à l'état haut afin d'activer le module Wi-Fi
- La pin « MCP\_RESET » est passée à l'état haut afin d'activer l'I/O expander
- L'ADC, le bus I2C, le module RFID ainsi que les deux timers sont initialisés
- Le mode polling du module RFID est activé
- Les différentes lignes du LCD sont effacées et la backlight de celui-ci est éteinte.
- Mise à jour de la machine d'état à APP\_STATE\_WAIT

**APP\_STATE\_SERVICE\_TASKS :**

- Lecture de la tension aux bornes de la batterie
- **SI** un badge est détecté par le module RFID :
  - **SI** la variable de vérification firstTime est à zéro :
    - Récupération de l'UID du badge
    - Led Wi-Fi allumée afin de simuler une communication Wi-Fi
    - Attente
    - Led Wi-Fi éteinte
    - **SI** le badge de l'utilisateur est autorisé :
      - Led d'autorisation allumée
      - Backlight du LCD allumée et message d'ouverture affiché
      - Ouverture de la porte
      - Led d'autorisation éteinte
    - **SINON** :
      - Affichage du refus sur le LCD
      - Led de non-autorisation allumée
      - Attente
      - Led de non-autorisation éteinte
    - Les différentes lignes du LCD sont effacées et la backlight de celui-ci est éteinte
    - Remise à zéro des flags du module RFID
  - **SINON** :
    - firstTime vaut zéro
    - Remise à zéro des flags du module RFID
    - Mise à jour de la machine d'état à APP\_STATE\_WAIT

**APP\_STATE\_WAIT :**

- Le processeur passe en mode IDLE

**6.10 LedControl**

La fonction LedControl commence par vérifier si l'action vaut 1 ou 0. Si elle est à 1 alors un masque est appliqué sur la variable statique ledsToLight afin d'allumer la LED désirée. Sinon, un masque est appliqué afin de l'éteindre.

Puis la communication I2C avec le MCP23008 débute. Le format de la trame est le suivant :

- Envoie un start suivi de l'adresse du MCP (0x42)
- Envoie l'adresse du registre IODIR (0x00)
- Envoie la configuration des I/O (0x00 afin de les configurer en sortie)
- Stoppe la communication I2C
- Envoie un start suivi de l'adresse du MCP (0x42)
- Envoie l'adresse du registre GPIO (0x09)
- Envoie la variable ledsToLight contenant les LEDs à allumer ou éteindre

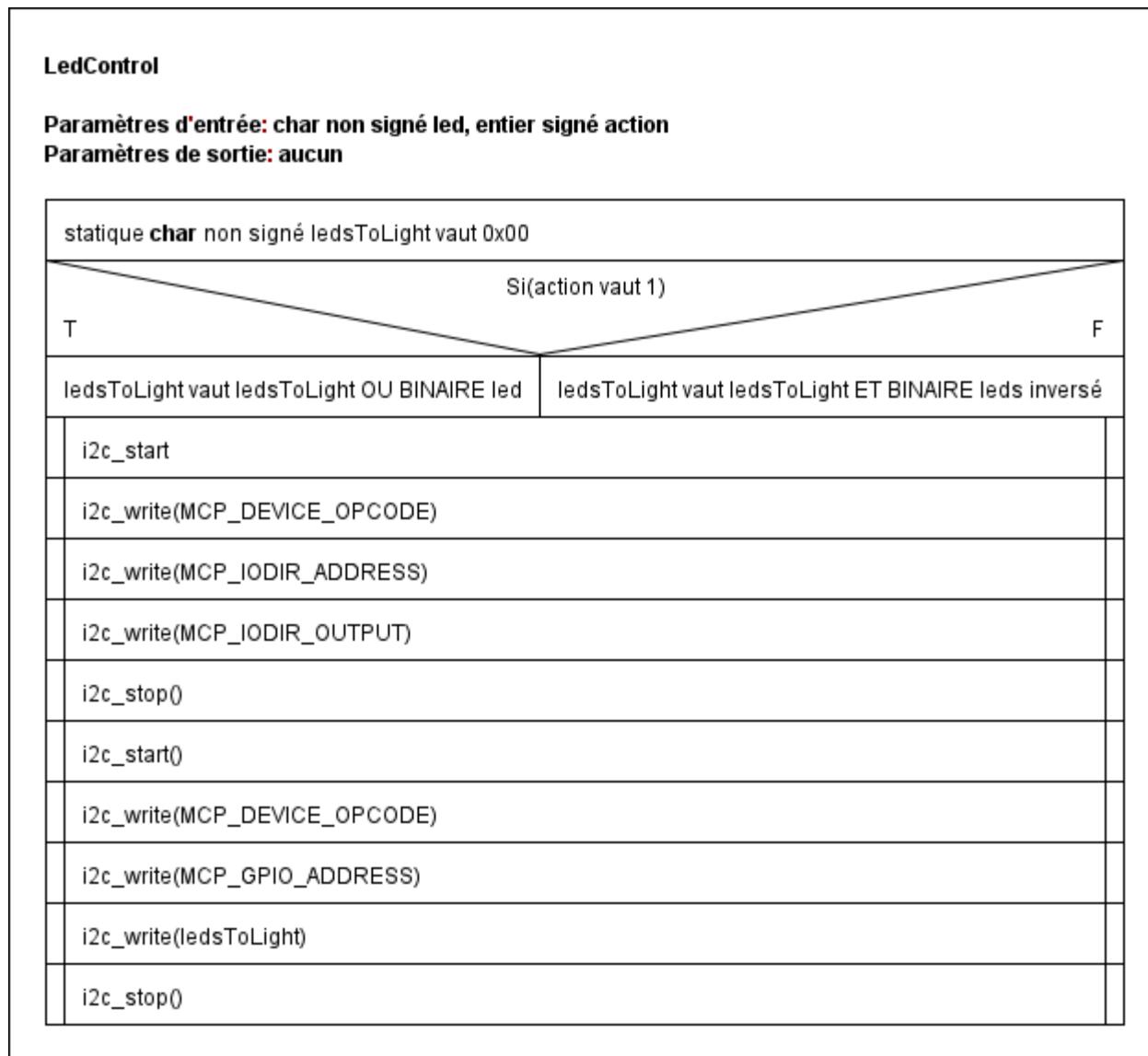


Figure 52 Structogramme LedControl

## 6.11 ReadBattery

La fonction ReadBattery débute par la déclaration des variables val\_ADC et batteryVoltage contenant respectivement la tension des accumulateurs en tick et en tension. La pin READ\_BAT est activée afin d'enclencher le circuit de lecture des batteries. La fonction lit la valeur de l'ADC et convertit celle-ci en tension. Si le niveau de tension est supérieur à 4 [V], la LED s'allumera en vert. Si elle est inférieure à 3.7 [V], la LED s'allumera en rouge. Sinon, une lumière bleue sera renvoyée par la LED indiquant un niveau de tension moyen. Enfin, la pin READ\_BAT est remise à un état bas et la tension aux bornes de la batterie est retournée.

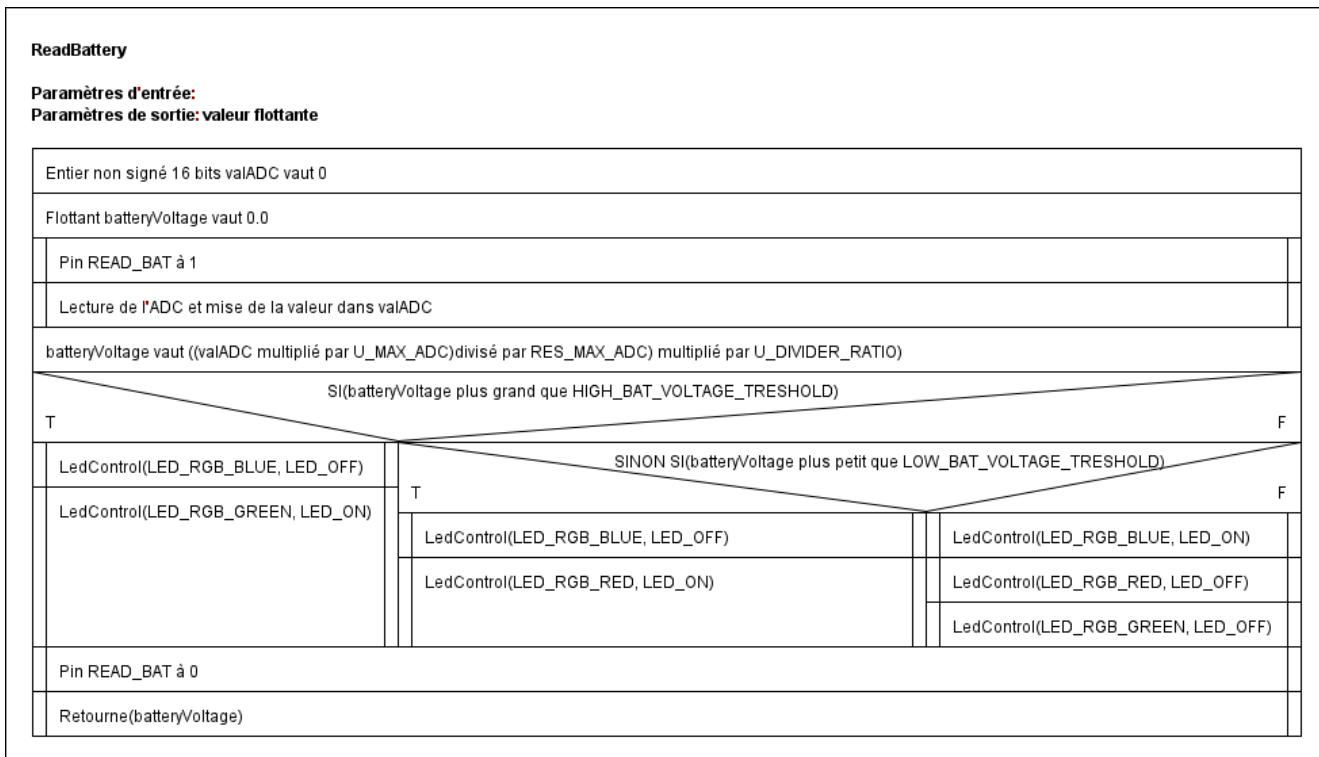


Figure 53 Structogramme ReadBattery

## 6.12 DoorOpening

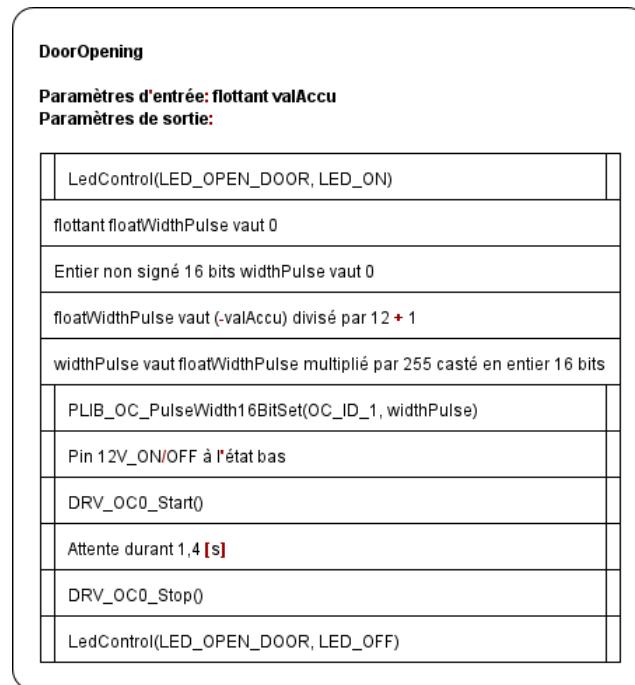


Figure 54 Structogramme DoorOpening

La fonction DoorOpening débute par l'activation de la LED indiquant l'ouverture de la porte. Deux variables sont déclarées. FloatWidthPulse contient la valeur flottante du rapport cyclique et widthPulse contient la même valeur en nombre de tick. Puis un calcul est effectué afin de déterminer le rapport cyclique en fonction de la tension de la batterie :

$$D = \frac{-V_{in}}{V_{out}} + 1$$

Cette valeur est convertie en nombre de tick en la multipliant par 255 et est castée en entier signé 16 bits. La fonction permettant de définir le rapport cyclique est appelée et le PWM est enclenché durant 1,4[s]. Enfin, la led d'ouverture est éteinte pour signifier la fin de la séquence d'ouverture.

## 7 Mesures

Le schéma de mesure est disponible en Annexe H et les mesures sont disponibles en Annexe L

### 7.1 PWM gâche

En mesurant à l'oscilloscope le PWM pilotant le DC-DC 12V de la gâche, j'ai pu relever les rapports cycliques des signaux avec une tension aux bornes de la batterie de 3.5[V] et de 4.2[V]. Voici les résultats obtenus :

Tension sur la batterie [V]	Rapport cyclique [%]	Durée du PWM [s]
3.5	70.82	1.4
4.2	65	1.4

Tableau 9 PWM en fonction de la tension

Sur le tableau ci-dessus il est possible de constater que le rapport cyclique mesuré correspond parfaitement à ce qui a été calculé lors du point 2.7.5.

