



Rapport de projet Triplex 2024

FX42

2	Définition du projet	5
2.1	Introduction	5
2.2	Fusex	5
2.3	Cansats	5
2.3.1	CanSat relai (Aramis)	5
2.3.2	CanSat parafoil (Athos).....	5
2.3.3	CanSat auto-rotatif (Porthos).....	5
2.3.4	Circuit générique	6
3	Conception du projet	6
3.1	Fusex	6
3.1.1	CAO.....	7
3.1.2	Méthode des éléments finie (MEF).....	7
3.1.3	Architecture globale	9
3.1.4	Stabtraj	10
3.1.5	Système de récupération	12
3.1.5.1	Retient Parachute:.....	12
3.1.5.2	Dimensionnement du parachute:.....	12
3.1.6	Système d'éjection	13
3.1.7	Minuterie.....	14
3.1.7.1	Schéma électrique de la minuterie.....	14
3.1.7.2	Partie Cansat	15

3.1.7.3	Partie Parachute.....	16
3.1.7.4	Code :.....	17
3.1.8	Système d'éjection	22
3.1.9	Système de récupération	22
3.1.10	Minuterie.....	24
3.1.11	Code :.....	25
3.1.11.1	Partie Cansat :.....	26
3.1.11.2	Partie Parachute :.....	27
3.1.12	Ailerons.....	27
3.1.13	Peau.....	28
3.1.14	Tests minif	29
3.1.15	Miniplex II :.....	31
3.1.16	Intégration.....	33
3.2	Cansats	34
3.2.1	CanSat relai (Aramis).....	34
3.2.1.1	4G LTE.....	34
3.2.1.2	Caméras	34
3.2.1.3	Architecture électrique.....	35
3.2.1.4	Architecture mécanique.....	36
3.2.1.5	Protocoles utilisés	37
3.2.1.6	Compression des données	37
3.2.1.7	Essais en vol.....	38
3.2.2	CanSat parafoil (Athos).....	39
3.2.2.1	Historique.....	39
3.2.2.2	Développement	39
3.2.3	CanSat auto-rotatif (Porthos).....	46
3.2.3.1	Autorotation	46
3.2.3.2	Incidence et répartition.....	48
3.2.3.3	Essais en soufflerie	50
3.2.3.4	Réduction du diamètre.....	52
3.2.3.5	Stabilité et centre de gravité	53
3.2.3.6	Mesure de la vitesse de rotation du CanSat	54
3.2.3.7	Mesure de la vitesse de rotation du rotor	54
3.2.3.8	Test au C'Space	55
3.2.4	Circuit générique	56

3.2.4.1	GPS.....	56
3.2.4.2	LoRa.....	61
3.2.4.3	Station sol (Hugo T.)	66

07/2024
Presentation CDR-Critical Design Review
Fusée Sonde ESL-1

09/01/2024

Sommaire	Erreur ! Signet non défini.
2 Définition du projet	5
2.1 Introduction	5
2.2 Fusex	5
2.3 Cansats	5
2.3.1 CanSat relai (Aramis)	5
2.3.2 CanSat parafoil (Athos).....	5
2.3.3 CanSat auto-rotatif (Porthos).....	5
2.3.4 Circuit générique	6
3 Conception du projet	6
3.1 Fusex	6
3.2 Cansats	34
3.2.1 CanSat relai (Aramis)	34
3.2.1.1 4G LTE	34
3.2.1.2 Caméras	34
3.2.1.3 Architecture électrique.....	35
3.2.1.4 Architecture mécanique	36
3.2.1.5 Protocoles utilisés	37
3.2.1.1 Compression des données	37
3.2.1.2 Essais en vol.....	38
3.2.2 CanSat parafoil (Athos).....	39
3.2.3 CanSat auto-rotatif (Porthos).....	46
3.2.4 Circuit générique	56
3.2.4.1 GPS.....	56
3.2.4.2 LoRa.....	61
3.2.4.3 Station sol (Hugo T.)	66

2 Définition du projet

2.1 Introduction

L'idée de la fusex Triplex vient d'une discussion avec des bénévoles de Planète Sciences au sujet du concours CanSat : le but final du projet est de proposer aux trois finalistes du concours CanSats un lanceur pour leur projet à la fin de la campagne. Cette solution pourrait ainsi attirer de nouveaux participants au concours.

Pour le premier lancement nous voulons lancer 3 CanSat qui communiquent entre eux avec l'un d'entre eux qui fait office de relai. Chaque CanSat remplit donc un rôle bien spécifique qui sera décrit plus tard.

Pour des raisons liées à notre école, nous avons dû découper le projet en 2, la fusex d'un côté et les CanSats de l'autre

2.2 Fusex

La Fusex a pour objectif de larguer 3 Cansats de manière suffisamment forte pour ne pas gêner le déploiement des différents systèmes de récupération. La Fusex doit donc accueillir en son sein trois emplacements suivant le cahier des charges Planètes Science. Elle sera équipée d'un pro 75 et à terme devra servir pour larguer les vainqueurs du concours Cansat.

2.3 Cansats

2.3.1 CanSat relai (Aramis)

Ce CanSat est équipé d'une Raspberry Pi 5, d'un récepteur LoRa, d'un modem 4G et d'une caméra. Son objectif est de relayer toutes les trames LoRa reçues des autres CanSats, y ajouter sa position GPS ainsi qu'un flux vidéo et envoyer le tout directement par 4G sur un serveur distant.

2.3.2 CanSat parafoil (Athos)

Athos est un CanSat auto-guidé porté par un parapente. Son but est relativement simple en théorie mais complexe en pratique: suivre le CanSat auto-rotatif grâce aux données GPS de ce dernier. Le tout en envoyant ses positions en direct au CanSat relais par LoRa. Il sera donc composé d'un microcontrôleur (Raspberry Pi Pico), d'un GPS, d'un module LoRa, d'une batterie au lithium et de servo moteurs contrôlant le parapente. Ce CanSat est un vrai défi, son asservissement, son déploiement ainsi que son contrôle nécessitent beaucoup de tests pour qu'il respecte le cahier des charges et ne dérive pas.

2.3.3 CanSat auto-rotatif (Porthos)

L'objectif de Porthos est de ralentir sa descente par l'auto-rotation de 3 pâles montées sur un rotor. Chacune des pâles est séparée en 2 parties : une partie créant la rotation (proche du rotor) avec une incidence de -30° , et une partie créant la portance (loin

du rotor) avec une incidence de 10° . Le CanSat sera aussi équipé d'un capteur à effet hall afin de récupérer la vitesse de rotation du rotor (en installant un aimant dessus), d'un gyroscope, d'un altimètre et d'une batterie. De plus, un GPS y sera installé afin de transmettre sa position par signal LoRa à Athos (qui doit le suivre) et à Aramis (qui doit retransmettre l'information au sol). Lors d'un test en soufflerie nous avons pu mesurer une portance de plus de 900 grammes avec un vent relatif de 14.75 m/s.

2.3.4 Circuit générique

Pour simplifier la programmation et mutualiser les coûts, nous avons décidé d'utiliser un seul circuit modulaire pour l'ensemble des CanSats ainsi que pour la fusée. Ce circuit doit pouvoir se localiser à l'aide d'un récepteur GNSS performant mais aussi communiquer à longue distance en utilisant la modulation LoRa. Il doit également être assez puissant pour remplir les tâches spécifiques à chaque CanSat.

3 Conception du projet

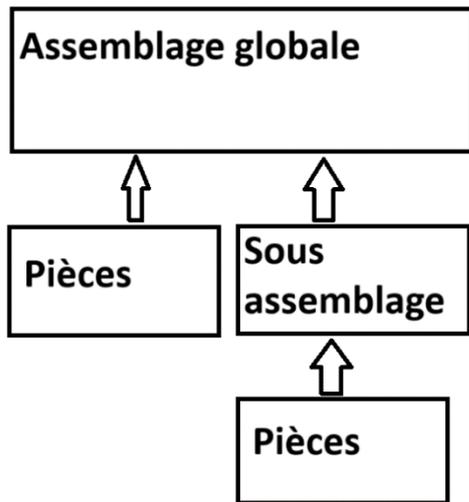
3.1 Fusex

Le projet Triplex consiste en la construction d'une fusée expérimentale réutilisable qui a pour but de lancer des cansats à une altitude relativement élevée. Ces cansats sont des mini-satellites issus du concours éponyme de Planète Sciences respectant un cahier des charges précis, cependant ce concours manque d'attractivité en France ces derniers temps. La raison évoquée par les organisateurs est qu'à l'inverse d'autres pays en Europe ceux-ci sont lancés à l'aide de drones. L'objectif du projet est donc de construire une fusée qui permet à d'autres personnes de pouvoir lancer des cansats afin de rendre le concours plus populaire.

Pour atteindre cet objectif, 10 élèves de l'ESTACA travaillent sur ce projet, 5 étudiants en troisième année construisent la fusée en elle-même et 5 étudiants en quatrième année ont pour objectif de construire 3 cansats test pour le premier vol, ces derniers sont également les initiateurs et les tuteurs du projet.

3.1.1 CAO

La CAO a une importance cruciale dans un projet, en effet c'est grâce à la conception assistée par ordinateur que nous pouvons suivre l'avancement dans la conception des différents sous-systèmes.



Nous fonctionnons dans un système de hiérarchie verticale

Les pièces sont faites séparément puis assemblées, les pièces ne sont pas faites en contexte.

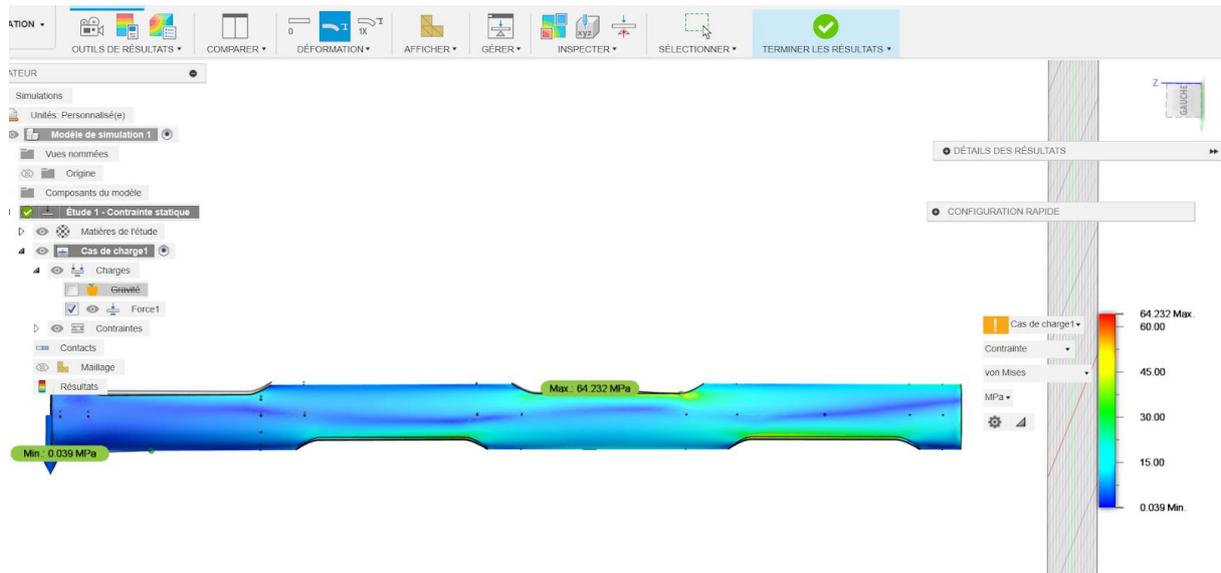
Comme la structure a été faite par différentes personnes, il est vital que la dernière version de la CAO soit en ligne.

Concernant l'assemblage, la fusée est constituée de 2 étages avec deux peaux différentes, le premier a une peau en carbone récupérée d'un ancien projet de l'ESO. Concernant le deuxième, une peau en fibre de verre a été commandée chez Air Esiea car ils possèdent une enrouleuse destinée à fabriquer des tubes en composite. Les deux sont reliés par une bague inter-étage usinée en aluminium afin de limiter au maximum la flèche. L'assemblage est également composé de 4 systèmes d'éjection dont 3 pour les cansats et un pour le parachute, de toutes les bagues permettant d'accueillir le moteur de la fusée et de tous les autres sous-systèmes présents dans la fusée. Le but étant d'avoir une CAO au plus proche de la réalité pour éviter tout problème lors de l'intégration.

3.1.2 Méthode des éléments finie (MEF)

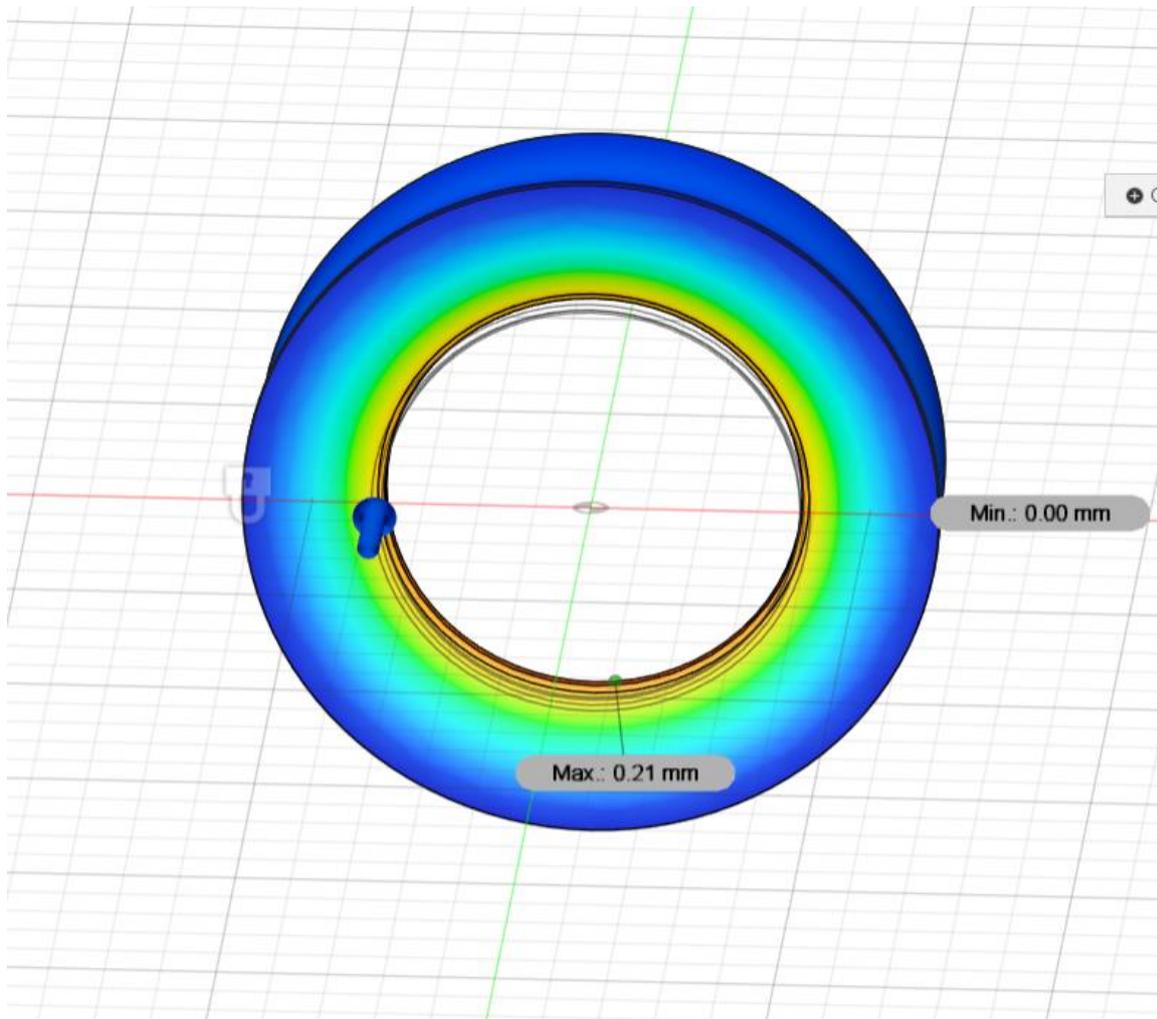
Durant la phase de conception des pièces qui sont soumises à des efforts importants, il est nécessaire de faire plusieurs études MEF afin de vérifier qu'elles résistent à tous les efforts mis en jeu. Pour le dimensionnement de 2 pièces nous avons réalisé une étude MEF.

