

Compte rendu

Réalisation d'une carte de développement

Introduction :

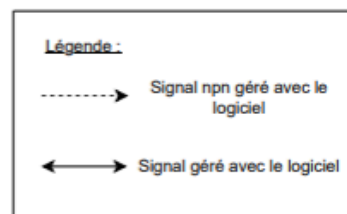
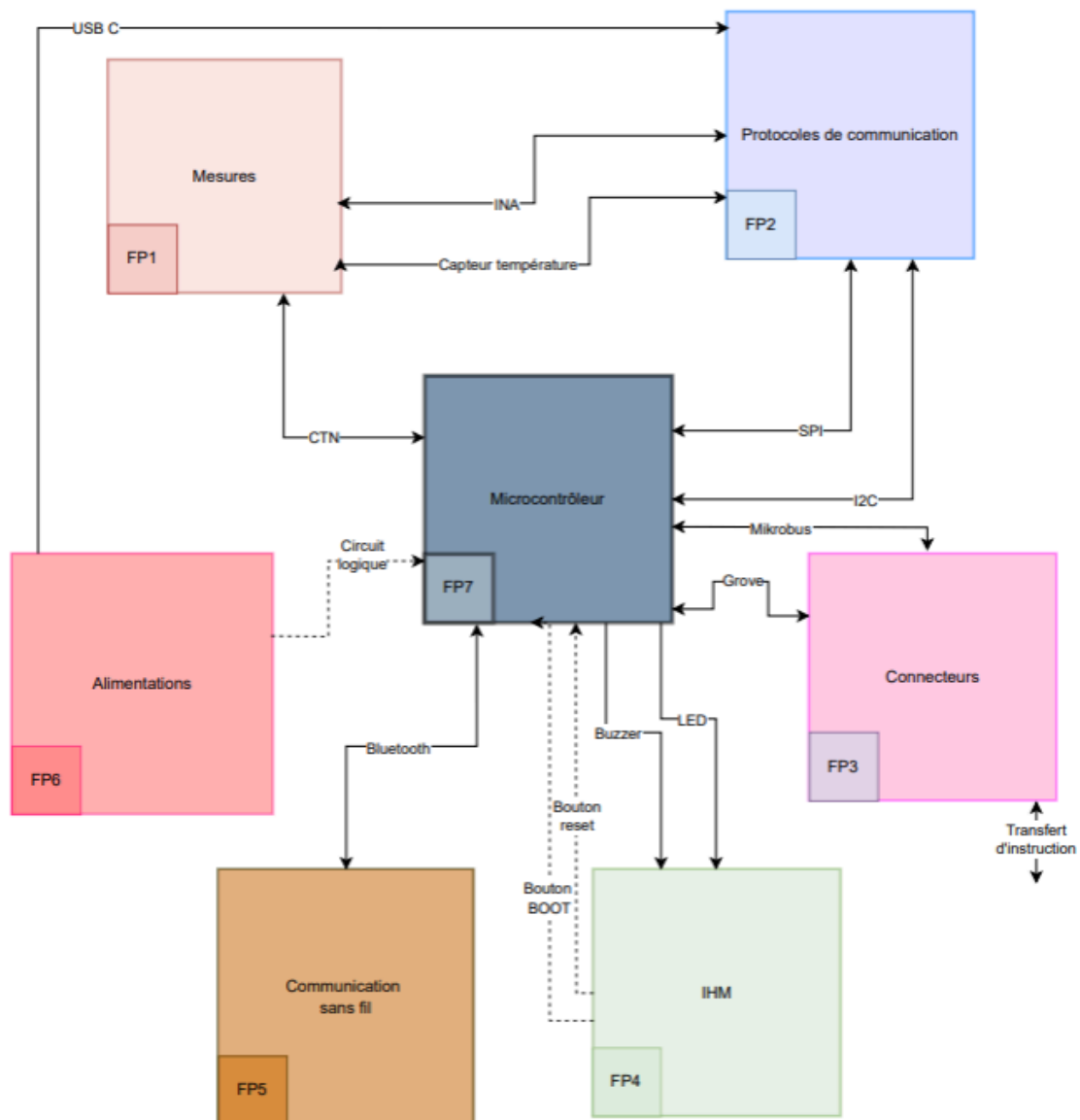
Contexte du projet

Objectif

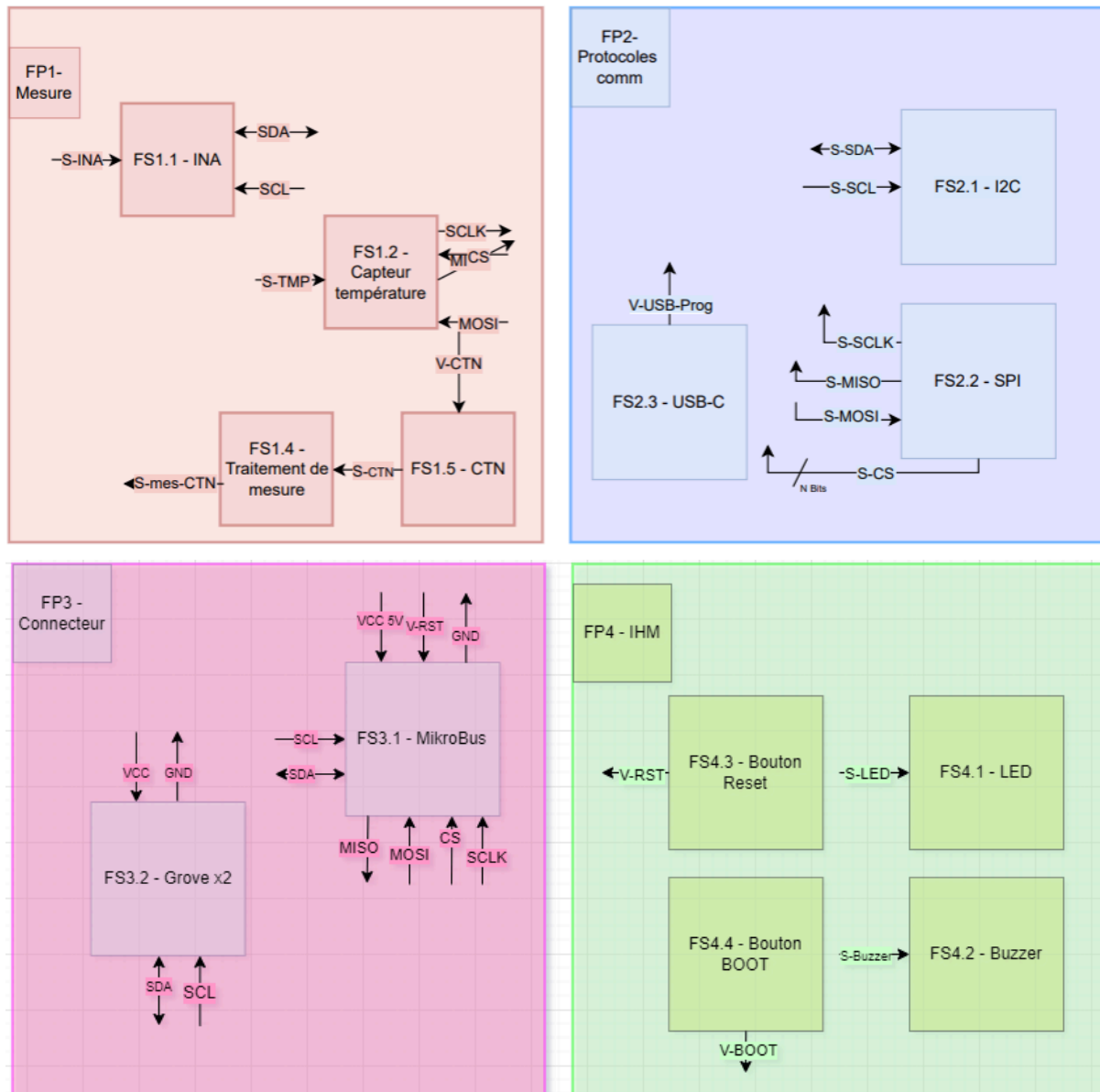
Le but de cette carte électronique est de servir de plateforme polyvalente et flexible pour différentes applications. Elle peut être alimentée de plusieurs manières, ce qui permet une adaptabilité en fonction des sources d'énergie disponibles. Elle est aussi dotée de capteurs pour mesurer le courant et la température, ce qui peut être utile pour le suivi et la surveillance de performances ou pour la protection de dispositifs. L'interface homme-machine (IHM), avec un buzzer, une LED, et une connectivité BLE, permet une interaction facile avec l'utilisateur. La possibilité de boot et reset matériel permet une gestion directe de la carte. Enfin, la compatibilité avec les modules MikroBus et Grove garantit une extension facile, ajoutant des fonctionnalités supplémentaires pour diverses applications.