## 7.2 UART RFID

### 7.2.1 Activation du polling

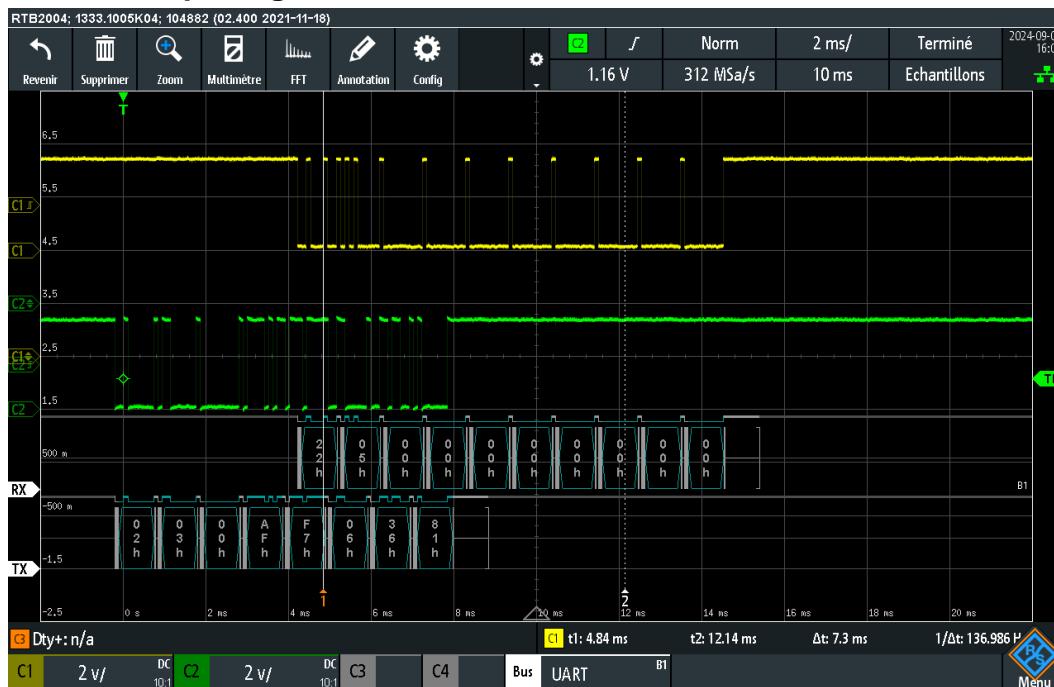


Figure 55 Trame UART polling module RFID

Sur la capture d'écran ci-dessus nous constatons que les paramètres permettant d'activer le polling définis dans la fonction CHU\_EnablePolling (voir Annexe G).

### 7.2.2 UID

La mesure ci-dessous montre la trame envoyée par le module RFID lorsqu'un badge est approché. Celle-ci contient l'UID du badge devant être lu par le microcontrôleur.



Figure 56 Trame UART UID RFID

### 7.3 I2C I/O expander

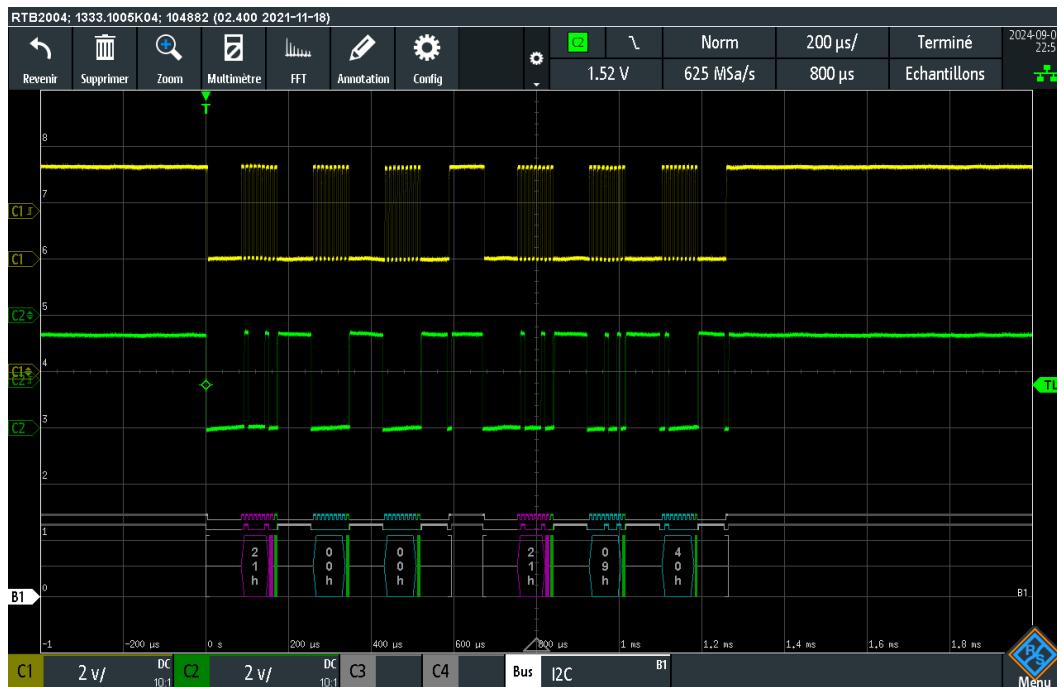


Figure 57 Trame I2C envoyée à l'I/O expander

La mesure ci-dessus montre la trame I2C transmise à l'I/O expander afin d'allumer les leds. Voici ce que signifient ces données :

0x21	0x00	0x00	0x21	0x09	0x40
Device opcode	IODIR address	IODIR outputs	Device opcode	GPIO address	GPIO states

Tableau 10 Explication de la trame

Plus d'explications sur le format de la trame sont disponibles au point 6.10.

## 8 Problèmes rencontrés

### 8.1 Hardware

#### 8.1.1 Connecteurs RFID et gâche

Lors de l'assemblage de mon PCB, je me suis rendu compte que les cosses prévues afin de câbler la gâche et le module RFID ne rentraient pas dans les connecteurs prévus à cet effet. J'ai donc directement soudé les fils de ces différents éléments au circuit

#### 8.1.2 Support d'accumulateurs

En essayant de mettre les accumulateurs dans le support j'ai remarqué que celui-ci était légèrement trop petit par rapport aux batteries. Cela n'est pas dû à une erreur de ma part car celui-ci est normalement prévu pour les accumulateurs que j'ai en ma possession car ils correspondent au format 18650.

### 8.2 Fréquence des badges

Lors de la phase de pré-étude du projet au point 2.5 j'ai déterminé la fréquence de la porteuse des badges fournis à l'aide d'un lecteur RFID de l'ES. Ceux-ci étant illisibles par le lecteur 150 [kHz], nous avons convenu avec M.Bovey que la fréquence de la porteuse des badges était donc de 13.56[MHz] et j'ai sélectionné le module RFID en conséquence. Cependant lors de la phase de programmation, je me suis rendu compte que les badges étaient toujours illisibles. J'ai donc procédé à toutes mes mesures à l'aide d'un badge 13.56[MHz].

### 8.3 Software

#### 8.3.1 Lecture de l'UID du badge

Malgré de nombreuses tentatives, je n'ai malheureusement pas réussi à lire l'UID des badges via mon software. J'ai uniquement réussi à lire la première moitié de la trame à cause d'un problème lié à la fifo du microcontrôleur. Je ne suis pas parvenu à régler ce bug dans le temps imparti, j'ai donc simulé le fonctionnement du système en validant les badges peu importe leur identifiant.

#### 8.3.2 Communication Wi-Fi

La communication Wi-Fi n'a pas encore été implémentée. Le module ESP32 a cependant été programmé avec succès.

#### 8.3.3 Ouverture de la gâche

A cause d'une erreur de conception hardware de ma part (voir point 4.1) et d'un retard de la commande Digikey contenant le MOSFET canal P choisi pour remplacer l'ancien transistor, j'ai pu tester la fonction d'ouverture de la gâche uniquement à la fin du projet, ce qui ne m'a pas laissé le temps de corriger le problème lorsque je me suis rendu compte que cela ne fonctionne pas.

## 9 État actuel du projet

### 9.1.1 Tâches effectuées

Voici les tâches du cahier des charges qui ont été effectuées durant ces 5 semaines :

- Pré-étude du projet et choix des composants
- Conception du schéma électrique
- Conception du PCB en se basant sur le schéma électrique
- Assemblage du PBC
- Corrections Hardware
- Configuration des timers / bus de données / OC
- Fonction permettant de communiquer avec l'I/O expander

- Fonctions permettant de communiquer avec le module RFID
- Fonction permettant de lire la tension aux bornes de la batterie
- Fonction générant un PWM afin d'ouvrir la gâche
- Programmation du module Wi-Fi ESP32
- Conception d'un boîtier

### 9.1.2 Tâches restantes

Cependant je n'ai malheureusement pas réussi à réaliser toutes les tâches demandées. Voici les tâches qu'il reste à faire afin d'être entièrement conforme au cahier des charges :

- Récupérer l'UID du badge
- Communiquer via Wi-Fi avec un serveur afin vérifier l'identité du badge
- Corriger l'ouverture de la gâche qui n'est pour l'instant pas fonctionnelle

## 10 Analyse de durabilité

### 10.1 Social

Le projet venant d'une demande spécifique de l'association Fixme, il paraît évident que celui-ci répond à un besoin de sécurité du frigo au sein du hackerspace. L'installer au sein de Fixme permettrait d'avoir un contrôle sur les ouvertures de la porte et ainsi d'empêcher quiconque ne serait pas autorisé à se servir de celui-ci. Le boîtier étant également facile à installer, son utilisation pourrait ne pas se limiter uniquement au frigo mais à toute porte ou tiroir nécessitant un contrôle d'accès, ce qui permettrait de renforcer le sentiment de sécurité au sein du groupe.

### 10.2 Économique

Economiquement le projet est dans la moyenne des systèmes de sécurité trouvables sur internet. S'il venait à être produit en masse, le coût de production diminuerait énormément et celui-ci deviendrait donc beaucoup plus rentable. Il serait possible de réduire le coût d'un exemplaire en remplaçant le système d'ouverture de la gâche « DC-DC 12V » par un chip contenant une alimentation à découpage.

### 10.3 Environnement

D'un point de vue environnemental, mon projet inciterait les membres de l'association à ouvrir la porte du frigo uniquement en cas de nécessité car celle-ci serait moins facile d'accès, et pourrait donc potentiellement réduire la consommation d'électricité du frigo. Afin de réduire d'avantage les coûts environnementaux du système, il faudrait vérifier la provenance des composants utilisés afin de moins polluer lors du transport, et faire fabriquer le PCB au sein d'une entreprise Suisse.

## 11 Apports du travail de diplôme

Ce travail de diplôme comportait trois éléments nouveaux pour moi ; l'utilisation d'un module RFID, l'utilisation d'un module Wi-Fi et la communication Ethernet. Bien que j'aie eu l'occasion d'en apprendre beaucoup sur ces trois éléments dans le cadre de mes études à l'ES-ETML, il s'agissait de la première fois que je mettais cela en pratique. Le module RFID m'a d'ailleurs causé beaucoup de difficultés ce qui m'a empêché de me pencher plus sérieusement sur les communications Wi-Fi et Ethernet. Cependant cela reste tout de même une expérience très positive à titre personnel car le travail de diplôme m'a permis de mettre en pratique diverses compétences apprises durant ces deux années. Cette expérience me sera probablement très utile dans la suite de ma carrière professionnelle.

## 12 Conclusion

En conclusion, ce projet de commande d'ouverture de frigo via RFID a été un véritable défi pour moi mais également une grande source d'apprentissage. Il s'agissait de la première fois où la conception d'un projet de cette envergure en si peu de temps m'était confiée. Le projet n'est pas totalement terminé mais je suis tout de même satisfait du travail que j'ai réussi à fournir au cours de ces 5 dernières semaines.

Tout d'abord lors de la phase de pré-étude j'ai été confronté à beaucoup de concepts tels que le RFID ou l'Ethernet que j'ai eu la chance d'étudier mais que je n'avais jamais eu l'occasion d'utiliser lors d'un projet. J'ai donc énormément documenté en très peu de temps afin d'être certain de saisir toutes les subtilités de ces éléments afin de pouvoir choisir les composants justement.

Une fois que les composants principaux ont été sélectionnés j'ai réalisé le schéma électrique du projet ce qui m'a pris beaucoup plus de temps que prévu en raison de la complexité de celui-ci. J'ai tout de même réussi à faire très peu d'erreurs. Les principales erreurs ont été commises lors de la conception du bloc DC-DC 12V servant à ouvrir la gâche. En effet, j'aurais pu utiliser un chip avec une alimentation à découpage intégrée ce qui m'aurait fait gagner du temps. De plus j'ai utilisé un MOSFET canal N pour le système d'activation de celui-ci au lieu d'un canal P.

Le retard lors de la conception du schéma électrique s'est répercuté sur la conception hardware du PCB, ce qui a fait que j'ai commandé le circuit avec quelques jours de retard. Cependant il n'y a pas eu d'erreurs majeures lors de cette étape.

La partie software est la partie durant laquelle j'ai rencontré le plus de problèmes. Je reconnais avoir passé trop de temps sur des aspects secondaires du code tels que l'affichage LCD ou les LEDs et pas assez sur la communication Wi-Fi par exemple. J'ai également eu de grandes difficultés avec le module RFID, je n'ai malheureusement pas réussi à obtenir l'UID des badges malgré de nombreuses tentatives. Cependant le module est fonctionnel, le mode de polling s'enclenche bel et bien et la trame contenant l'ID est envoyée correctement.

