**Télémétrie Tigresse**

Table des matières

[Contexte 1](#_Toc103786538)

[Besoins 1](#_Toc103786539)

[Contraintes 2](#_Toc103786540)

[Problématiques 2](#_Toc103786541)

[Solutions proposées 2](#_Toc103786542)

[Charge utile : 2](#_Toc103786543)

[Localisation : 2](#_Toc103786544)

[Communication : 3](#_Toc103786545)

[Enregistrement : 3](#_Toc103786546)

[Processing : 3](#_Toc103786547)

[Alimentation : 3](#_Toc103786548)

[Station sol : 4](#_Toc103786549)

[Choix du hardware : 6](#_Toc103786550)

[Données transmises : 7](#_Toc103786551)

# Contexte

Dans le cadre du CSpace … t’as compris, la flemme. Avec un délais court de développement, de nombreux raccourcis ont été pris lors de la conception de ce système.

Cette documentation est clairement une ébauche.

# Besoins

La télémétrie doit permettre un suivi en temps réel et la récupération de la fusée après sa mission.

Elle sera constituée d’une charge utile et d’une station sol.

Un magnétomètre 3 axes devra être implémenté pour échantillonner les variations magnétiques pour l’expérience des résonnances de Schumann.

# Contraintes

Interdiction d’utiliser un récepteur télémétrique en zone pyrotechniques

Une image contenant texte

Description générée automatiquement

Alimentation indépendante, avec interrupteur de mise en marche indépendant.

Autonomie de 3h minimum.

Le Taux d’Onde Stationnaires (TOS) doit être inférieur à 2 pour la fréquence d’émission (avec connectique BNC).

Arrêtez l’émission 10 minutes après le lancement (Impératif)

# Problématiques

GPS : high rate / CoCom limit (< 18.000m & < 1000 knots)

Précision GPS

Perte de de signal GPS & Downlink

# Solutions proposées

## Charge utile :

La charge utile doit permettre l’envoie de donnée nécessaire au suivi et à la récupération de la fusée. Elle doit également permettre de valider et enregistrer les performances de celle-ci.

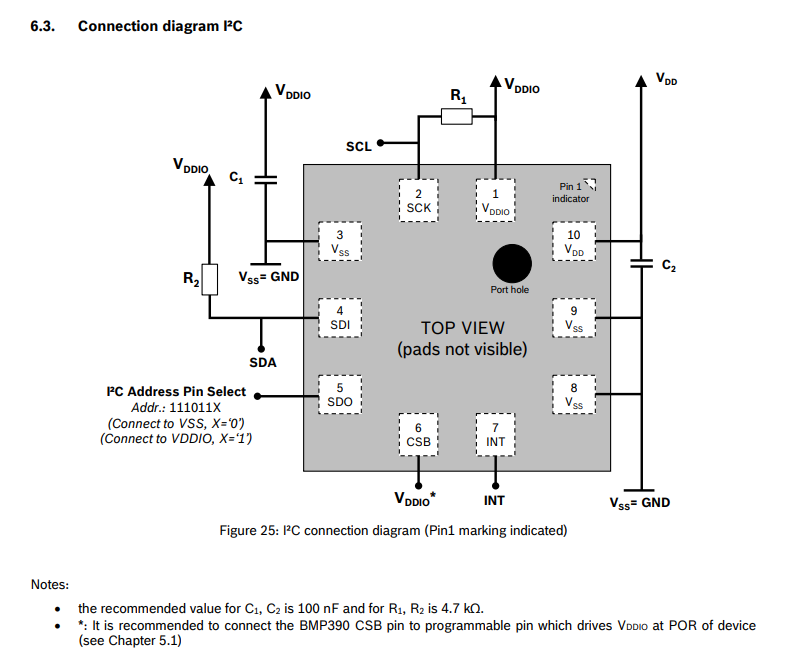
### Localisation :

Afin de permettre la localisation de la fusée dans l’espace, la CU devra embarquer un GPS, une centrale inertielle et un altimètre. Une fusion de ces 3 éléments doit permettre la localisation du lanceur à des fins de performances, et de tracking(si implémenté) par la station sol.