Le premier élément est la peau en fibre de verre, étant donné que la peau est structurante il est très important d'avoir une peau possédant une flèche très faible. Cependant, cette peau doit pouvoir accueillir les 3 cansats, le parachute et par conséquent les 4 trappes associées à ces derniers. Nous avons donc effectué une comparaison entre différentes dispositions de trappe sur le tronçon à l'aide du logiciel fusion 360. La conclusion de l'étude a révélé qu'une disposition avec des trappes disposées une à une à 180° était la meilleure pour minimiser la flèche de la fusée.



Nous avons réalisé une deuxième étude MEF sur la bague reprise de poussée du moteur qui comme son nom l'indique a pour rôle de transmettre la force maximale de 1286 N du Pro-75 au reste de la fusée. Il est donc crucial d'avoir une pièce bien dimensionnée pour des raisons de sécurité. À la suite de cette étude nous avons décidé d'ajouter une pièce en renfort afin d'être sûr de la bonne tenue mécanique de cette bague lors du décollage.

MEF de la bague reprise de poussée



3.1.3 Architecture globale

L'architecture de la fusée est une étape à ne pas négliger. Elle détermine toutes les étapes d'intégrations, les dimensions de la fusée et bien d'autres paramètres.

Par définition nous avons une fusée qui sera grande car nous devons respecter le cahier des charges du concours cansat qui impose un certain volume autour du satellite, cette contrainte engendre un gros impact car plus la fusée est longue plus elle est sensible à la flèche. Il faut donc faire attention à ne pas mettre les pièces lourdes en haut de la fusée. De plus, il y a certaines contraintes données par les cansats que nous transportons. En effet, nous devons positionner tous les cansats dans la peau en fibre de verre qui est un matériau qui laisse passer les ondes nécessaires au bon fonctionnement de ces derniers. Nous

avons placé le cansat 4G tout en haut de la fusée car il possède une caméra qui devra filmer l'éjection du cansat Parafoil et également car il est léger. Etant donné que le cansat parafoil doit être filmé par le cansat 4G il est obligatoirement en troisième position dû à la disposition des trappes. Nous avons également décidé de placer le cansat auto-rotatif au plus bas car c'est le cansat qui est éjecté en premier et qui doit être suivi par le cansat parafoil. Enfin le parachute est finalement au deuxième emplacement, cela évite les risques d'accrochage avec les ailerons et ça permet également de limiter le bras de levier entre le parachute et le centre de gravité de la fusée qui pourrait causer des basculements violents de la fusée pendant sa récupération.

Toutes ces décisions ont été actées en prenant en compte la position du centre de gravité qui est essentielle à la bonne stabilité de la fusée.

Erreurs CAO

L'une des erreurs les plus fréquentes lors de la création d'une CAO est de réaliser une pièce ou un assemblage sans anticiper les potentielles modifications à ceux-ci. Faire des esquisses ou des systèmes facilement modifiables par l'intermédiaire de cote paramétrée par exemple peut faire gagner un temps non négligeable durant toute la durée d'un projet.

La pièce de reprise de poussée a été mal conçue car la mise en plan envoyée à l'usineur n'a pas été revue par différents membres du groupe. Lorsqu'il y a des commandes de pièces importantes il faut toujours faire intervenir plusieurs personnes voir même demander des avis à des personnes externes au projet afin d'éviter ce type d'erreurs.

3.1.4 Stabtraj

Le logiciel Stabtraj se compose de deux parties, il permet d'assurer la bonne stabilité de la fusée et de prévoir la trajectoire de la fusée.

Pour assurer que la fusée soit stable il faut déterminer les dimensions des ailerons afin d'avoir une marge statique et une portance respectivement dans leurs intervalles fournis dans le cahier des charges.



STABILITO

Stabilité de fusée à ailerons

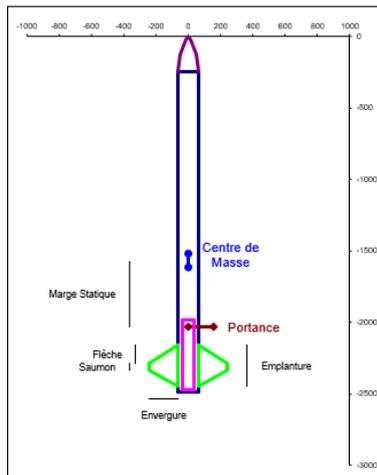
Remplir les cases jaunes

Fusée	
Nom	Triplex
Club	ESO
Type	Fusée expérimentale.
Masse	10448 g sans propu
Centre de Masse	1414 mm sans propu
Longueur totale	2483 mm

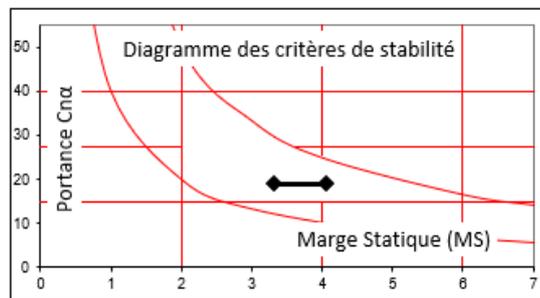
Propulseur	
Type	Original (Pro75-3G C)
Position du bas	2469,3 mm

Coiffe	
Forme	Ogivale (pointue)
Hauteur	245 mm
Diamètre	125 mm

Ailerons



13/05/2024	Min	Résultats	Max
Finesse	10	19,9	35
Portance	15	19,0	40
Marge Stat	2 D	3,31 D	4,06 D
Couple	40	62,7	77,0
XCp		2031 mm	2031 mm
MS /L		17% L	20% L
STABLE			



Checksum : propu OK

v3.4.2

La marge statique correspond à la distance entre le centre aérodynamique de la fusée et son centre de masse. C'est pourquoi il faut bien choisir l'architecture du lanceur car cela impacte directement la position du centre de masse et par conséquent la stabilité de la fusée. En effet, un centre de masse trop bas aura tendance à rendre la fusée instable tandis qu'une position trop haute de celui-ci rend la fusée surstable. Pour déterminer la position du barycentre de la fusée nous avons utilisé le logiciel Solidworks qui nous permet de fixer la masse de chaque composant de la fusée. Par la suite, en utilisant l'outil "Propriétés de masse" nous obtenons la position du centre de gravité ainsi que la masse totale de la fusée. Une fois que nous avons toutes ces informations, il suffit de rentrer ces données dans le tableur et de modifier la géométrie des ailerons jusqu'à ce que la fusée rentre dans la zone de stabilité. Bien évidemment, le changement de géométrie des ailerons modifie la masse totale et la position du barycentre du lanceur mais en effectuant plusieurs itérations on tombe sur une solution stable dans un grand nombre de configurations.

Pour ce qui est de la trajectoire, il faut bien veiller à avoir une vitesse en sortie de rampe supérieure à 20 m/s et de ne pas avoir une vitesse chute entre 5 m/s et 15 m/s comme indiqué dans le cahier des charges des fusées expérimentales du C'Space.

Erreurs Stabtraj:

Il faut faire attention à l'origine utilisée pour le positionnement du centre de gravité. Dans Stabtraj, le centre de masse est positionné à partir du bout de l'ogive tandis que sur

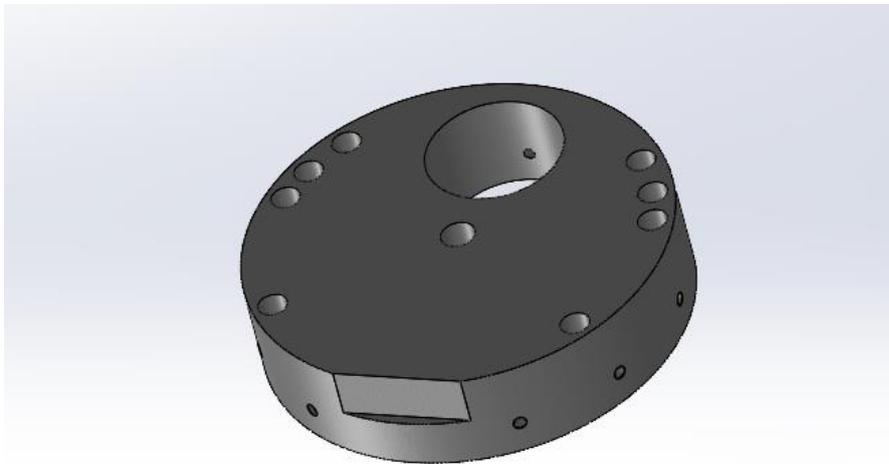
Solidworks il est placé à partir de l'origine de l'assemblage. Il faut donc bien faire attention lors de la mise en place du bâti de l'assemblage de bien mettre celui-ci au niveau de l'origine en utilisant Solidworks pour calculer la position de barycentre.

Il est très important de faire toute l'étude de stabilité sans propulseur car Stabtraj prend en compte le changement de masse du propulseur mais il est également plus simple d'évaluer le propulseur adéquat pour la fusée. Il est plus simple de déterminer la position du centre de gravité de manière empirique car il est peu probable d'avoir un propulseur chargé à disposition.

3.1.5 Système de récupération

Cette partie sera exclusivement réservée à la récupération en elle-même, le système d'éjection étant le même pour les cansats et pour le parachutes, celui-ci sera abordé dans la partie consacrée au système d'éjection.

3.1.5.1 Retient Parachute:



Pour accrocher le parachute à la fusée nous avons choisi d'utiliser un eye bolt (anneau de levage). Le mousqueton du parachute passe dans l'œil tandis que la tige est retenue par la plaque en métal servant d'écrou. La bague en pivot par rapport à l'eye bolt est volontairement surdimensionnée pour des raisons de stabilité.

3.1.5.2 Dimensionnement du parachute:

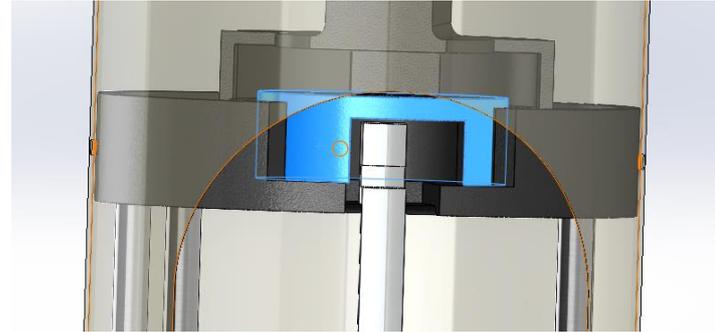
Dans la conception du parachute, il est capital de prendre une fourchette en termes de surface dans stabtraj. La masse globale étant difficilement ostensible et donc le centre de gravité. Notre fourchette est définie en fonction de la vitesse maximale et minimale définie

dans le cahier des charges Planètes sciences. En effet la fusée ne doit pas descendre trop vite au risque d'abîmer ses composants ni trop lentement au risque de dériver. Donc la surface de notre parachute dépend de cette fourchette et des approximations sur la masse et son centre.

3.1.6 Système d'éjection



Afin de simplifier la réalisation nos systèmes d'éjection pour le parachute et les cansats sont les mêmes à une différence près : le système pour le parachute prend moins de hauteur du a des tiges plus courtes. C'est un système de cloche. Concrètement la pièce surlignée en bleu est en liaison pivot par rapport au bâti et est fixée à un moteur. Celui-ci en fonction de la minuterie va tourner pour permettre de faire passer ou non le doigt gris lié à la trappe.



Du tissu lycra agissant comme un élastique va appliquer une force permettant d'éjecter latéralement l'objet contenu à une vitesse importante. Le lycra a été choisi car il est possible de le coudre à l'opposé des élastiques en caoutchouc et sera donc plus facilement intégrable.

Le fait qu'on puisse le coudre permet aussi d'attacher le lycra à la base, cela permet d'éviter que tout élément éjectable se glisse hors du système d'éjection et mette en péril la mission.

Les tiges subissent des efforts importants en supportant la tension du Lycra c'est pourquoi il a été choisi qu'elles seraient en aluminium.

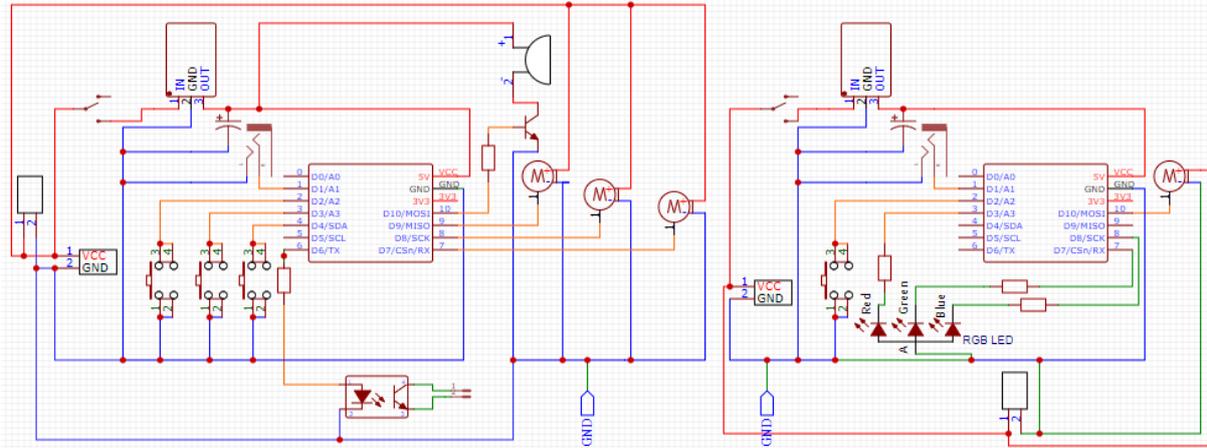
3.1.7 Minuterie

La minuterie est un des éléments les plus importants de la fusée car elle permet d'éjecter les trois Cansats ainsi que le parachute. Elle possède un état fermé (système d'éjection calibré) et un état ouvert (trappe ouverte et système éjecté). Dans la conception, celle-ci se découpe en deux parties : le schéma électrique et l'algorigramme du code.

