

CONCEPTION ELECTRONIQUE

PROJET CARTE DE DÉVELOPPEMENT ESP32

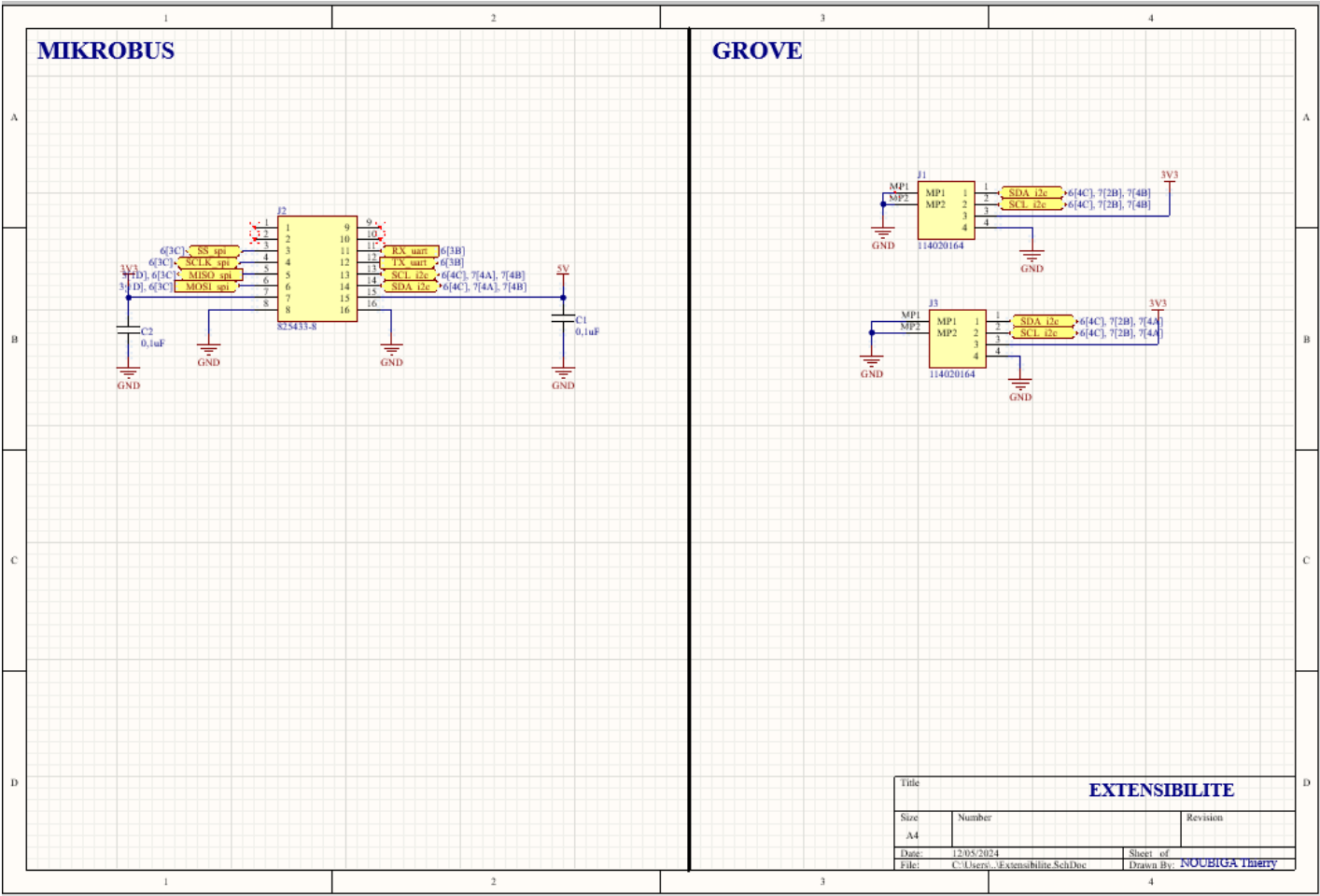
1. INTRODUCTION

Dans le cadre de notre cours de conception électronique, il nous a été demandé de réaliser une carte de développement avec un microcontrôleur esp32-PICO-D4 à l'aide du logiciel de conception électronique altium designer.

2. ANALYSES FONCTIONNELLES

1) EXTENSIBILITE

Sur le schéma ci-dessous nous avons un MIKROBUS et un GROVE qui sont des modules d'extensions électroniques ou sont connecte des protocoles de communication I2C, UART et SPI. Sur ce schéma, on voit les connexions des interfaces de communication standard (SPI, I2C, UART) qui seront utilisées pour relier différents modules ou capteurs à l'ESP32-PICO-D4, le microcontrôleur principal de notre projet.