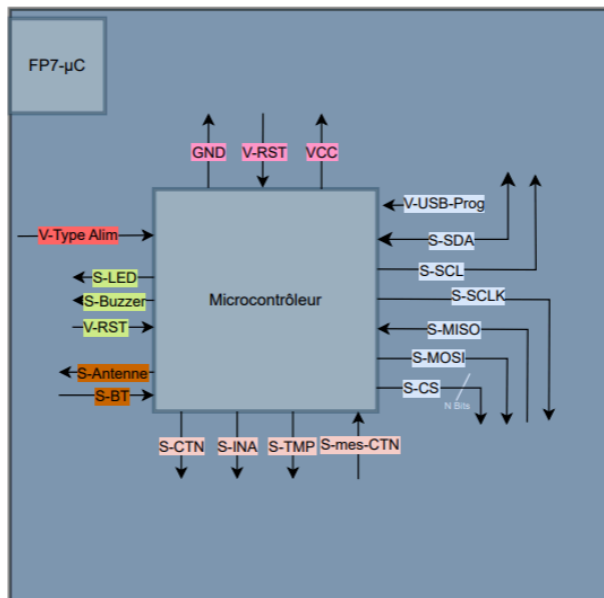
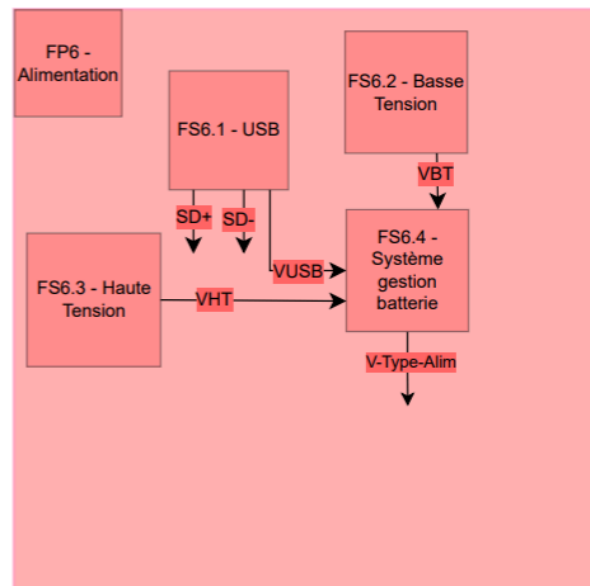
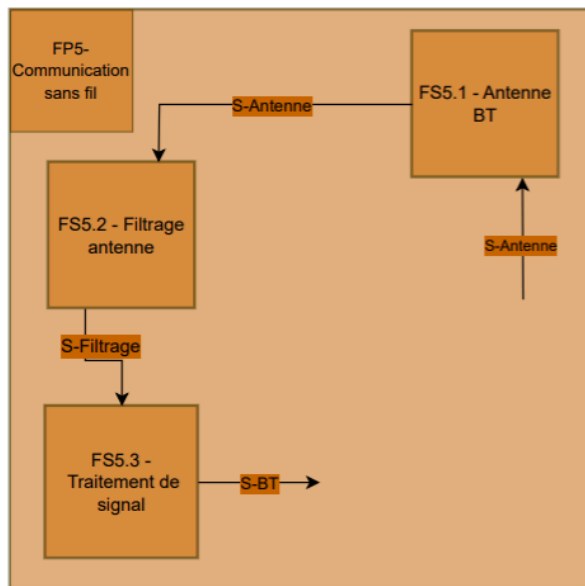
Jalon 1 : Analyses fonctionnelles

AF1



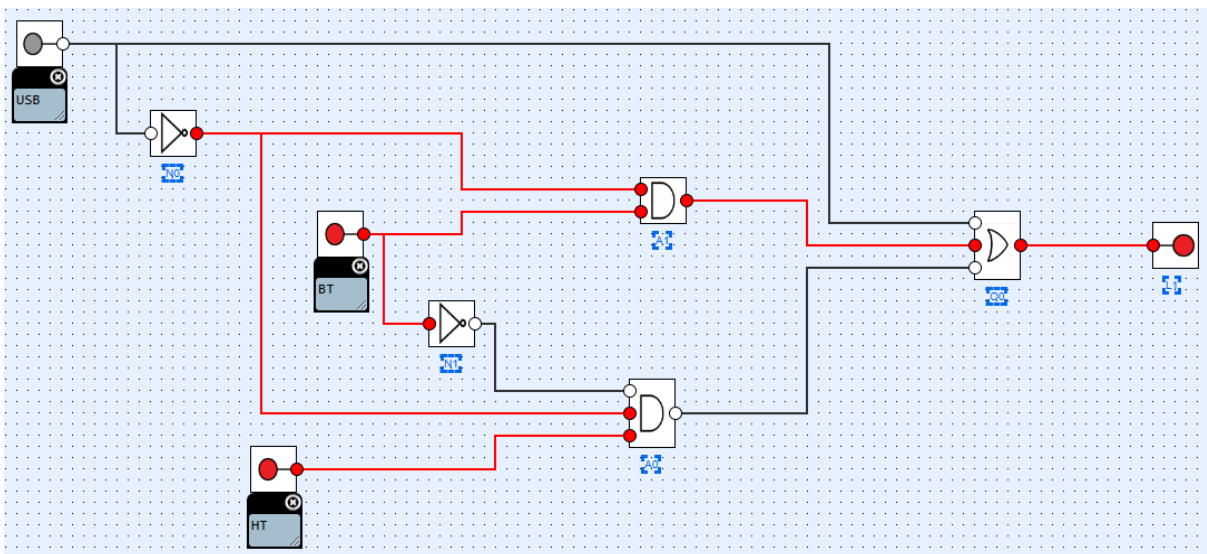
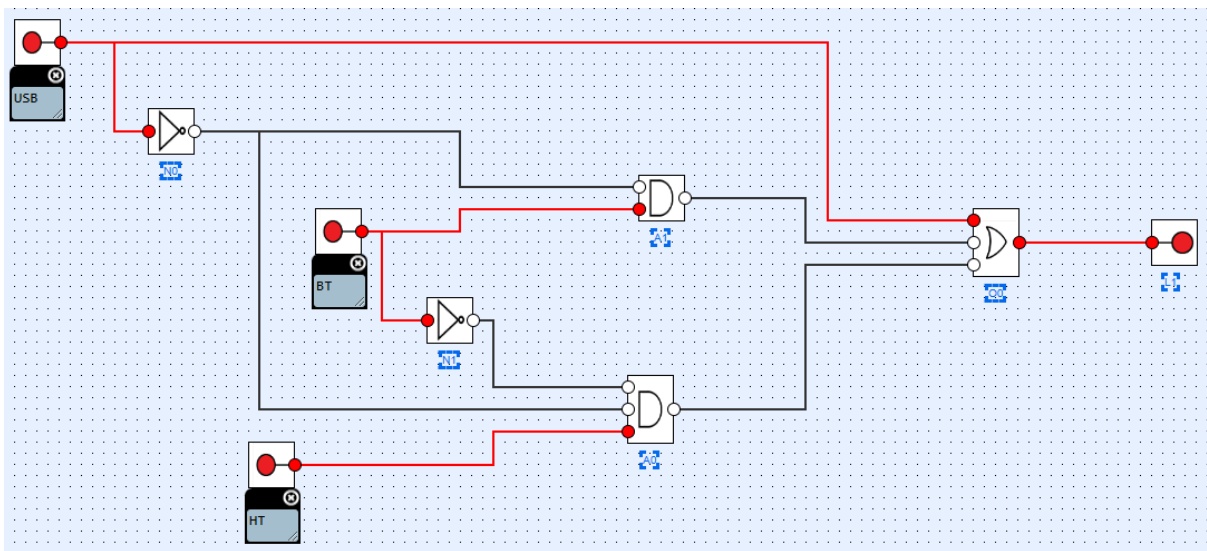
AF2