3.1.7.1 Schéma électrique de la minuterie

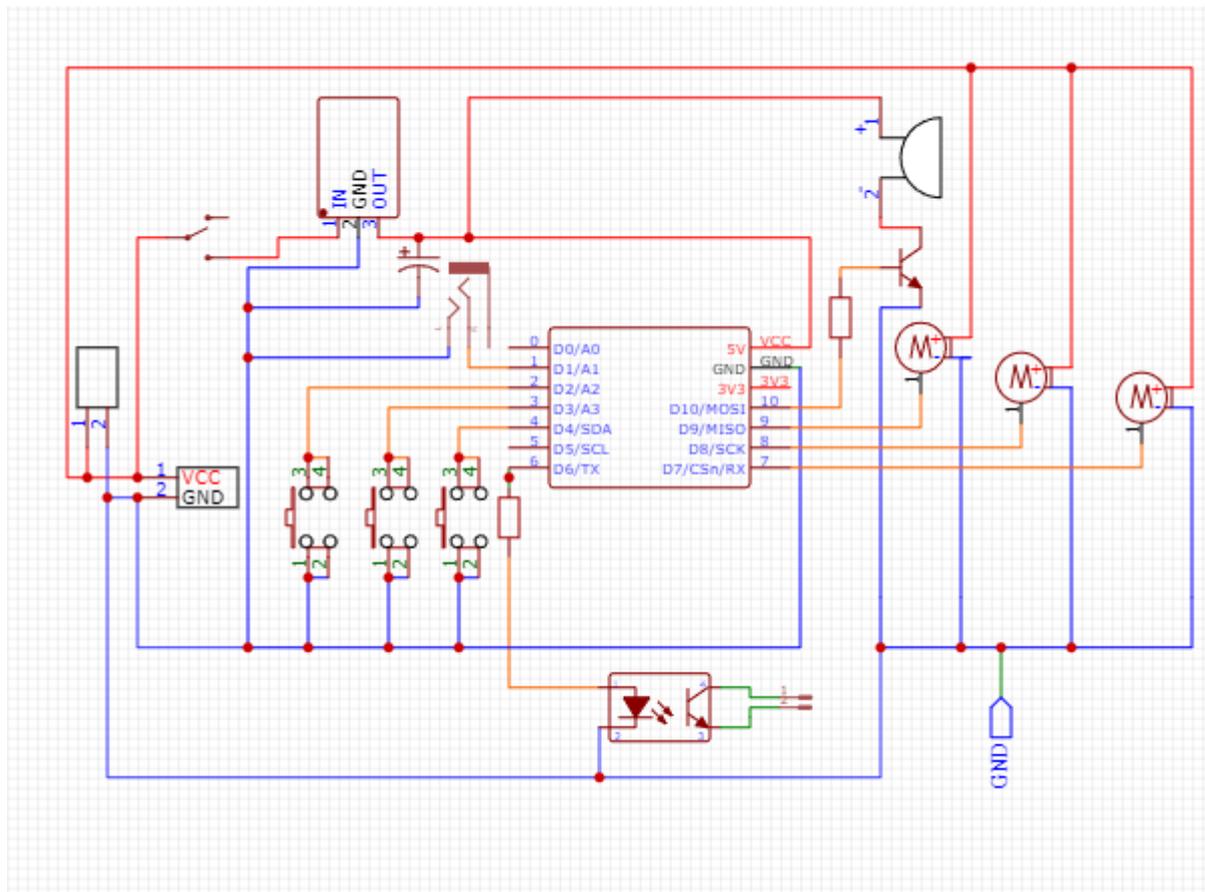
La minuterie de triplex doit permettre l'activation des systèmes d'éjections au moment souhaité que ce soit pour les cansats ou pour le parachute. Pour ce faire, le schéma électrique est séparé en deux parties indépendantes conformément aux cahiers des

charges Planète Sciences avec une partie minuterie parachute et de l'autre une partie minuterie Cansats.



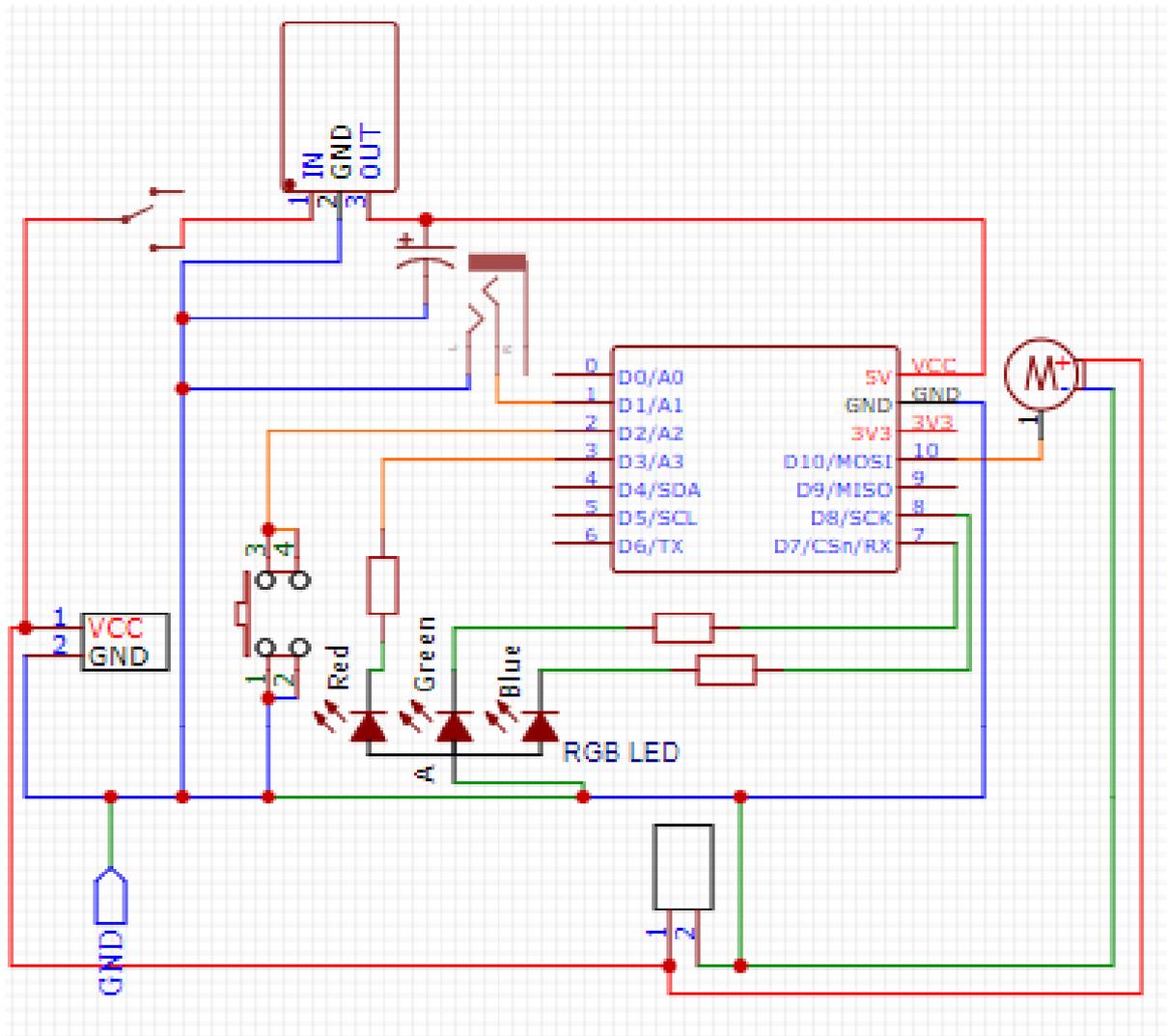
3.1.7.2 Partie Cansat

La minuterie Cansat consiste à alimenter 3 servomoteurs nécessaires à l'éjection des cansats. Une batterie de 7,4V alimente les moteurs, le buzzer et un microcontrôleur. Avant d'atteindre ce dernier composant, la tension est abaissée à 5V avec un abaisseur de tension 7805. Le microcontrôleur choisi est un Raspberry 2040 blindé en raison de sa taille et de sa performance, elle est blindée car l'un des cansats pour le premier vol utilise la 4G, c'est donc une sécurité pour éviter toute interférence. A cette RP 2040 sont liés les moteurs, une prise jack détecte le décollage en changeant d'état, des boutons permettent de contrôler manuellement les servomoteurs et le buzzer via un transistor. Le buzzer émet différents sons en fonction de l'état du système. Les différents modes de fonctionnement du système et des composants seront expliqués ultérieurement dans la partie code de ce rapport. On notera également la présence d'un composant au pin 5 de la RP, celui-ci n'est pas utile au fonctionnement de la minuterie et l'est uniquement aux cansats du premier vol de Triplex, sa fonction ne sera donc pas expliquée dans ce rapport.



3.1.7.3 Partie Parachute

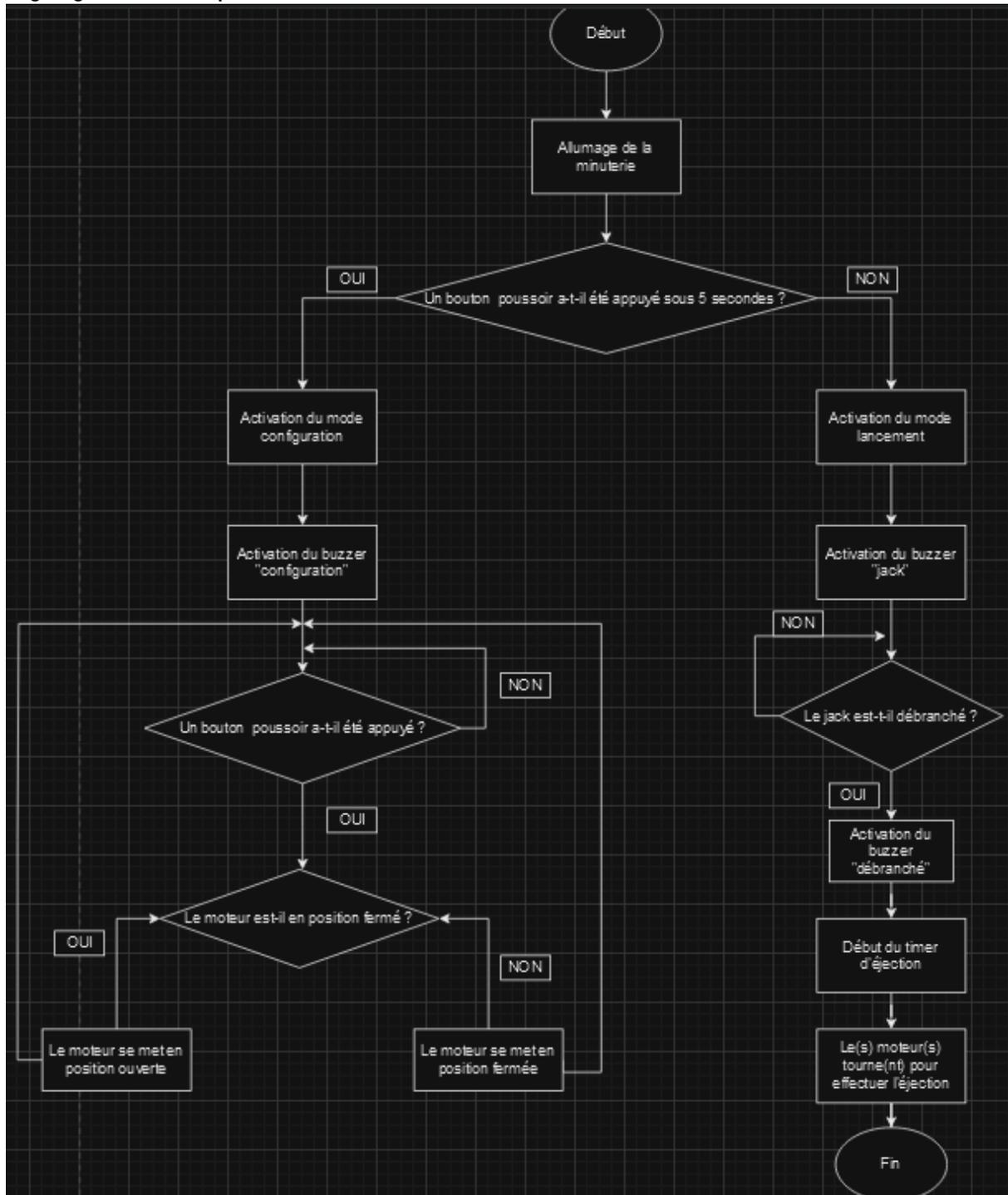
La minuterie Parachute est très semblable à la minuterie Cansats. Les seules différences étant la présence d'un seul moteur ainsi que le remplacement du buzzer par une led RGB comme indicateur de l'état du système. A l'image de la partie Cansats, les différents modes de fonctionnement du système et des composants seront expliqués ultérieurement dans la partie code de ce rapport.



3.1.7.4 Code :

Pour le code, il est important d'avoir un algorithme avant de commencer à coder afin de comprendre comment le code va se structurer et quel est son but pour le système.

Algorithme complet de la minuterie

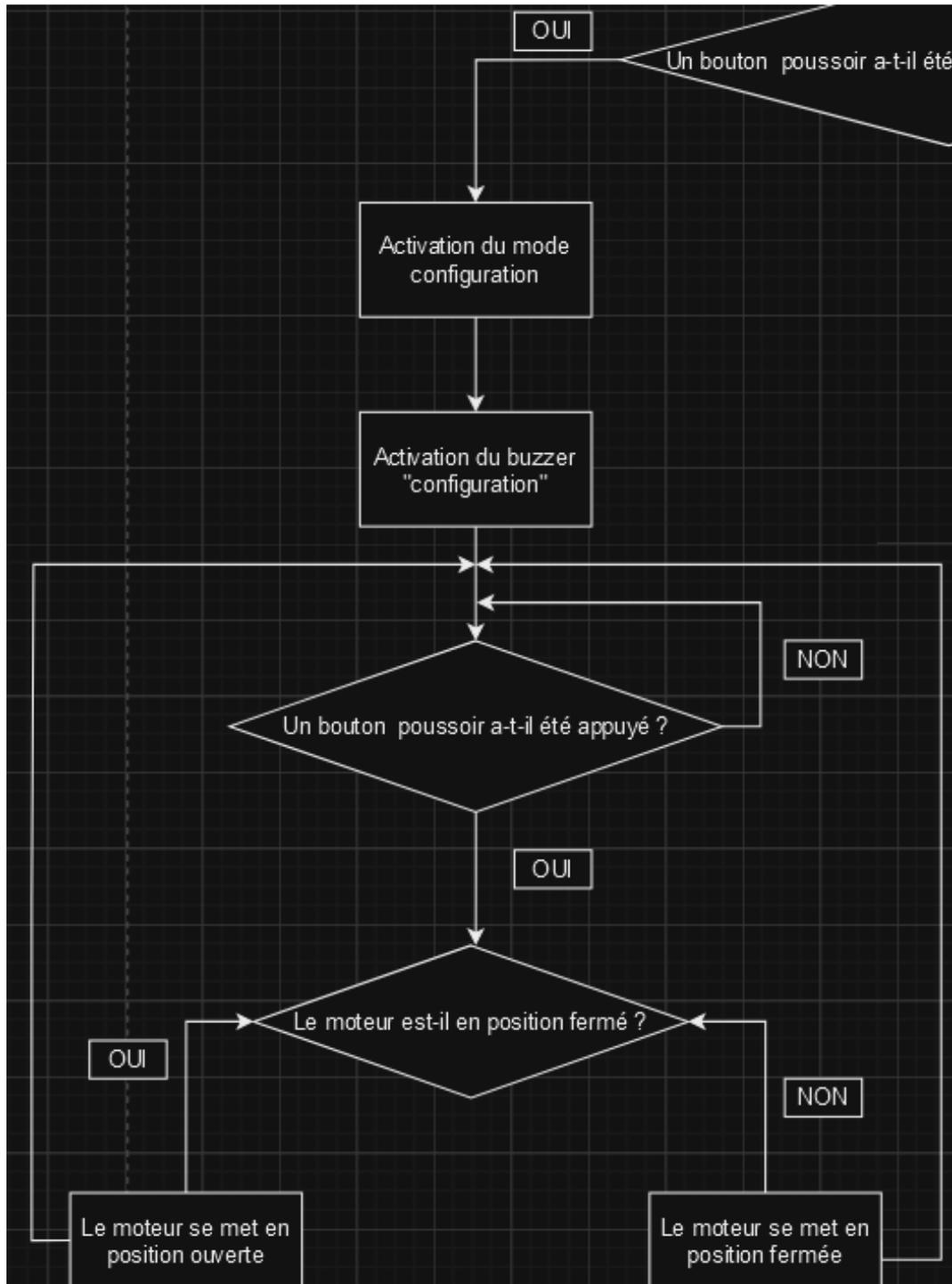


Lors de la mise en route de la minuterie, une interruption de 5 secondes se fait. Après ces 5 secondes, si un bouton poussoir est appuyé ou non, la minuterie passe dans l'un de ses deux modes. Une chose importante à noter est qu'une LED RGB remplace le moteur pour la minuterie parachute. Cependant, son "fonctionnement" dans l'algorithme reste similaire au buzzer.