EXTENSIBILITE

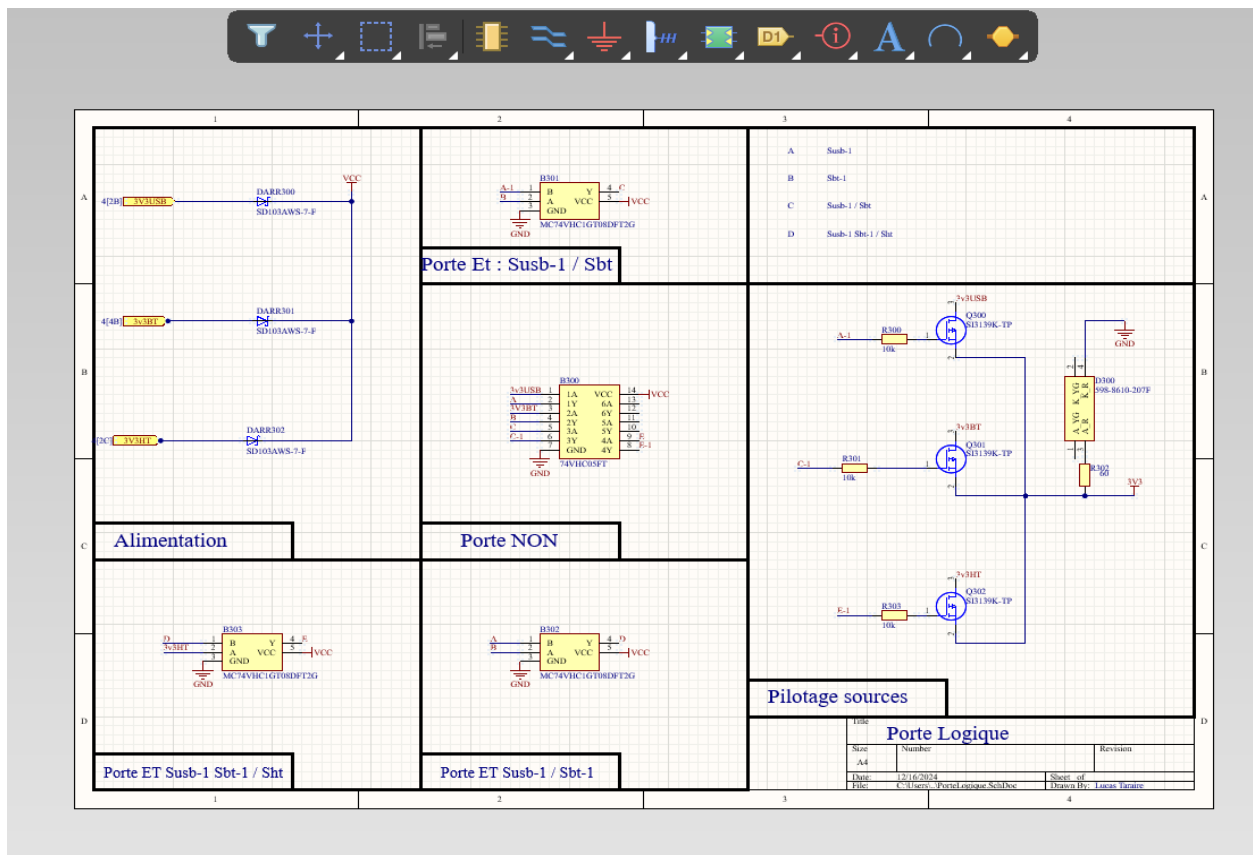
2) PORTES LOGIQUES

Description : Schéma comprenant des portes logiques (AND et NOT) pour gérer les signaux. Les transistors MOSFET (SI3139K-TP) sont utilisés pour piloter des sorties en logique.

Composants clés :

- MC74VHC1GT08DFT2G : Porte AND.
- SI3139K-TP : MOSFET N-channel.
- Résistances : 10kΩ.