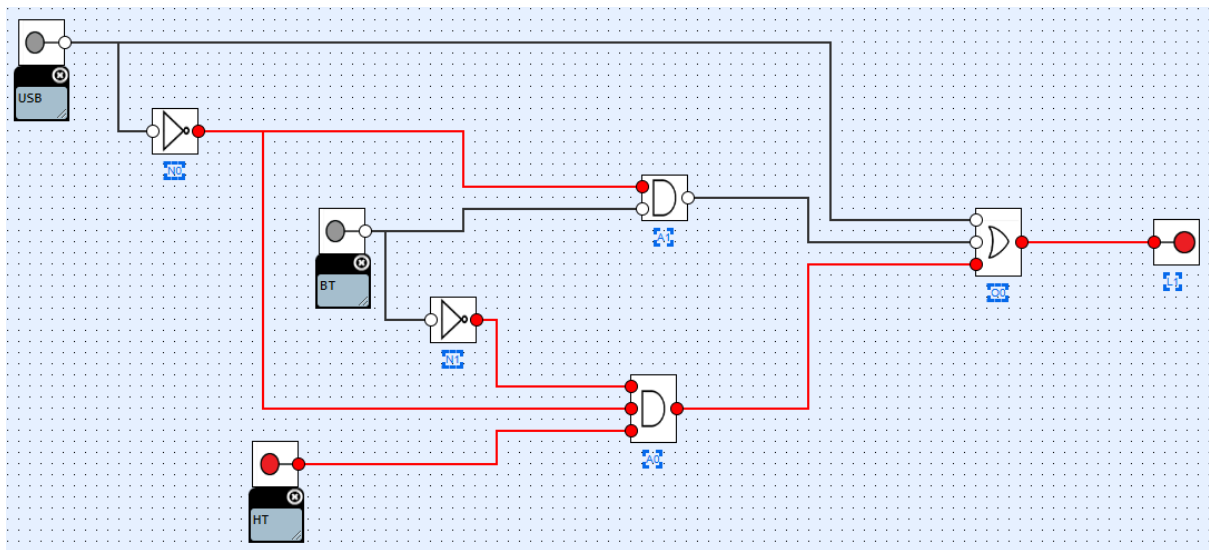




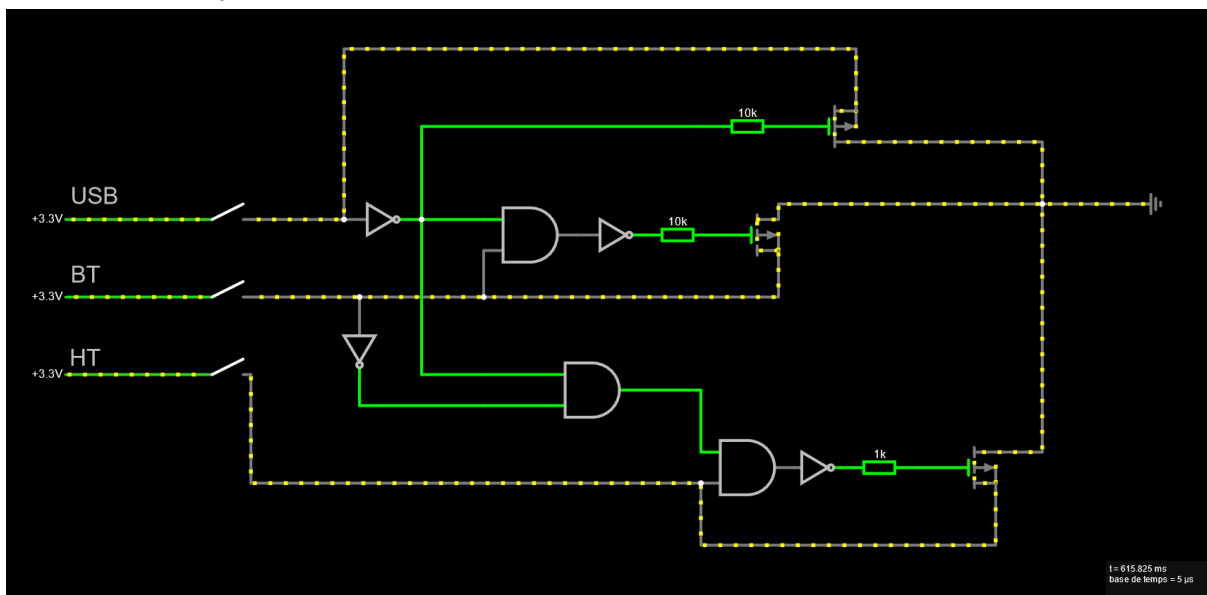
Jalon 2 : Réalisation schématique

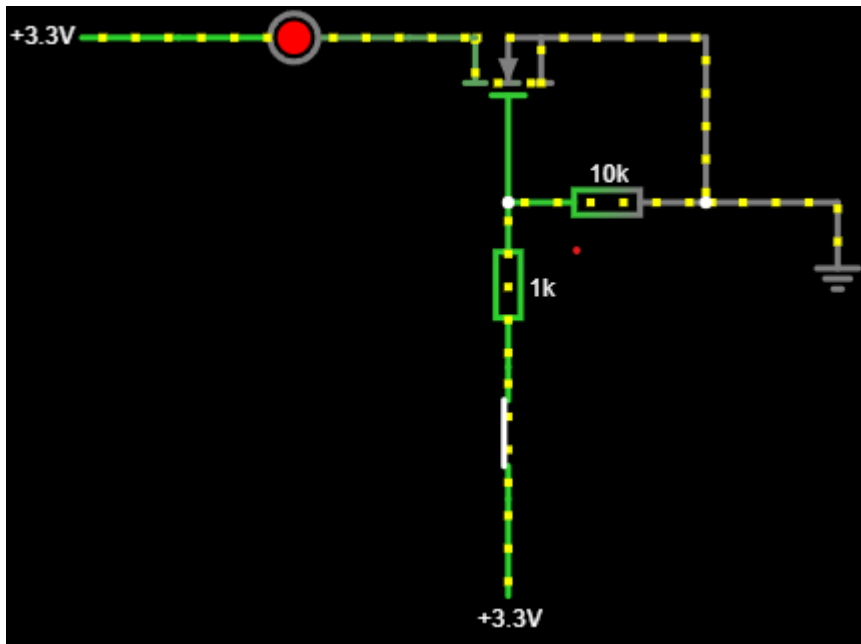
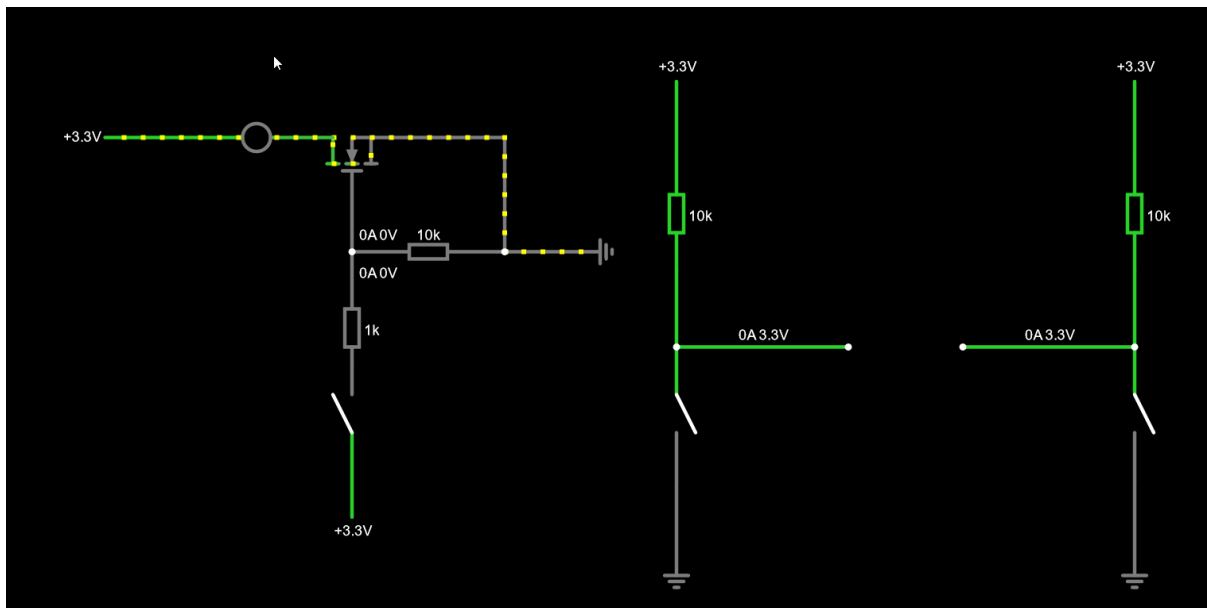
Pour la réalisation schématique, nous avons commencé par effectuer des simulations afin de valider notre conception. Dans un premier temps, la simulation de l'alimentation (USB, BT et HT) a été réalisée à l'aide d'ISim, ce qui nous a permis de vérifier le bon fonctionnement des sources d'énergie et leur logique de commutation.