Malgré ces difficultés, une partie du cahier des charges est tout de même remplie et l'état actuel du circuit devrait permettre d'obtenir un produit entièrement fonctionnel en apportant quelques changements au software.

Lausanne, le 08.09.2024

Antonio Do Carmo

**Annexe A Cahier des charges**

**Annexe B Planification**

**Annexe C Journal de travail**

**Annexe D Procès-verbaux**

**Annexe E Schémas électriques**

**Annexe F Fichiers de fabrication**

**Annexe G Software**

**Annexe H Schéma de mesure**

**Annexe I Feuille de modifications**

**Annexe J Mode d'emploi**

**Annexe K Boîtier plans d'assemblage**

**Annexe L Mesures**

les écoles supérieures

# CAHIER DES CHARGES

ETML-ES

## DIPLOME

### Commande Ouverture Frigo

N° projet Diplôme 2407

#### Mandataire

Entreprise/Client:	Fixme	Département:	SLO
Demandé par (Prénom, Nom):	Ph. Bovey	Date:	31.07.2024

## 1 Objectif - Cahier des charges

Le but de ce travail de diplôme est de réaliser une commande d'ouverture de frigo, à l'aide d'un badge **RFID**<sup>1</sup>, qui pilotera une gâche magnétique. Le système devra pouvoir communiquer avec une base de données (*non incluse dans ce diplôme*), qui informera si le badge lu, correspond à un membre de l'association de **Fixme**<sup>2</sup> ; si oui, l'ouverture du frigo sera autorisée ; pour cela il faudra implémenter une communication soit à l'aide d'un module **ESP32**<sup>3</sup> (WiFi), soit à l'aide d'une communication Ethernet.

Le système électronique sera équipé d'un petit affichage permettant de visualiser l'ID du badge RFID, voir par la suite, d'afficher d'autres messages (indication) comme le nom du membre de l'association, autorisation, etc... . Un système de leds, sera mis en place pour savoir si le frigo est ouvert / fermé, lecture de la carte RFID (lecture – permission), indication lumineuse si batterie trop faible.

Le diplômant devra étudier si le système électronique peut être alimenté par accumulateurs (commande de la gâche, communication, affichage, ...), avec la contrainte suivante : que le système devrait être alimenté au minimum pendant une semaine de manière automne (alimentation par batteries).

Après études et décision :

- si la réponse est oui : le diplômant devra mettre en place un système de recharge via port USB ou connecteur basse tension pour bloc secteur. Lorsque la batterie sera en dessous d'un certain seuil (à définir), un message sera envoyé soit par Ethernet si cette connexion est active, ou par Wifi pour indiquer que le système doit être rechargeé.
- si la réponse est non : le diplômant devra trouver un bloc secteur adéquat (tension et ampérage), et prévoir les coupures d'électricité, c'est-à-dire un système sur accu ou pile, permettant d'envoyer une alerte pour indiquer qu'il y a une coupure d'électricité par Wifi (ESP32).

Quelque soit la solution choisie : le système électronique devra pouvoir se mettre en veille, et être réveillé par la lecture du badge RFID.

<sup>1</sup> RFID :

Radio Frequency Identification

<sup>2</sup> Fixme – hackerspace :

<https://fixme.ch/>

<sup>3</sup> ESP32 :

<https://www.espressif.com/en/products/modules>





Le diplômant prendra aussi en compte que le système électronique pourra évoluer par la suite, avec l'ajout de module(s) supplémentaire(s) (I/O, communication(s) série(s), etc...).

Le système développé devra être fixé sur la porte du Frigo, pour cela prévoir des magnets (petits aimants) en guise de pieds de fixation.

Pour simuler le frigo, une glacière portative fera office de Frigo.

#### **Partie Hardware**

Le diplômant devra concevoir une carte électronique basé sur un microcontrôleur de son choix ; cette carte doit contenir au minimum :

- Un microcontrôleur
- Alimentation => étude de cas (par accus ou par bloc secteur)
- Gâche (magnétique et/ou électrique)
- Lecteur RFID
- Affichage
- Indications lumineuses – Leds
- Module RF ESP32 pour transmission de données
- Module Ethernet pour transmission de données
- ...

Il devra soit choisir un boîtier du commerce et l'adapter (design des ouvertures pour leds – affichage – emplacement lecteur RFID, accus, etc...) ou le designer lui-même.

**Attention :** prévoir des emplacements pour des magnets (aimant) permettant de fixer le module sur la porte du Frigo.

#### **Partie Firmware**

L'étudiant réalisera un Firmware (partie microcontrôleur) permettant de faire au minimum les tâches suivantes :

- Lecture Badge RFID
- Commande de la gâche
- Communication
  - uC et module RFID
    - Récupération ID du badge RFID
  - uC et module ESP
    - envoi et réception de datas (ID badge, validation [Ok – Not Ok], ...)
  - uC et module Ethernet
    - envoi et réception de datas (ID badge, validation [Ok – Not Ok], ...)
  - uC et affichage
    - communication
    - envoi de data (ouverture porte – ID badge RFID, ...)
- Communication UART
  - Configuration module ESP
- Indication lumineuse
  - Ouverture porte frigo
  - Lecture Badge
  - Validation Badge
  - Communication Wifi
  - ...



Le diplômant devra aussi configurer le module ESP32 :

- Wifi – mode client

#### **Partie Software**

Le diplômant devra garantir la réception et la transmission de donnée sur :

- Module ESP (Wifi – mode client - ...)
- Configuration d'un routeur Wifi



### 1.1 Données en lien avec l'objectif – les grandes lignes

- Recherche et implémentation de solutions **Hardware** :
  - Microcontrôleur (préférence fabriquant Microchip ou fabriquant ST Electronics)
  - Alimentation – Etude de cas / recherche de solution :
    - Par Accumulateur
      - Type accumulateur / nb de cellules / ...
      - Puissance à fournir / durée / ...
      - Système de recharge pour accumulateur
    - Par Bloc Secteur
      - Tension et ampérage adéquat
      - Connecteur
      - ...
  - Attention :** les tensions subsidiaires au montage (interne) devront être réalisées par des alimentations à découpage DC-DC
  - Lecteur RFID
    - Analyse Fréquence badge
    - Choix module
    - Communication avec le uC
    - ...
  - Gâche
    - Magnétique ou Electrique
    - Système de commande de la gâche
    - ...
  - Module de communication ESP32
    - Chip ou module
    - Paramétrage du module (boot)
    - Communication – Configuration – UART
      - Avec le uC
      - Extérieur – configuration
    - ...
  - Module de communication Ethernet
    - Chip ou module
    - Communication avec le uC
    - ...
  - Affichage
    - Alphanumérique
    - Minimum 2 lignes
    - Communication avec le uC
    - ...
  - Led(s) – indication lumineuse
    - Ouverture du frigo
    - Lecture badge RFID
    - Etat batterie
    - Autorisation (badge reconnu – badge non reconnu)
    - Wifi Connecté
    - ...
- Recherche de composants électroniques autour des composants décrits ci-dessus : AOP, résistances, condensateur, self, autres
- Réalisation de schématique(s) complet(s) et d'un PCB - à réaliser sous ALTIUM de préférence
- Recherche (achat) ou réalisation d'un boîtier sous SolidWorks



- Recherche et implémentation de solutions au niveau **Firmware**
  - Communication entre uC et module lecteur RFID
    - Choix protocole de communication (SPI, I2C, UART, ...)
    - Lecture ID – Badge RFID
    - ...
  - Commande gâche
    - Configuration I/O
    - ...
  - Communication entre le uC et l'ESP32
    - Choix protocole de communication (SPI, I2C, UART, ...)
    - Configuration trames
      - Longueur trame – début et fin trame – taux de transfert
      - Datas (ID badge – message texte (ok – not ok), ...)
      - ...
  - Communication entre le uC et module Ethernet
    - Choix protocole de communication (SPI, I2C, UART, ...)
    - Configuration trames
      - Longueur trame – début et fin trame – taux de transfert
      - Datas : ID badge – message texte (ok – not ok), ...
      - ...
  - Communication entre le uC et l'affichage
    - Choix protocole de communication (SPI, I2C, UART, ...)
    - Datas : ID badge – message texte (ok – not ok), ...
    - ...
  - Configuration module ESP32 :
    - Wifi
      - Mode client
      - SSID
      - Password
      - Protocole communication => envoi de datas
      - Datas ID badge – message texte (ok – not ok), ...
      - ...
  - Commande de leds
    - Configuration I/O
      - Ouverture de la porte du Frigo
      - Réception ID du badge RFID
      - Validation du badge RFID – membre ou non
      - Connection Wifi
      - Etat batterie
      - ...
  - ...
- Recherche et implémentation de solutions au niveau **Software**
  - Réception et émission de datas (ID badge RFID – Validation Membre Fixme)
    - Wifi
    - Ethernet
- Démontrer par différentes simulations et mesures que les parties Hardware, Firmware et Software ont été bien implémentées.



## 2 A l'issue du projet de diplôme, l'étudiant fournira (liste non exhaustive) :

- Fichiers sources de CAO électronique du PCB réalisé (ALTIUM) + configuration logiciel (version utilisées)
- Tout le nécessaire pour fabriquer un exemplaire hardware : fichiers de fabrication (GERBER) / liste de pièces avec références pour commande (BOM) / implantation (prototype) / modifications, etc
- Fichiers sources de programmation microcontrôleur (.c / .h) + configuration logiciel (version utilisées)
- Fichiers sources d'application externe au système électronique + configuration logiciel (version utilisées)
- Tout le nécessaire pour programmer le microcontrôleur (logiciel ou fichiers .hex), sous format numérique => utilisation de la structure de projet fourni par l'ES
- Tout le nécessaire pour programmer des applications (logiciel ou fichiers .hex) sous format numérique => utilisation de la structure de projet fourni par l'ES
- Tout le nécessaire à l'installation de programmes sur PC ou autres environnements utilisés durant le travail de diplômes
- Un rapport de diplôme contenant :
  - Les concepts du design **Hardware**
    - Études des différents systèmes à implémenter (explications)
    - Choix des composants et dimensionnement de ceux-ci (calculs / simulation / autre)
    - Réalisation schématique / PCB / boitier / montage de la carte
  - Les concepts du design **Firmware**
    - Structogramme / flowchart / Pseudocode
    - Explication des algorithmes mise en place
    - Démonstration par calculs / outils de debug / des résultats obtenus
    - Validation des concepts mis en place
  - Les concepts du design **Software**
    - Structogramme / flowchart / Pseudocode
    - Explication des algorithmes mise en place
    - Démonstration par calculs / outils de debug / des résultats obtenus
    - Validation des concepts mis en place
  - Tests & Validation :
    - Méthodologie de tests
    - Mesure(s)
    - Validation des résultats
    - Correction(s) apportée(s) au design (Hardware / Firmware / Software)
  - Estimation des coûts pour le design développé (un prototype)
  - Etat d'avancement & problème(s) rencontré(s)
  - Conclusion
  - Bibliographie / webographie / autre sources
- Les annexes :
  - Calculs détaillés des concepts
  - Listings complets des parties Firmware & Hardware que vous avez implémenté
  - Schématique + plan d'implémentation complète du PCB
  - Dessin / schématique du boitier
  - Mesures



- Pages utilisée des différents datasheets ou documentations exploités
- Mode d'emploi du système développé pendant le diplôme
  - Journal de travail
  - PV de séances hebdomadaires
  - Un prototype

### **3 Autres demandes / contraintes / conseils**

- Planifier dans le détail les travaux demandés.
- Se référer au planning régulièrement, vérifier son avancement, rédiger son journal de projet quotidiennement.
- Commencer à rédiger le rapport de diplôme le plus tôt possible, et régulièrement tout au long du travail de diplôme.
- Prendre du temps, préparer sa réflexion, rechercher des apports théoriques et des exemples pratiques, envisager plusieurs possibilités avant de finaliser une solution.
- Numéroter et dater tous les documents
- En cas de problème (retard, objectif à revoir, difficulté rencontrée, etc.), se référer à l'enseignant et au mandataire au plus vite.
- Toutes les décisions importantes, tant au niveau technique qu'organisationnel, doivent être posées par écrit dans le PV de séance, le rapport de diplôme et /ou figurer dans le journal de projet, après discussion avec l'enseignant / le mandataire.