Fonction principale : Pilotage de sources 3.3V via des portes logiques



PORTE LOGIQUE

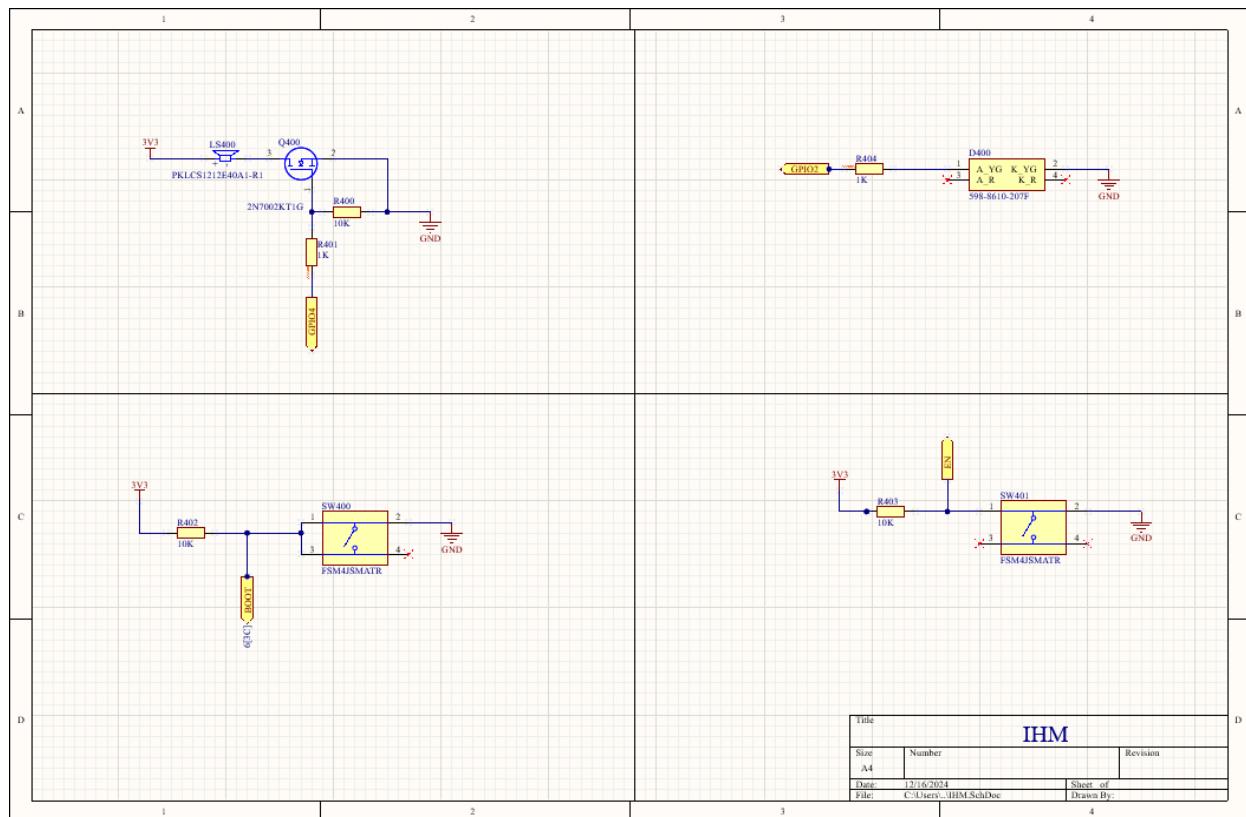
3) IHM (Interactions humain-machine)

Description : Schéma d'une interface utilisateur avec un buzzer, deux boutons-poussoirs, et une LED bicolore.

Composants clés :

- PKLCS1212E40A1-R1 : Buzzer piézoélectrique.
- FSM4JSMATR : Boutons-poussoirs.
- 598-8610-207F : LED bicolore.
- 2N7002KT1G : Transistor MOSFET pour piloter le buzzer.

Fonction principale : Contrôle des boutons (RESET/BOOT) et du buzzer via des GPIOs.