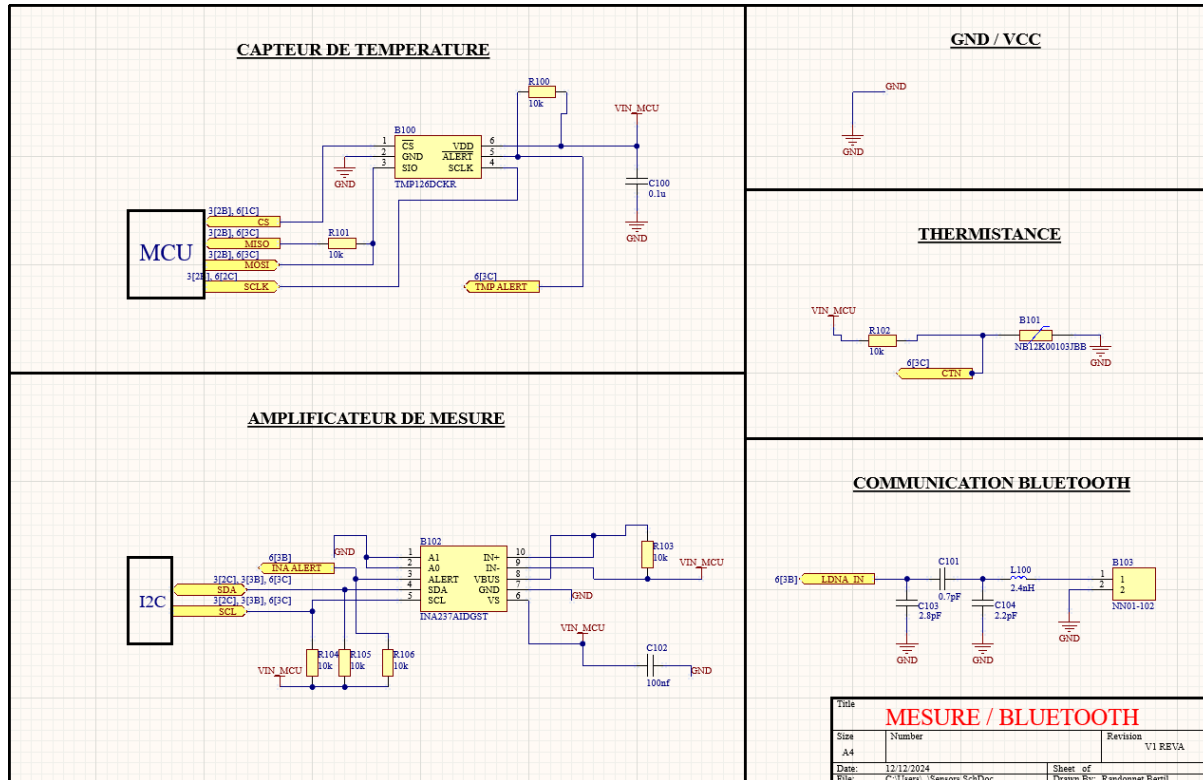
Ensuite, nous avons approfondi les tests en utilisant l'outil Falstad pour simuler de manière plus visuelle et dynamique les différents états des circuits





Altium :

Schématique sensors :

**1. Capteur de température**

Ce bloc utilise le capteur **TMP126DKCR**, qui communique via un bus SPI avec le microcontrôleur (MCU). Voici les points clés :

Broches SPI (CS, MOSI, MISO, SCLK) : Connectées au MCU pour transmettre les données de température.

Broche ALERT : Sert à signaler au MCU des conditions spécifiques, comme une surchauffe.

- **Résistance de pull-up (R101, 10 kΩ) :** Maintient la ligne de communication en état stable lorsqu'elle n'est pas utilisée.
- **Condensateur de découplage (C100, 0.1 μF) :** Filtre les bruits parasites sur l'alimentation du capteur.
- **Alimentation :** Le capteur est alimenté par le **VIN_MCU**, une source stabilisée à 3,3 V.

Ce sous-système permet de mesurer la température et d'envoyer les données au MCU pour traitement.

[Annexe 1 : TMP126DKCR](#)

2. Amplificateur de mesure

Le bloc amplificateur utilise le circuit **INA237AIDGST**, qui est un amplificateur de précision capable de mesurer des tensions et des courants. Points essentiels :

- **Entrées IN+ et IN-** : Reçoivent le signal différentiel à amplifier, provenant du circuit mesuré.
- **Sortie vers le bus I2C (SCL, SDA)** : Les données amplifiées sont envoyées au MCU pour analyse via un bus I2C.
- **Résistances de pull-up (R104, R106)** : Stabilisent les lignes SDA et SCL en évitant les états flottants.
- **Condensateur de découplage (C102, 100 nF)** : Réduit les bruits électriques sur l'alimentation de l'amplificateur.

Ce circuit permet de mesurer avec précision des courants ou tensions faibles.

[Annexe 2 : INA237AIDGST](#)

3. Thermistance

La thermistance **NBI12K00103JBB** est utilisée pour des mesures analogiques de température. Fonctionnement :

- **Résistance pull-up (R102, 10 kΩ)** : Forme un diviseur de tension avec la thermistance, permettant de convertir les variations de résistance en signal analogique mesurable.
- **Connexion au VIN_MCU** : Fournit une alimentation stable à la thermistance.

Ce composant fournit une mesure alternative ou redondante de la température.

