

Dossier d'architecture

CREATION DE LA PARTIE ELECTRONIQUE D'UNE FUSEE

Leeroy MOOTHERY

A destination du Président du Jury

Grace NDOKO OUNOUNOU

Valentin Marchetti

20/01/2023

SOMMAIRE

| | |
|--|-----------|
| INTRODUCTION | 2 |
| ANALYSE CAPELLA | 2 |
| CHAPITRE 1 : CARTE DE TELEMETRIE | 4 |
| 1. INTRODUCTION DE LA CARTE | 5 |
| 1.1 PARTIE TELEMESURE..... | 5 |
| 1.2 PARTIE CAPTEURS..... | 7 |
| 1.2.1 ETUDE DES COMPOSANT..... | 7 |
| 1.2.2 BILAN ENERGETIQUE..... | 11 |
| 1.3 SCHEMATIQUE SUR EasyEDA..... | 12 |
| 1.4 CAS PRATIQUE DE LA CARTE | 14 |
| 1.4.1 TEST REALISEES..... | 14 |
| 1.5 TACHES A VENIR | 16 |
| CHAPITRE 2 : CARTE OBC (ON BOARD COMPUTER)..... | 17 |
| 1. CARTE OBC (ON-BOARD COMPUTER)..... | 18 |
| 2. OBJECTIFS DE LA CARTE OBC..... | 18 |
| 3. SYNOPTIQUE FONCTIONNEL | 19 |
| 3.1 SYSTEME AVEC CONNECTEUR JACK..... | 19 |
| 3.2 SERVO-MOTEUR + SYSTEME A PIGNON CREMAILLERE..... | 20 |
| 4. DESCRIPTION TECHNIQUE DE LA CARTE | 20 |
| 5. SCHEMA DES CIRCUITS DE LA CARTE OBC..... | 20 |
| 6. SUITE ET PERPECTIVES..... | 23 |
| 6.1 REALISER LE PROTOTYPE ET FAIRE UNE PREUVE DE CONCEPT (POC) | 23 |
| 6.2 CONCEPTION DU PCB ET ROUTAGE..... | 24 |
| CHAPITRE 3 : CARTE DE PUISSANCE | 25 |
| 1. OBJECTIF DE LA CARTE | 26 |
| 2. BILAN DE PUISSANCE DE LA CARTE SEQUENCEUR..... | 26 |
| 3. SOLUTION APPORTEE | 26 |
| 3.1 CHOIX DES COMPOSANTS..... | 26 |
| 3.1 SCHEMATIQUE | 28 |
| 3.2 PCB..... | 29 |
| 3.3 CARACTERISTIQUE DE LA CARTE | 30 |
| 3.4 CHOIX DE LA BATTERIE..... | 30 |
| 4. AXES D'AMELIORATION | 31 |
| CONCLUSION | 33 |
| ANNEXE SCHEMATIQUE DE LA CARTE OBC | 34 |

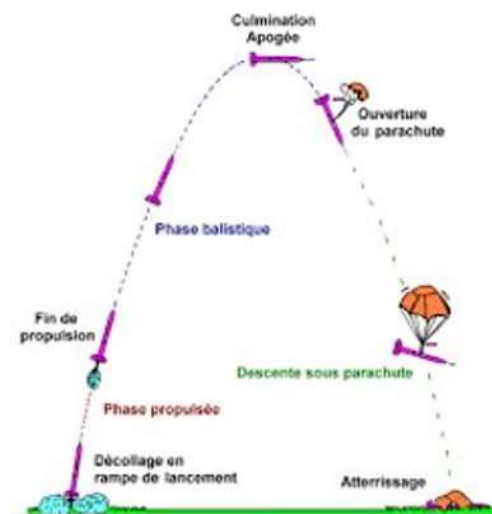
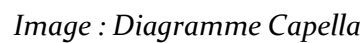
Introduction

Le projet Fusionic consiste en la création de 3 cartes électroniques à intégrer dans une fusée sur une durée d'1 ans. Les 3 cartes à créer sont les suivantes : une carte séquenceur ou ordinateur de bord ayant pour but de déclencher l'ouverture d'une trappe et, ainsi, libérer un parachute pour l'atterrissage de la fusée. La seconde carte à créer sera une carte de puissance qui aura pour rôle d'assurer l'alimentation de l'ordinateur de bord pendant le vol. Enfin, la dernière carte sera une carte de télémétrie ayant pour rôle de récolter des informations concernant l'environnement de la fusée (accélération, altitude, pression et température). Cette carte aura également pour but de transmettre ces informations à l'ordinateur de bord.

Il est également à noter que ce projet est évolutif. Cela signifie qu'à l'origine, le module à concevoir était destiné à une fusée pour le challenge CSPACE qui devait voler jusqu'à environ 3000mètre d'altitude. En revanche, il est aujourd'hui question d'une fusée sonde réalisée par plusieurs écoles qui ira jusqu'à une centaine de kilomètre d'altitude. Il faut donc prendre en compte que les prototypes réalisés jusqu'alors sont voués à évoluer car les conditions de fonctionnement ont changé. Nos cartes ont ainsi pour vocation de réaliser une prémices du projet final, de se familiariser avec les logiciels de CAO, les protocoles de communications au sol et l'architecture d'un tel projet.

Analyse Capella

Afin de comprendre à peu près comment nos cartes seront interconnectées entre elles et de voir comment elles seront structurées, nous avons fait une analyse fonctionnelle avec le logiciel capella. Cette analyse nous a permis de voir de façon un peu plus détaillée la façon dont on s'y prendra pour pouvoir atteindre les différents objectifs de nos cartes. Le diagramme ci-dessous est celui que nous avons obtenu après analyse.



CHAPITRE 1 : CARTE DE TELEMETRIE

1. INTRODUCTION DE LA CARTE

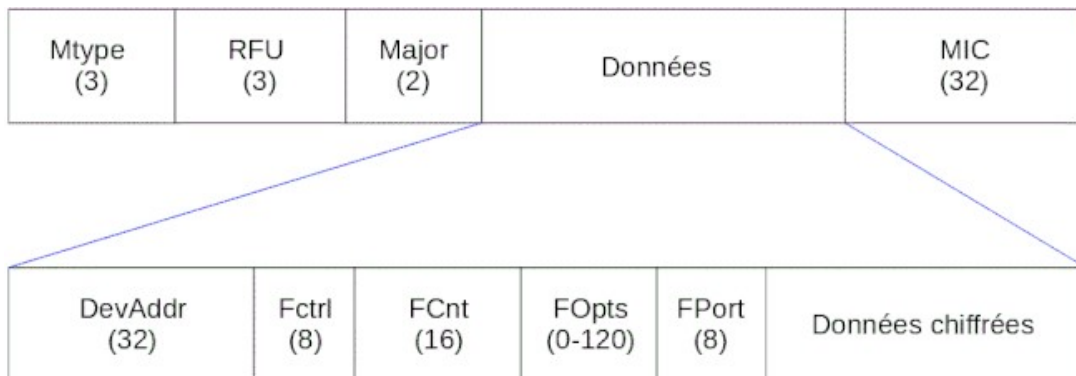
1.1 PARTIE TELEMESURE

Notre carte de télémétrie est une combinaison d'un module de télémétrie, elle s'occupe de la communication de la fusée avec le sol et d'une série de capteurs pour caractériser l'environnement de la fusée.