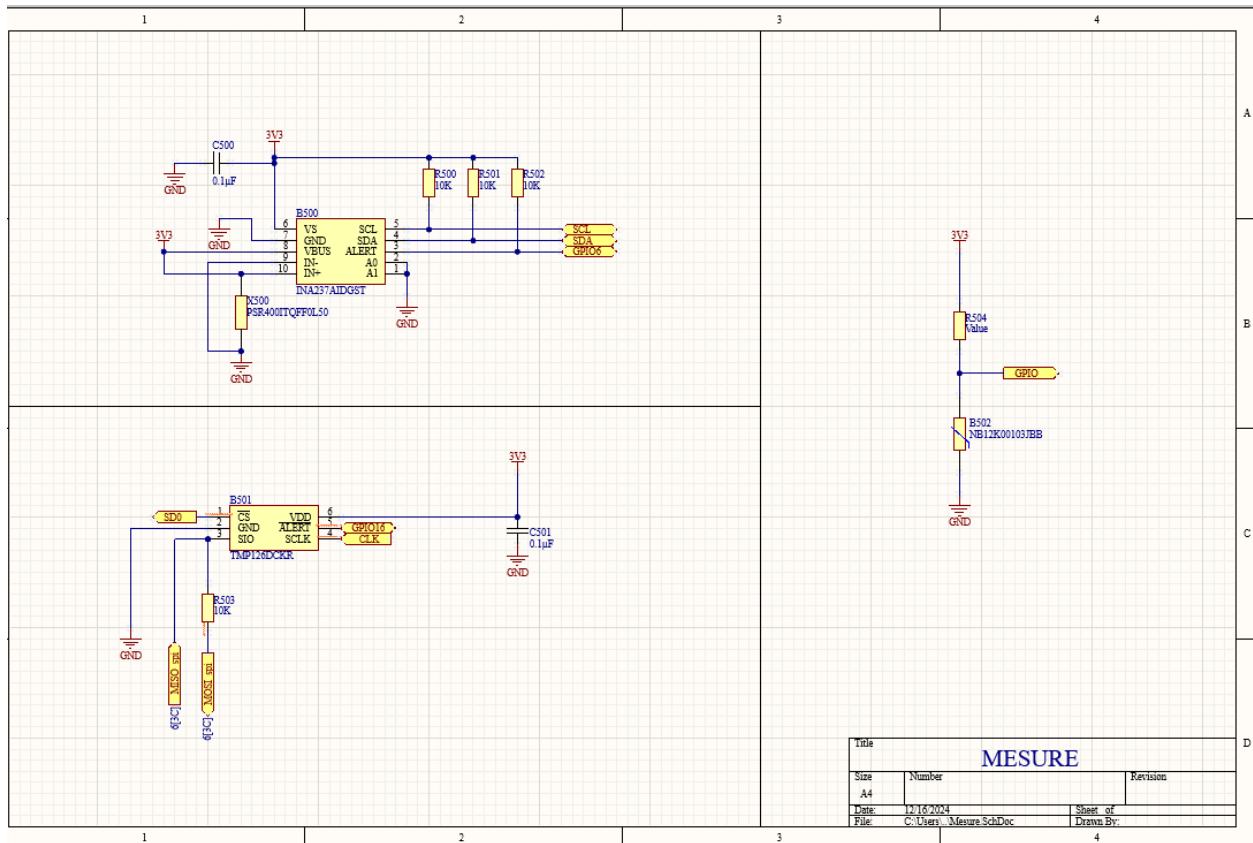
INTERFACE HOMME-MACHINE

Description : Schéma pour la mesure de tension et courant via un capteur INA237 et une résistance shunt.

Composants clés :

- INA237AIDGST : Moniteur de puissance I2C.
- PSR400ITQFF0L50 : Résistance shunt.
- TMP126DCKR : Capteur de température SPI.

Fonction principale : Surveillance du courant, tension, et température.