[Annexe 3 : NBI12K00103JBB](#)

4. Communication Bluetooth

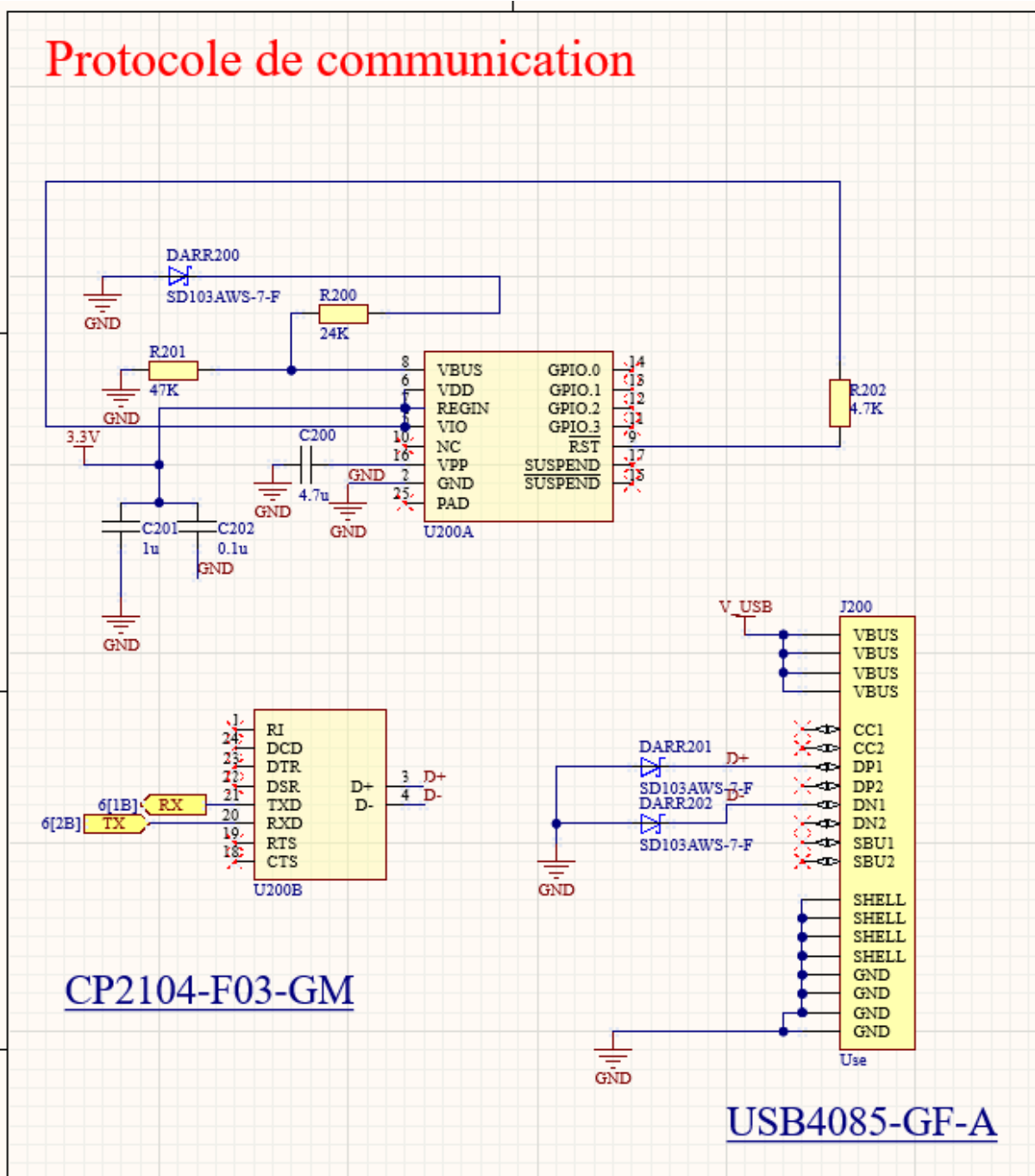
Le bloc Bluetooth utilise un module **NN01-102** pour établir une communication sans fil. Points essentiels :

- **LDNA_IN** : Une ligne d'entrée pour l'antenne Bluetooth.
- **Circuit d'accord d'antenne (C101, L100, C103, C104)** : Optimise les performances de l'antenne en adaptant l'impédance.
 - **Condensateurs (C101, C103, C104)** : Assurent le filtrage des hautes fréquences.
 - **Inductance (L100, 2.4 nH)** : Contribue à l'accord en série avec les condensateurs.
- **Alimentation via VIN_MCU** : Alimente le module Bluetooth.

Ce bloc permet d'établir une communication sans fil entre l'ESP32 et un appareil externe, comme un smartphone ou un ordinateur.

Annexe 4 : NN01-102

Schématique protocole de communication :



1. USB4085-GF-A (Connecteur USB)

Ce composant représente le connecteur USB principal (J200). Points importants :

- **Broches VBUS et GND** : Assurent l'alimentation (+5 V) pour les circuits connectés à l'USB.
- **Broches D+ et D-** : Lignes de données différentielles utilisées pour la communication USB.
- **SHIELD** : Connexion à la masse pour l'écran du câble USB, réduisant les interférences électromagnétiques.

Ce bloc fournit l'interface physique pour connecter la carte à un port USB externe, permettant l'alimentation et l'échange de données.

[Annexe 5 : USB4085-GF-A](#)

2. CP2104-F03-GM (Convertisseur USB-UART)

Le **CP2104** (U200B) est un convertisseur USB vers UART qui permet de connecter un périphérique USB à une interface série. Fonctionnement :

- **Entrées D+ et D-** : Reçoivent les données du connecteur USB (J200) via le bus différentiel.
- **Sorties TX et RX** : Fournissent une interface série (UART) pour communiquer avec le MCU.
- **Broches supplémentaires (RI, DCD, DTR, etc.)** : Peuvent être utilisées pour des fonctions avancées de contrôle de flux série.
- **Alimentation** : Alimentée en 3,3 V depuis le régulateur intégré.
- **GND** : Référence de masse commune.

Le **CP2104** est un élément clé pour établir une liaison série entre la carte et un ordinateur ou tout autre périphérique compatible USB.

[Annexe 6 : CP2104-F03-GM](#)

3. Régulateur de tension et circuit passif

Ce sous-circuit alimente et stabilise les tensions nécessaires pour le fonctionnement du protocole USB :

- **SD103AWS-7-F (DARR200, diode)** : Protège contre les surtensions sur les lignes d'alimentation et évite les retours de courant.
- **Résistances (R200, R201, R202)** :
 - R200 (24 kΩ) : participe à la configuration de la tension du régulateur.
 - R201 et R202 (47 kΩ, 4,7 kΩ) : jouent un rôle dans le découplage et la stabilisation des signaux.
- **Condensateurs (C200, C201, C202)** :

- C200 (4,7 μ F) : Assure le filtrage de l'alimentation pour éviter les fluctuations de tension.
- C201 et C202 (0,1 μ F, 1 μ F) : Réduisent les bruits haute fréquence.

Ce bloc garantit que le convertisseur USB-UART et les composants associés reçoivent une alimentation propre et stable.