Le mode configuration consiste à calibrer le système d'éjection, c'est-à-dire d'avoir le servomoteur dans sa position fermée avec un CanSat (ou le parachute) dans la fente avec la trappe fermée.

Lorsque la minuterie entre dans le mode configuration, le buzzer le notifie à l'utilisateur avec une séquence de son précise. Ensuite, l'utilisateur peut régler chaque système d'éjection à l'aide des boutons poussoirs correspondant aux servomoteurs. Lorsqu'un bouton est appuyé, le servomoteur correspondant passe simplement d'une position à l'autre (s'il est en position ouverte alors il passe en position fermé). Ce mode est infini tant que la batterie n'est pas débranchée pour ne pas laisser de contraintes de temps aux opérateurs des systèmes d'éjections.

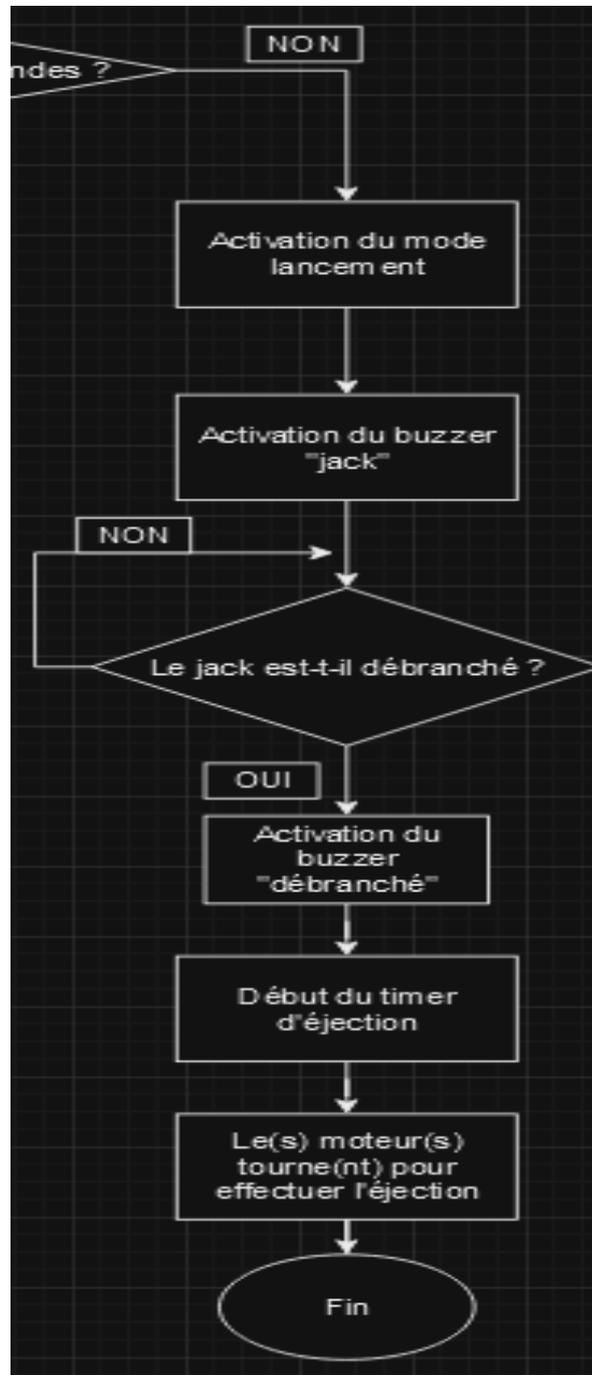
Partie Configuration de la minuterie



Lorsque la minuterie entre dans le mode lancement, le buzzer notifie si le jack est branché ou non. Il a une séquence de son beaucoup plus rapide lorsque le jack est débranché pour que des erreurs liées au débranchement du jack avant la mise à feu soient identifiées le plus vite possible. Tant que le jack n'est pas débranché, le code est dans une boucle infinie.

Dès que le jack est débranché, le buzzer émet un son rapide et un timer commence. Après ce timer, chaque système se met en position ouverte pour faire une éjection.

Partie Lancement de la minuterie



Réalisation

3.1.8 Système d'éjection

Le système est principalement composé de pièces en plastique imprimées en 3D. Les principales difficultés rencontrées sont au niveau des tolérances laissées pour le guidage et le frottement entre 2 pièces en plastique.

Une autre difficulté de la réalisation du système d'éjection est la mise en place d'insert. C'est une pièce métallique utilisée en complément des vis permettant de plus répartir des efforts. Il permet aussi un taraudage dans des pièces en PLA. Pour la placer, il est nécessaire de faire chauffer l'insert puis enfoncer ce dernier sans ajouter de plastique chaud à l'intérieur, sinon la vis ne peut entrer et tourner. L'impression des pièces et la découpe des tiges n'a pas posé de problèmes particuliers.

3.1.9 Système de récupération



Les dimensions du parachute hémisphérique unique composant ce système ont été calculées au moyen d'un tableur excel ayant déjà servi pour une fusée expérimentale antérieure. Hugo Teixeira nous a assisté dans le dimensionnement de ce dernier. Un parachute hémisphérique a été une nouveauté pour le groupe, en effet seuls des parachutes circulaires ont été faits précédemment. La particularité de ces parachutes est

leur structure composée de plusieurs gores. Les gores sont des pièces de forme vaguement trapézoïdale permettant, lorsqu'elles sont toutes mises bout à bout, de créer une demi-sphère. Cette façon de faire est bien plus efficace étant donné le coefficient de traînée plus important pour ce type de parachutes comparé à un simple disque.



Coutures du parachute et des sangles (en jaune)

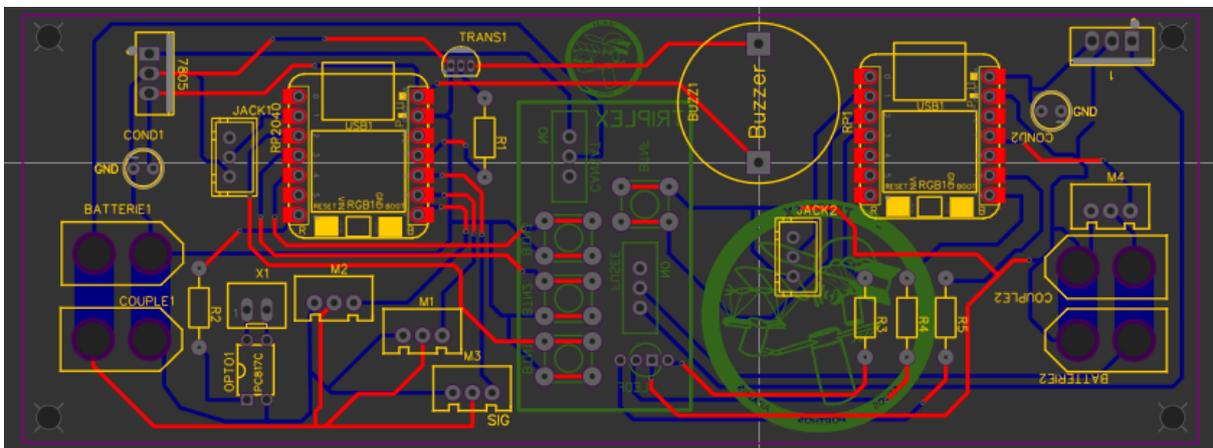
Le parachute a été découpé dans une toile en nylon au moyen d'un patron en carton. Les gores ont été cousues bout à bout avec trois lignes de coutures pour chaque jonction. Ensuite ont été rajoutées les sangles permettant l'attache des suspentes du parachute. Pour finir la cheminée, le trou en haut du parachute qui laisse l'air s'échapper, a été renforcé par un bout de tissu semblable à de la soie.

Le parachute étant composé de huit gores nous avons également attaché huit suspentes à chaque couture inter-gore. La longueur des suspentes est de l'ordre du diamètre du parachute, et elles se rejoignent toutes au niveau d'un émerillon. Cet émerillon transmet tous les efforts vers une unique sangle plus grosse que les précédentes qui rejoint le système eye bolt de la fusée.

Le parachute est éjecté suivant le même principe que notre charge utile. Ce dernier est tout de même attaché solidement au reste de la fusée grâce à un système eye bolt, qui est logé au biais d'une tige filetée dans un bague en aluminium. La bague est elle-même attachée à la peau en fibre de verre au moyen de 8 vis M4 d'une longueur de 16 mm. La tige filetée transfère les efforts du eye bolt à la bague en aluminium par son filetage, mais aussi grâce à une grosse rondelle en aluminium également de l'autre côté de cette bague.

3.1.10 Minuterie

Le schéma électrique conçu est ensuite mis en place sur une pcb où sont soudés les composants. Les composants sont disposés sur les deux faces, la majorité d'un côté (en jaune sur le schéma) tandis que les composants devant interagir avec l'utilisateur (la LED, les interrupteurs et les boutons, en vert) sont placés sur un petit espace rectangulaire de l'autre côté, le but étant de pouvoir y accéder via une trappe. De plus, on relie les moteurs, placés aux niveaux des systèmes éjections, avec des câbles via des JST.



La réalisation de la minuterie s'est faite principalement en binôme. Une personne qui soude et une qui vérifie.

La majorité a été faite au fer à souder et pour les plus gros composants la finition a été faite au pistolet à air chaud.



3.1.11 Code :

En réalisation, le code s'est fait dans le langage Micropython sur le logiciel Thonny pour la RP-2040. Une des plus grandes particularités du code est l'utilisation d'une classe pour coder les servomoteurs. Ceux-ci doivent être commandés en PWM (Pulse with Modulation). Ils ont donc besoin d'une fréquence. Le constructeur de cette classe est le suivant :

```
#Classes
class Servo :

    def __init__(self, pin, position_ouverte, position_fermé):
        self.pin = PWM(Pin(pin), freq)
        self.position_initiale=position_ouverte
        self.position_finale=position_fermé
        self.fermeture()
```

Ici, le constructeur a besoin d'un Pin , d'une position ouverte (le CanSat est lancée), d'une position fermée (le Cansat est dans la fusée) et il se met en position fermé de base. Ensuite, la classe possède plusieurs méthodes :

- Une méthode pour passer le servomoteur en position ouverte appelé "ouverture".
- Une méthode pour passer le servomoteur en position fermée appelé "fermeture".
- Une méthode permettant d'envoyer une position au servo moteur.

-Une méthode permettant de convertir des "duty" en angle pour avoir une position (car les servomoteurs marchent en duty).

-Une méthode dictant l'éjection (position ouverte) après un timer. Celle-ci est utilisée dans le mode lancement après que le jack soit débranché.

-Une méthode appelée toggle qui permet de régler le moteur à l'aide des boutons : Si le servo est en position ouverte et que le bouton est appuyé, alors le servo se met en position fermé et vice-versa. Cette méthode est utilisée uniquement dans le mode configuration.

Le mode configuration s'active lorsqu'un bouton est appuyé après 5 secondes du début de la minuterie. Pour cela, la fonction utilisée est "If any button.value() == 0 for button in buttons" qui s'active lorsque n'importe quel bouton dans une "liste" de boutons est appuyé. Le mode configuration consiste d'une boucle while True qui est une boucle infinie. Pour passer en mode lancement, il faut redémarrer la minuterie.

Le mode lancement se lance si aucun bouton n'a été appuyé après 5 secondes. Celui-ci est une boucle infinie while True qui lit une valeur au niveau du pin du jack pour savoir s'il est débranché ou non (à l'aide d'un If). Dès que le jack est débranché, un timer se lance (la durée est une variable globale et dépend de la trajectoire de la fusée ainsi que de la charge qui va être éjectée). A la fin de ce timer, la fonction éjection est appelée, ce qui "ouvre" le système d'éjection.

Voici ensuite les différences entre les deux parties de la minuterie dans le code :

3.1.11.1 Partie Cansat :

```
0 def buzzer_toggle() :
1     temps=time.ticks_ms()
2     if STATUT_BUZZER == BUZZER_CONFIG and (temps % 1000 < 100) :
3         buzzer.on()
4     elif STATUT_BUZZER == BUZZER_JACK_BRANCHE and (temps % 1000 < 200) :
5         buzzer.on()
6     elif STATUT_BUZZER == BUZZER_JACK_DEBRANCHE and (temps % 200 < 20) :
7         buzzer.on()
8     elif STATUT_BUZZER == BUZZER_LANCEMENT and (temps % 1000 < 500) :
9         buzzer.on()
0     else :
1         buzzer.off()
```

La partie Cansat possède trois temps d'éjections différents pour chaque CanSat ce qui se traduit par 3 timers qui se lancent au même moment dans la boucle de lancement, c'est-à-dire après que le jack est débranché. De plus, un buzzer doit être codé dans cette partie-là. Un buzzer est codé en rapport cyclique, ce qui va influencer le son qu'il va émettre ainsi que la durée de celui-ci. La fonction du buzzer est la suivante :

Dans cette fonction, le buzzer possède 4 états :

- Un état configuration où le buzzer va émettre un son court avec une période un peu longue.
- Un état où le jack est branché. Dans cet état, le buzzer va émettre un son court avec une période un peu longue.
- Un état où le jack est débranché. Dans cet état, le buzzer va émettre un son court avec une période très courte. Cet état permet de faire comprendre à tous les opérateurs autour de la minuterie que le jack a été débranché et que les systèmes d'éjections vont être activés.
- Un état éjecté où le buzzer va émettre un son long avec une période courte. Cet état permet de localiser la fusée après son atterrissage.

Chaque état est connu dans le code à l'aide d'une variable globale appelée STATUT_BUZZER qui est modifiée à chaque passage d'état dans la minuterie.

3.1.11.2 Partie Parachute :

La partie parachute possède une led RGB qui permet de connaître les états de la minuterie:

- La led est rouge lorsque la minuterie vient d'être allumée
- La led est bleue lorsque le mode configuration est activé.
- La led est violette lorsque le mode lancement est activé.
- La led ne s'éteint que lorsque l'éjection se fait.

Erreurs

Il est nécessaire de bien faire attention au branchement de la batterie sur la PCB. La partie batterie est femelle il faut donc mettre une partie Mâle sur la PCB.

Une erreur a aussi été faite lors de la conception de la PCB sur une polarité d'un transistor.

L'accumulation de ses erreurs nous a conduit à refaire les soudures sur une carte neuve. Il est donc vital de prendre soin des parties abordées plus tôt, notamment quand le temps vient à manquer.

3.1.12 Ailerons

Les ailerons de Triplex ont été créés à partir de quatre trapèzes d'aluminium maintenus en place par deux bagues en POM-H.

La forme des ailerons a été décidée grâce au logiciel Stabtraj de planète sciences. L'aluminium nous a été fourni par M. Faux, responsable de l'atelier de l'ESTACA. La plaque de 4mm offerte était un reste qui ne lui était plus nécessaire, ce qui représentait une charge budgétaire en moins sur le projet.

Le POM-H aussi nommé delrin est un matériau dérivé du plastique. Choisi pour sa résistance mécanique, sa facilité d'usinabilité, et aussi pour réduire la masse de la fusée, il répondait au cahier des charges qui était dressé.

Les ailerons sont maintenus par des vis aux bagues en quatre points chacun. Les bagues sont elles-mêmes vissées à la peau en carbone. Le système entier est conçu de façon à

être facilement glissé dans la base de la fusée. Ceci a pour but de faciliter l'intégration de ce système mais aussi de réunir la durée et la complexité de futures réparations.