MESURE

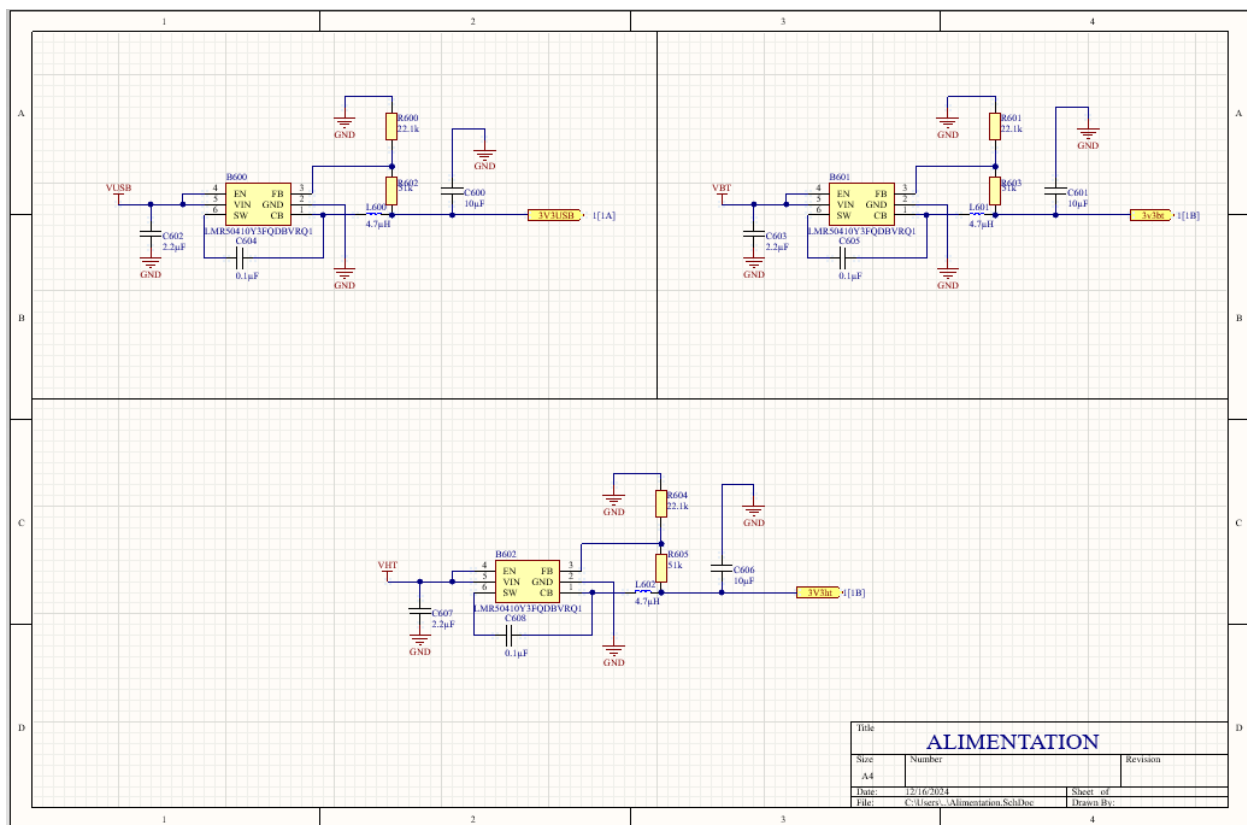
5) ALIMENTATION

Description : Circuit d'alimentation basé sur des régulateurs DC-DC LMR50410 pour générer différentes tensions.

Composants clés :

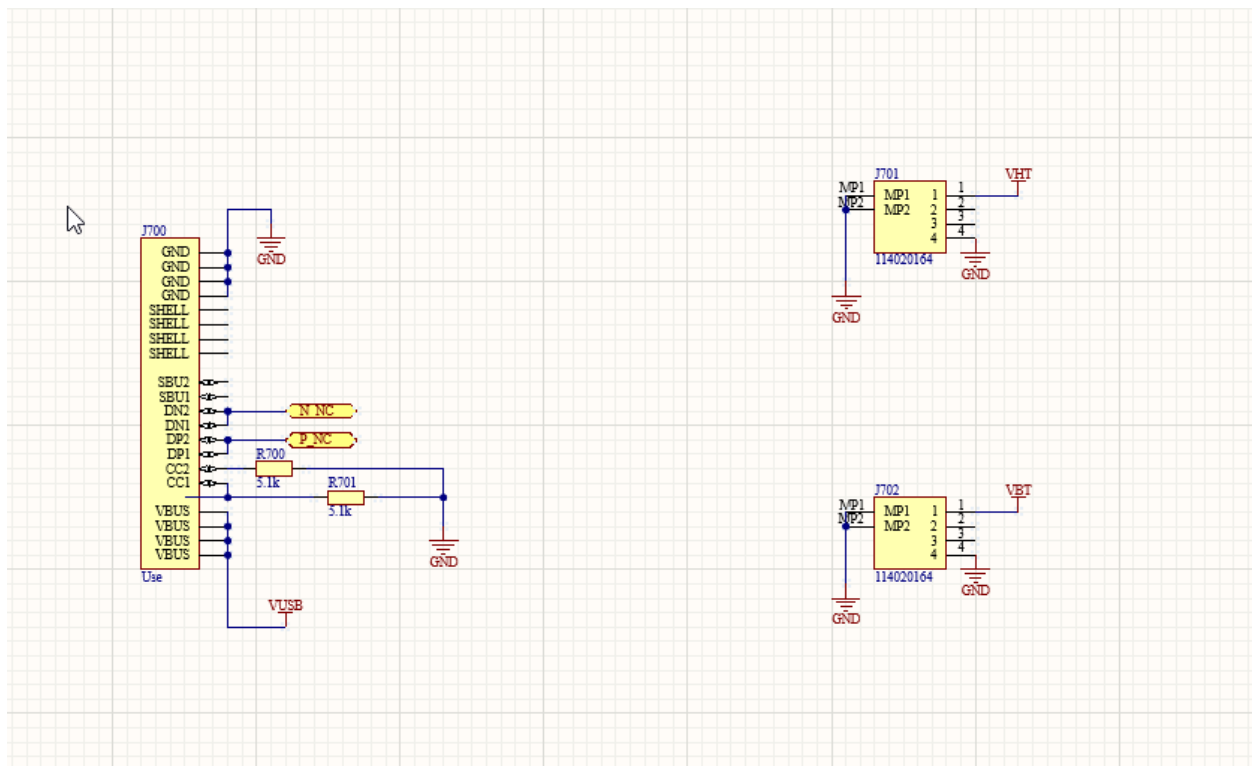
- LMR50410Y3FQDBVRQ1 : Régulateur buck 36V \rightarrow 3.3V.
- Condensateurs : 10 μ F, 0.1 μ F, 2.2 μ F.

Fonction principale : Fournir 3.3V pour différentes sections du circuit.