### **4 Documents de références**

#### Vidéo

- Envoi de donnée via ESP32 (wifi)
- <https://www.youtube.com/watch?v=13Crtvr85IE>

#### Pages web :

- Articles

#### Réseau Wifi et ESP32 :

[https://www.wikidebrouillard.org/wiki/Configurez\\_le\\_r%C3%A9seau\\_Wifi\\_sur\\_un\\_ESP](https://www.wikidebrouillard.org/wiki/Configurez_le_r%C3%A9seau_Wifi_sur_un_ESP)

- Documentation sur l'ESP32

#### Wifi :

[https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/network/esp\\_wifi.html](https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/network/esp_wifi.html)

- Exemple de gâches

<https://www.opo.ch/fr/b/assa-abloy?position=4>  
<https://www.opo.ch/fr/contacts-magnetiques-u-appliques/p/bp6617601.html?listerPage=BRAND&brand=assa-abloy&position=4>



<https://www.opo.ch/fr/contacts-magnetiques-dmcm-20u/p/bp6617504.html?listerPage=BRAND&brand=assa-abloy&position=1>

- **Exemple lecteur RFID**

<http://duinoedu.com/store1/-san-fil/446-dlid.html>

- **Exemple module Ethernet**

<https://www.az-delivery.de/fr/products/enc28j60-netzwerkmodul>

-

Date	Début	Fin	Durée	Tâche	Détails	Problèmes	Solutions
05.08.2024	10h00	11h15	1h15	Présentation travail de diplôme			
05.08.2024	11h15	11h50	0h35	Lecture de la documentation	Cahier des charges, grille d'évaluation, etc		
05.08.2024	11h50	12h00	0h10	Préparation des fichiers	Création du projet Altium		
05.08.2024	12h50	13h50	1h00	Préparation des fichiers	Planification, journal de travail, rapport		
05.08.2024	13h50	14h10	0h20	Discussion avec M. Bovey	Precisions sur le cahier des charges + questions	[1] Possible d'avoir le matériel du TP RFID? [2] Mise à disposition de la gaciérie? [3] Présence des enseignants? [4] Quand est la 1ère séance, que préparer?	
05.08.2024	14h10	14h20	0h10	Préparation des fichiers	Planification		
05.08.2024	14h20	14h40	0h20	Séance sur la création des fichiers Altium	Par M. Moreno		
05.08.2024	14h40	15h30	0h50	Préparation des fichiers	Planification		
05.08.2024	15h30	19h00	3h30	Pré-étude	Choix de la gâche	Les gâches magnétiques classiques nécessitent une grande puissance et sont en fonctionnement permanent => Pas possible de les faire fonctionner sur batterie et facilement bypassable en débranchant le boîtier du secteur	J'ai trouvé ce modèle de gâche
05.08.2024	19h00	20h00	1h00	Pré-étude	Recherche de modules Wi-Fi et Ethernet		
06.08.2024	8h40	9h40	1h00	Pré-étude	Lecture de la documentation sur le RFID		
06.08.2024	9h40	10h00	0h20	Pré-étude	Lecture du rapport du projet 2312		
06.08.2024	10h00	10h20	0h20	Pré-étude	Recherche d'un affichage		
06.08.2024	10h20	11h30	0h50	Pré-étude	Tentative de lecture des badges RFID fournis	Impossible de les lire avec le kit de l'ES et avec des applications dédiées sur smartphone	
06.08.2024	12h30	14h10	1h40	Rapport de projet	Rédaction du rapport avec les composants choisis		
06.08.2024	14h10	14h30	0h20	Pré-étude	Tentative de lecture des badges RFID fournis	A l'aide de l'application Tag Reader j'ai pu lire un des deux badges fournis	
06.08.2024	14h30	15h30	1h00	Rapport de projet	Rédaction du rapport avec les composants choisis		
06.08.2024	15h30	16h10	0h40	Préparation des fichiers	Préparation du word pour le PV du landomain		
06.08.2024	16h10	17h00	0h50	Rapport de projet	Rédaction du rapport avec les composants choisis		
07.08.2024	8h30	10h00	1h30	Rapport de projet	Rédaction de l'introduction du rapport		
07.08.2024	10h00	10h40	0h40	Séance avec M. Bovey	Séance de pré-étude	Voir PV de la séance	
07.08.2024	10h40	11h00	0h20	Réécriture du PV de la séance	PV du 07.08.2024		
07.08.2024	11h00	11h30	0h30	Rapport de projet	Tableau comparatif des gâches		
07.08.2024	13h00	13h20	0h20	Réécriture du PV de la séance	Corrections du PV		
07.08.2024	13h20	16h30	3h10	Rapport de projet	Réécriture d'un nouveau LCD	Le LCD choisi précédemment doit être alimenté en 5V mais tout le reste de mon circuit fonctionne en 3.3V	J'ai trouvé ce modèle de LCD
07.08.2024	16h30	17h20	0h50	Pré-étude	Réécriture d'un nouveau LCD		
07.08.2024	17h20	18h20	1h00	Rapport de projet	Réécriture des chapitres sur les différents		

07.08.2024	18h20	19h00	0h40	Pré-étude	Ajout du système de recharge et de lecture de l'accumulateur sur le schéma bloc
08.08.2024	8h45	10h00	1h15	Pré-étude	Choix du microcontrôleur
08.08.2024	10h00	11h30	1h30	Pré-étude	Chronogramme ouverture
08.08.2024	12h30	14h10	1h50	Pré-étude	Estimation de la consommation
08.08.2024	14h10	14h50	0h40	Schéma électrique	Début du schéma électrique
08.08.2024	14h50	15h00	0h10	Discussion avec M.Bovey	Au sujet de l'estimation de la consommation
08.08.2024	15h00	18h00	3h00	Schéma électrique	Blocs microcontrôleur, Alimentation, Affichage
09.08.2024	8h40	11h30	2h50	Schéma électrique	Footprints + modèles 3D
09.08.2024	12h30	13h40	1h10	Schéma électrique	Footprints + modèles 3D
09.08.2024	13h40	16h30	2h50	Schéma électrique	
11.08.2024	15h00	19h00	4h00	Rapport de projet	Rédaction des chapitres sur les différents rapports de projets
12.08.2024	8h40	9h10	0h30	Revue de schéma	Rédaction des chapitres sur les différents rapports de projets
12.08.2024	9h10	10h40	1h30	Revue de schéma	Rédaction des chapitres sur les différents rapports de projets
12.08.2024	10h40	11h40	1h00	Schéma électrique	Rédaction des chapitres sur les différents rapports de projets
12.08.2024	12h30	20h30	8h00	Schéma électrique	Rédaction des chapitres sur les différents rapports de projets
13.08.2024	8h40	9h40	1h	Gâche	Finalisation des blocs Wifi, Ethernet, Affichage, Alimentation
13.08.2024	9h40	11h30	1h50	Schéma électrique	Test de tension et courant sur la gâche
13.08.2024	12h30	16h30	4h00	Schéma électrique	Finalisation des blocs Wifi, Ethernet, Affichage, Alimentation
14.08.2024	8h40	10h00	1h20	Schéma électrique	Pas assez de pins sur le uC
14.08.2024	10h00	11h00	1h00	Pré-étude	Voir PV
14.08.2024	11h00	11h30	0h30	Schéma électrique	J'ai trouvé ce modèle.
14.08.2024	12h20	14h30	1h50	Schéma électrique	
14.08.2024	14h30	16h00	1h30	Schéma électrique	
14.08.2024	16h00	18h00	2h00	Commande des composants	Vérification du schéma avec JAR
15.08.2024	8h40	10h10	1h30	Schéma électrique	Ajout des composants sur saphir
15.08.2024	10h10	11h30	1h20	Design du PCB	Corrections du schéma
15.08.2024	12h30	16h00	3h00	Design du PCB	Début du layout du PCB
15.08.2024	16h00	16h20	0h20	Discussion avec M.Bovey	Layout du PCB
15.08.2024	16h20	19h10	2h50	Design du PCB	Fin du layout PCB
16.08.2024	8h40	11h30	2h50	Design du PCB	Début du routage
16.08.2024	12h30	18h00	5h30	Design du PCB	Routage du PCB
17.08.2024	13h10	16h10	3h00	Design du PCB	Routage du PCB
19.08.2024	8h40	10h00	1h20	Design du PCB	Revue du PCB
19.08.2024	10h00	10h20	0h20	Discussion avec M.Bovey	Revue du PCB
19.08.2024	10h20	11h30	1h10	Design du PCB	Routage du PCB
19.08.2024	12h30	19h00	6h30	Design du PCB	Routage du PCB
19.08.2024	21h00	23h00	2h00	Design du PCB	Routage du PCB
20.08.2024	8h30	10h00	1h30	Design du PCB	Finalisation du PCB
20.08.2024	10h00	10h30	0h30	Discussion avec M.Morales	Revue du PCB
20.08.2024	10h30	11h00	0h30	Design du PCB	Finalisation du PCB et commande
21.08.2024	8h30	11h30	3h00	Software	Pinning et configuration

21.08.2024	12h30	16h30	4h00	Software	Lecture de la batterie
22.08.2024	8h30	11h30	3h00	Software	Lecture de la batterie
22.08.2024	12h30	16h30	4h00	Software	Fonction de lecture de la batterie et d'ouverture de la gâche
23.08.2024	8h30	11h30	3h00	Software	Lecture de la batterie et d'ouverture de la gâche
23.08.2024	12h30	16h30	4h00	Software	Lecture de la doc et des librairies
23.08.2024	8h00	9h20	1h20	Schema électrique	Lecture de la doc et des librairies
23.08.2024	9h20	9h40	0h20	Visite de l'expert	Avancement sur le rapport de diplôme
23.08.2024	9h40	11h30	1h50	Software	Séance avec M.Versin
23.08.2024	12h30	16h30	4h00	Software	Fonction de lecture de la batterie et d'ouverture de la gâche
26.08.2024	8h30	11h30	3h00	Hardware	Assemblage du PCB
26.08.2024	12h30	13h00	0h30	Mesures	Test de l'alimentation
26.08.2024	13h00	20h00	7h00	Hardware	Assemblage du PCB
27.08.2024	8h30	11h30	3h00	Hardware	Assemblage du PCB
27.08.2024	12h30	17h30	5h00	Hardware	Assemblage du PCB
28.08.2024	8h30	9h30	0h30	Software	Tests programmation sur la carte
28.08.2024	9h30	10h00	0h30	Discussion avec M.Bovey	Séance rapide
28.08.2024	10h00	11h30	1h30	Software	Fonction de lecture de la batterie et d'ouverture de la gâche
28.08.2024	12h30	17h00	4h30	Software	Fonction de lecture de la batterie et d'ouverture de la gâche
29.08.2024	8h30	11h30	3h00	Hardware	Câblage du LCD
29.08.2024	12h30	17h30	5h00	Software	Fonction pour le LCD
30.08.2024	8h30	11h30	3h00	Software	Fonction pour l'I/O Expander
30.08.2024	12h30	16h00	3h30	Software	Fonction pour l'I/O Expander
30.08.2024	16h00	16h30	0h30	Discussion avec M.Bovey	Séance classique
30.08.2024	16h30	17h30	1h00	Software	Fonction pour l'I/O Expander
01.09.2024	8h30	9h30	1h00	Software	Fonction pour l'I/O Expander
01.09.2024	9h30	10h00	0h30	Discussion avec M.Bovey	Séance rapide
02.09.2024	10h00	11h30	1h30	Software	Fonction pour le RFID
02.09.2024	12h30	18h00	5h30	Software	Fonction pour le RFID
03.09.2024	8h30	11h30	3h00	Software	Fonction pour le RFID
03.09.2024	12h30	22h00	9h30	Rapport de projet	Fonction pour le RFID
04.09.2024	8h30	11h30	3h00	Software	Suite du rapport
04.09.2024	12h30	16h00	3h30	Software	Fonction pour le RFID
04.09.2024	16h00	16h30	0h30	Discussion avec M.Bovey	Séance rapide
04.09.2024	16h30	22h00	5h30	Software	Fonction pour le RFID
05.09.2024	8h30	11h30	3h00	Software	Programmation de l'ESP32
05.09.2024	12h30	18h00	5h30	Software	Fonction pour le RFID
05.09.2024	18h00	22h00	4h00	Rapport de projet	Suite du rapport
06.09.2024	8h30	11h30	3h00	Modifications	Modifications DC-DC12V et assemblage boîtier

				Software		Fonction pour le RFID
06.09.2024	12h30	17h30	5h00		Mesures	Mesures des différentes trames
06.09.2024	17h30	18h30	1h00		Rapport de projet	Suite du rapport
06.09.2024	18h30	22h30	4h00		Rapport de projet	Suite du rapport
07.09.2024	13h00	0h00	11h00		Rapport de projet	Suite du rapport
07.09.2024	00h00	2h00			Rapport de projet	Suite du rapport
08.09.2024	13h00	0h00	11h00		Rapport de projet	Suite du rapport
09.09.2024	0h00	4h30	4h30		Rapport de projet	Fin du rapport



## PV de séance

### Procès-verbal du 07.08.2024

#### Présents

- M. Bovey (PBY)
- M. Do Carmo (ADC)

#### État des lieux

- Avancement phase pré-étude du projet :
  - Sélection composants principaux : gâche, module Wi-Fi, module Ethernet, affichage, module RFID.
  - Recherche documentations techniques sur les différents éléments cité ci-dessus
- Rédaction rapport : chapitres traités : introduction - pré-étude.

#### Problèmes rencontrés

- Alimentation :
  - Batterie inutilisable ? Pour des gâches magnétiques à contact
    - Consommation en courant élevée pour maintenir le champ électromagnétique
- Gâche :
  - Modèle adéquat ?
  - Magnétique ?
  - Electromagnétique ?
- Badge RFID :
  - Fréquence porteuse badge ?
    - 150kHz ? 13.56MHz ? information sur badge officiel Fixme ?
    - Selon réponse : deux modèles de lecteur
- Validation des composants principaux avec le client (PBY)) => Go – No Go
  - Gâche
  - Lecteur RFID
  - Affichage LCD 2 lignes – 16 segments
  - Contrôleur Ethernet
  - Module ESP

#### Solutions proposées

- Alimentation :
  - Si autre gâche : possibilité utilisation batterie
    - Ne nécessite pas un courant constant pour maintenir le frigo fermé
    - Nécessite uniquement un pic de courant de 1,4[A] durant max 5[s] pour ouvrir la porte.
    - Tension d'alimentation pouvant descendre jusqu'à 3[V].