La fabrication des ailerons se sépare en plusieurs étapes :

- Découpe des formes brutes des ailerons dans une plaque d'aluminium de 4mm d'épaisseur au moyen d'une scie sauteuse guidée par des coins en aluminium. Pendant ce temps un assistant maintenait la plaque en position en appuyant dessus de tout son poids. L'utilisation d'étaux aurait pu être une meilleure option, mais le matériel disponible ne convenait pas à la taille de plaque que nous avons.
- Finition à la fraiseuse avec un plan et des côtes précises déduites du calcul StabTraj. Le travail de fraisage a été effectué par M. Faux.
- Peaufinage des bords d'attaque et de fuite au moyen de limage et de papier de verre pour réduire la traînée de la fusée.

3.1.13 Peau

Pour la réalisation de la peau, nous avons séparé la peau de la fusée en deux tronçons. Un tronçon destiné à accueillir le moteur de la fusée et le second en fibre de verre qui transportera les cansats et le parachute. Comme nous l'avons dit précédemment, le premier tronçon a été récupéré et le second a été fabriqué par l'association Air Esiea.

Nous nous sommes donc occupés de préparer les deux tronçons. Dans le tronçon en carbone nous avons découpé les fentes par lesquelles les ailerons doivent passer car ceux-ci sont fixés à l'aide de bague interne à la peau. Cette étape demande une certaine précision car les fentes doivent être perpendiculaires entre elles, de la bonne largeur, de la bonne longueur, etc... Ces contraintes nous ont conduit à établir un protocole nous permettant d'assurer toutes ces contraintes.

L'objectif de ce protocole est dans l'unique but de ne rien laisser au hasard lorsque nous faisons des tâches irréversibles aussi critiques.

Voici quelques exemples des mesures que nous avons prises avant de se lancer dans la découpe de ces fentes :

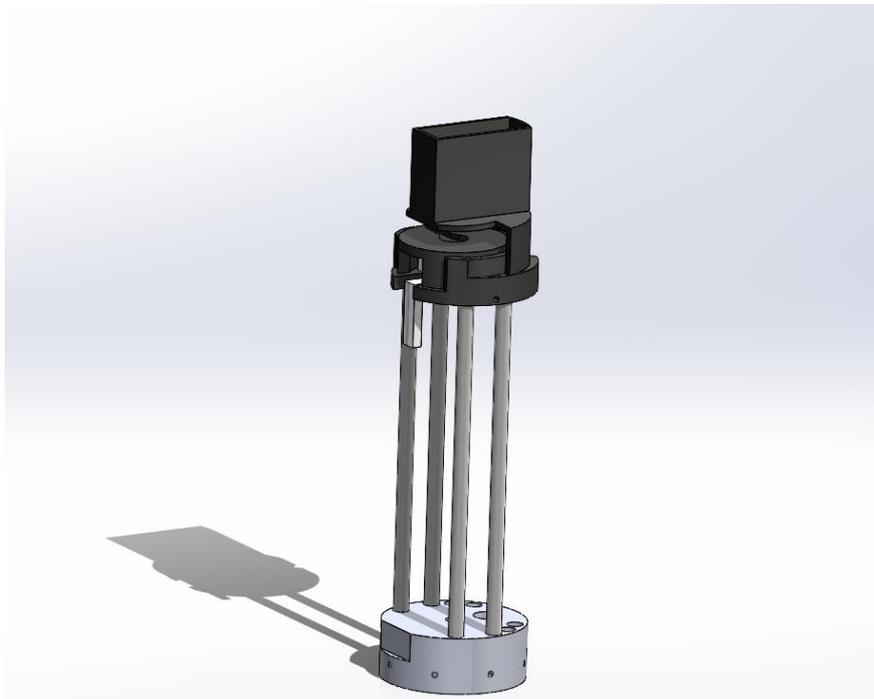
- Afin d'assurer la perpendicularité nous avons imprimé un guide venant se glisser autour de la peau avec des trous espacés exactement de 90° nous permettant donc de placer des marques au niveau de chaque trou.
- Nous avons décidé de découper ces fentes à la dremel, cependant il est facile de faire des écarts avec cet outil donc afin de nous assurer de faire des coupes droites et de ne pas faire d'écart nous avons placé des cornières qui nous indique le chemin à suivre. Ils agissent comme des barrières de sécurité et nous permettent d'assurer qu'on ne fait pas des fentes trop larges.

- Avant de découper les fentes sur le tronçon nous avons effectué des tests sur des chutes de tube de carbone afin de valider notre protocole. Cette étape a été capitale dans la bonne réalisation du travail.

Pour le tronçon en fibre de verre, le plus gros travail était de découper chaque trappe, cependant nous avons délégué cette tâche à Air Esiea qui a utilisé une CNC pour faire les découpes sur le tube. Cependant suite à des difficultés techniques, une petite partie du travail a dû être fini à la main. Nous avons également fait tous les trous permettant la fixation des systèmes internes dans le tube et avons finis par le ponçage du tube.

3.1.14 Tests minif

Durant le projet on procède à une série de tests afin de vérifier la fonctionnalité de nos systèmes. Cependant, pour aller plus loin et valider nos systèmes les plus critiques, nous effectuons un vol de mini fusée dans l'objectif que le système assure sa fonction dans les conditions les plus proches du vol. C'est aussi une bonne manière de prouver aux contrôleurs du C'space qui décideront de la qualification de la fusée que nos systèmes ont déjà "volés" et qu'ils sont sûrs.



Nous avons effectué un seul vol de minifusée baptisé "Miniplex I" qui avait pour but de valider le système d'éjection que nous utilisons pour éjecter les cansats et le parachute. Cette mini fusée a été réalisée en collaboration avec le groupe 4A car ils devaient

également tester leur système GPS. Nous nous sommes principalement occupés du système de récupération de la minifusée. Pour ce faire, nous avons modélisé à l'aide d'une CAO un nouveau système d'éjection en plus petit exemplaire qui avait pour but d'éjecter le parachute de la minifusée. Nous avons par la suite imprimé en 3D et réalisé le système pour le lancement.



Malheureusement, Miniplex I a fait un vol "torche" cela signifie qu'il y a eu une éjection nominale du parachute cependant celui-ci ne s'est pas déployé correctement. Le problème venait d'une erreur de procédure car le parachute qui était attaché à l'aide d'un scotch afin qu'il ne s'emmêle pas avec ses suspentes n'a pas été libéré avant le lancement. Ce test a tout de même été positif pour notre système d'éjection qui s'est exécuté à la perfection. Nous avons également pu détecter plusieurs problèmes dans notre système lors de l'intégration de celui-ci dans la minifusée ce qui nous a permis d'améliorer le système final de la fusée expérimentale.



Erreurs test minif

Ce test nous a appris l'importance de suivre une chronologie de lancement afin d'éviter les oublis malencontreux qui pourraient faire échouer le lancement de la fusée finale.

3.1.15 Miniplex II :

L'objectif de cette minifusée est de tester l'entièreté de la minuterie Triplex c'est-à-dire la partie CanSat et la partie Parachute ensemble. Un seul servo moteur permet une éjection mais les trois autres sont reliés à une autre PCB, faite par le chef de projet Triplex, pour regarder s'ils fonctionnent à l'aide d'un enregistrement des réceptions de signal.

La deuxième version de Miniplex possède plusieurs changements par rapport à la précédente. L'un des plus notables est le fait que les 3A ont principalement fait cette minifusée. Il y a quand même des éléments qui ont été récupérés de Miniplex I : la peau, le parachute, les ailerons et la pièce qui permet de retenir le moteur viennent de Miniplex I. Un des autres changements notables est l'ogive de cette minifusée (cf Annexe 5) puisque celle-ci est beaucoup plus longue que l'ogive de la version originale. Cela permet de rallonger la peau de Miniplex, raccourcie suite à son vol torche, pour abriter la PCB 4A et fixer le support de minuterie.



Malheureusement, celle-ci n'a pas pu voler à cause de multiples reports (temps désastreux, intégration pas prête à temps, refus de lancer sans la présence de tous les membres, examens) jusqu'au moment où le vol ne pouvait plus être effectué puisqu'il n'y avait plus de pyrotechnicien à disposition. L'expérience fut tout de même positive puisque cette mini fusée a permis d'entraîner les 3A à une gestion de câbles ainsi qu'une gestion de "crise" pour résoudre des problèmes arrivant le jour du vol.

3.1.16 Intégration

L'intégration est une étape critique pendant la vie du projet qui consiste à assembler tous les sous-systèmes réalisés auparavant. Cela consiste dans notre cas à tout installer dans la peau de la fusée et à s'assurer que tout fonctionne bien. C'est l'aboutissement du projet mais c'est également l'étape des imprévus ou des problèmes de montage qui n'ont pas été anticipés pendant la phase de conception.

L'une des grosses difficultés que nous rencontrons dans cette phase est l'alignement des trous de vis de chaque élément. Nous avons en effet choisi dans la phase de conception de fixer tous les éléments de la fusée à l'aide de vis qui passent à travers la peau. Cependant, cela implique d'aligner le trou de vis avec le perçage que nous réalisons dans la peau. Cela peut paraître trivial au premier abord mais c'est bien plus compliqué que ça en a l'air lorsqu'il faut faire des alignements à l'aveugle dans une peau en carbone opaque.

Avant d'assembler les éléments, il faut les préparer à l'assemblage en réalisant du ponçage et des perçages. Lors de la réalisation de trous ou de découpes sur une pièce, il est important de ne pas négliger l'attention portée au matériel. En mécanique, il est normal de casser des outils surtout lorsque l'on débute mais une attention toute particulière est nécessaire lors de la manipulation de ceux-ci, surtout lors de la fabrication de pièces en aluminium.

3.2 Cansats

3.2.1 CanSat relai (Aramis)

Étant relativement proche des autres CanSats et de la fusée, il n'aura aucun mal à capter toutes les transmissions LoRa provenant de ces systèmes et contenant leurs positions respectives ainsi que d'autres informations propres à leur mission. Ces données ainsi que la position GPS d'Aramis sont récupérées par le circuit générique, sont transmises par USB à la Raspberry Pi. Elle va pouvoir ensuite les enregistrer sur sa carte SD, les compresser et les envoyer par 4G LTE. Le CanSat va également envoyer par 4G un flux vidéo en direct de sa caméra grand angle.

3.2.1.1 4G LTE

Pour qu'Aramis puisse communiquer à travers Internet, nous l'avons équipé d'un modem 4G LTE. Ce module, d'abord un A7670E fonctionnant sur les bandes LTE Cat-1 (moins énergivores que les Cat-4 standards) n'était pas adapté cause de coupures aléatoires suivies d'un redémarrage. Nous sommes donc passé sur le modem SIM7600E-H déjà testé par le passé sur le ballon B.A.E de l'ESO.



Ce modem de la marque SIMCOM est autorisé à émettre sur les bandes de fréquences de la 4G LTE car le fabricant lui-même est certifié. Nous n'avons pas besoin de licence de radio-amateurs pour avoir le droit d'émettre sur ces bandes, simplement un module certifié et non-modifié.

Le modem est connecté à la Raspberry Pi et utilise le protocole RNDIS pour communiquer avec lui. Pour que la connexion se fasse correctement, il faut suivre les étapes détaillées dans ce gist : <https://gist.github.com/RedBlaze42/ed223e8fd3c49d82bb948d643f275b1f>

3.2.1.2 Caméras

Ce CanSat a la particularité d'être sa propre trappe, de cette façon, il a une vue sur l'extérieur pendant la montée de la fusée mais également pendant la descente du CanSat. Il sera donc équipé de deux caméras pour enregistrer et diffuser l'intégralité du vol.

La première caméra est une Raspberry Pi Camera V2 qui permet de filmer jusqu'en 2k avec un champ de vision assez serré de 62.2°x48.8°. Cette caméra de bonne qualité visuelle sera uniquement utilisée pour enregistrer une vidéo sur sa carte SD puis nous envoyer l'extrait du lancement par 4G une fois au sol, probablement assez lentement vu le débit vidéo (15 Mb/s).

La seconde caméra vient d'Aliexpress et est équipée du capteur OV5647 très connu pour son bas prix. Cette caméra a l'avantage d'avoir une lentille grand angle (160°) mais un piqué très moyen. Nous allons donc enregistrer la vidéo de cette caméra à basse résolution (720p) et transmettre le flux en direct par 4G. Le débit vidéo choisi est de 1 Mb/s pour privilégier la fiabilité du flux plutôt que sa qualité.

Nous avons décidé de ne pas utiliser la Raspberry Pi Camera V3, qui aurait pu être un choix logique car elle est récente et plus performante. Néanmoins, elle a besoin d'une fréquence de bus CSI-2 plus importante ce qui engendre un brouillage des fréquences GNSS. De plus,

elle est équipée d'un autofocus magnétique qui est réputé pour faire vibrer la lentille quand elle subit des accélérations.

L'encodage va se faire en software avec l'encodeur x264 pour garantir le meilleur rapport qualité/bande passante au détriment de l'utilisation CPU.

3.2.1.3 Architecture électrique

- Batterie 3S 28 Wh

Avec le module 4G et l'ordinateur mono-carte, ce CanSat est gourmand en énergie, c'est pourquoi il utilise une batterie constituée de 3 accumulateurs Li-Ion au format 18650 agencés en série. Elle fournit une tension nominale de 10.8V avec une charge totale de 2.6 Ah amenant sa capacité totale d'énergie à 28 Wh. L'autonomie finale dépend beaucoup du taux d'utilisation de la Raspberry et de la 4G qui peuvent varier du simple au triple mais est comprise entre 2 et 6h, suffisant pour tenir la préparation au lancement, le lancement et le transfert de données supplémentaires post-vol par 4G. Cette batterie intègre un circuit de protection qui protège la chimie de la batterie des surintensités, surtensions et sous-tensions. Ce circuit est obligatoire pour le C'Space afin d'éviter des incendies. Cette batterie nous a été offerte dans le cadre d'un partenariat avec GoTronic



Figure 1 - Batterie 3S 28 Wh

- Régulateur UBEC 5V 7A

Ce régulateur de tension à découpage permet d'abaisser la tension de 10.8V à la tension nécessaire par les autres composants de 5V de efficacemnt et avec une excellente stabilité. Il peut convertir jusqu'à 35W de puissance ce qui est utile en cas de pic de consommation.



Figure 2 - Régulateur UBEC 5V 7A

3.2.1.4 Architecture mécanique

L'architecture d'Aramis présente une différence majeure par rapport aux deux autres CanSats : la trappe doit rester attachée au CanSat et la caméra installée à bord doit pouvoir filmer l'intégralité du vol, même avant l'éjection. Idéalement, toutes les trappes devraient rester attachées aux CanSats, mais dans le cas d'Athos et de Porthos, cela pourrait les déstabiliser voire rendre inefficaces leurs systèmes de descente.

Ainsi, pour Aramis, nous n'avons pas simplement décidé d'accrocher la trappe, mais de la rendre une partie intégrante du CanSat en créant quatre pièces qui s'assemblent :

- Le corps principal, conforme aux dimensions réglementaires du cahier des charges du C'Space.
- Le rack électronique, qui se glisse dans le corps principal et auquel sont attachés les composants électriques.
- La trappe, divisée en deux parties :
 - La trappe supérieure
 - La trappe inférieure : équipée d'une ouverture pour la caméra

Les deux parties de la trappe peuvent s'emboîter dans le corps principal, et deux axes en aluminium traversent les deux parties de la trappe ainsi que le corps principal pour fixer et solidifier l'ensemble.