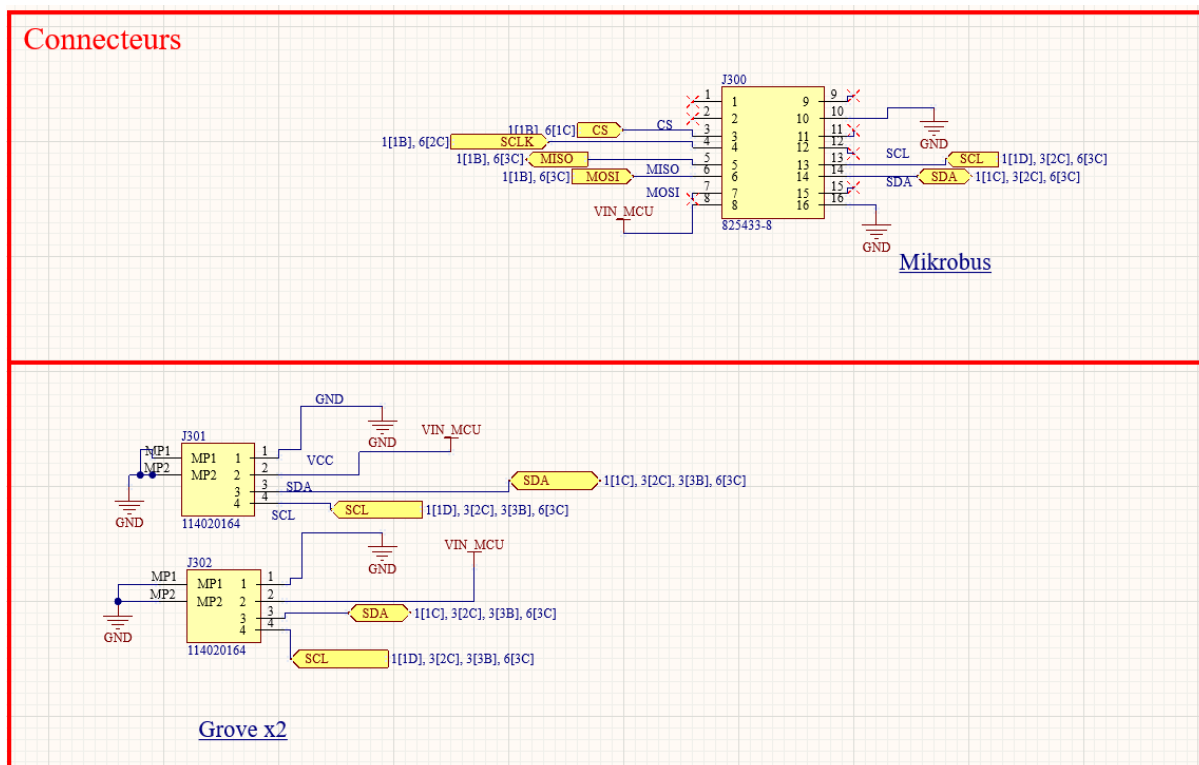
Annexe 7 : SD103AWS-7-F

4. Communication UART

Les lignes **TX** et **RX** établissent la communication série :

- **TX (Transmitter)** : Transmet les données du convertisseur USB-UART vers le microcontrôleur.
- **RX (Receiver)** : Reçoit les données en provenance du microcontrôleur.

Schématique connecteurs :



1. Connecteur Mikrobus (J300)

Le connecteur Mikrobus est conçu pour accueillir des modules MikroElektronika. Il fournit une interface standard pour une communication rapide et facile entre la carte et des périphériques externes.

- **Broches SPI (CS, MISO, MOSI, SCLK) :**
 - **CS (Chip Select) :** Permet de sélectionner un périphérique spécifique sur le bus SPI.
 - **MISO (Master In Slave Out) :** Ligne utilisée pour transférer des données du périphérique esclave au maître.
 - **MOSI (Master Out Slave In) :** Ligne utilisée pour transférer des données du maître au périphérique esclave.
 - **SCLK (Serial Clock) :** Génère l'horloge nécessaire pour la communication SPI.
- **Broches I²C (SCL, SDA) :**
 - **SCL (Serial Clock Line) :** Fournit l'horloge pour le protocole I²C.
 - **SDA (Serial Data Line) :** Utilisée pour transférer des données bidirectionnelles entre le maître et l'esclave.
- **Broches d'alimentation (VIN_MCU, GND) :**
 - **VIN_MCU :** Fournit une alimentation au module connecté.
 - **GND :** Sert de référence pour la masse.

Ce connecteur est polyvalent et peut accueillir plusieurs modules, comme des capteurs ou des périphériques de communication, grâce à ses lignes SPI et I²C.

[Annexe 8 : Mikrobus](#)

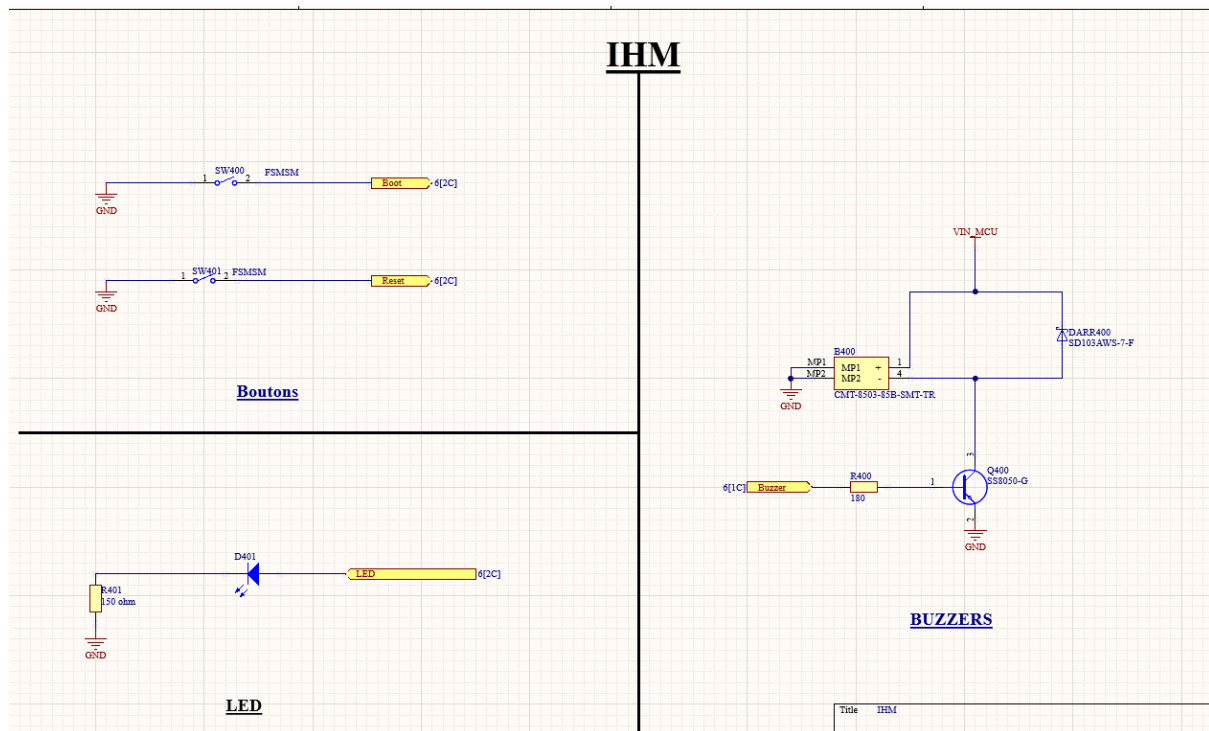
2. Connecteurs Grove (J301 et J302)

Les connecteurs **Grove** (J301 et J302) sont utilisés pour des modules compatibles avec le système Grove. Ces connecteurs permettent une connexion facile avec des capteurs ou des périphériques. Chaque connecteur suit une configuration standard avec 4 broches :

- **VCC :** Fournit l'alimentation pour les modules connectés (tirée de VIN_MCU).
- **GND :** Assure la masse pour les modules connectés.
- **SDA (Serial Data Line) :** Transmet les données pour le protocole I²C.
- **SCL (Serial Clock Line) :** Fournit l'horloge pour la communication I²C.

Les deux connecteurs sont parallèles, permettant la connexion de plusieurs modules Grove en même temps. Cela favorise l'évolutivité du système et la facilité d'utilisation.

[Annexe 9 : Groove](#)

Schématique IHM :**1. Boutons**

Cette section contient deux boutons poussoirs **SW400** et **SW401**.

Composants principaux :

- **SW400** (Boot) : Bouton connecté à la masse (**GND**).
- **SW401** (Reset) : Deuxième bouton également relié à la masse.
- Les deux boutons sont connectés à des signaux étiquetés **Boot** et **Reset**.

Fonctionnement :

- **SW400** permet de déclencher la fonction **Boot**.
- **SW401** permet de déclencher la fonction **Reset**. Ces boutons sont généralement utilisés pour forcer un redémarrage ou initialiser une carte MCU.

2. LED

Cette section comprend une **LED** pour signaler l'état du système.

Composants principaux :

- **D401** : Diode LED.
- **R401** (150 ohm) : Résistance de limitation de courant pour protéger la LED.
- **GND** : La LED est reliée à la masse pour assurer un fonctionnement sécurisé.

Fonctionnement :

- Lorsque le signal **LED** (étiqueté **6[2C]**) est activé, la LED s'allume en passant un courant contrôlé à travers **R401**.
- Cela permet d'indiquer visuellement un état spécifique du système.