- Gâche
  - [Modèle magnétique](#) : pas adéquat – garantir courant permanent
  - [Modèle Electromagnétique](#) : travail en impulsion juste pour l'ouverture
- Badge RFID
  - Besoin du badge officiel Fixme – pour déterminer type de porteuse pour badge
- Validation composants :
  - Gâche : voir ci-dessus
  - Lecteur RFID : dépend test avec Badge Fixme
  - [Affichage LCD](#) 2 lignes – 16 segments
  - [Contrôleur Ethernet](#) – implémenter composants du module directement sur PCB (diminution des couts => ~22 chf pour le module VS ~13 chf pour les composants séparément)

### Décisions prises

- Alimentation :
  - Possibilité avec batterie – car gâche électrique validée
- Gâche :
  - [Modèle Electromagnétique](#) – validé (plus adéquat avec utilisation batterie)
- Badge RFID
  - Besoin du badge officiel Fixme pour déterminer type de porteuse pour badge
- Validation composants :
  - Gâche : PBY doit commander la gâche
  - Lecteur RFID : dépend test avec Badge Fixme
  - Affichage LCD => Ok pour modèle proposé
  - [Contrôleur Ethernet](#) => Ok pour solution proposée

### Suite du projet / objectifs - jusqu'au 12 août

- Alimentation :
  - Déterminer nb cellules à utiliser
  - Courant max
  - Courant moyen
  - Courant standby
  - Durée batterie
- Finaliser la pré-étude
  - Rédaction
  - Recherche documentation
- Sélectionner et commander tous les composants nécessaires au système
- Réaliser le schéma électrique de la carte

### Prochaine réunion:

12.08.2024, 9h00, Salle R110 ETML-ES

Revue du schéma électrique

14.08.2024, 14h00, Salle R110 ETML-ES

Séance classique

**Destinataires de ce PV**

- Grégoire Rossier, Doyen de l'ETML-ES
- Philippe Bovey, Maître de diplôme

Lausanne le 07.08.2024

Antonio Do Carmo

# PV de séance

## Procès-verbal du 12.08.2024

### Présents

- M. Bovey (PBY)
- M. Do Carmo (ADC)

### État des lieux

- Avancement phase pré-étude du projet :
  - Sélection composants principaux : alimentation, microcontrôleur
  - Recherche documentations techniques sur les différents éléments cité ci-dessus
- Rédaction rapport : chapitres traités : conception
- Consommation : Estimation de la consommation faite (90h d'autonomie avec une batterie 3400mAh)
- Avancement sur le schéma électrique : EthWifiRfid, Affichage, Alimentation, Microcontrôleur.

### Problèmes rencontrés

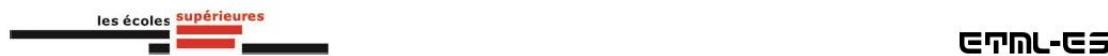
- Alimentation :
  - Recharge de l'accu via un USB-C ?
  - Difficultés à trouver un convertisseur DC-DC pour la gâche
  - Un accu classique ne suffit pas pour 1 semaine d'autonomie
- Affichage :
  - LCD seulement 2 lignes VS 4 dans le kit de l'ES
- Rapport :
  - Possibilité de fusionner la partie pré-étude et la partie schéma électrique ?

### Solutions proposées

- Alimentation :
  - Réaliser le convertisseur DC-DC directement sur le système
  - Utilisation de 2 accumulateurs
- Affichage :
  - Ce LCD choisi => utilisé dans d'autres projets et compatible avec les librairies du Kit ES

### Décisions prises

- Alimentation :
  - OK pour l'USB-C
  - Designer un convertisseur step-up sur le circuit



- Rapport :
  - OK pour la fusion des deux parties
- Revue schéma électrique :
  - Affichage :
    - Remplacer les 3 leds de la batterie par une RGB
    - LED bicolore pour la validation du badge
  - Alimentation :
    - Ajouter LED de statut MCP73871
    - Ajouter condensateur sur pin 2 MCP73871
    - Ajouter thermistance sur pin 5 MCP73871
    - Ajouter régulateur DC-DC 3,3V et 12V
  - EthWifiRfid :
    - Implémenter RFID et Ethernet
    - Ajouter connexion Wifi => uC

#### ***Suite du projet / objectifs - jusqu'au 14 août***

- Schéma électrique :
  - Finaliser les blocs : EthWifiRfid, Affichage, Alimentation, Microcontrôleur
- Ajouter à la commande tous les composants nécessaires au système

#### ***Prochaine réunion:***

14.08.2024, 10h00, Salle R110 ETML-ES

Séance classique

#### ***Destinataires de ce PV***

- Grégoire Rossier, Doyen de l'ETML-ES
- Philippe Bovey, Maître de diplôme

Lausanne le 14.08.2024

Antonio Do Carmo

# PV de séance

## Procès-verbal du 14.08.2024

### Présents

- M. Bovey (PBY)
- M. Do Carmo (ADC)

### État des lieux

- Revue de schéma effectuée Lundi 12.08.2024
- Schéma :
  - EthWifiRfid, Affichage, Alimentation finis => corrections apportées
- Tests effectués sur la gâche

### Problèmes rencontrés

- Microcontrôleur :
  - Pins I/O insuffisants

### Solutions proposées

- Microcontrôleur :
  - I/O expander ou connecter certaines LEDs à l'ESP32

### Décisions prises

- Microcontrôleur :
  - Ajouter un I/O expander au système
- Avancement :
  - Retard par rapport à la planification
    - Nouvelles deadlines pour le projet

### Suite du projet / objectifs - jusqu'au 21 août

- Finir le schéma électrique pour le 15.08.2024
- Faire la BOM et commander les composants pour le 15.08.2024
- Commencer et finir le design du PCB pour le 19.08.2024
- Commencer la conception du boîtier pour le 19.08.2024
- Finir le chapitre conception du rapport pour le 19.08.2024



- 

***Prochaine réunion:***

16.08.2024, 15h30, Salle R110 ETML-ES  
19.08.2024, 15h00, Salle R110 ETML-ES  
21.08.2024, 09h00, Salle R110 ETML-ES

Revue design PCB  
Finalisation PCB  
Séance classique

***Destinataires de ce PV***

- Grégoire Rossier, Doyen de l'ETML-ES
- Philippe Bovey, Maître de diplôme

Lausanne le 14.08.2024

Antonio Do Carmo

# PV de séance

## Procès-verbal du 21.08.2024

### Présents

- M. Bovey (PBY)
- M. Do Carmo (ADC)

### État des lieux

- Revue de PCB effectuée Lundi 19.08.2024
- Design du PCB fini :
  - Corrections sur le schéma (driver de gate ajouté + alim 3,3V corrigée)
- BOM finie
- Commande des composants effectuée
- Commande du PCB effectuée
- Rapport :
  - Pré-étude finie
  - Schéma presque fini

### Problèmes rencontrés

- Rapport :
  - Question sur le ratio pour la fin de charge de l'accu
  - Pourquoi des résistances 5k1 sur l'USB-C ? Norme ?

### Solutions proposées

### Décisions prises

- Rapport :
  - La tension de l'accu a atteint sa tension nominal - système de limitation du courant - car l'accu se charge que le courant décroît
  - Documentation sur l'USB-C trouvée

### Suite du projet / objectifs - jusqu'au 28 août

- Structogrammes et machine d'état
- Réaliser le pining du microcontrôleur
- Paramétrier les différents drivers
- Importer les librairies
- Commencer le code



**Prochaine réunion:**

23.08.2024, 11h30, Salle R110 ETML-ES  
28.08.2024, 09h00, Salle R110 ETML-ES

Revue code  
Séance classique

**Destinataires de ce PV**

- Grégoire Rossier, Doyen de l'ETML-ES
- Philippe Bovey, Maître de diplôme

Lausanne le 22.08.2024

Antonio Do Carmo

## PV de séance

### Procès-verbal du 30.08.2024

#### Présents

- M. Bovey (PBY)
- M. Do Carmo (ADC)

#### État des lieux

- Début du code :
  - Pining et configuration
  - Diagramme d'états
  - Fonction lecture de la batterie
  - Fonction ouverture de la gâche
- PCB reçu et assemblé
- Rapport :
  - Pré-étude finie
  - Schéma fini

#### Problèmes rencontrés

- Support d'accu pas adapté

#### Solutions proposées

- Commencer le soft en alimentant via l'alim de labo

#### Décisions prises

#### Suite du projet / objectifs - jusqu'au 4 septembre

- Structogrammes et amélioration machine d'état
- Finaliser le code (RFID, Wi-Fi, I/O Expander, ...)
- Concevoir le boîtier

#### Prochaine réunion :

02.09.2024, 11h30, Salle R110 ETML-ES  
04.08.2024, 09h00, Salle R110 ETML-ES

Revue code  
Séance classique



Ecole supérieure, école des métiers Lausanne  
Rue de Sébeillon 12  
CH-1004 Lausanne  
PV\_300824 .docx

[www.etml.ch](http://www.etml.ch)  
| 1 / 2 |

**Destinataires de ce PV**

- Grégoire Rossier, Doyen de l'ETML-ES
- Philippe Bovey, Maître de diplôme

Lausanne le 28.08.2024

Antonio Do Carmo

## PV de séance

**Procès-verbal du 05.09.2024**

### Présents

- M. Bovey (PBY)
- M. Do Carmo (ADC)

### État des lieux

- Code :
  - Communication MCP23008 finie
  - Début config Wi-Fi
  - Programmation RFID en cours
- Rapport :
  - Hardware fini à 90%
  - Modifications finies

### Problèmes rencontrés

- Badge Fixme non reconnu par le module RFID
- Blocage code RFID

### Solutions proposées

- Continuer le code avec le badge ETML-ES

### Décisions prises

### Suite du projet / objectifs - jusqu'au 9 septembre

- Finaliser le projet

### Prochaine réunion :

### Destinataires de ce PV

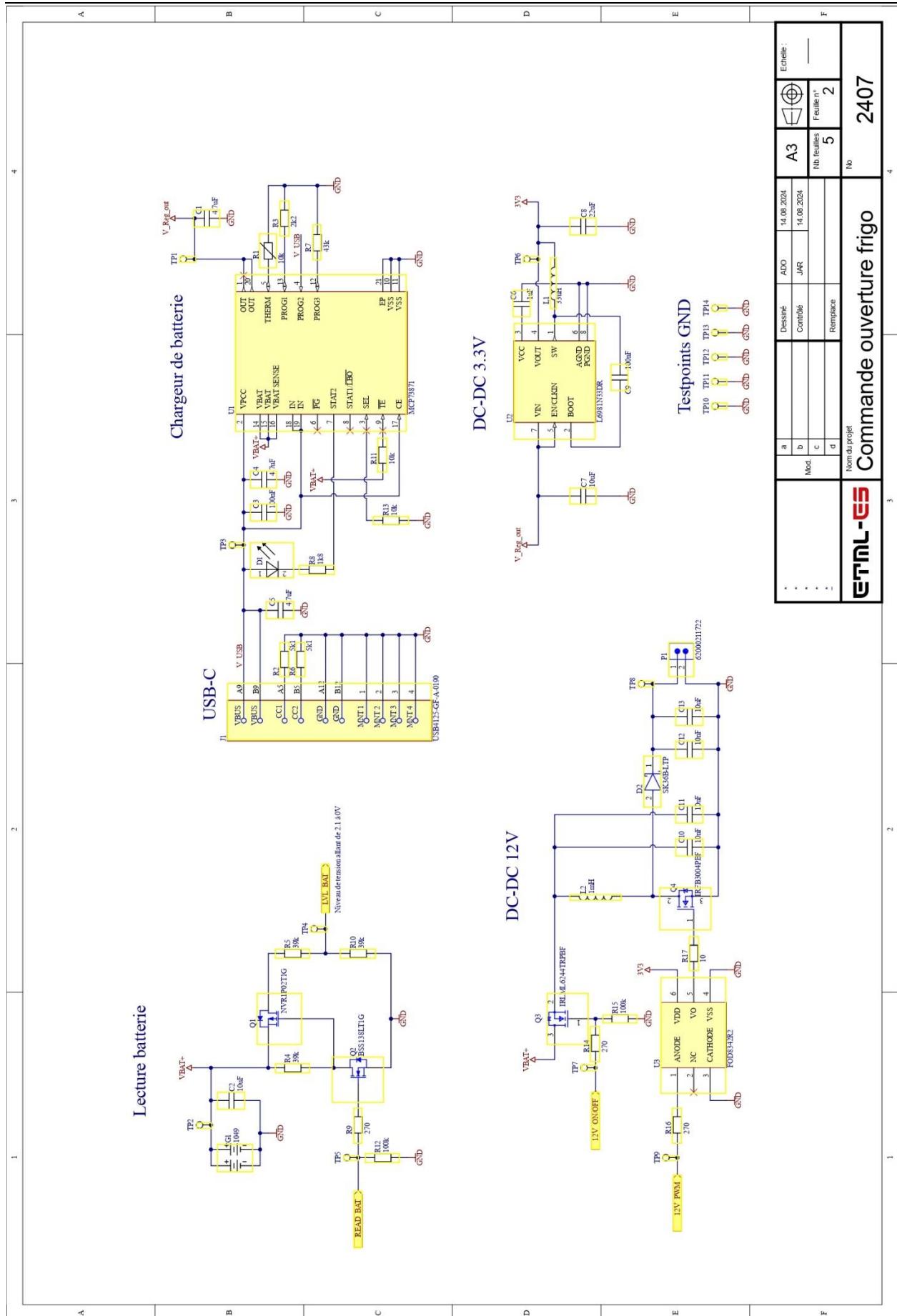
- Grégoire Rossier, Doyen de l'ETML-ES
- Philippe Bovey, Maître de diplôme