Afin de réaliser cette communication entre la fusée et le sol nous utilisons la technologie LORA, cette technologie est utilisée afin d'envoyer des données à faible débit sur de grande distance.

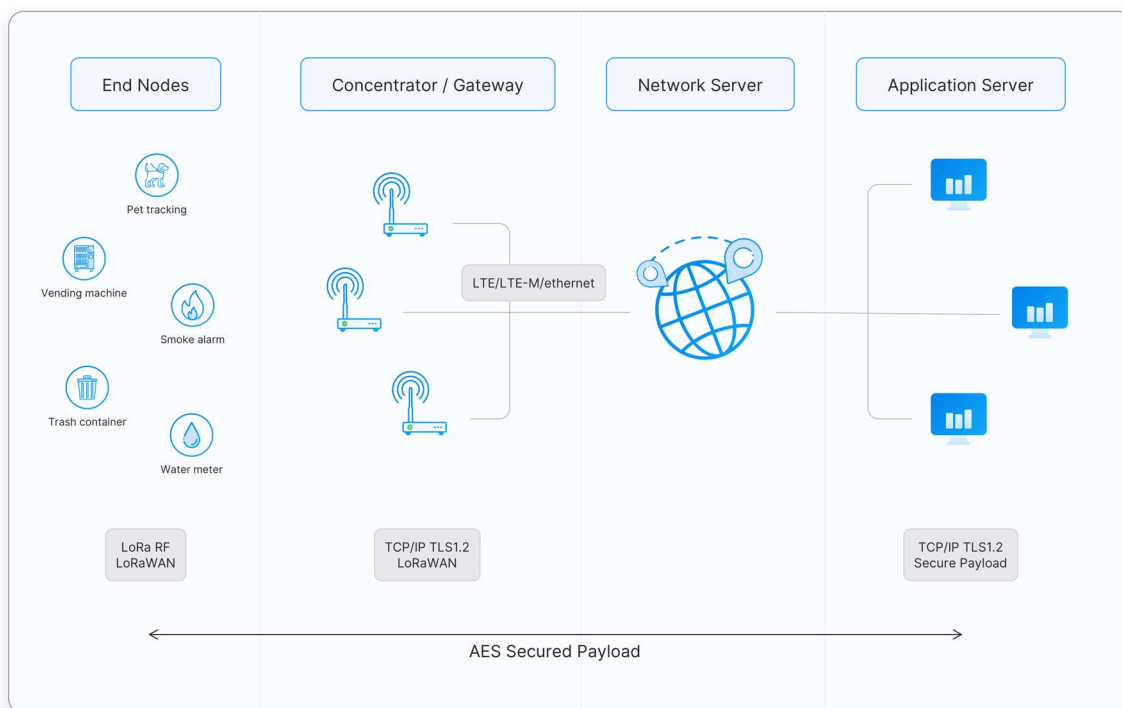
Plus précisément nous ne utilisons la technologie LORAWAN, c'est est un protocole de réseau qui est construit au-dessus de la modulation LORA, il est conçu pour connecter des objets électroniques à distance sur un réseau régional, national ou mondial.

Voici la trame de LORAWAN qui est composé de 5 parties. Parmi ces parties nous avons les données chiffrées, c'est une sous partie consacrée aux données de nos capteurs.



Trame LORAWAN

Voici l'architecture LORAWAN qui est composé de 4 étapes, avec l'end nodes, le gateway, le network server et enfin l'application server.



Architecture LORAWAN

A travers cette architecture nous pouvons comprendre que des End nodes (Périphériques terminaux, capteurs) envoient des messages à des gateway (Passerelles). Les gateway ont pour tâches de transférer les messages au network server (serveur de réseau). Le Network server est le logiciel gérant tout le réseau, il envoie les informations à l'application server (serveur d'application) qui va exécuter un traitement sécurisé des informations et les faire visualiser.

Canaux par défaut EU863-870Mhz

| Fréquence du canal (MHz) | Bande passante (kHz) | Débit de données LoRa | Débit binaire |
|--------------------------|----------------------|-----------------------|----------------|
| 868.10 | 125 | DR0 – DR5 | 0,3 à 5 kbit/s |
| 868.30 | 125 | DR0 – DR5 | 0,3 à 5 kbit/s |
| 868,50 | 125 | DR0 – DR5 | 0,3 à 5 kbit/s |

EU683-870 débit de données

Débit de données Configuration (SF + BW) Débit binaire (bit/s)


0 LoRa : SF12 / 125 kHz 250

1.2 PARTIE CAPTEURS

Ensuite, à travers la caractérisation de l'environnement de notre fusée, nous utiliserons différents capteurs pour mesurer l'accélération de la fusée lors du lancement (à l'aide d'un accéléromètre), l'altitude de la fusée (à l'aide d'un capteur de pression), l'état de la fusée (à l'aide d'un magnétomètre et d'un gyroscope) ainsi que les températures à l'intérieur et à l'extérieur de la fusée (à l'aide de capteurs de température).

1.2.1 ETUDE DES COMPOSANT

Choix et caractéristiques des composants



carte un microcontrôleur Pycom ESP32, LoPy4

| LoPy4 |
|--|
| Microcontrôleur : ESP32 |
| Alimentation : |
| -Tension : 3.3 à 5.5V |
| -Courant : 133mA |
| Interface : |
| -8E/S analogiques 12 bits |
| -2 x bus série, I2C, 2 x SPI et I2S |
| .Interface pour lecteur de carte microSD |
| Mémoire : |
| -Flash : 8MB |
| -RAM : 4MB |
| Prix : 48.90Euros TTC |

| LORA |
|---|
| Fréquence : 868MHz |
| Bande Passante : 125kHz |
| Puissance de sortie : 13dBm en transmission |
| Facteur d'étalement : 12 |
| Sensitivité : -137dBm |
| Bit rate : 293bit/s |
| Courant d'alimentation mode transmetteur : 28mA |
| Distance théorique : 46km |
| Distance expérimentale : 5615km |

La est composé d'un module LORA, et d'un processeur pour l'IOT (Internet Of Things). Composant robuste et résistant aux intempéries il est très efficace pour l'intégration d'un projet électronique à l'intérieur d'une fusée. En revanche le processeur ESP32 à d'haute performance, nous pourrions à travers la carte que télémétrie utilisé un processeur moins puissances, avec une puissance de calculs moins puissante afin de minimiser les consommations d'énergie. Enfin, il peut fournir jusqu'à 1.2A à des sorties (nos capteurs par exemple).

| Antenne molex |
|---|
| Polarisation : Linéaire |
| Diagramme de rayonnement : Omnidirectionnel |
| Impédance : 50 Ohms |



Antenne molex

| |
|-----------------------|
| Gain : 0.3dBi |
| Rendement total > 55% |

L'antenne molex est une antenne utilisée pour améliorer la portée et la fiabilité des communications sans fils à une fréquences de 868MHz.