ALIMENTATION

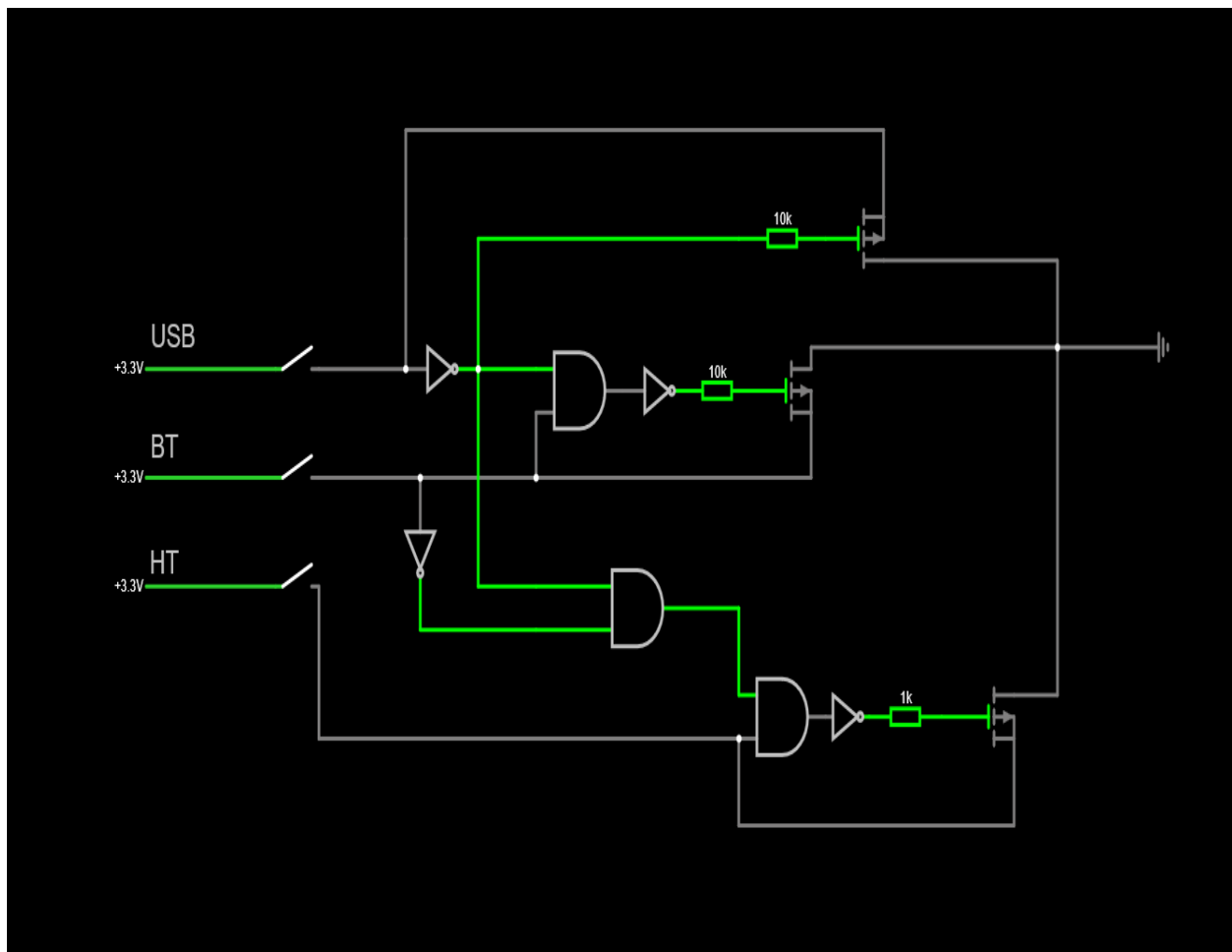
- **Description** : Connecteurs USB et GPIO pour l'alimentation et les communications.
- **Composants clés**:
 - Connecteurs USB4085-GF-A.
 - Résistances 5.1k Ω pour les lignes CC1/CC2.
- **Fonction principale** : Connecter et gérer l'alimentation via USB et GPIO.