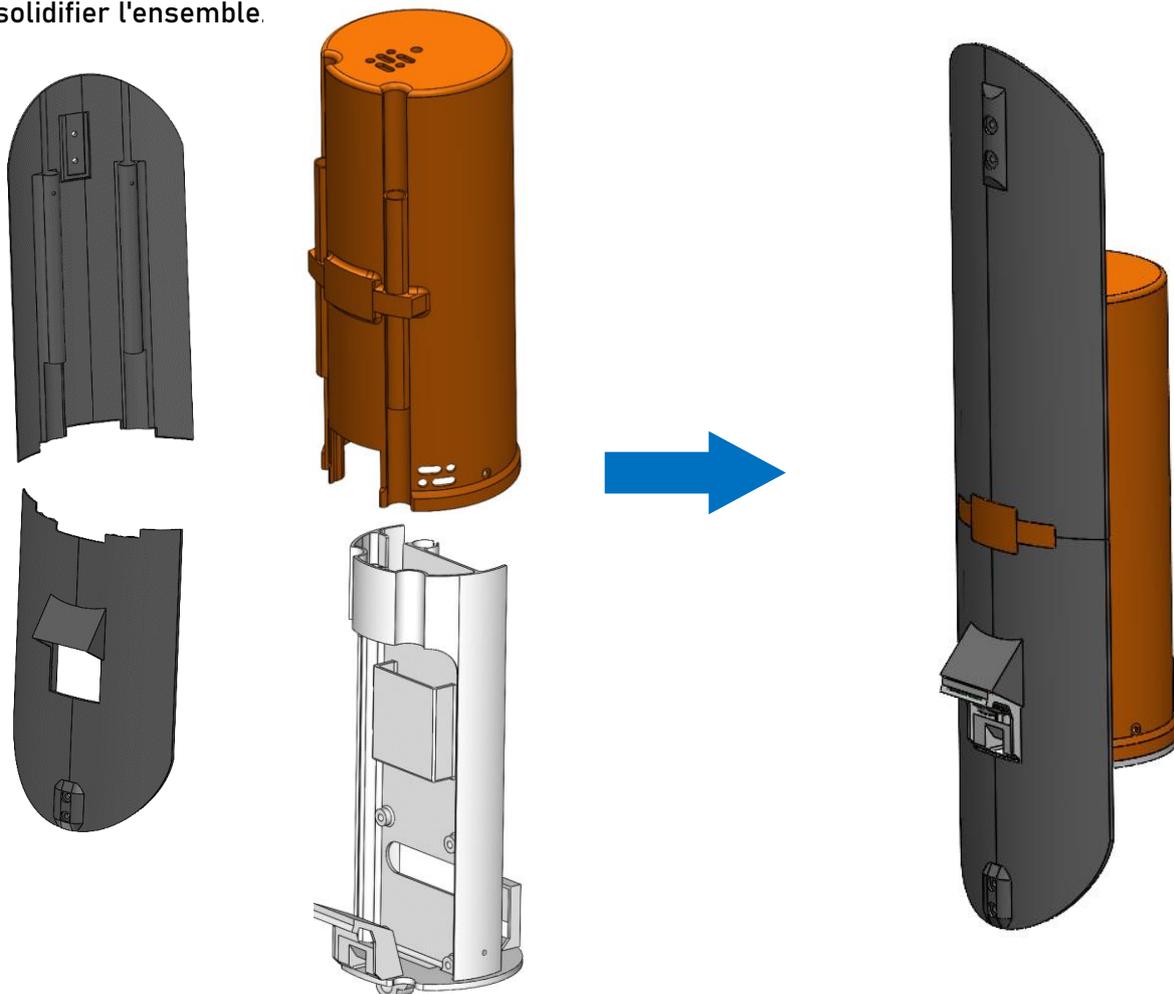


Figure 3 - Modèle 3D d'Aramis

3.2.1.5 Protocoles utilisés

La connexion 4G n'étant pas garantie à 100%, des pertes de paquets peuvent arriver. C'est à dire qu'il est possible qu'un paquet de données n'arrive pas à destination. Le protocole TCP (Transmission Control Protocol), l'un des plus utilisés, détecte cette perte et demande à l'émetteur une retransmission du paquet. Ce système permet de garder l'intégrité totale des données mais peut causer des problèmes de congestion et sur un temps de vol aussi court, engendrer un délai trop important.

Nous utiliserons donc, pendant le vol, des protocoles réseaux basés sur le protocole UDP (User Datagram Model) qui est "sans connexion" c'est-à-dire que l'émetteur peut envoyer des paquets sans attendre de retours.

Pour la transmission vidéo, nous utiliserons le protocole SRT (Secure Reliable Transmit) qui est un protocole de couche "Applicative" du modèle OSI, c'est-à-dire qu'il gère les données par trames et utilise le protocole UDP pour le transport, donc pour le découpage et l'adressage des paquets.

Il ajoute des fonctionnalités au protocole UDP pour améliorer la fiabilité de la transmission. Par exemple, il utilise un système de retransmissions optionnelles : en cas de pertes de paquets, le récepteur demande à l'émetteur une retransmission dans un temps imparti. Si la retransmission n'a pas lieu à temps, toute ou juste une partie de l'image est abandonnée et le récepteur passe à la construction de la suivante. Cela permet d'éviter que des images soient bloquées en attente de la retransmission d'un paquet précédemment perdu. L'intégrité des données est donc partiellement assurée par la couche "Applicative" du réseau.

Pour les données de vol brutes, nous utiliserons notre propre protocole de couche "Applicative", utilisant le protocole UDP pour le "Transport".

Après le vol, si la connexion le permet, nous aurons le temps de nous connecter au CanSat en utilisant le protocole SSH (Secure Shell Protocol) pour le contrôler à distance et lui demander de nous transmettre les données accumulées avec un protocole garantissant l'intégrité des données. De la même manière que le SRT, le SSH est un protocole de la couche "Applicative", mais utilisant le protocole TCP pour le "Transport". L'intégrité des données est donc garantie dès la couche de "Transport".

3.2.1.6 Compression des données

Une fois enregistrées sur la carte SD, les mesures d'Aramis et des autres cansat sont compressées avant d'être envoyé par 4G. L'algorithme de compression utilisé est Zstandard. Il apporte plusieurs avantages. Premièrement il permet une compression rapide et efficace, il peut compresser un paquet de 350 octets en 250 octets en ne prenant que 100µs de temps CPU.

Dans ces 250 octets de sortie il a également ajouté des sommes de contrôles qui permettent de détecter toute corruption qui a pu apparaître sur la transmission 4G. Cette somme de contrôle est créée par l'algorithme XXH64 qui donne une somme de contrôle de grande



Figure 4- Antenne relais qui sera utilisé pour le vol

qualité, avec moins de collision possible que le classique CRC32. Il est également plus rapide que certains algorithmes de sommes de contrôles cryptographique, les benchmarks montrent qu'il est aussi rapide que la copie depuis la RAM.

3.2.1.7 Essais en vol

En Avril 2024, nous avons lancé une minifusée de test « Miniplex 4G ». Elle faisait 80mm de diamètre, 1.1m de haut pour 1.6 kg. À cause de son poids, elle est allée à

Pendant le vol, la transmission 4G a été perdue lors de l'accélération mais a pu être regagnée très vite et les données qui n'ont pas pu être envoyées pendant la montée se sont transmises d'un coup. L'entièreté de la vidéo du vol ainsi que des données de capteurs tests ont pu être récupérées sans perte pendant la descente par le serveur de réception.

Les photos que vous voyez proviennent du flux en direct. L'ordinateur embarqué n'était pas aussi puissant que celui qui est dans Aramis, d'où la qualité moindre des images. Le test étant concluant, nous avons pu continuer la fabrication de la version finale.



3.2.2 CanSat parafoil (Athos)

Athos est un CanSat auto-guidé porté par un parapente. Son but est relativement simple en théorie mais complexe en pratique : suivre le CanSat auto-rotatif grâce aux données GNSS de ce dernier tout en envoyant ses positions en direct au CanSat relais par LoRa. Il sera donc composé d'un module comprenant un microcontrôleur (ESP32), un GNSS et un émetteur-récepteur LoRa d'une batterie au lithium et de deux servo-moteurs contrôlant le parapente. Ce CanSat est un vrai défi, son asservissement, son déploiement ainsi que son contrôle nécessitent beaucoup de tests pour qu'il respecte le cahier des charges et ne dérive pas.

3.2.2.1 Historique

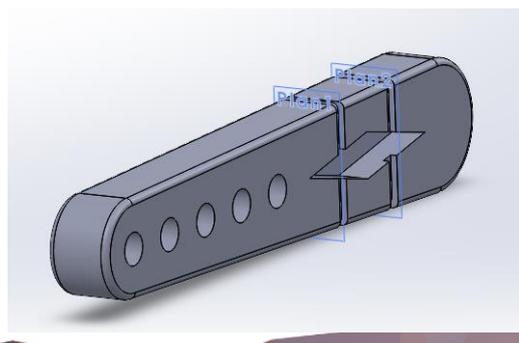
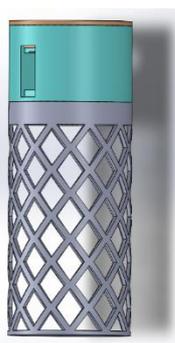
Comme les autres CanSat, Athos se base sur une méthode itérative pour son développement. C'est-à-dire que plusieurs prototypes ont été fabriqués, possédant chacun des différences et améliorations par rapport à son prédécesseur. Athos a eu 6 versions avant d'arriver à sa version la plus aboutie. Le CanSat et ses prototypes ont été focalisés uniquement dans la mission principale (contrôle de trajectoire avec parapente) bien que ce dernier soit inscrit au concours CanSat.

3.2.2.2 Développement

Comme énoncé précédemment il y a eu 6 prototypes avant qu'Athos se retrouve au C'Space. Voici donc un résumé du développement de ces itérations :

- 1^{er} et 2^{ème} prototypes

Ici nous étions au balbutiement du projet, le but était juste de savoir comment contrôler le parapente. C'est ainsi qu'un cylindre aux dimensions du concours vit le jour. Pour vérifier le contrôle nous lançons le prototype du haut du 4^{ème} étage de l'école en tirant à gauche ou à droite le parapente. Malheureusement après de nombreux essais infructueux, des améliorations non-efficaces (les « oreilles ») et des crashes mortels pour Athos v1 et v2 nous avons déterminé que cette méthode posait des problèmes et ne contrôlait pas efficacement le CanSat. Dans ces problèmes rencontrés nous retrouvons : des problèmes de déploiement du parapente à cause d'emmêlement de suspentes ou de la hauteur de largage trop basse ou les deux et un problème de dimensions, en effet, les oreilles dépassaient des dimensions autorisées pour le concours.

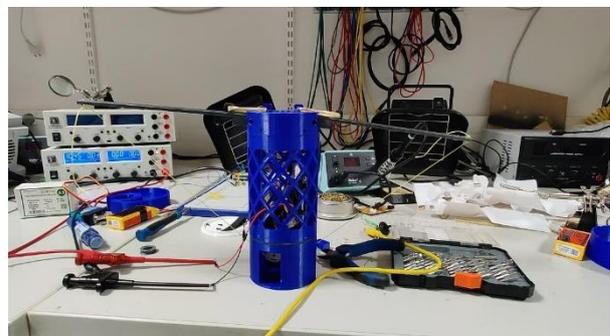


- 3^{ème} prototype

Le 3^{ème} prototype change assez radicalement de ses prédécesseurs, en effet, les oreilles ne sont plus et font place à des bras déployables (grâce à des élastiques forçant l'ouverture), augmentant considérablement l'envergure du parapente. Il a été remarqué que le parapente se déployait bien mieux plus les points d'attaches étaient éloignés. Cela permet d'éviter ainsi d'avoir le parapente qui se replie sur lui-même aux extrêmes. Pour ce 3^{ème} prototype nous gardons cependant le même système pour faire tourner le CanSat, c'est-à-dire, que nous tirons un côté ou un autre du parapente pour pouvoir changer la direction d'un côté ou de l'autre. Le changement pour ce prototype opère dans la réalisation, notamment avec la prise en compte des bras déployables. Désormais les servos moteurs se situe au plus bas du CanSat et possèdent des poulies permettant d'augmenter la course des bras. Nous avons donc réglé le deuxième problème des prédécesseurs sur les dimensions du CanSat.

Pour tester ce prototype nous changeons de lieu et nous nous rendons au viaduc des Fauvettes situé à 30min de l'école. Ce viaduc présente l'avantage d'être plus haut que l'école (environ 25m), possède un terrain relativement dégagé en face (il y a cependant la forêt et donc des arbres aux extrémités du viaduc) et est accessible facilement au public.

Lors de ces tests nous avons essayé de tester au maximum, aussi bien la direction mais aussi le déploiement en essayant de lancer le CanSat avec le parapente plié « en condition similaire du largage depuis la fusée ». Les résultats des tests sont mitigés, concernant la direction nous remarquons des changements mais ces derniers sont tellement légers que nous avons du mal à déterminer si cela était grâce à notre système ou à cause du vent. Pour la partie déploiement nous avons fait 2 essais, 1 échec et 1 réussite, donc nous étions également déçus.

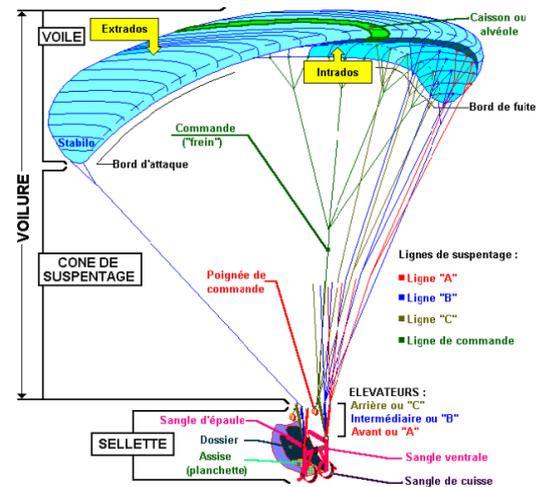


- 4^{ème} & 5^{ème} prototype

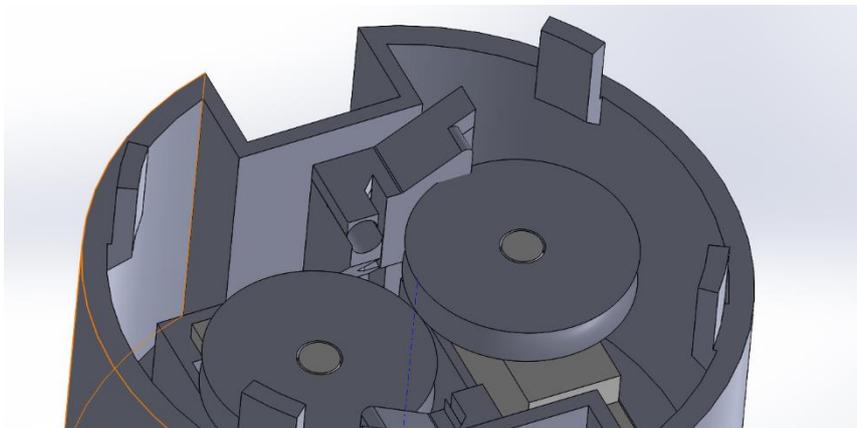
Après les essais décevants du 3^{ème} prototype nous avons pris du recul et avons fait des recherches sur le contrôle de la direction pour les parapentes. Nous utilisons un cerf-volant ressemblant aux réels parapentes car cela était la seule alternative que nous avons trouvé qui soit économique (10/15€ sur AliExpress), petit en dimensions (1m40 d'envergure) et qui ne demandait pas d'en fabriquer un soit même (perte de temps considérable).

Pour pouvoir se mouvoir les parapentistes utilise un système assez différent de celui utilisé sur les anciens prototypes du CanSat. Le principe se base sur des suspentes dites de commande qui sont attachées au bord de fuite du parapente. Si vous souhaitez tourner à

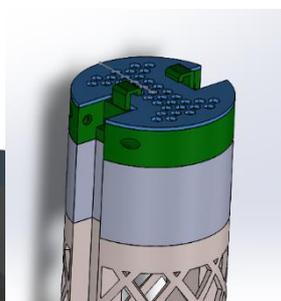
droite il faut tirer ces suspentes ce qui baissera le bord de fuite droite, engendrant une grosse trainée de ce côté et ainsi un virage. A l'instar des ailerons d'avions mais avec une voile.