3. Buzzers

Cette partie contrôle un **buzzer** pour produire un signal sonore.

Composants principaux :

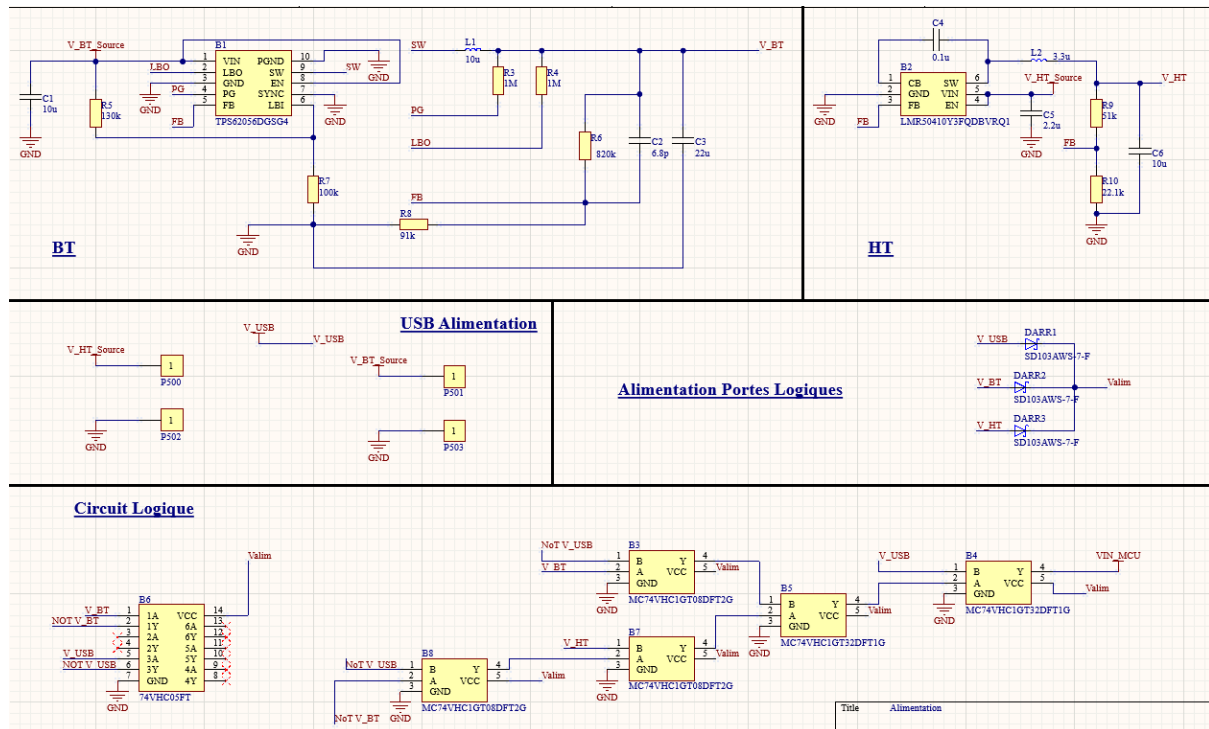
- **B400** : Relais **CMT8503-SSB-SMT-TR** pour activer le buzzer.
- **Q400** (SS8550-G) : Transistor PNP utilisé comme interrupteur pour contrôler l'alimentation du buzzer.
- **R400** (180 ohm) : Résistance pour limiter le courant traversant la base du transistor.
- **DARR400** (SD103AWS-7-F) : Diode pour la protection contre les surtensions (anti-retour de courant).
- **Buzzer** : Signal étiqueté **6[1C]** qui active le buzzer.

Fonctionnement :

1. Le relais **B400** permet de basculer l'alimentation vers le buzzer.
2. Le transistor **Q400** agit comme un interrupteur pour contrôler l'activation du buzzer.
3. La résistance **R400** limite le courant entrant dans la base du transistor pour protéger le circuit.
4. La diode **DARR400** protège contre les pics de tension indésirables.

Lorsque le signal **Buzzer** (6[1C]) est activé :

- Le transistor se met en conduction.
- Le buzzer est alimenté, générant un signal sonore.

Schématique alimentation :**1. Alimentation BT (Bloc BT)**

Ce bloc utilise le **TPS62050DGS4**, un régulateur abaisseur (DC-DC) pour générer une tension **V_BT**.

Composants principaux :

- **Entrée (V_BT_Source)** : Alimentation d'entrée pour le régulateur.
- **C1 (10uF)** : Condensateur pour stabiliser la tension d'entrée.
- **R5 (100k)** et **R7 (100k)** : Résistances pour configurer la tension de sortie via la broche FB (Feedback).
- **L1 (2.2uH)** : Inductance pour le filtrage de la sortie.
- **C2 (6.8pF)**, **C3 (22uF)** : Condensateurs pour réduire les ondulations à la sortie.
- **R8 (91k)**, **R6 (820k)** : Résistances pour le contrôle du feedback.

Fonctionnement : Le régulateur abaisse la tension d'entrée pour fournir une tension stable **V_BT**. Les résistances de feedback (FB) permettent de définir cette tension de sortie.

[Annexe 10 : TPS62050DGS4](#)

2. Alimentation HT (Bloc HT)

Ce bloc utilise le **LMR50410Y3FDBVRQ1**, un autre régulateur abaisseur pour générer une tension **V_{HT}**.

Composants principaux :

- **Entrée (V_{HT_Source})** : Source d'alimentation d'entrée.
- **C4 (0.1uF)** et **C5 (2.2uF)** : Condensateurs pour le filtrage côté entrée et sortie.
- **L2 (3.3uH)** : Inductance pour stabiliser la tension de sortie.
- **C6 (10uF)** : Condensateur de sortie pour réduire les ondulations.
- **R9 (51k)**, **R10 (22.1k)** : Résistances pour le feedback.

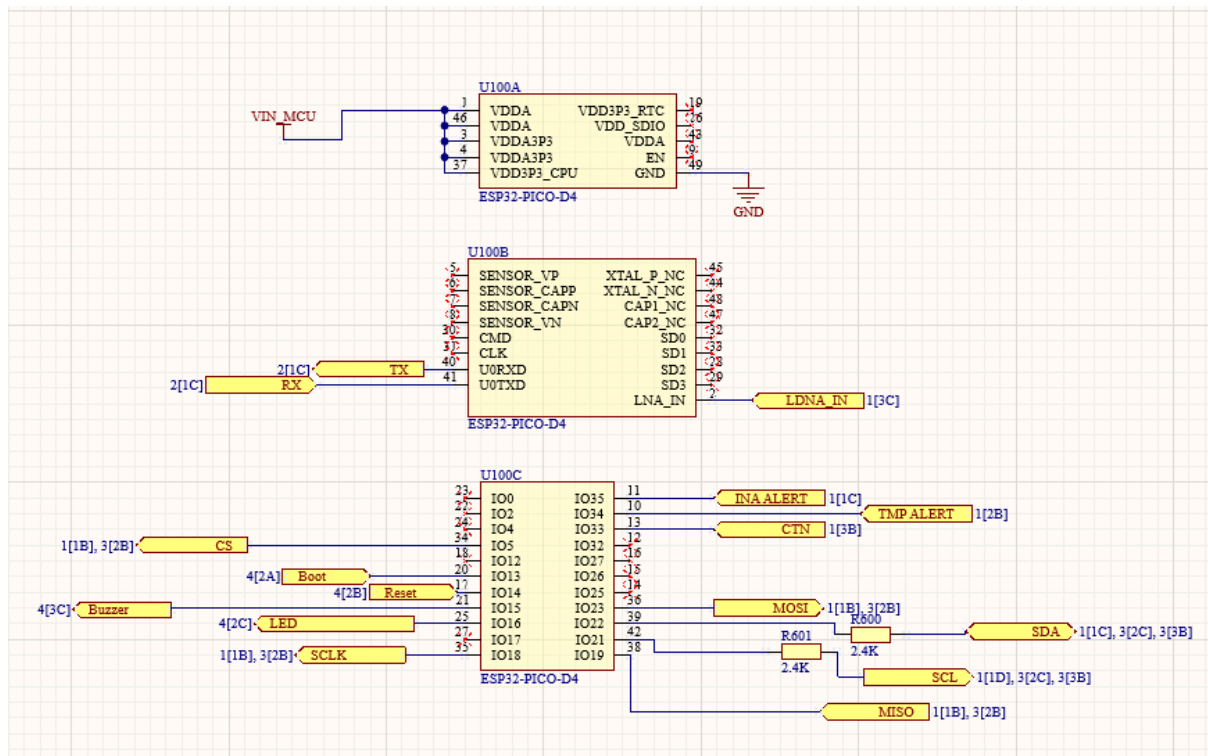
Fonctionnement : Le régulateur ajuste la tension d'entrée pour produire une tension de sortie **V_{HT}**, définie par les résistances de feedback.