Pytrack 2.0

| Pytrack 2.0 (carte d'extension LoPy4) |
|--|
| Alimentation : |
| -3.5V – 5.5V |
| -14mA active |
| Prix : 18.23Euros TTC |

| GPS |
|---------------------------------|
| Bande fréquence : 1575.2MHz |
| Acquisition < 1s |
| Sensibilité : |
| -Suivi : -167dBm |
| -Acquisition : -149dBm |
| Données : |
| -Taille : 16 octets |
| Altitude max mesurable : 18000m |

La carte Pytrack 2.0 est une carte d'extension de la carte LoPy4 permettant de lui rajouter des fonctionnalités. En effet elle permet de lui rajouter un port micro USB, un lecteur de carte SD et un GPS. Plus la sensibilité d'un GPS à une valeur basse mieux il recevra le signal GPS. Le GPS à une sensibilité d'acquisition de -25+dBm c'est-à-dire une très bonne capacité à acquérir un signal la première fois et à une sensibilité de suivi de -167dBm, c'est-à-dire une très capacité à suivre un signal déjà acquis. Ces caractéristiques nous montrent que ce composant convient parfaitement à l'acquisition de donnée GPS pour une fusée.



Accéléromètre, IMU
 MPU9250

| MPU9250 |
|---|
| Alimentation : |
| -Tension : 3 à 5V |
| -Courant : 3.7mA |
| Interface : I2C |
| Données : |
| -Taille : 6 octets |
| -Résolution : 16bits |
| Gamme du gyroscope : $\pm 2000^\circ/s$ |
| Gamme de l'accéléromètre : $\pm 16g$ |
| Gamme de magnétomètre : $\pm 4800\mu T$ |
| Prix : 18.23Euros TTC |

Le MPU9250 est un capteur à 3 fonctions, c'est-à-dire qu'il constitue lui uniquement 3 différents capteurs, un accéléromètre, un gyroscope et un magnétomètre. Il permet de connaître l'état de la fusée durant son vol.

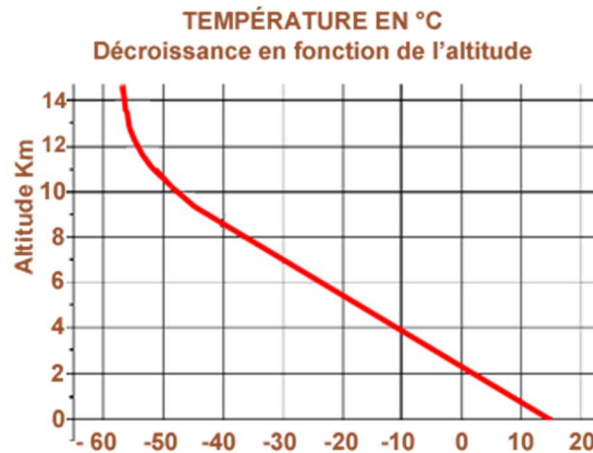


Capteur de température
 DS18B20

| DS18B20 |
|--|
| Alimentation : |
| -Tension 3 à 5.5V |
| -Courant : 1.5mA |
| Interface : 1Wire |
| Données : |
| -Taille : 2 octets |
| -Résolution : 9-12bits |
| Gamme de mesure : -55 à $+125^\circ C \pm 0.2^\circ C$ |
| Prix : 6.40Euros TTC |

Le DS18B20 est un capteur de température utilisant l'interface 1-wire. Cette interface est très compliquée d'utilisation. En effet dans la réalisation du code, cette interface implique une connaissance poussée en développement de carte, et sur l'interface 1-wire spécifiquement. Ce capteur a une gamme de mesure de -55 à $+125^\circ C \pm 0.2^\circ C$ ce qui est totalement en accord avec les températures environnantes de notre fusée. A travers Durant sa phase vol l'environnement de la fusée se modifiera, notamment la température qui variera. A l'aide de la formule $PV = nRT$, il est possible de retrouver une altitude en fonction de la température.

D'après la figure ci-dessous, nous pouvons voir qu'obtenir une précision de $\pm 0.2\text{kPa}$ est assez précis pour mesurer l'altitude en fonction de la température.



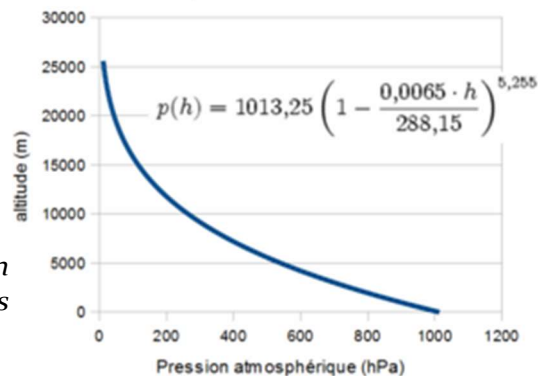
Courbe de l'altitude en km en fonction de la température en degré d'après AstroSurf.com



Baromètre MPX4250AP

| MPX4250AP | | |
|------------------------------------|-----|--|
| Alimentation : | | |
| -4.85 à 5.35V | | |
| -Courant consommé | 7mA | |
| typiquement et 10mA max | | |
| Interface : Bridge | | |
| Données : | | |
| -Taille : 16 octets | | |
| -Résolution : 9-12bits | | |
| Gamme de mesure : | | |
| +20 à +250kPa $\pm 0.35\text{kPa}$ | | |
| Prix : 21.47Euros TTC | | |

Pour le capteur MPX4250AP, nous avons un capteur permettant de mesurer sur une gamme de mesure de +20 à +250kPa $\pm 0.35\text{kPa}$ (3.5hPa), sachant que notre fusée théoriquement volerait jusqu'à 3000 mètres d'altitude, d'après la figure ci-dessous, notre capteur est assez sensible afin de pouvoir mesurer l'altitude de notre fusée pendant sa phase de vol



Graphique 1: Courbe de l'altitude en m en fonction de la pression en hPa d'après Lavionnaire.fr

1.2.2 BILAN ENERGETIQUE

| Symbole | Description | Min. | Typ. | Max. | Unités |
|---------|-----------------------|----------|------|----------|--------|
| V | Regulated 3.3V supply | | 3.3 | | V |
| VIL | Input low voltage | -0.3 | | 0.25*3.3 | V |
| VIH | Input high voltage | 0.75*3.3 | | 0.3+3.3 | V |
| VOL | Output low voltage | 0.1*3.3 | | | V |
| VOH | Output high voltage | 0.8*3.3 | | | V |

Valeurs maximales des paramètres électriques

| Composants | Commentaires | Min. | Typ. | Max. | Unités |
|------------|------------------------|----------|-------|---------|--------|
| MPU9250AP | Normal mode | | 3.7 | | mA |
| MPU9250AP | Low-Power mode | | 19.8 | | μA |
| DS18B20 | Input high voltage | 0.75*3.3 | | 0.3+3.3 | mA |
| DS18B20 | Output low voltage | 0.1*3.3 | | | nA |
| MPX4250AP | Output high voltage | 0.8*3.3 | | | mA |
| Lopy4 | Avec transmission LORA | | 133 | | mA |
| Lopy4 | Deep sleep | | 19.5 | | mA |
| Total | | | 19.1 | | mA |
| Total | | | 144.4 | | mA |