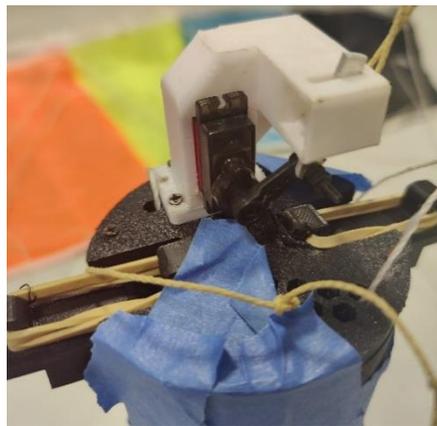
Nous avons donc maintenant deux types de suspentes : de "poids" et de "commande". Désormais il est inutile d'avoir les servos moteurs aussi bas, c'est pour cela que l'étage de contrôle avec les servos moteurs se situe dorénavant tout en haut du CanSat et les poulies sont orientées parallèles au sol pour optimiser la place et pour maximiser la taille des poulies. En sortie de poulie la suspente est dirigée, par un petit "tunnel", vers le haut du CanSat et donc le parachute.



Comme pour les anciens prototypes, le 4ème a été largué depuis le viaduc des Fauvettes. Lors de ces tests nous nous sommes rendu compte que ce système marchait extrêmement bien et il est devenu donc évident qu'il allait être gardé pour les prochaines itérations.



La différence entre le 4ème et le 5ème prototype est la présence d'un système de largage depuis un drone. Permettant ainsi des vols beaucoup plus long (largage à 100m) et relativement moins loin pour aller le larguer (30min à pied de l'école) dans un endroit beaucoup plus adapté et sûre (champs). Le drone nous a été gracieusement prêté par Arthur Gerst, président de la Siera. Merci encore à lui car nous avons abimé son drone et il n'a rien dit.



Pour le 5ème prototype les essais étaient mitigés. Nous utilisons la LoRa présente à bord pour pouvoir télécommander le CanSat. Les tests étaient mitigés pour plusieurs raisons : la météo, nous n'avions pas eu de chances et il pleuvait quasi systématiquement quand nous allions pour tester. D'ailleurs pour cette raison je voulais particulièrement remercier tous les membres des CanSat : Hugo A. en priorité, Timothé, Tristan et Hugo T. sur la fin de l'année où la météo était plus clémente. Un deuxième problème qui engendrait un échec récurrent des tests est l'emmêlement des suspentes qui lors du largage, le plus embêtant est que ce problème n'était pas solvable en l'état car nous ne pouvions pas faire autrement que de laisser le parapente trainer en dessous du CanSat. Mais après de nombreux essais nous avons réussi à contrôler assez correctement le CanSat !

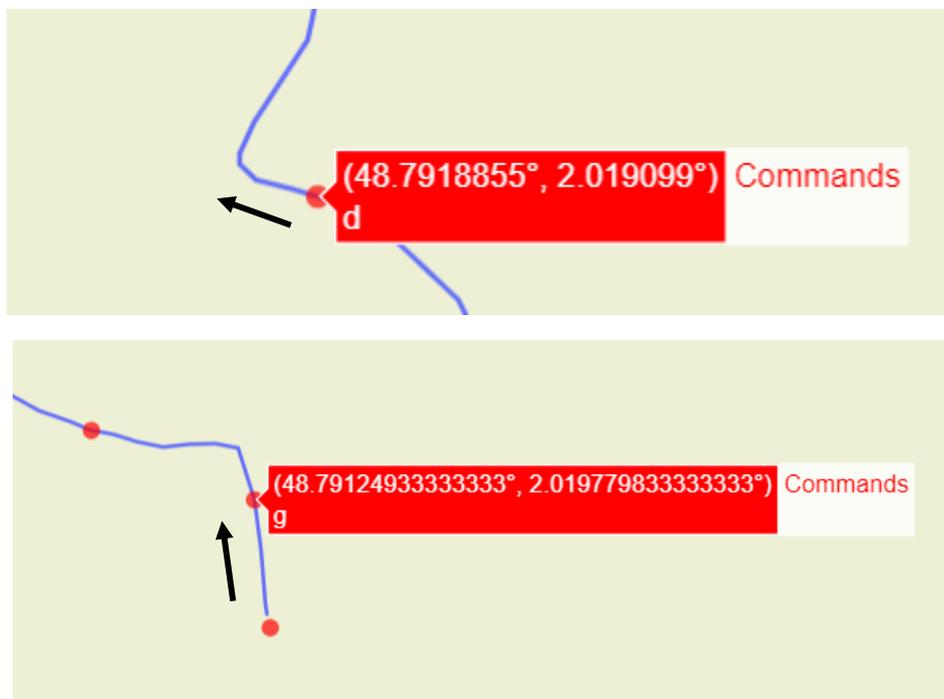


Figure 6 5ème prototype après les tests

Quand nous contrôlions le CanSat par télécommande, la commande consistait à baisser un côté pendant quelques secondes à fond puis revient au neutre. Le but final était de varier se temps pour tourner à l'angle voulu. Au final nous n'avons jamais fait varier ce temps car nous étions en fin d'année et je n'avais plus la motivation d'effectuer ces tests seul pendant plusieurs heures, d'autant plus que j'étais sur d'autres projets.

Les vols sont disponibles dans cette playlist non répertoriée :

<https://youtube.com/playlist?list=PLGmEpvQDSshVf2WVyBF2wpbRPS9ijGYbr&si=2wplvmhD2x0nKQqP>



Il y a eu sur les dernières semaines avant le C'Space un élan de reprise pour faire la partie software vu que la partie mécanique était fixée. Pour l'électronique, ça n'a pas été évoqué mais comme les autres CanSat du projet on retrouve une batterie, le GPS tracker avec GPS, LoRa, microcontrôleur intégré, et un BMP280.

Pour le programme final j'ai été complètement carry par Nicolas Blin qui n'est pas identifié dans le projet mais qui m'a grandement aidé. Nous n'avions pas pu faire de grands tests à 100m de haut comme prévu pour ajuster le PID mais il a réussi à avoir des résultats cohérents. Les paramètres n'ayant pas été ajustés nous sommes partis au C'Space avec un programme assez rudimentaire mais intégré partiellement au code global.

Pour parler rapidement du C'Space, cette fois-ci j'ai beaucoup été aidé par Tristan qui m'a beaucoup épaulé sur Athos en partie sur l'intégration du code dans le code général. Après plusieurs galères, notamment pour le concours, un vol a été effectué mais un seuil d'altitude avait été mis avant de déclencher le contrôle de direction et ce seuil n'a jamais été atteint car nous avons utilisé l'altitude du GPS plutôt que celui du BMP qui ne nous inspirait pas confiance quelques minutes avant le vol. Athos a donc juste plané. Le déploiement a été sans accro et petit fun fact : nous sommes arrivés le plus proche de la cible des autres candidats du concours.

Mais peut-être qu'Athos reviendra... C'était une chouette aventure qui sur la fin a eu de nombreux problèmes en grosse partie à cause d'un manque d'investissement de ma part au profit d'autres projets.

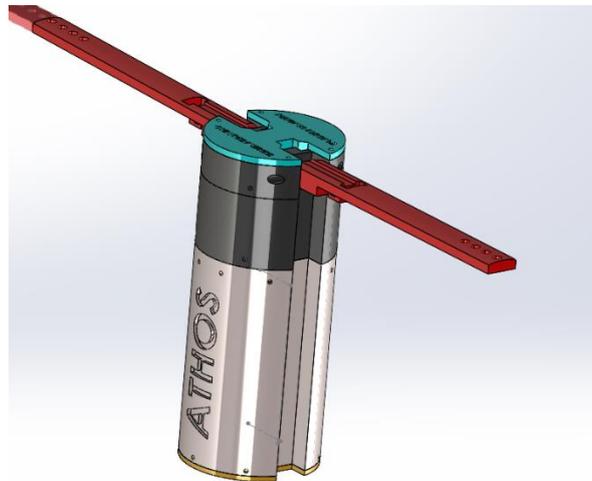




Figure 7 Version finale du CanSat Athos

3.2.3 CanSat auto-rotatif (Porthos)

L'objectif de Porthos est de ralentir sa descente par l'auto-rotation de 3 pales montées sur un rotor. Chacune des pales est séparée en 2 parties : une partie créant la rotation (proche du rotor) et une partie créant la portance (loin du rotor). Le CanSat sera équipé d'un capteur à effet hall afin de récupérer la vitesse de rotation du rotor (en installant un aimant dessus) ainsi que d'une centrale inertielle comprenant un gyromètre qui mesurera la rotation propre au CanSat.

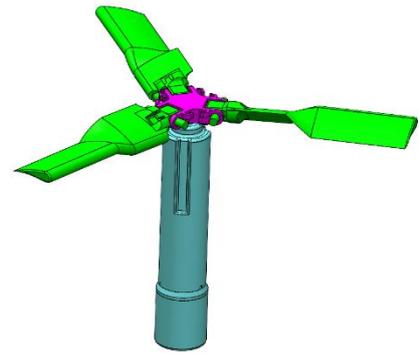


Figure 8 - Modèle 3D de Porthos en CAO sur Solidworks

De plus, nous y installerons un altimètre, une batterie et un GNSS afin de transmettre sa position par signal LoRa à Athos (qui doit le suivre) et à Aramis (qui doit retransmettre l'information au sol).

3.2.3.1 Autorotation

Le principe d'autorotation est majoritairement utilisé par les hélicoptères en cas de panne moteur. Si l'altitude est suffisante, la chute provoque un flux d'air ascendant qui permet de faire tourner les pales. L'hélicoptère descend ainsi plus doucement et peut atterrir sans dégâts.

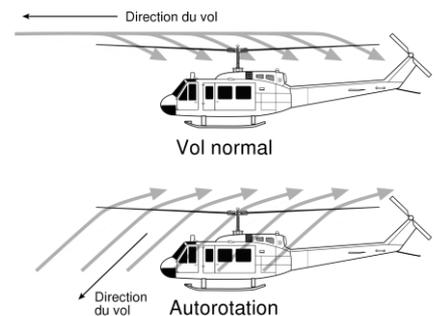


Figure 9 - Fonctionnement de l'autorotation sur un hélicoptère

Pour provoquer ce mouvement, notre pale doit être séparée en 2 sections :

- La section "auto-rotative", située près du rotor, induit la rotation en présentant une incidence négative par rapport au plan horizontal du CanSat. En effet, le vent relatif traversant la pale peut se projeter sur deux composantes distinctes : une composante verticale, provoquée par la chute du CanSat, et une composante horizontale, engendrée par la rotation de la pale. Cela signifie que l'inclinaison adéquate de cette section de la pale permettra au vent relatif de générer une force, principalement de traction (mais également de portance), entraînant ainsi le rotor de manière similaire au fonctionnement d'une éolienne.

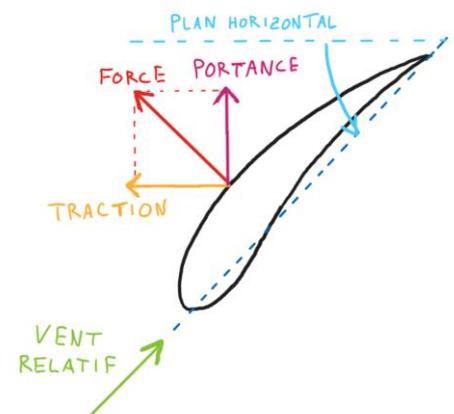


Figure 10 - Section Auto-Rotative

- La section "portante", éloignée du rotor, présente une incidence légèrement positive par rapport au plan horizontal du CanSat. Cet angle, combiné à la rotation, permet grâce à l'écoulement sur la pale, de produire une force de portance ainsi qu'une traînée légère, similaire au fonctionnement de l'aile d'un avion.

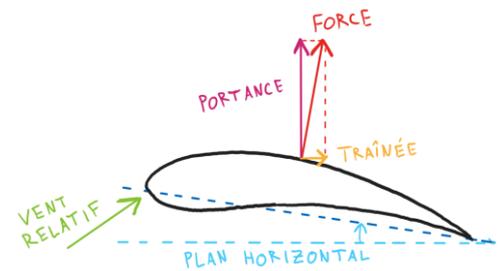


Figure 11 - Section Portance

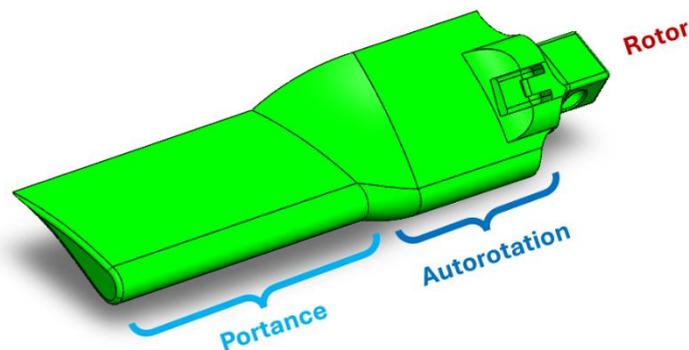


Figure 12 - Modèle 3D d'une pale en CAO sur SolidWorks

Le profil que nous avons sélectionné reste constant le long de la pale. Il est basé sur un profil NACA 5510 caractérisé par une cambrure prononcée, une faible épaisseur, et une grande flèche (c'est-à-dire un écart important entre la corde et la ligne moyenne). Ce choix de profil vise à optimiser la portance au détriment de la solidité, un critère qui n'est pas particulièrement crucial dans notre décision, étant donné que les pales ne sont soumises qu'à peu de contraintes pendant le vol. De plus, ce profil est capable de générer de la portance même en présence d'une incidence nulle.

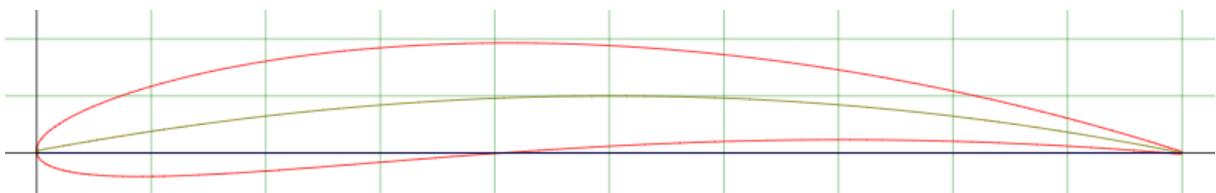


Figure 13 - Profil NACA 5510

Les trois pales, attachées au rotor, sont elles-mêmes reliées au corps par un axe monté serré dans 2 roulements à billes, permettant la rotation. Ce nombre de pales s'explique par le fait que chaque pale génère des perturbations après son passage. Ainsi, un excès de pales augmenterait la traînée produite et ralentirait le rotor.

Dans le cas d'un turbopropulseur, plus il est puissant, plus le nombre de pales augmente permettant d'avoir une meilleure poussée, le moteur pouvant compenser la traînée générée. Dans notre configuration, sans moteur, l'objectif était de minimiser au maximum la traînée tout en trouvant un équilibre pour générer suffisamment de portance.

À l'origine, il n'y avait qu'un seul roulement maintenant l'axe en place. Cependant, cela entraînait un léger effet de rotule entre le rotor et le corps du CanSat, ce qui aurait pu le déstabiliser. En ajoutant un deuxième roulement, nous avons pu transformer cet effet de rotule en un simple pivot, évitant ainsi toute déstabilisation.