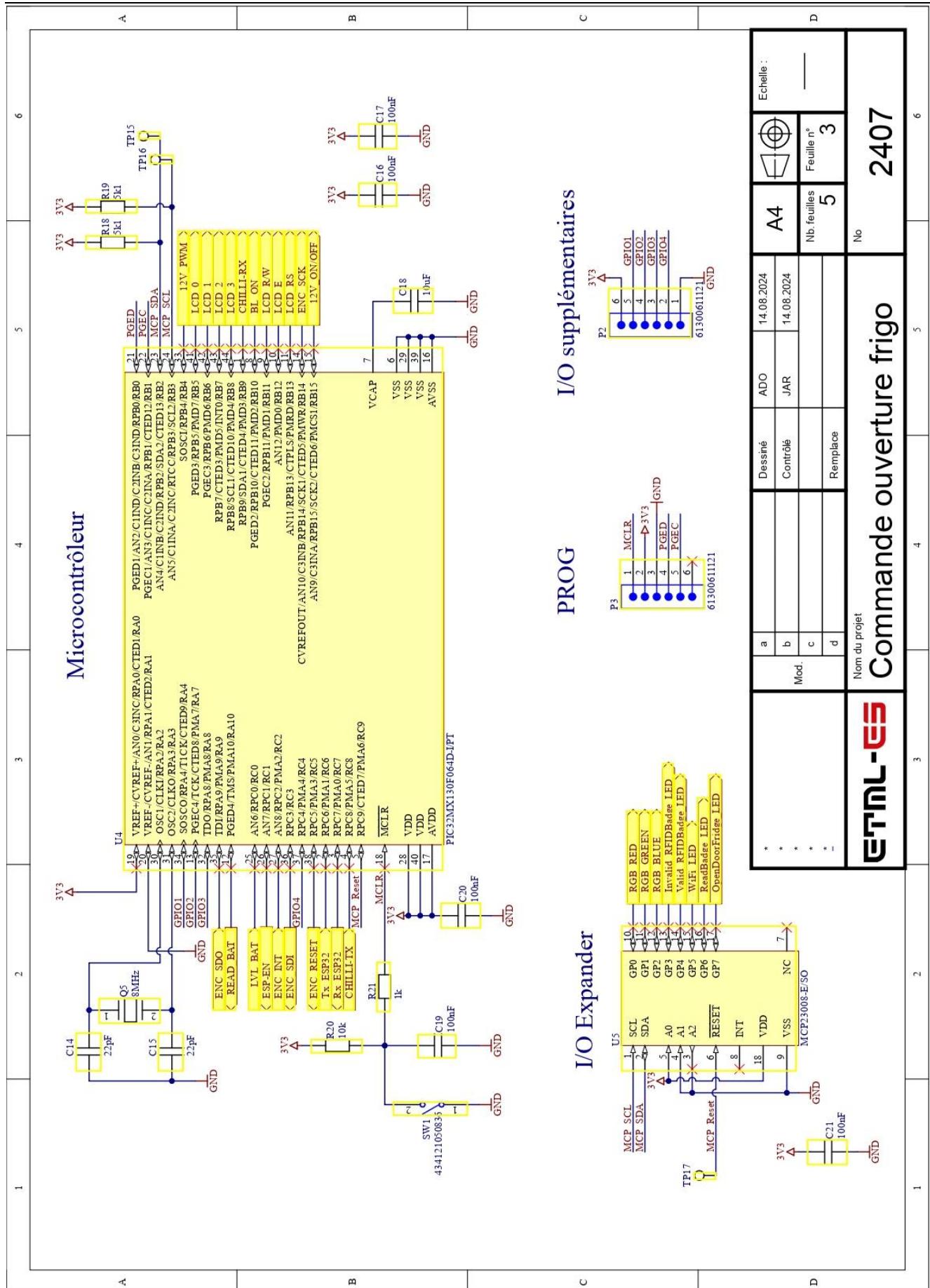


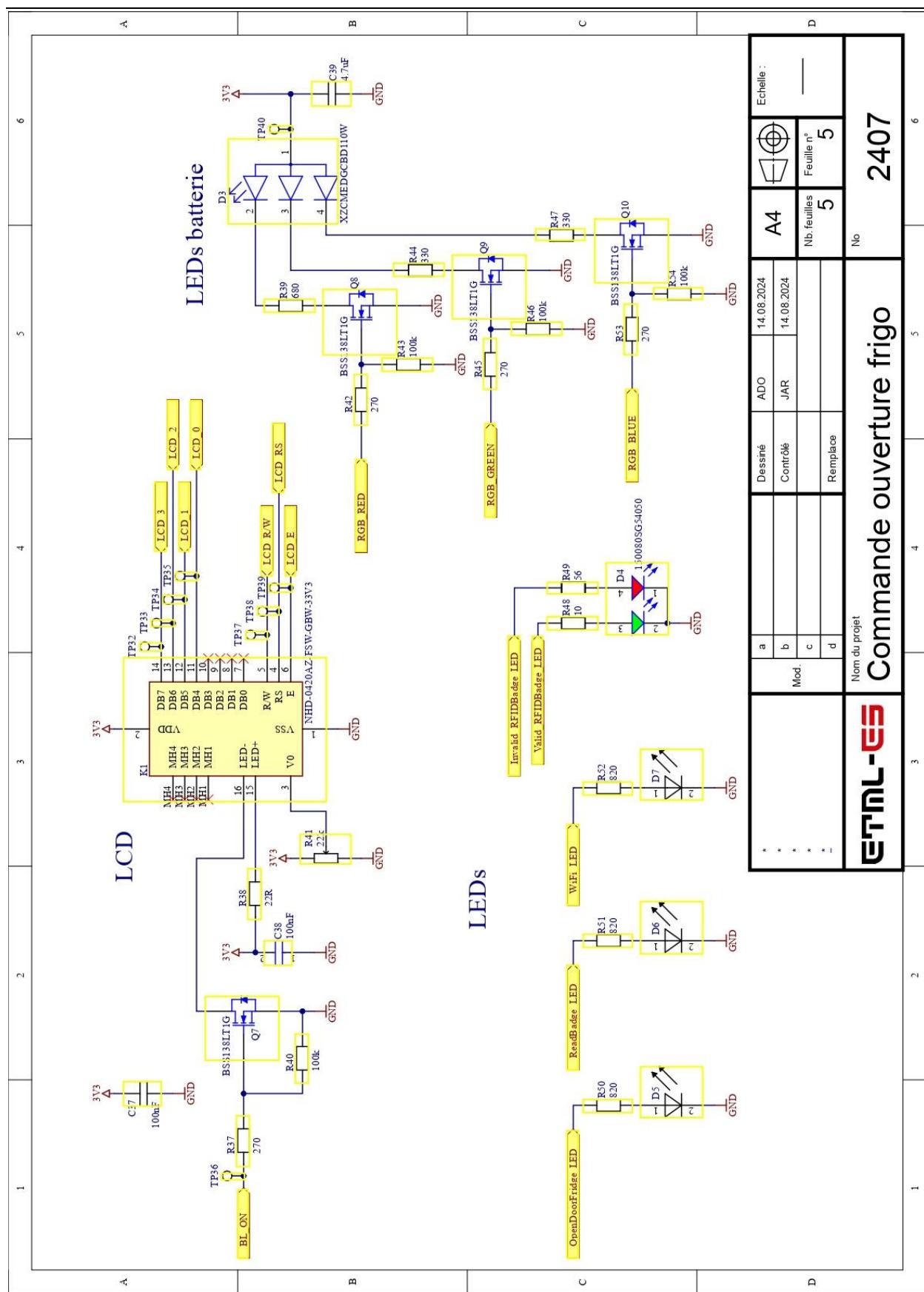


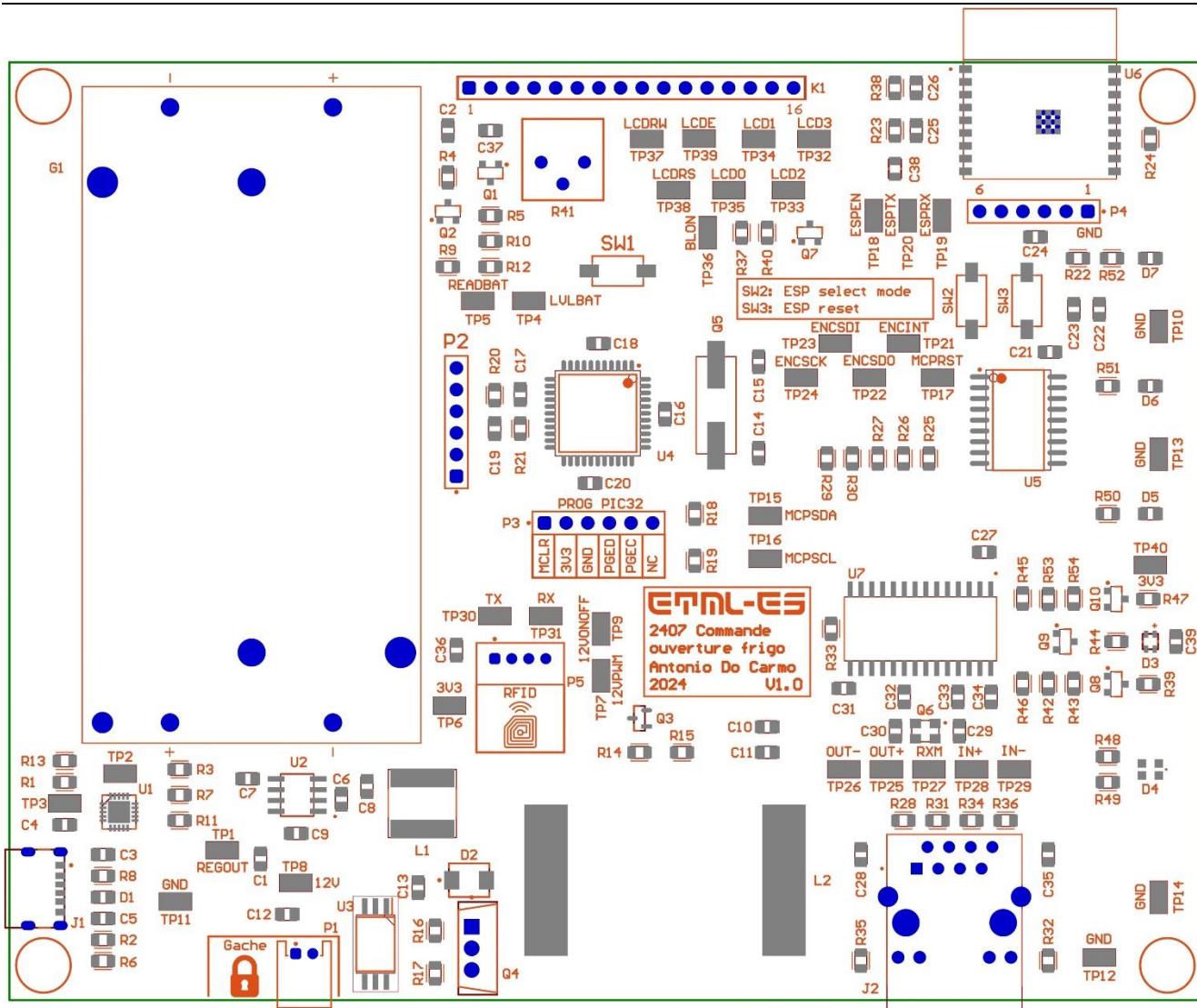
Lausanne le 05.09.2024

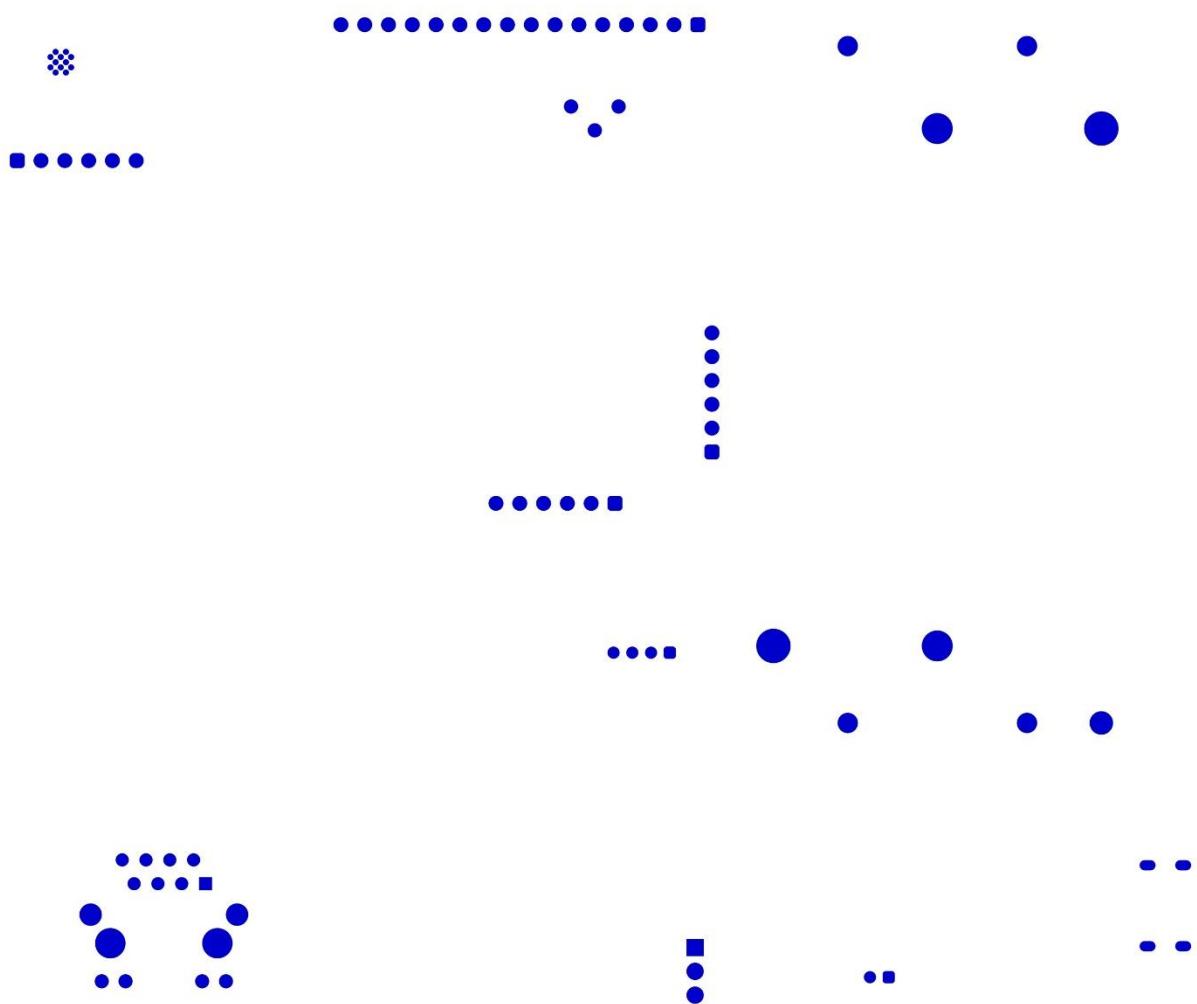
Antonio Do Carmo

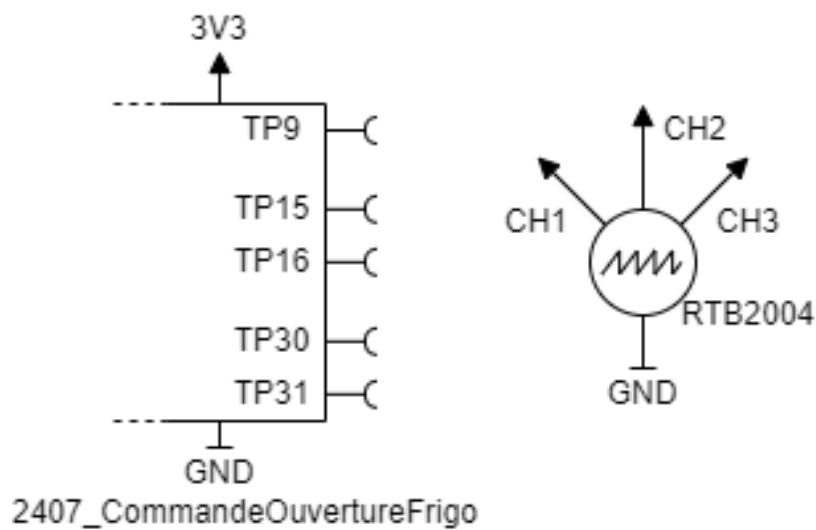












## Projet ETML-ES – Modification – « Hard/Soft »

*A remplir par l'initiateur*

<b>PROJET:</b>	Commande ouverture Frigo		
<b>Entreprise/Client:</b>	Fixme	<b>Département:</b>	SLO
<b>Demandé par (Prénom, Nom):</b>	P.Bovey	<b>Date:</b>	08.09.2024
<b>Objet (No ou réf, pièce, PCB...)</b>	2407		
<b>Version à modifier:</b>	A		

*A remplir par l'exécutant*

<b>Auteur (ETML-ES):</b>	Antonio Do Carmo	<b>Filière:</b>	SLO
<b>Nouvelle version:</b>	B	<b>Date:</b>	09.09.2024

### 1 Description ou justification

Hardware :

- DC-DC 12V : MOSFET canal N utilisé -> doit être remplacé par MOSFET canal P
- Support d'accumulateurs : Le support est trop petit pour les accumulateurs prévus
- Fréquence des badges : Le module RFID ne lit pas les badges fournis -> erreur sur la fréquence de la porteuse

Software :

- Corriger fonction ouverture de la gâche
- Corriger lecture badges RFID
- Implémenter communication Wi-Fi
- Implémenter communication Ethernet

### 2 Référence conception

Pour le hardware : K:\ES\PROJETS\SLO\2407\_CommandeOuvertureFrigoRFID

### 3 Détail des modifications

#	Description	Fait	Approuvé
1	Modification du MOSFET dans le bloc DC-DC 12V	NOK	ADO
2	Choisir un autre support d'accumulateurs	NOK	ADO
3	Choisir un autre module RFID	NOK	ADO
4	Corriger fonction ouverture de la gâche	NOK	ADO
5	Corriger lecture badges RFID	NOK	ADO
6	Implémenter communication Wi-Fi	NOK	ADO
7	Implémenter communication Ethernet	NOK	ADO

### 4 Remarques

Les modifications n'ont pas été effectuées par manque de temps

## 5 Convention de nommage et liens

2407-CommandeOuvertureFrigo-MOD-V1.pdf – première modification pour ce projet

### 5.1 Stockage du fichier

Ce fichier (2410\_PvSimulator-MOD-V1.pdf) sera stocké à la racine du dossier **/doc** de ce projet  
(K:\ES\PROJETS\SLO\2407\_CommandeOuvertureFrigoRFID\doc\2407\_CommandeOuvertureFrigoRFID -MOD-V1.pdf)

**Ecole supérieure**  
Électronique

ETML-ES  
Salle R110

Diplôme - 2407  
Commande Ouverture Frigo

# Mode d'emploi

Réalisé par :

Antonio Do Carmo

**Étape 1 :**

Insérer les accumulateurs dans le support prévu à cet effet

**Étape 2 :**

Brancher le module RFID, le LCD et la gâche au circuit

**Étape 3 :**

Refermer le boîtier

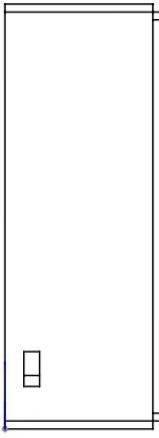
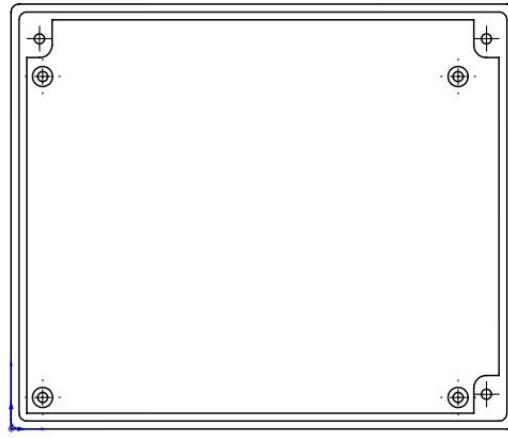
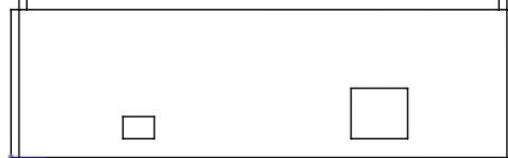


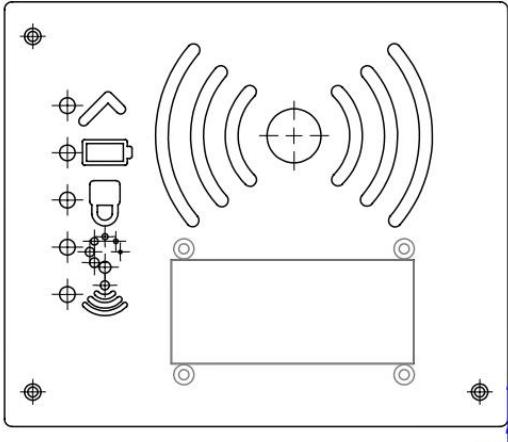
**Étape 4 :**

Placer le boîtier et la gâche sur la porte du frigo

**Étape 5 :**

Passer le badge RFID sur la surface du boîtier comportant le logo RFID  
afin d'ouvrir la porte

	4	3	2	1	
F				F	
E				E	
D				D	
C				C	
B				B	