**Le GPS sera de type  377-MAX-7Q-0**

**La centrale inertielle sera de type MPU9250** (toutes les IMU lowcost ont quasi les même specs dont autant pas se faire chier) et comprend le magnétomètre.

**L’altimètre sera de type BMP390**, pour une résolution ~25cm dans un cas ideal



### Communication :

Une TTGO LoRa est souvent utilisé. Cependant, d’autres solution sont également disponible, comme l’utilisation d’un TX3H-869.5-10 avec un récepteur SDR. Un récepteur de type SDR impliquerait une plus grosse architecture (ordi et OS) au niveau de la station sol, mais reste envisageable a moindre coup.

Pour du 869MHz, le module **TX3H-869** à une portée jusqu’à 3Km, et un débit de 10Kbps pour 450mW

Malgré la grosse consommation et le prix, ce module présente l’avantage d’une forte puissance d’émission.

### Enregistrement :

Un stockage de masse devra permettre la journalisation des données. Il pourra s’agir d’une carte SD ou d’une EEPROM (comme sur la raspi pico). Etant donné le choix de l’OBC, il s’agira d’une carte SD.

### Processing :

L’OBC a déjà été choisi. Il est donc imposé, il s’agira d’une carte **Adafruit Feather 32u4 Adalogger.**

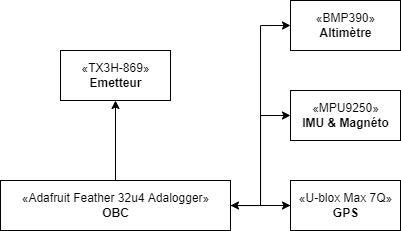
### Alimentation :

|  |  |  |
| --- | --- | --- |
| Module | Courant max | Tension d’alimentation |
| **Adafruit Feather 32u4 Adalogger** | 300mA | 3.3V |
| **TX3H-869** | 350mA | 5V |
| **BMP390** | 4µA | 3.3V |
| **MPU9250 (shield)** | 4mA | 5V |
| **MAX-8Q-0** | 70mA | 3V |

Comme la régulation se fait avec des régulateurs linéaires et pour une alimentation nominale de 7.4v, on trouve :

Pour 3h de fonctionnement, il faudrait donc une batterie de 2.17 mAh

Nous avons donc opté pour des batteries de type Li-ion car elles possèdent une capacité massique intéressante pour notre application.



## Station sol :

La stations sol a plusieurs objectifs. Afficher les données brutes de l’état du lanceur, extrapoler la position du lanceur dans l’espace grâce à une fusion. Permettre de localiser et récupérer la fusée.

Il faudra également enregistrer les données reçues.

L’interface sera donc constituée de 3 sections :

* Données du lanceur (altitude, vitesse, stages, )
* Données de localisation par rapport à la station sol

Pour les données du lanceur, il faudra afficher :

* L’état de la liaison GPS
* L’état de la liaison télémétrique

Une image contenant texte

Description générée automatiquement

* L’altitude et vitesse actuelle



* Les étapes du lancement (Standby, Lift off, MECO, chute deployment, landing)



* La projection du vol

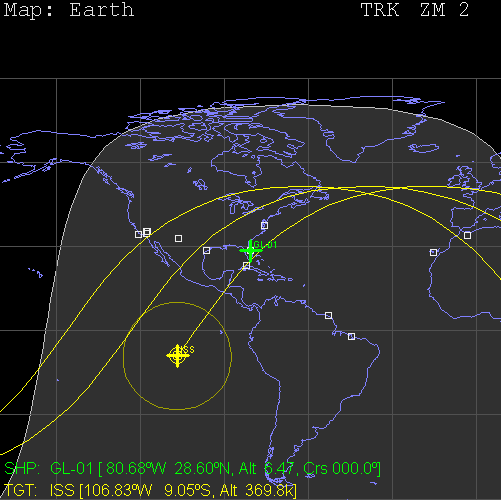
Une image contenant texte, équipement électronique