Courant de consommation à 25°C

1.3 SCHEMATIQUE SUR EasyEDA

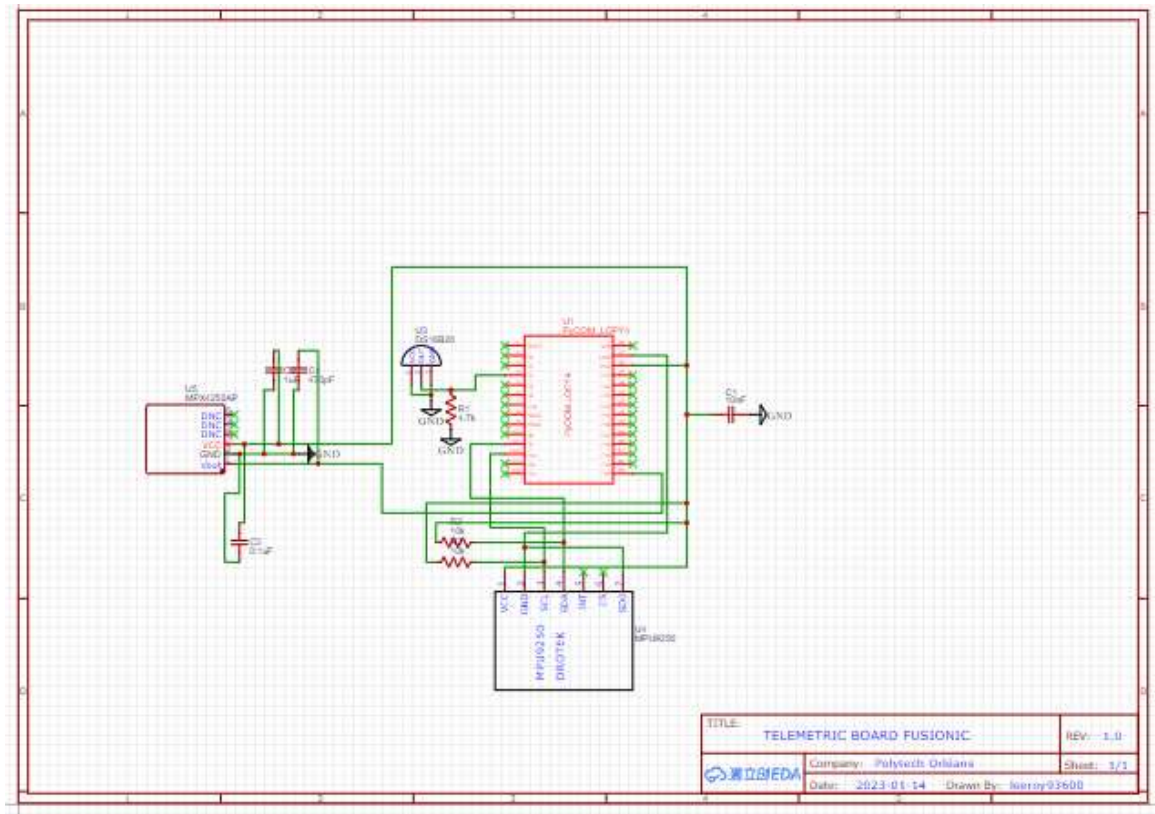
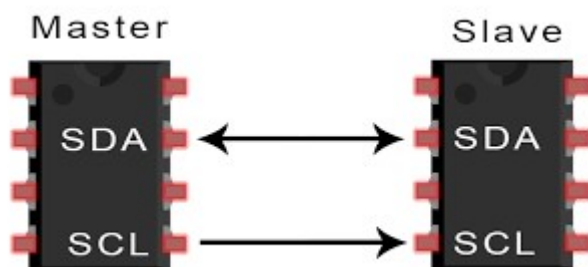
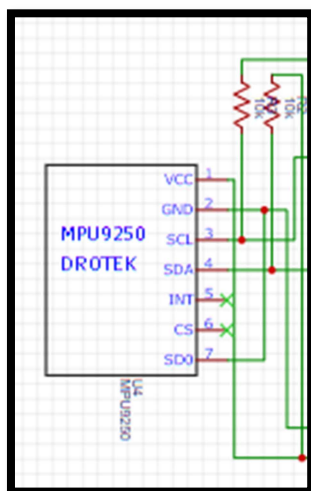


Image 1 : Schématique carte télémétrie sans Pytrack 2.0

A travers cette partie je vais vous présenter la réalisation de la schématique de ma carte de télémétrie, en revanche la carte d'extension Pytrack 2.0 ne figure pas sur le schéma car elle est conçue pour s'imbriquer directement à la LoPy4 formant un unique composant

MPU9250



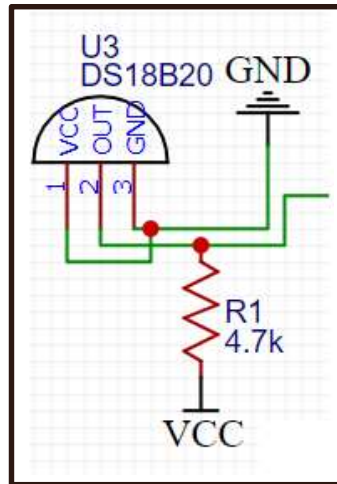
Branchement communication I2C

SDA : Ligne de données série, partagée par tous les périphériques sur le bus.

SCL : Horloge série, dirige le débit de données pour la transmission de données

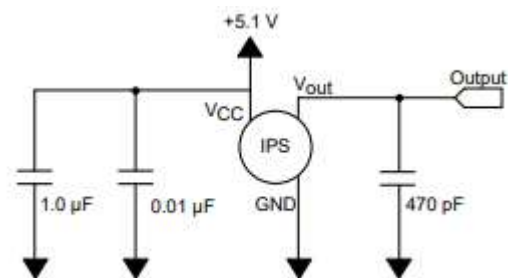
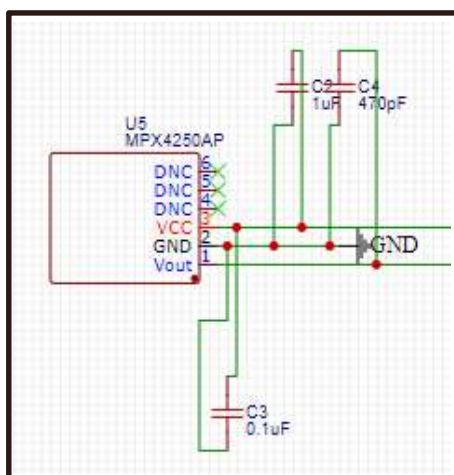
Deux résistances de pull up de 10kOhm sont également nécessaire car elles permettent de mettre le signal de sortie à l'état haut sinon nous obtenions un signal flottant.