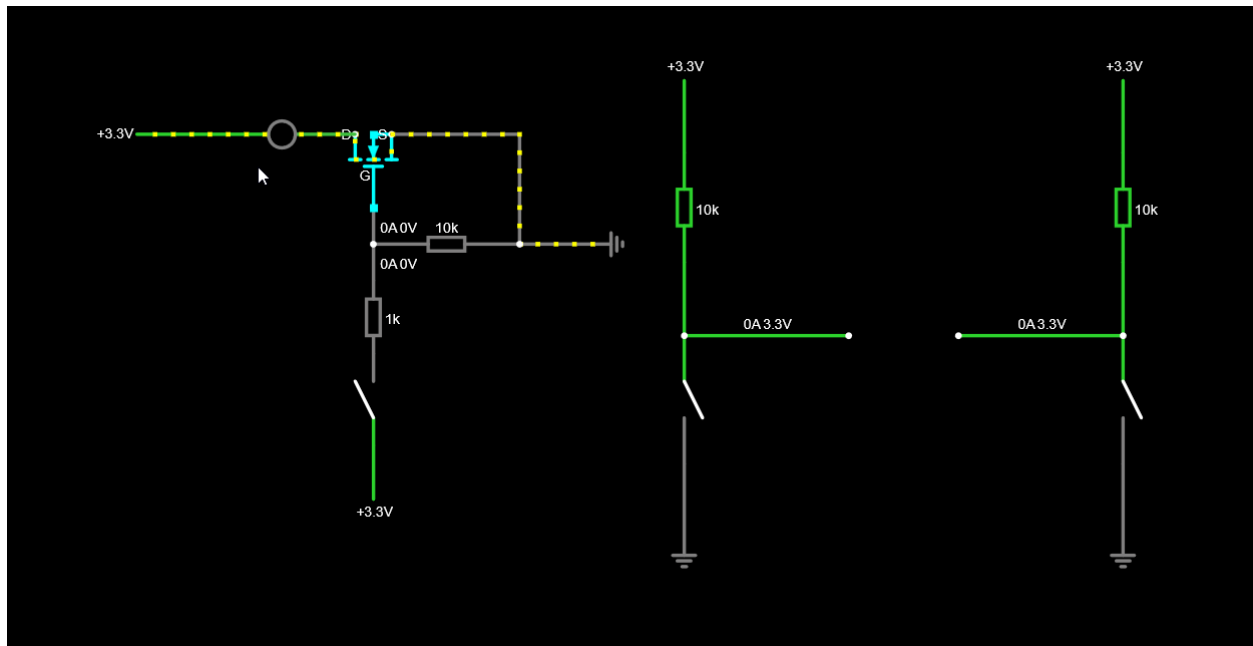
SOURCES

-
- COMMUNICATION BLUETOOTH**
- MICROCONTROLEUR**
- MCU et Communication Bluetooth
- Size A4 Number Revision
- Date: 12/16/2024 Sheet of 1
- File: D:\View - MCU et Communication Schematic By: NOUBIGA Thierry