[Annexe 11 : LMR50410Y3FDBVRQ1](#)

3. USB Alimentation

Cette section inclut trois connecteurs **P500**, **P501**, et **P502** pour distribuer les différentes tensions d'alimentation (**V_{USB}**, **V_{BT_Source}**, **V_{HT_Source}**) ainsi que la masse **GND**.

Rôle : Ces connecteurs permettent de fournir des sources d'alimentation externes pour tester et alimenter d'autres composants du circuit.

Schématique MCU :**1. Alimentation du MCU**

- **Broches d'alimentation :**
 - **VIN_MCU** : Alimentation principale fournie au microcontrôleur, connectée aux broches **VDDA3P3**, **VDDA**.
 - Les broches **VDDA**, **VDD33_CPU** alimentent respectivement les parties analogiques et numériques de l'ESP32.
 - La **masse (GND)** est commune à tout le circuit et connectée aux broches de référence **GND**.
- **Stabilité :**
 - Le fait d'alimenter correctement ces broches assure un fonctionnement stable et performant du module ESP32.

2. Broches de Communication Série (UART)

- **UART0 :**
 - **TX (Transmission)** : Broche **2[1C]** utilisée pour envoyer des données série.
 - **RX (Réception)** : Broche **2[1C]** utilisée pour recevoir des données série.
- **Utilisation :** Ces broches servent à la programmation et au débogage de l'ESP32 via UART.

3. Interface SPI (Serial Peripheral Interface)

- **CS (Chip Select)** : Broche permettant de sélectionner un périphérique SPI.

- **SCLK (Serial Clock)** : Broche **1[1B]**, **3[2B]** qui génère l'horloge pour la communication SPI.
- **MOSI (Master Out Slave In)** : Broche **MOSI** permet l'envoi de données du maître vers l'esclave.
- **MISO (Master In Slave Out)** : Broche **MISO** pour recevoir les données de l'esclave.
- **Utilisation** : SPI est utilisé pour connecter des composants externes comme des capteurs, écrans, ou mémoires externes.

4. Interface I2C (Inter-Integrated Circuit)

- **SCL (Serial Clock Line)** : Broche d'horloge **1[1D]**, **3[2C]**, **3[3B]**.
- **SDA (Serial Data Line)** : Broche de données **1[1C]**, **3[2C]**, **3[3B]**.
- **Pull-up Résistances** : On observe des résistances **R601** et **R602** de **2.4kΩ** sur SDA et SCL pour garantir des niveaux logiques corrects.
- **Utilisation** : L'I2C permet de connecter des périphériques comme des capteurs INA ALERT ou TMP ALERT.

5. Boutons et Entrées/Sorties (GPIO)

- **Boot et Reset** :
 - **Boot** : Connecté à **4[2A]**, permet de mettre le MCU en mode programmation (Flash).
 - **Reset** : Connecté à **4[2C]**, réinitialise l'ESP32.
- **Buzzer** :
 - Connecté sur **GPIO** dédié, activé par un signal logique.
- **LED** :
 - Connectée via une résistance à la broche **GPIO**, utilisée pour des signaux visuels simples.

6. Autres Broches

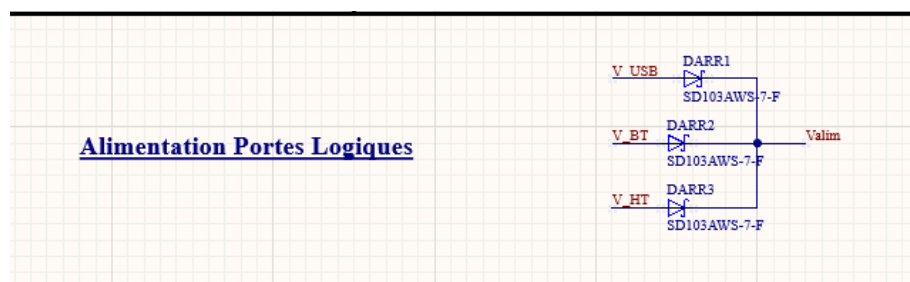
- **INA ALERT et TMP ALERT** :
 - Connectés à des GPIO pour remonter des alertes provenant de capteurs externes.
- **LDNA_IN** :
 - Entrée numérique pour récupérer un signal externe.

Problèmes Rencontrés :

- 1) Alimentation du circuit analogique : comme le circuit analogique est celui qui doit déterminer l'alimentation du microcontrôleur on devait trouver une solution pour alimenter les blocs logiques, sans avoir l'idée de quelle alimentation est active ou pas.
- 2) Antenne Bluetooth :
Ayant des connaissances limitées en radiofréquence, l'antenne Bluetooth représente un véritable défi pour notre équipe. Nous avons étudié la fiche technique du composant et mené des recherches approfondies sur le web afin de mieux comprendre les principes de conception et d'acquisition liés à cette antenne.
- 3) PCB : Lors du routage, nous avons pu rencontrer différentes erreurs soit un total approximatif de 2000 erreurs notamment des erreurs "Clearance", "Solder Mask"

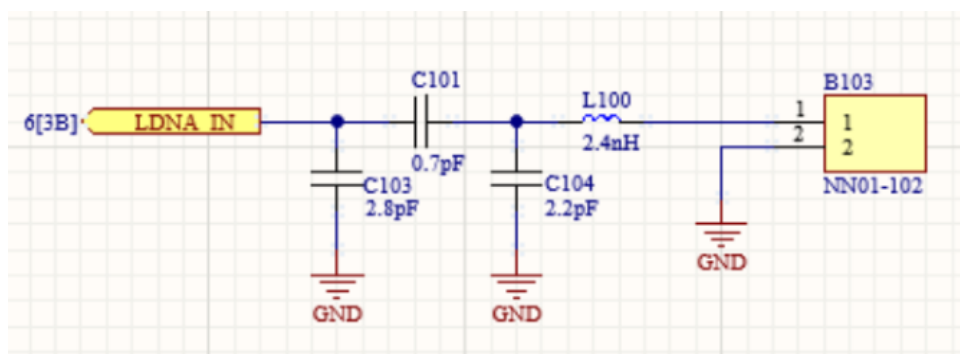
Solution :

- 1) Avoir toutes les sources comme possibilité d'activer les blocs logiques. De ce fait on devrait avoir les alimentations au même niveau d'alimentation (3V3) il fallait aussi prendre en compte le courant et d'éventuels retour du courant. C'est pour cela que les diodes ont été rajoutées. Et à la fin nous avons eu ce schéma :



- 2) Pour exploiter efficacement les signaux d'une antenne Bluetooth, l'utilisation d'un filtre est indispensable. Nous avons opté pour un filtre passe-bande, car notre objectif se concentre sur une plage de fréquences bien spécifique.

Après de nombreuses recherches et analyses, nous avons identifié le circuit qui répondait à nos besoins, comme décrit dans l'annexe 04.



- 3) Nous avons étudié les types d'erreurs une par une et nous avons pu clear la majorité d'entre eux.

Une mauvaise stratégie d'entrée nous a malheureusement bloqué pour finaliser la correction et dernières erreurs, sans refaire le routage de toute la carte PCB.