DS18B20



En 1-Wire la réalisation du schéma est très simple car nous devons connecter uniquement le pin de donnée à une entrée numérique sur notre microcontrôleur et le reste nous le branchons à la masse. Nous devons mettre une résistance de 4.7kOhm à notre branche de donnée afin d'assurer une connexion fiable entre le capteur et la LoPy4.

MPX4250AP

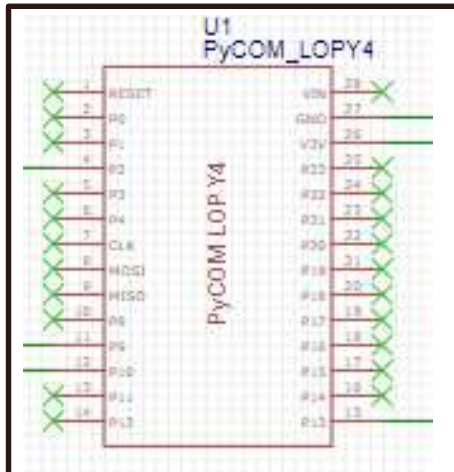


Branchement datasheet

Afin d'effectuer le branchement du MPX4250 AP il suffit de brancher la masse avec la masse de la LoPy4, de brancher le VCC avec le pin 3V3, et enfin le signal de sortie nous l'avons

branché à une sortie numérique. Afin de placer les condensateurs, nous nous sommes aidés de la datasheet.

LoPy4



Pins SDA / SCL extrait datasheet LoPy4

| | | | | | | | |
|----|----|-----|-----------------------|----|---|---|----------|
| 11 | 12 | P9 | SDA | 2* | • | • | JTAG TDI |
| 12 | 13 | P10 | SCL (I2C) / CLK (SPI) | 2* | • | • | JTAG TCK |

Nous voyons que laLoPy4 à bien les pins adaptés à une communication I2C.

1.4 CAS PRATIQUE DE LA CARTE

1.4.1 TEST REALISEES

Dans cette partie, on s'intéresse principalement aux tests réalisés. La carte LoPy4 est composé d'un microcontrôleur Pycom ESP32, la programmation de cette carte se fera à l'aide du langage de programmation Python.

Pour ce faire j'ai développé mes codes Python à l'aide de l'IDE Visual Studio Code. A l'intérieur de ce logiciel, j'ai dû ajouter une extension nommée « Pymakr » afin de pouvoir développer des codes sur ma carte. J'ai dû également utiliser la plateforme nommée « Node.js » afin de pouvoir exécuter mes codes.

J'ai alors débuté par le GPS. Voici ci-dessous les résultats du terminal que j'ai obtenu.


```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
(47.84541, 1.928373)
(47.84541, 1.928373)
(47.84541, 1.928373)
(47.8454, 1.928378)
(47.84541, 1.928378)
(47.84541, 1.928378)
(47.84541, 1.928378)
(47.84547, 1.928425)
(47.84548, 1.928415)
(47.84548, 1.92841)

```

Ln 44, Col 4 Spaces: 4 UTF-8 LF Python Go Live

Résultats codes python GPS (Longitude, Latitude)

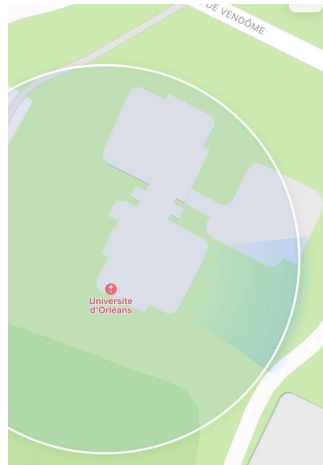
En déplaçant ma carte nous voyons quelques variations de la longitude et de la latitude, très minime, cela témoigne de la précision de la position à laquelle le GPS a localiser la carte.



Résultats localisation Google Maps (Longitude, Latitude)

| x | Longitude (°N) | Latitude (°E) |
|-------------|----------------|---------------|
| Pytrock 2.0 | 47.84548 | 1.928373 |
| Google Maps | 47.84550 | 1.92840 |

Le tableau ci-dessus nous montre que les coordonnées récupéré avec notre GPS sont excellente. Les erreurs se manifestent due à des erreurs de précisions sur google maps quand je choisie ma position. En effet afin de trouver sa longitude et sa latitude nous devons sélectionner notre position avec notre doigt.



Carte Google Maps

Après avoir mesurer notre position à l'aide du GPS, nous avons travaillé sur le capteur de température néanmoins nous n'avons pas réussi à l'élaboration des codes python. Ces difficultés s'expliquent car l'interface du capteur DS18B20 est 1-wire. Il est très compliqué de travailler avec cette interface, mon responsable de projet ma alors conseillé de changer de capteur.

1.5 TACHES A VENIR

La prochaine tâche serait donc de trouver un nouveau capteur de température. Ensuite nous aimerons utiliser la transmission de donnée en LORAWAN, pour cela nous devons procéder à l'aide du site web « The Things Network » permettant l'enregistrement de données.

Enfin, nous continuons à développer les codes afin d'obtenir l'acquisition de données par la totalité de nos capteurs.

CHAPITRE 2 : CARTE OBC (ON BOARD COMPUTER)

1. Carte OBC (On-Board Computer)

L'OBC : ordinateur de bord, également appelé ordinateur embarqué ou séquenceur, est un type de micro-ordinateur qui est intégré à la fusée. Elle est la carte maîtresse du dispositif électronique. Elle doit gérer la procédure de vol de la fusée de la rampe de lancement à l'atterrissage.

2. Objectifs de la carte OBC

Dans le cadre de notre projet, la carte OBC a principalement deux objectifs :

- Son objectif premier est de détecter le décollage de la fusée. Pour ce faire, elle indique d'abord, à travers une DEL (diode électroluminescente) de couleur verte, que la fusée est installée sur la rampe de lancement et est prête à être lancée. Ensuite, elle détecte le décollage de la fusée et l'indique aussi à travers une DEL de couleur jaune cette fois-ci.
- Son deuxième objectif, est de commander l'ouverture de la trappe afin que le parachute embarqué à bord de la fusée soit déployé lorsque cette dernière atteint l'apogée. Le temps prévu pour que la fusée atteigne l'apogée est de 20 secondes.

En soi, les deux objectifs sont entièrement liés, puisqu'il n'est possible d'atteindre l'apogée que si la fusée décolle.