Description générée automatiquement

* Un journal de vol

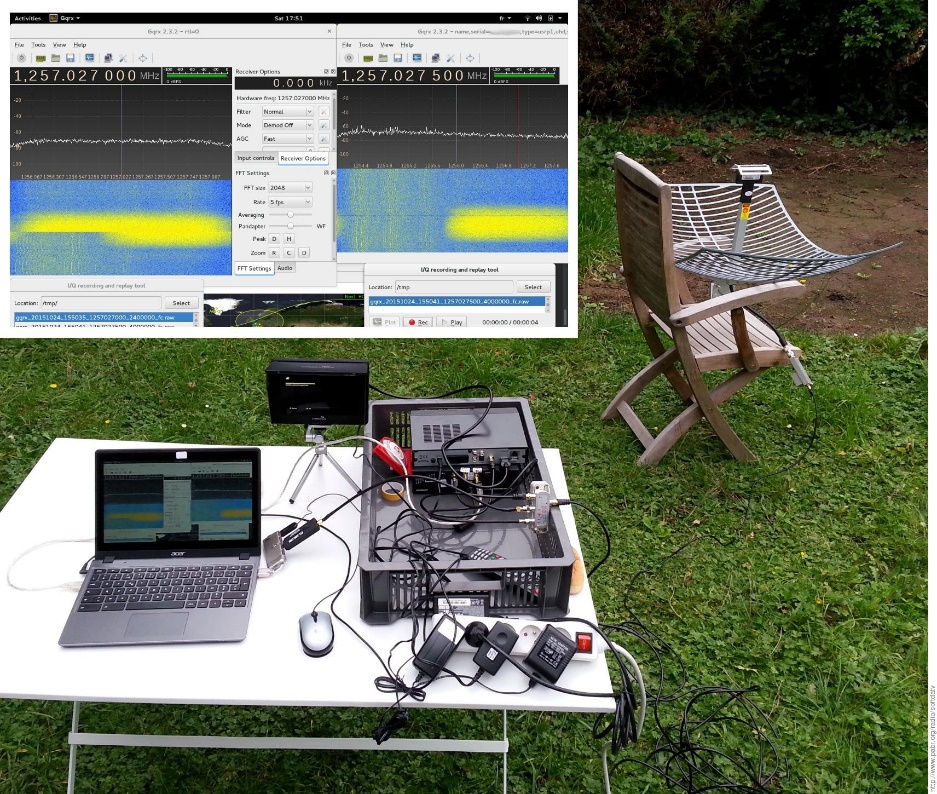
Pour les données de récupération :

* La trajectoire du vol effectué sur une surface par rapport à notre position
* La distance et l’orientation par rapport à la position finale de la fusée



### Choix du hardware :

Pour la station sol, 2 solutions s’offres à nous. Dans le premier cas il s’agirait d’utiliser un ordinateur et un récepteur SDR :



Ou d’utiliser une solution full hardware, ce qui rendrais l’installation plus fiable, mais plus complexe à développer :



# Données transmises :

Les données qui transiteront entre la fusée et la station sol seront les suivantes :

* Raw GPS (permet la récupération de la fusée en cas de dérive de la fusion de données)
* Attitude de la fusée
* Fusion de données (Altimètre, GPS, IMU) pour la localisation dans l’espace
* Trigger de stage (décollage, ouverture trappes …)
* \*optionnel\* Altitude, Vitesse, température

# Design du nose cone :

Le type de design sera de type puissance. Le script python noConeGen.py permet de générer la courbe sous forme de dxf pour créer le cone.

# Carte télémétrie version 1 :

Après assemblage de la première version de la carte électronique sur PCB, quelques erreurs ont été relevé.

* Le montage mosfet du buzzer n’est pas correct. Une résistance a été temporairement branché en parallèle au buzzer afin de permettre à celui-ci de fonctionner correctement.
* Attention au sens de branchement de la puce GPS !!

# Calibration IMU

Pour la calibration IMU, utiliser le programme MPU9250CAL.ino et reporté les valeurs dans le programme principal.