A	<div style="display: flex; justify-content: space-between;"> <div style="flex: 1;"> <small>SAUF INDICATION CONTRAIRE: LES COTES SONT EN MILLIMÈTRES ETAT DE SURFACE: TOLERANCES: LINEAIRES: ANGULAIRES:</small> </div> <div style="flex: 1; text-align: center;"> <b>FINITION:</b>            2407         </div> <div style="flex: 1; text-align: center;"> <b>CASSER LES ANGLES VIFS</b> </div> <div style="flex: 1; text-align: center;"> <b>NE PAS CHANGER L'ÉCHELLE</b> </div> <div style="flex: 1; text-align: center;"> <b>REVISION</b> </div> </div>			A	
A	<b>AUTEUR</b> ADO	<b>NOM</b> SIGNATURE	<b>DATE</b> 08.09.24		
A					
A					
A					
A					
A	<b>MATERIAU:</b>  <b>MASSE:</b>			<b>TITRE:</b>  <b>Commande ouverture frigo</b> <b>Boitier base</b>	<b>NO. DE PLAN</b>  <b>A4</b>
A				<b>ECHELLE: 1:2</b>	<b>FEUILLE 1 SUR 1</b>
	4	3	2	1	

	4	3	2	1																																								
F				F																																								
E				E																																								
D				D																																								
C				C																																								
B				B																																								
A	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2">SAUF INDICATION CONTRAIRE: LES COTES SONT EN MILLIMÈTRES ETAT DE SURFACE: TOLERANCES: LINEAIRES: ANGULAIRES:</td> <td rowspan="2" style="text-align: center;">FINITION:  2407</td> <td rowspan="2" style="text-align: center;">CASSER LES ANGLES VIFS</td> <td style="text-align: center;">NE PAS CHANGER L'ECHELLE</td> <td rowspan="2" style="text-align: center;">REVISION</td> </tr> <tr> <td>AUTEUR</td> <td>NOM ADO</td> <td>SIGNATURE</td> <td>DATE 08.09.24</td> </tr> <tr> <td>VERIF.</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>APPR.</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>FAB.</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>QUAL.</td> <td></td> <td></td> <td>MATERIAU:</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td>MASSE:</td> <td></td> <td></td> </tr> </table>			SAUF INDICATION CONTRAIRE: LES COTES SONT EN MILLIMÈTRES ETAT DE SURFACE: TOLERANCES: LINEAIRES: ANGULAIRES:		FINITION:  2407	CASSER LES ANGLES VIFS	NE PAS CHANGER L'ECHELLE	REVISION	AUTEUR	NOM ADO	SIGNATURE	DATE 08.09.24	VERIF.						APPR.						FAB.						QUAL.			MATERIAU:						MASSE:			TITRE:  Commande ouverture frigo  Couvercle boitier
SAUF INDICATION CONTRAIRE: LES COTES SONT EN MILLIMÈTRES ETAT DE SURFACE: TOLERANCES: LINEAIRES: ANGULAIRES:		FINITION:  2407	CASSER LES ANGLES VIFS	NE PAS CHANGER L'ECHELLE	REVISION																																							
AUTEUR	NOM ADO			SIGNATURE		DATE 08.09.24																																						
VERIF.																																												
APPR.																																												
FAB.																																												
QUAL.			MATERIAU:																																									
			MASSE:																																									
	4	3	2	1																																								