3. Synoptique fonctionnel

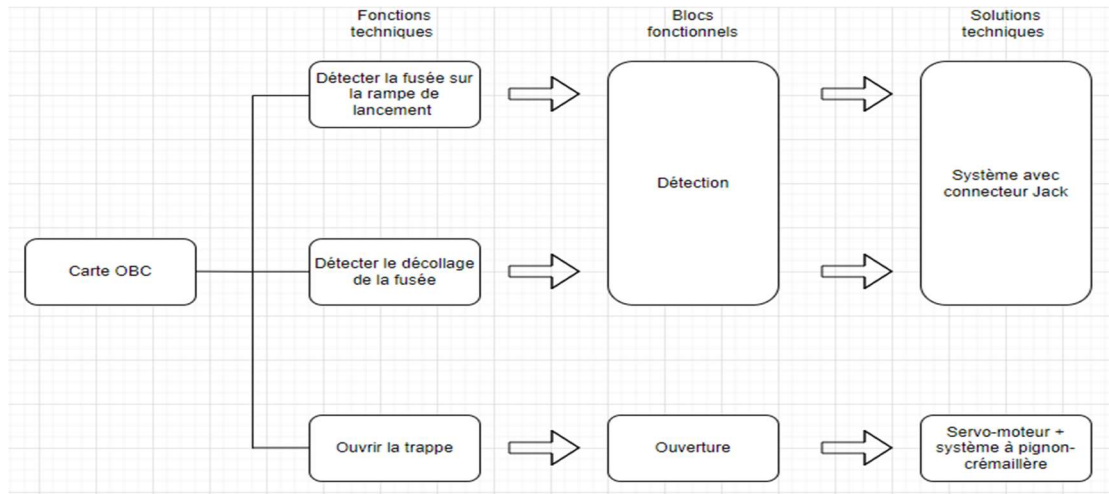


Figure 1 : Diagramme fonctionnel - Source : Hélioth NDOKO

Les solutions techniques citées ci-dessus sont interfacées entre elles via un microcontrôleur. Et pour ce qui est des contraintes que nous avons sur la carte OBC, elles sont toutes groupées sur le fait que la carte doit tenir dans la fusée durant toutes les phases de vol ; c'est-à-dire du décollage à l'atterrissage. Donc elle doit être circulaire et doit faire 10 cm de diamètre.

3.1 Système avec connecteur Jack

Ce système est composé de deux connecteurs jacks mâle et femelle. Le connecteur mâle est en court-circuit et le connecteur femelle est quant à elle reliée au microcontrôleur. Le système est tel que lorsque la fusée est installée sur la rampe de lancement, le connecteur mâle, dont la partie en court-circuit est attachée à la rampe de sorte à l'arracher au décollage, est enfiché au connecteur femelle. Ce qui permettra au microcontrôleur de voir le court-circuit et à travers un programme informatique, de comprendre que la fusée est sur la rampe de lancement. Ainsi donc, lorsque le connecteur mâle est arraché, le microcontrôleur ne voit plus le court-circuit et comprend que la fusée a décollé.

3.2 Servo-moteur + système à pignon crémaillère

Ce système permettra d'ouvrir la trappe grâce à la crémaillère qui sera entraîné en mouvement de translation grâce au pignon qui lui sera en rotation grâce au servo-moteur. Le servo moteur est quant à lui commander par le microcontrôleur. Pour rappel, l'ouverture de la trappe se fait à l'apogée et permet de déclencher le parachute pour un atterrissage en douceur de la fusée.

4. Description technique de la carte

Comme dit un peu plus haut, la carte doit être circulaire et doit faire 10 cm de diamètre. Elle intègre un microcontrôleur qui permettra à travers les différentes solutions techniques de faire la détection de la fusée sur la rampe, la détection du décollage et l'ouverture de la trappe afin de réaliser les fonctions techniques. Elle est embarquée à la fusée.

5. Schéma des circuits de la carte OBC

Le schéma des circuits de la carte OBC se trouve en annexe de ce document.

Les principaux composants présents sur la carte sont :

- Le microcontrôleur : Raspberry Pi RP2040S, son rôle est de faire tous les calculs de la carte. C'est le cerveau de la carte OBC. Elle lit et agit en fonction des informations que lui renvoie le système de détection. Lorsque le décollage est détecté, il lance un compte à rebours de 20 secondes avant de commander l'ouverture de la trappe.



Image : Microcontrôleur RP2040

Microcontrôleur RP2040

Alimentation : 1.8 à 5.5 V

Microprocesseur : ARM Cortex-M0+ Dual
 Core à 133 MHz

Interface compatible PWM : oui

- Un connecteur USB-B : Permettant uniquement de programmer le microcontrôleur.



Image : Connecteur USB type B femelle

USB_MICRO-type B femelle

Alimentation

Courant nominale : 0.5A

- Un régulateur de tension : le NCP1117, son rôle est de réguler la tension de 5 Volts à 3,3 Volts.

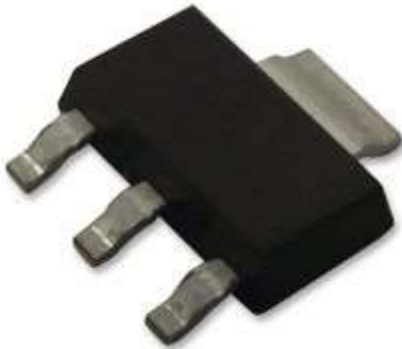


Image : NCP1117-3-3_SOT223

NCP1117-3-3-SOT223

Type de sortie : fixe

Alimentation :

Tension entrée : 3.5 – 20V

Tension de sortie fixe : 3.3V

Courant de sortie : 1A

- Un oscillateur : Le ABL5-12.000MHz permettant de définir une fréquence à compléter en relisant la datasheet.



Image : Cristal de quartz

Le ABLS-12.000MHz

Fréquence : 12MHz

Résistance ESR : 50 Ohms

- Un transistor MOSFET : Le IRF520, faisant la commutation et commandant ainsi le servo-moteur.



Transistor MOSFET IRF520

Tension VDS : jusqu'à 100 V

VGS-th (max) : 4 V

IDS : jusqu'à 9.2 A

Il faut préciser que le moteur à commander est le HS-422 et n'est pas présent sur la carte, donc ne sera pas présenter ici. Ce moteur nécessite un signal PWM (pour Pulse Width Modulation en anglais) d'environ 3 à 5 volts d'amplitude et un courant variable de 150 mA à 800 mA en fonction de s'il fonctionne à vide ou non.

- Une mémoire Flash : ce composant, cette mémoire flash est celui qu'utilise le microcontrôleur RP2040 pour comme mémoire flash et mémoire programme. Car celui-ci n'intègre pas de mémoire à son bord.



Image : Mémoire flash W25Q128JVSIM

Mémoire flash W25Q128JVSIM

Alimentation :

Fonctionnement minimum : 2.7 V

Fonctionnement maximum : 3.6 V

Taille mémoire : 128Mbit

Type d'interface : Quadruple SPI

Temps d'accès aléatoire max : 6ns

Température de fonctionnement : -40 à +85 °C

Voilà à peu près les principaux composants se trouvant sur la carte OBC.

Pour l'instant, aucun test n'a encore été fait pour la carte OBC avec tous ces composants du fait que, le microcontrôleur que nous avons, est endommagé.

6. SUITE ET PERSPECTIVES

Dans la suite du projet, ce que nous envisageons de faire après réception d'un microcontrôleur en bon état et qui fonctionne c'est, dans un premier temps :

6.1 REALISER LE PROTOTYPE ET FAIRE UNE PREUVE DE CONCEPT (POC)

En effet, réaliser la preuve de concept nous permettra non seulement, de valider les choix des solutions techniques faites, de les tester mais aussi de valider cette première partie du projet et d'avancer. Dans le cas où, une solution choisie ne fonctionnait pas, elle nous permettra de nous en rendre compte et de reconsidérer nos choix.