8) SIMULATION DANS FALSIAD

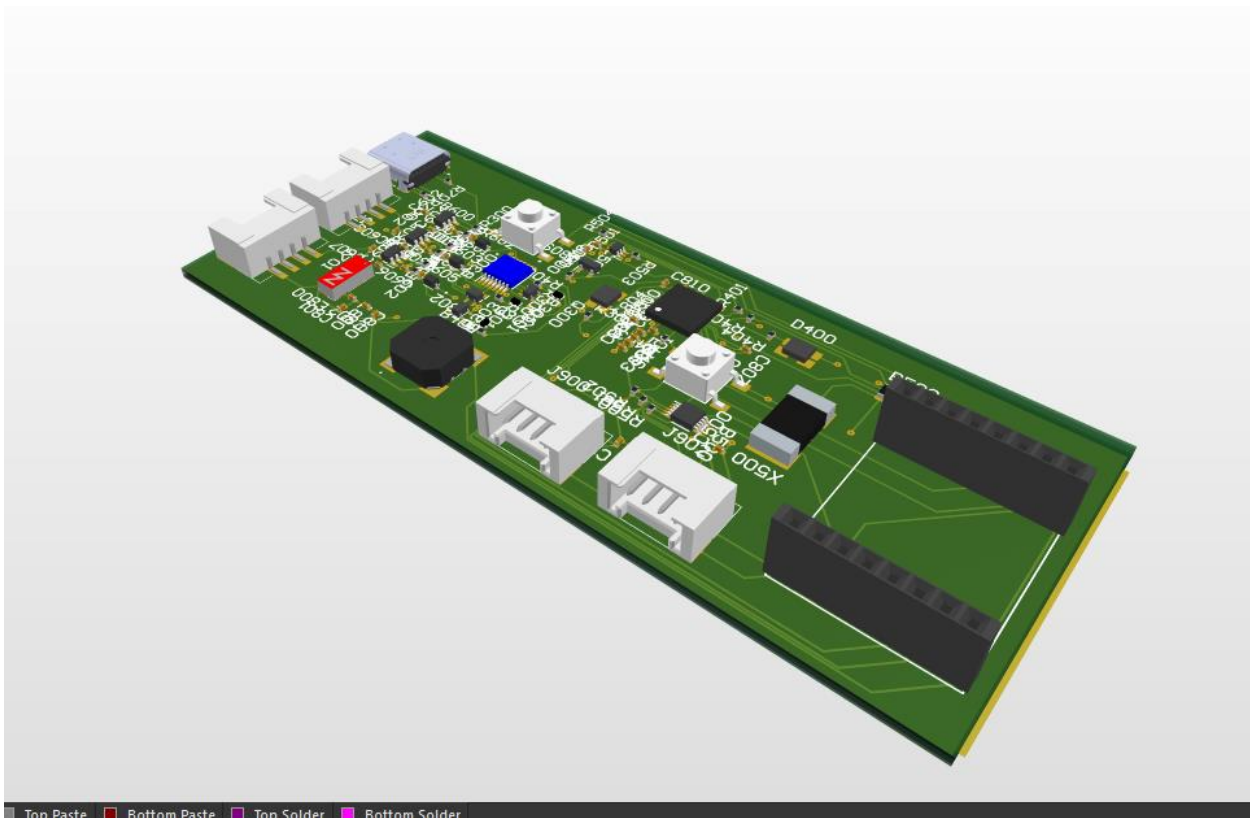
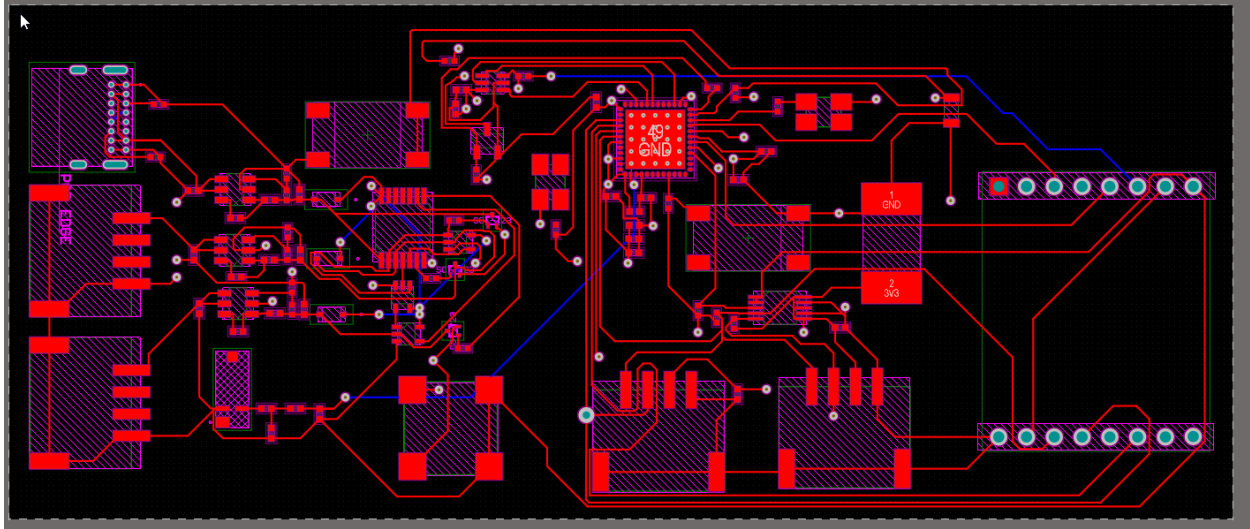


CIRCUIT D'ALIMENTATION



INTERFACE HOMME-MACHINE

4. PRINTED BOARD CIRCUIT



5. PROBLEMES ET SOLUTIONS

a. PROBLEMES

Au cours de la réalisation de ce projet de groupe, nous avons dû faire face à plusieurs problèmes, dont nous allons énumérer deux dans les lignes qui suivent.

Lors de la création de la schématique pour le système logique, j'ai rencontré des difficultés à organiser les différentes connexions entre les entrées (USB, BT, HT), les portes logiques (AND, NOT) et les sorties vers les MOSFETs. Le problème principal était lié à:

1. **La complexité croissante des connexions** : Les multiples portes logiques ajoutaient beaucoup de câbles, rendant la schématique difficile à lire.
2. **Conflits d'activation des MOSFETs** : À certaines conditions, plusieurs sorties logiques pouvaient activer les transistors simultanément, causant un dysfonctionnement.

Causes identifiées:

- Les portes logiques AND et NOT créaient des sorties qui ne respectaient pas toujours les priorités définies dans le cahier des charges (exemple : prioriser l'USB lorsque toutes les entrées sont actives).
- Une mauvaise gestion des connexions et du routage des **signaux** compliquait la lisibilité et introduisait des erreurs logiques.

b. SOLUTIONS

1. **Optimisation de la logique:**

- Analyse et simplification des conditions logiques à l'aide d'une table de vérité pour éviter les conflits entre les sorties.
- Utilisation de portes NOT pour inverser les signaux lorsque nécessaire afin d'activer les MOSFETs correctement.

2. Organisation de la schématique:

- Réorganisation des entrées (USB, BT, HT) sur la gauche, des portes logiques au centre, et des MOSFETs sur la droite.
- Ajout de net labels pour éviter des connexions encombrantes et améliorer la lisibilité.
- Annotations claires pour expliquer chaque étape du système logique