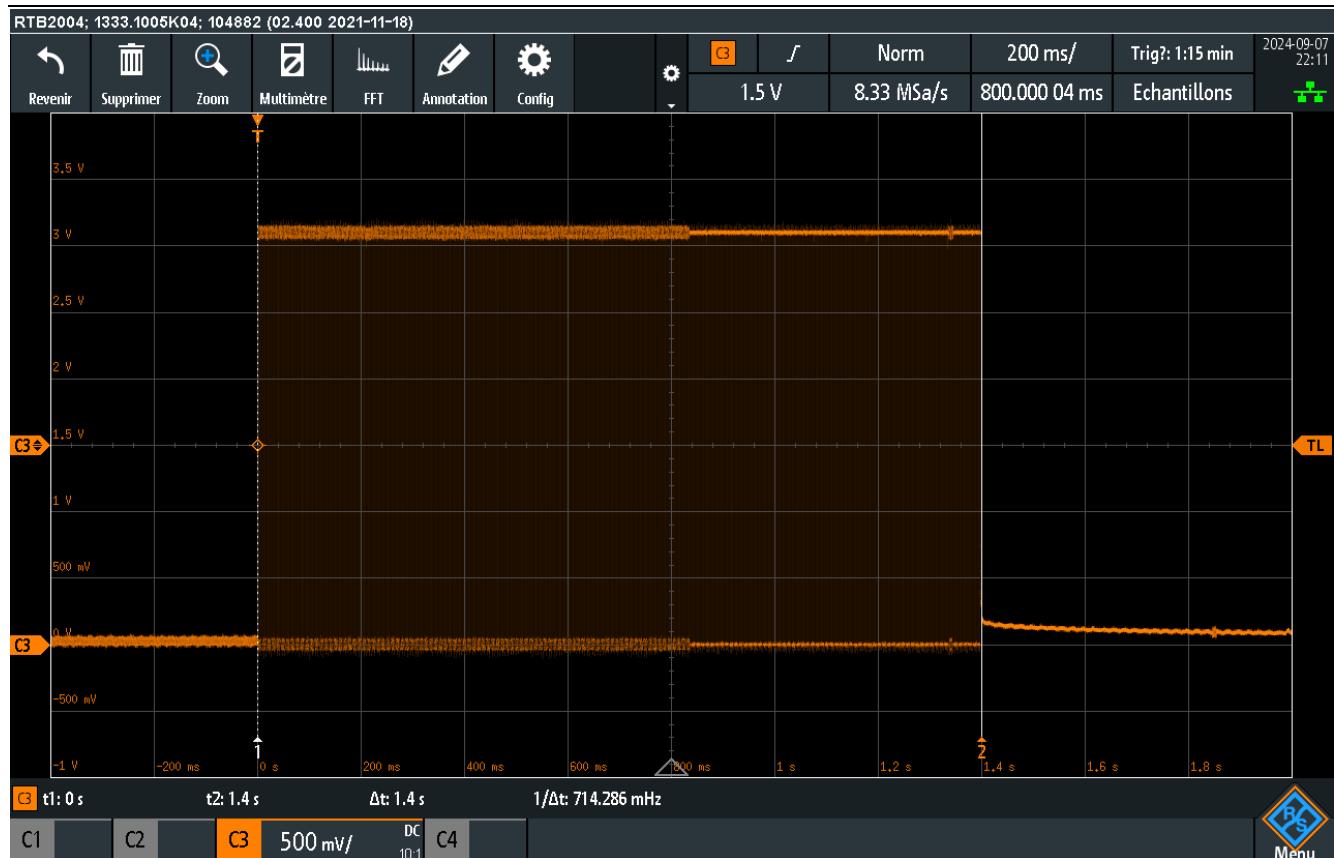


Figure 58 Durée enclanchement de la gâche



Figure 59 Rapport cyclique à 3.5[V]



Figure 60 Rapport cyclique à 4.2[V]

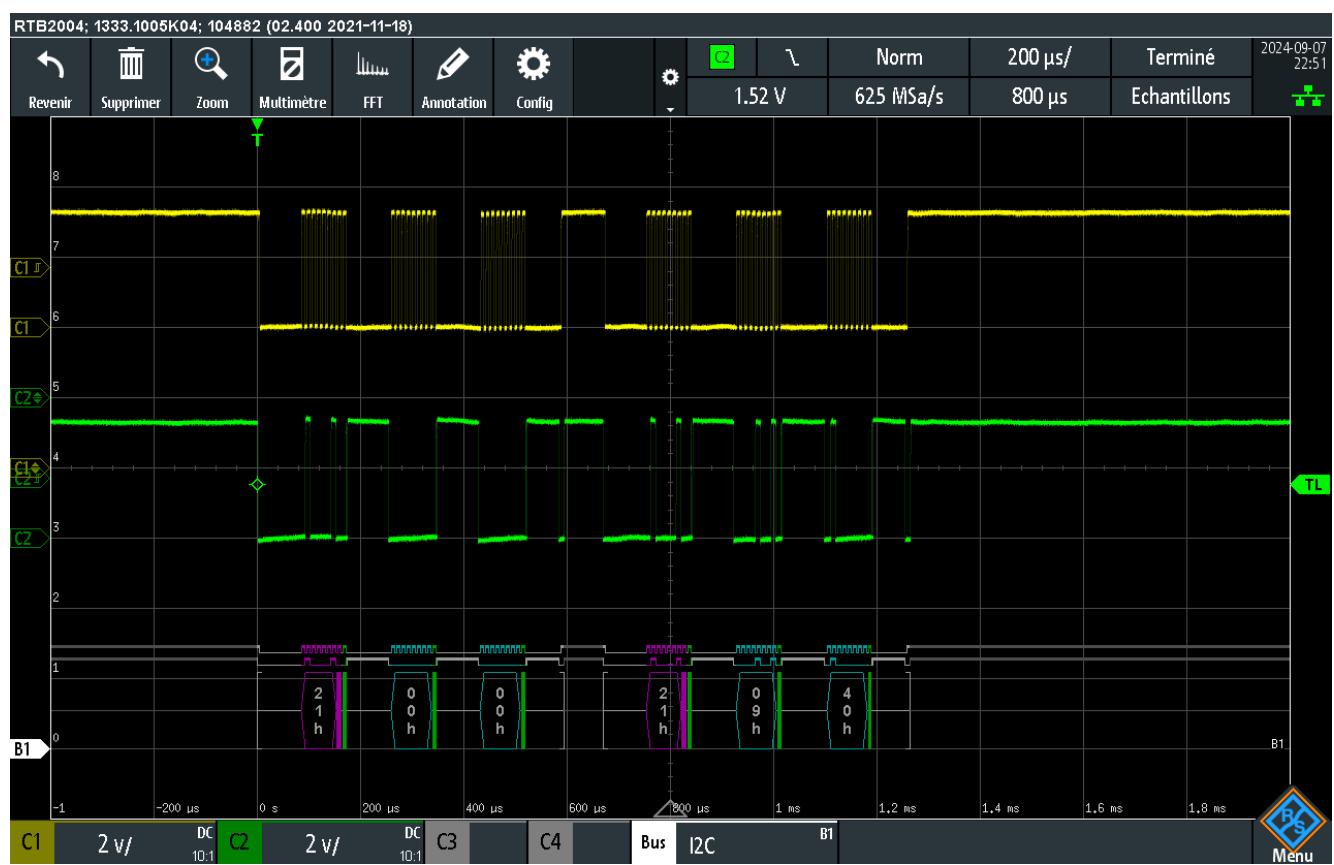


Figure 61 Trame I2C

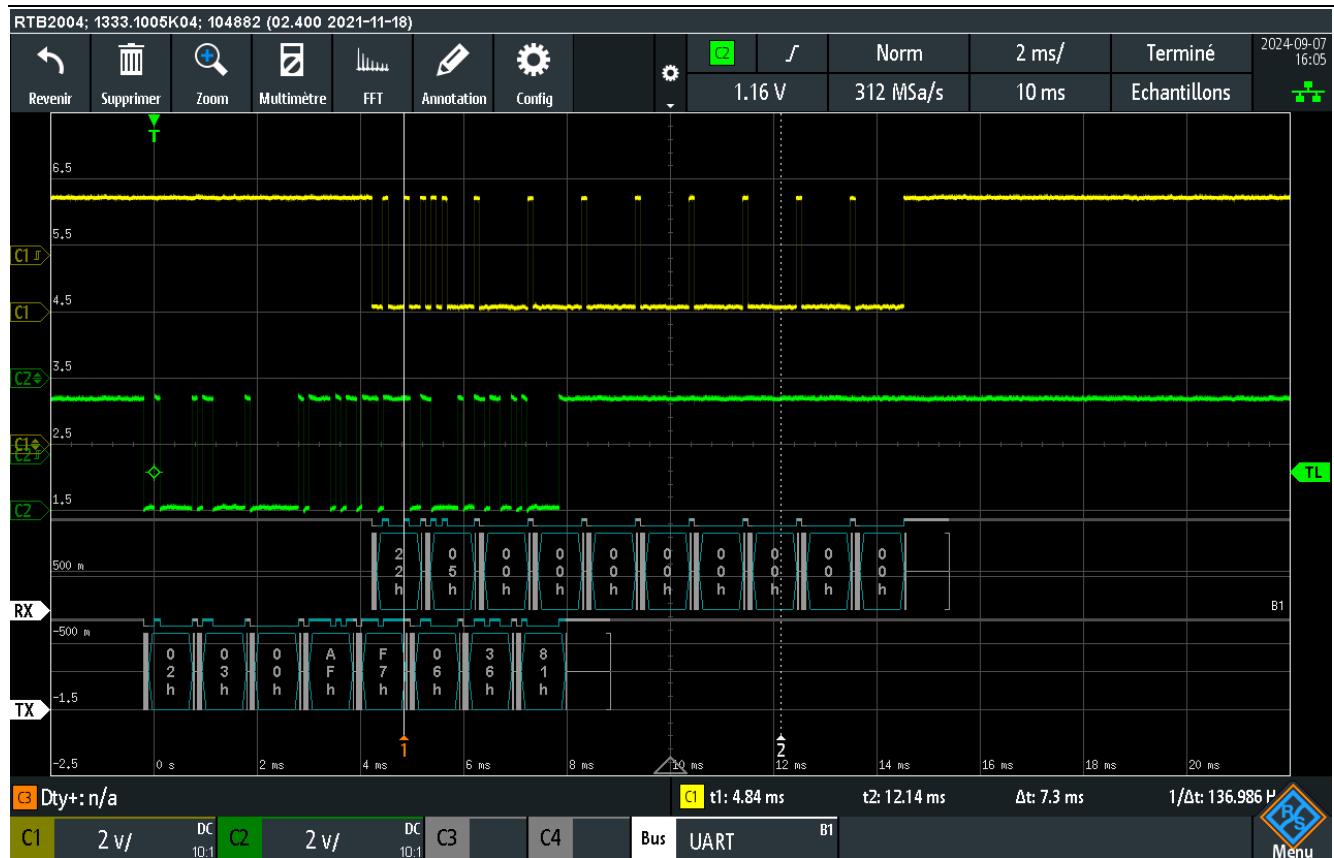


Figure 62 Commande de polling

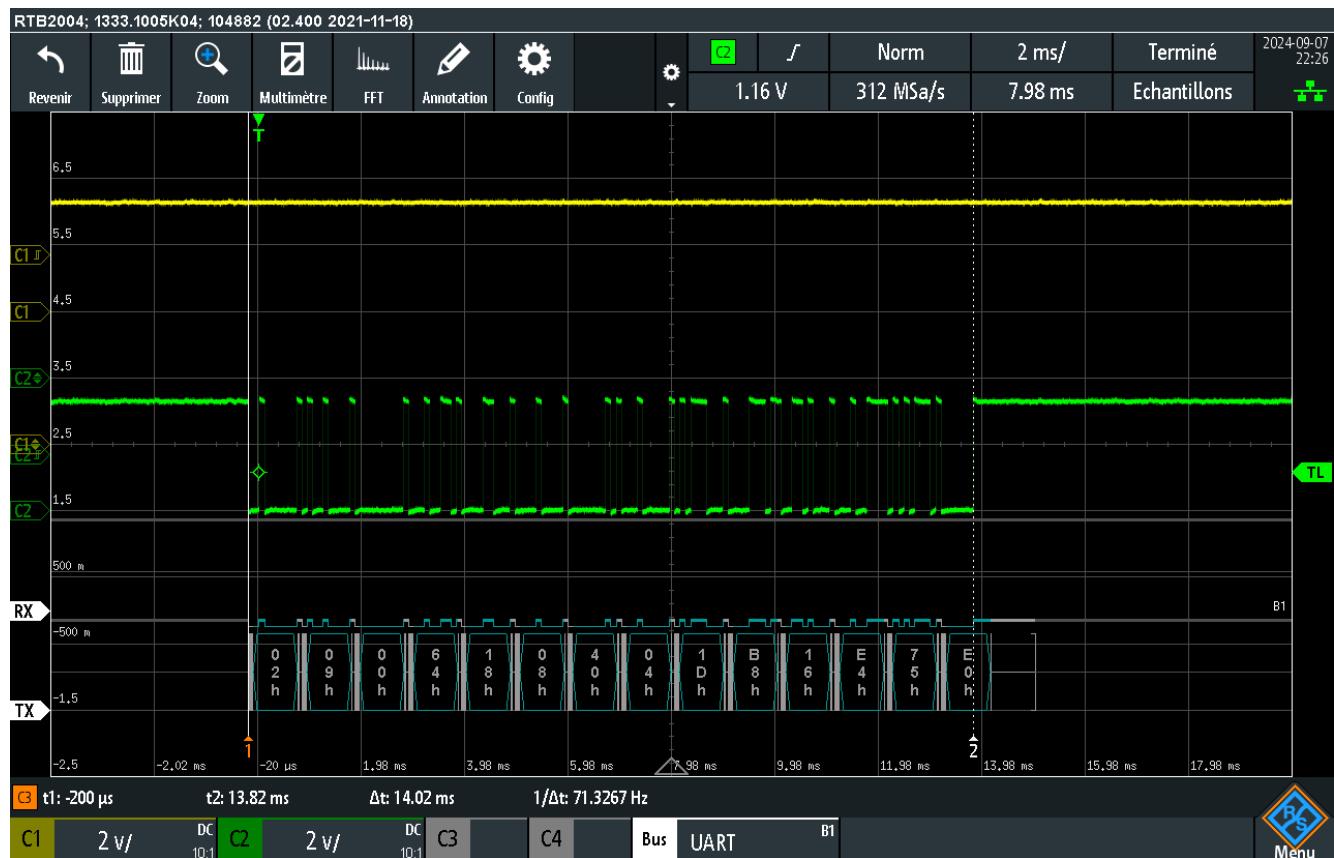


Figure 63 UID RFID