Et dans l'éventualité où nos solutions fonctionneraient correctement, la preuve de concept, nous permettra de faire des tests sur la précision, le bon fonctionnement de notre conception et bien

entendu, sur la conformité par rapport aux attentes du client. La preuve de concept, nous servira aussi à voir dans quelle mesure, on pourra optimiser et améliorer notre carte pour la rendre encore moins énergivore et plus écologique. C'est après que pourra passer à la conception du PCB, du routage de la carte et certainement son impression.

6.2 CONCEPTION DU PCB ET ROUTAGE

On a choisi de mettre la conception du PCB après l'étape de validation des tests afin d'être sûr qu'il n'y aura pas d'erreur sur le PCB que nous allons commander et ou imprimer. Sachant que le livrable final que nous avons à déposer pour ce projet est une carte physique en bonne état et fonctionnant correctement en respectant toutes les exigences qu'a défini le client.

CHAPITRE 3 : CARTE DE PUISSANCE

1. OBJECTIF DE LA CARTE

Le rôle de cette carte est, comme nous l'avons vu précédemment, d'alimenter la carte séquenceur du module. Il sera ainsi question de réaliser un bilan énergétique de l'ordinateur de bord afin de définir son besoin en apport d'énergie (puissance, tension, courant) et, par la suite, réaliser une carte répondant à ce besoin.

2. BILAN DE PUISSANCE DE LA CARTE SEQUENCEUR

| Components | Utility | Standby consommation | In use consommation |
|---------------------|--------------------|-------------------------|---------------------|
| Raspberry pi RP2040 | Microcontroller | 0,39mA | 6mA |
| NCP1117 | 5V – 3V3 Regulator | -- | 10mA |
| HS422 | Servomotor | 8mA | 150mA |
| Total | | 18,39mA | 166mA |

Figure n°: tableau récapitulatif des coûts en courant de l'OBC

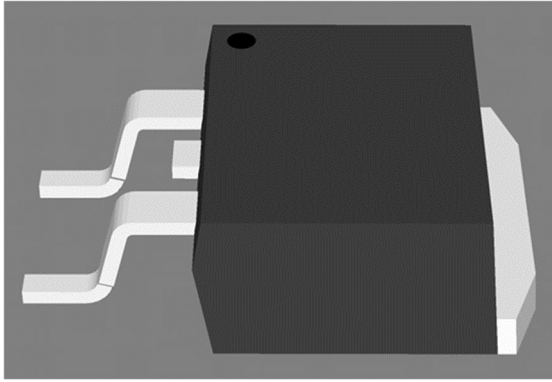
En plus des informations contenues dans le tableau ci-dessus, il faut également noter que le servomoteur ne sera pas en utilisation constante. Il ne sera en action que lorsqu'il sera nécessaire d'activer la trappe pour libérer le parachute. Le régulateur sera en fonctionnement constant et la Raspberry aussi. Il pourrait d'ailleurs être intéressant d'utiliser cette dernière en mode interruption pour baisser la consommation. Enfin, nous savons que pour leur bon fonctionnement, la Raspberry doit être alimentée en 5V et le servomoteur en 6V.

Nous avons désormais les informations nécessaires pour réaliser le choix de nos composants.

3. SOLUTION APPORTEE

3.1 CHOIX DES COMPOSANTS

Figure n° : Régulateur 6V L7806 :



Output Voltage : 6V

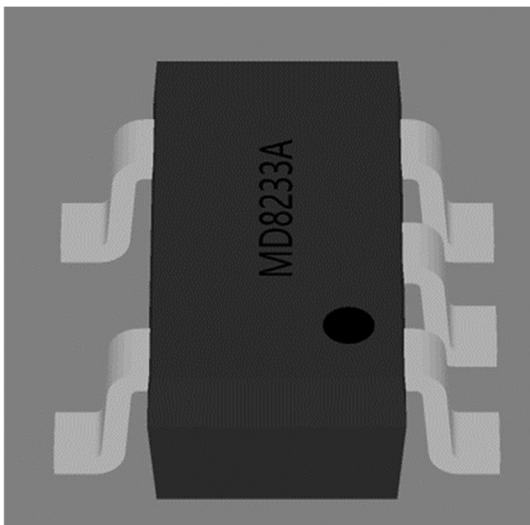
Output Current : 1,5A

Quiescent Current : 6mA

Ce régulateur serait un bon choix pour réaliser l'alimentation du servomoteur, car il correspond aux caractéristiques requises par ce composant. En plus de ça, ce composant a une consommation et des

pertes en courant très faible.

Figure n° : Régulateur 5V MD5150



Output Voltage : 5V

Output current : 300mA

Quiescent Current : 500nA

A l'instar du précédent régulateur, ce composant nous permettrait de d'alimenter correctement la Raspberry en ayant une nouvelle fois un courant suffisant, une consommation et des pertes minimales.

Maintenant que nous avons les composants nécessaires à notre alimentation, nous allons pouvoir nous intéresser à la schématique de la carte de puissance. **Schématique**

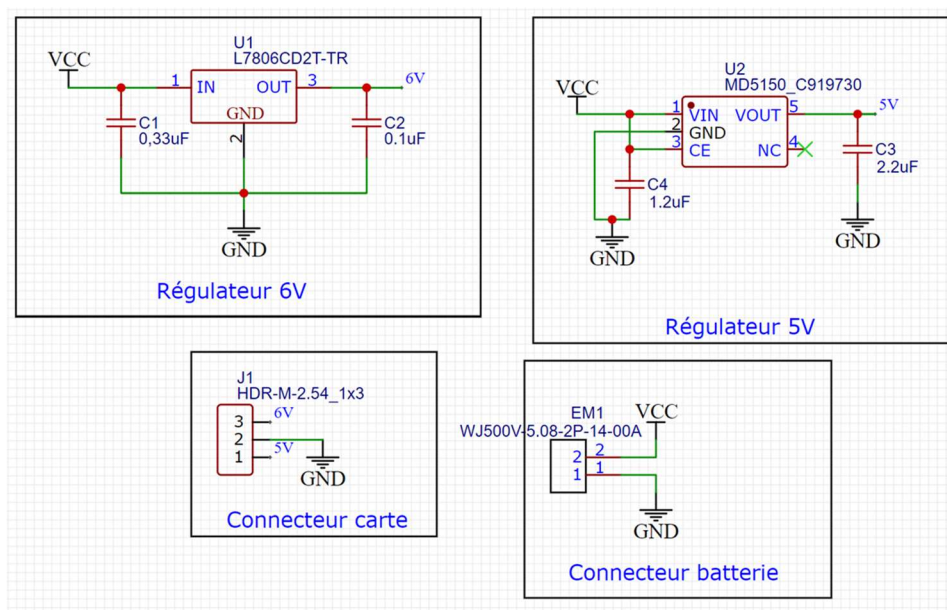


Figure n° : Schématique de la carte de puissance

Les câblages des régulateurs sont réalisés avec des capacités de découplage dont les valeurs sont déterminées avec la datasheet. Nous avons choisi le même connecteur que sur la carte OBC pour des questions de praticités et de disponibilité. Enfin, pour faire la connexion avec la batterie, nous avons choisis un bornier qui pourrait être remplacé par une prise jack si nécessaire.

3.2 PCB

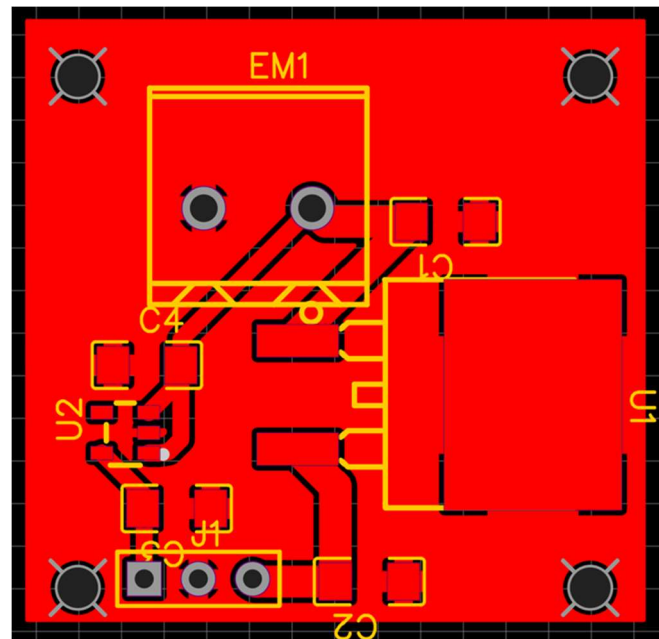


Figure n° : PCB de la carte de puissance

Une fois la schématique réalisé, on peut passer à la réalisation du PCB. On retrouve nos composants vus précédemment :

- EM1 : Bornier
- U1 : Régulateur 6V
- U2 : Régulateur 5V
- J1 : Connecteur
- C1, C2, C3 et C4 les 4 capacités de découplage.

La différence de taille entre les deux régulateurs s'explique par la différence de courant fournie par chacun d'entre eux. Le plan de masse correspond à toute la partie rouge du PCB. Enfin, on a 4 pas de vis pour fixer ce circuit sur la carte de puissance.

3.3 CARACTERISTIQUE DE LA CARTE

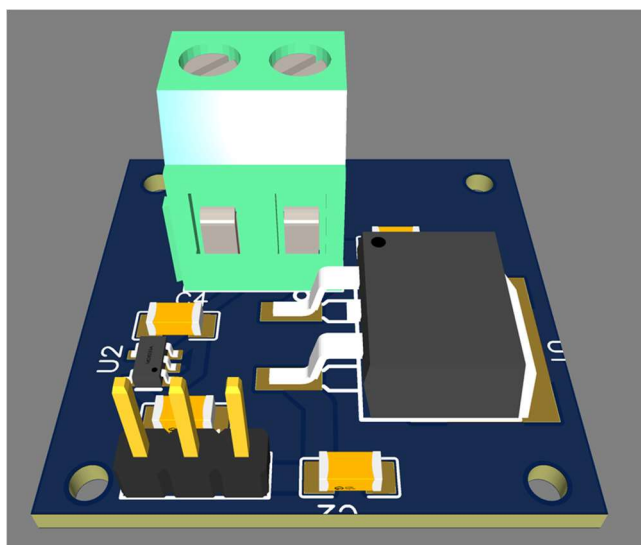


Figure n° : Visualisation 3D de la carte de puissance

Finalement, on obtient une carte avec deux tensions de sortie de 6 et 5V. La sortie en courant correspond à 1,5A pour le 6V et 300mA pour le 5V. On a également une faible consommation générale sur la carte et très peu de perte de courant (~6mA). Pour finir, la carte obtenue mesure 29mm par 29mm ce qui est plus qu'adapté à notre carte imposée de 100mm x100mm.

3.4 CHOIX DE LA BATTERIE

Maintenant, que notre carte est conçue, il nous reste à choisir la source d'alimentation principale qui s'occupera de fournir la puissance nécessaire au bon fonctionnement des deux cartes. Pour choisir une telle batterie, il faut réaliser un nouveau bilan prenant en compte cette fois-ci les composants consommateurs des deux cartes.

| Composants | Output current | Consommation pour 3H |
|------------|---------------------------------|-----------------------------|
| RP2040 | 6 mA | 18 mAh |
| NCP1117 | 10 mA | 30 mAh |
| HS422 | 8 mA en standby ou 150 mA actif | 95 mAh (30min actif sur 3h) |
| L7806 | 6 mA | 18 mAh |

| | | |
|--------|--------|---------|
| MD5150 | 500 nA | 1,5 uAh |
| TOTAL | | 161 mAh |

Nous avons choisi de réaliser cette étude en calculant la consommation de nos composants sur une période de 3h afin d'avoir de la marge. La consommation de l'HS 422 n'est calculée que sur 30 minutes car ce composant n'est utile que lors du déploiement du parachute. Il est donc inutile comme nous l'avons vu précédemment, que ce composant soit en fonctionnement toute la durée du vol. Nous arrivons ainsi à un "coût" total de 161 mAh. Grâce à cette donnée, nous allons pouvoir effectuer le choix de notre batterie.



Figure n° : Photo d'une pile PP3 9V

Nous nous sommes orientés vers une pile type PP3 9 Volts pour plusieurs raisons. La première, son apport en tension est suffisant pour fournir le 6V et le 5V requis. La seconde, sa capacité oscille entre 400 et 600 mAh ce qui est bien supérieur à nos 161 mAh requis. Enfin, en termes de dimensions, cette pile mesure 47,5mm par 25,5 mm par 16,5mm ce qui est adapté à notre carte.

4. AXES D'AMÉLIORATION

Dans l'optique d'améliorer et de rendre cette carte plus performante, nous avons pensé à quelques pistes à explorer :

- Ajouter un module en entrée permettant de mesurer la température afin d'avoir une donnée supplémentaire et de surveiller une éventuelle surchauffe.
- Déplacer le régulateur 3v3 de la carte OBC sur la carte de puissance pour centraliser la partie puissance et libérer de l'espace sur la carte séquenceur.
- Refaire notre PCB de sorte à ce qu'elle soit circulaire pour être plus adapté à la plaque de 100 x 100mm.
- Chercher une source d'alimentation plus adaptée (moins couteuse/moins d'espace/apport en énergie plus adapté à notre besoin).

Conclusion

Au cours de ce semestre et de cette première moitié de projet, nous aurons pu nous familiariser avec de nouveaux outils (logiciel CAO, langage de programmation, protocole de communication). Nous avons également pu réaliser nos trois cartes en simulation. Pour la suite du projet, il nous faudra redéfinir le cahier des charges du projet et adapter nos travaux avec les nouveaux besoins.

ANNEXE SCHEMATIQUE DE LA CARTE OBC