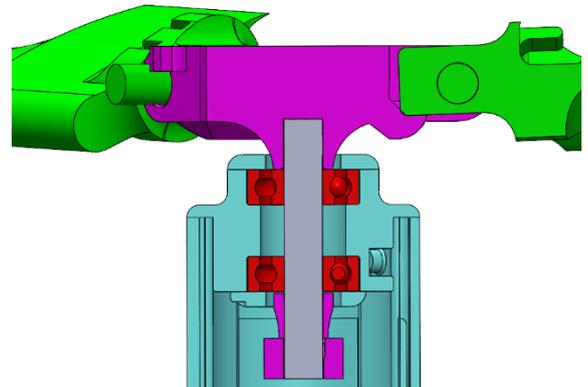


Figure 14 - Vue en coupe du montage du rotor, de l'axe et des roulements à billes

3.2.3.2 Incidence et répartition

Dans l'objectif d'optimiser la portance générée par chacune des pales, plusieurs caractéristiques doivent être définies, notamment les angles d'incidence de chaque section et la répartition de la surface de la pale entre les deux sections.

- Incidence de la section portance :

Cet angle n'a d'influence que sur la portance générée par cette section. Plus il est prononcé, plus la portance augmente, entraînant simultanément une augmentation de la traînée. Bien que le profil que nous avons choisi nous permette de générer de la portance sans angle d'incidence, nous avons choisi d'appliquer un angle de 10° , théoriquement capable de tripler la portance par rapport à un angle de 0° . La traînée induite demeurant négligeable.

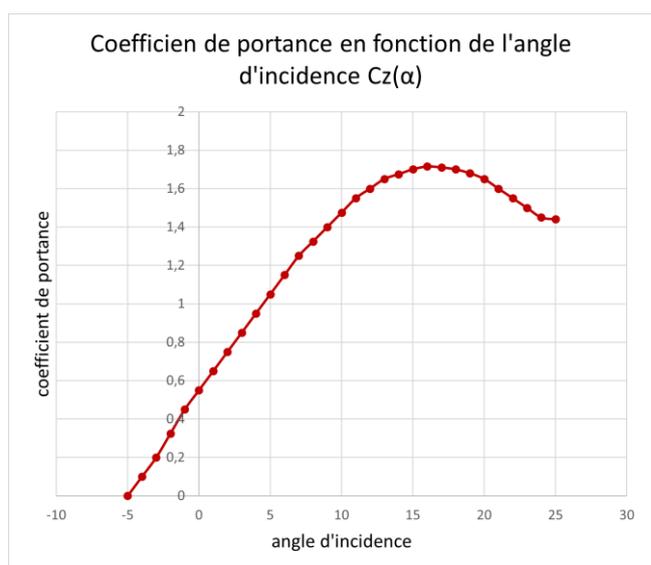


Figure 15 - Courbe du coefficient de portance C_z en fonction de l'incidence

- Incidence de la section auto-rotative :

Cet angle a une influence non seulement sur la portance de cette section, mais surtout sur la traction générée, et donc sur la vitesse à laquelle le rotor va tourner. Or, plus la vitesse augmente, plus la portance générée par la section "portance" est grande. Ainsi, cet angle doit permettre de faire tourner le rotor le plus rapidement possible, en le freinant le moins possible. Nous devons donc choisir un angle au plus proche du vent relatif, favorisant ainsi une traction et une portance maximales. Ne pouvant ni connaître ni mesurer l'angle du vent relatif, nous avons initialement fixé cet angle à -45° , tout en étant conscients que nous pouvions le faire varier lors des tests en soufflerie.

- Répartition des 2 sections :

Cette répartition de la pale entre les deux sections est cruciale et influence largement la portance générée. En effet, plus la section "auto-rotative" est grande (c'est-à-dire, plus sa surface est grande), plus le rotor tourne rapidement, ce qui augmente la portance générée par la section "portance". Cependant, à mesure que la section "portance" diminue, la portance générée diminue également. Nous devons donc trouver un équilibre optimal. Cette caractéristique a été mesurée en fonction du pourcentage que représente la section "auto-rotative". Ainsi, lorsque nous mentionnons une répartition de 40%, cela signifie que 40% de la longueur de la pale est dédiée à la section auto-rotative, tandis que le reste (60%) est alloué à la section portance. Il est important de noter que cette caractéristique est également ajustable lors des tests.

Dans l'objectif de concevoir et tester plusieurs pales avec les caractéristiques précédemment mentionnées qui diffèrent, nous avons mis au point une notation permettant de les identifier rapidement :

Pâle AA-BB CC%

avec > AA : incidence de la section auto-rotative (en valeur absolue)

> BB : incidence de la section portance

> CC : pourcentage de la longueur dédiée à la section auto-rotative

Par exemple une pale ayant -45° d'incidence sur la section auto-rotative, 10° d'incidence sur la section portance et dont la longueur de la section auto-rotative représente 40% de la longueur de la pale sera nommée : Pâle 45-10 40%

3.2.3.3 Essais en soufflerie

Dans le but de déterminer les valeurs optimales pour les incidences et la répartition, nous avons réalisé des tests dans la soufflerie de l'ESTACA. Nous avons ainsi mis en place le système de test suivant :

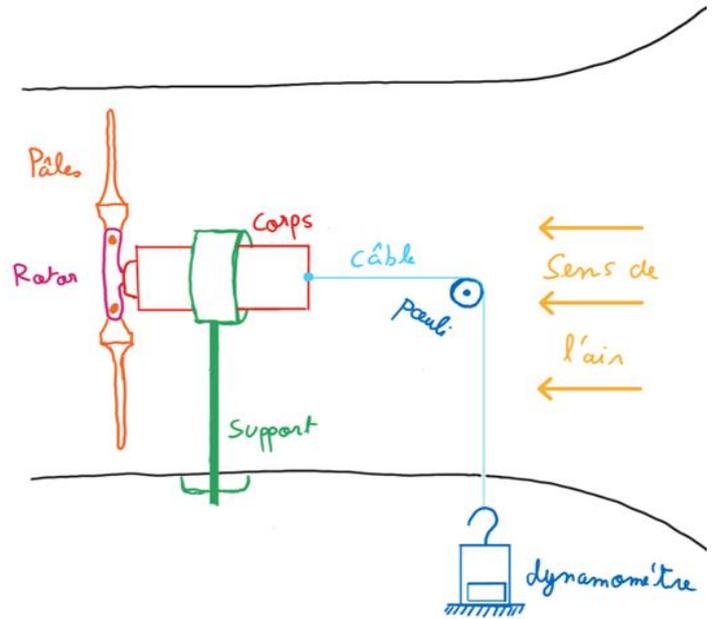


Figure 16 - Schéma du montage de test

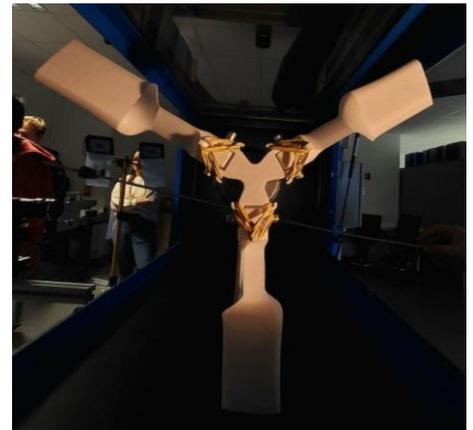


Figure 17 - Photos du montage de test

L'objectif était de tester plusieurs pales avec des incidences et des répartitions différentes. Un dynamomètre était relié à l'arrière du CanSat par un câble afin de mesurer la portance exercée. Les différentes pales utilisées pendant le test étaient les suivantes :

45-10 10% / 45-10 30% / 45-10 30% red / 30-10 30% / 45-10 50%

Le terme "red" signifie "réduit". Pour ce type de pale, nous avons testé la réduction de la corde du profil sur la section auto-rotative. Nous souhaitions vérifier si, en ayant une surface plus petite et donc en tournant moins vite, nous pourrions maintenir une portance suffisante. Ainsi, nous avons pu tester chacune de nos pales à différentes vitesses de l'écoulement et nous avons obtenu les courbes suivantes :

Portance en fonction de la vitesse de l'écoulement L(v)

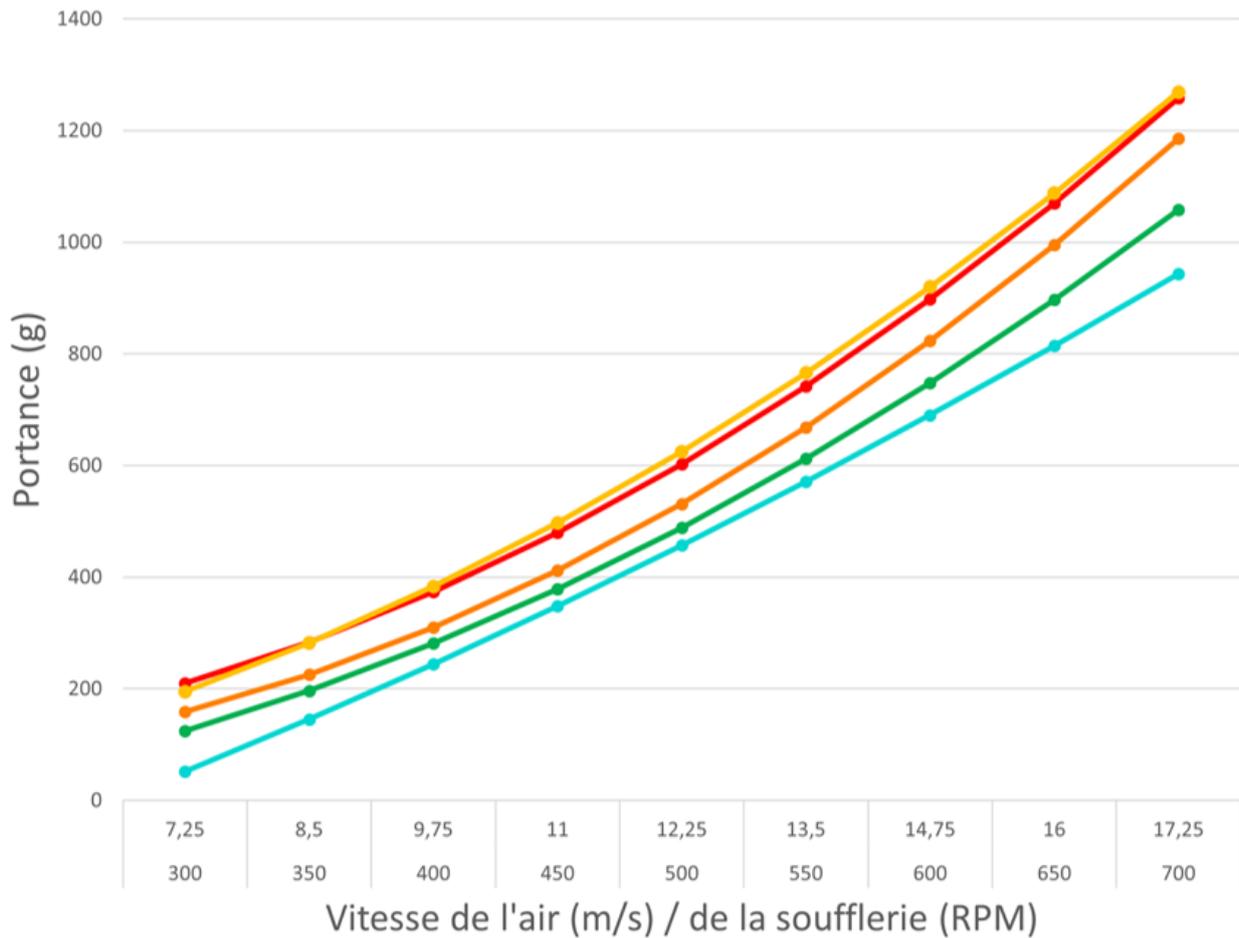


Figure 18 - Résultat du test en soufflerie

Au vu des résultats, on peut remarquer que la répartition a une influence significative sur la portance générée, et que la plus optimale est à 30%. En ce qui concerne la version "réduite", elle fonctionne moins bien que la version non réduite, même si elle reste plus efficace que les autres. Enfin, on peut constater que la version avec une incidence de la section autorotative moins prononcée (-30°) présente une portance légèrement supérieure. Cela peut s'expliquer par le fait que le vent relatif possède un angle plus proche des -30° que des -45°.

Selon le cahier des charges du C'Space, les CanSat doivent tomber à une vitesse comprise entre 5 et 15 m/s et ne doivent pas peser plus de 1 kg. En lisant la courbe jaune (30-10 30%), celle de la version la plus optimale, et en effectuant une interpolation linéaire, on trouve qu'à 10 m/s la portance est de 3,924 N (400 g * 9.81) et à 15 m/s elle est de 9,320 N (950 g * 9.81). Pour ralentir, Porthos doit être capable de supporter son propre poids, alors que son poids actuel est de 550g. Ainsi, on peut conclure que la pale 30-10 30% est suffisante, c'est donc celle que nous utiliserons pour le projet.

3.2.3.4 Réduction du diamètre

D'après le cahier des charges du C'Space, les CanSat doivent avoir une hauteur maximale de 200 mm et un diamètre maximal de 80 mm. Si la hauteur n'est pas un problème particulier, limitant simplement la longueur des pales, le diamètre pose quant à lui davantage de difficultés. En effet, lorsque Porthos est à l'intérieur de la fusée, ses pâles sont repliées le long du corps, occupant beaucoup d'espace et nous obligeant à avoir un corps avec un diamètre plus petit.

Bien que notre première idée ait été de simplement réduire le diamètre du corps, nous nous sommes rapidement rendu compte que le bord d'attaque de la section auto-rotative de la pale, ayant un angle négatif, venait heurter le corps. Les pales ne pouvaient alors pas se replier complètement et nous avons toujours un diamètre trop grand. Nous avons alors envisagé une nouvelle forme de corps avec des espaces pour les bords d'attaque des sections auto-rotatives, permettant de replier complètement les pales sans perdre trop d'espace.

Lors du redimensionnement, nous avons été confrontés à un autre problème. Pour déployer les pâles lors de l'éjection, un système d'élastiques reliant un crochet au sommet de chaque pale à des crochets situés aux bords du rotor a été mis en place. Ce système permet de tendre les élastiques en repliant les pâles pendant l'installation dans la fusée, puis, au moment de l'éjection, de déplier automatiquement les pâles. Cette solution mécanique évite de recourir à un déploiement électrique qui pourrait ne pas fonctionner en cas d'erreur de code, de problème électrique ou de câble endommagé.

Cependant, dans la première version, les crochets sur les pâles dépassaient, les rendant fragiles et augmentant le diamètre lorsque les pâles étaient repliées. Pour réduire ce diamètre et éviter la casse des crochets, ils ont été installés à l'intérieur des pâles. Cette modification a rendu l'installation du système légèrement plus complexe, mais elle a permis d'éviter les problèmes mentionnés précédemment.

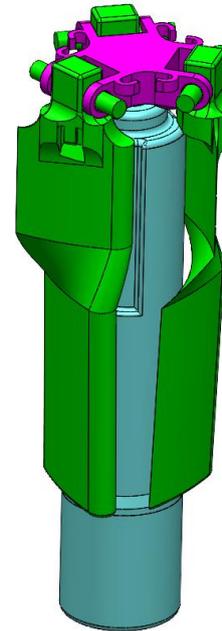


Figure 19 - Modèle 3D du CanSat en position replié (à l'intérieur de la fusée)

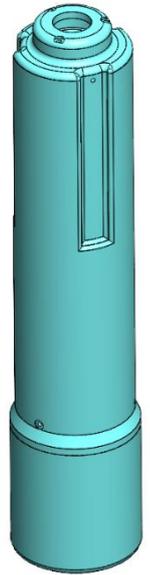


Figure 20 - Modèle 3D du corps redimensionné du CanSat

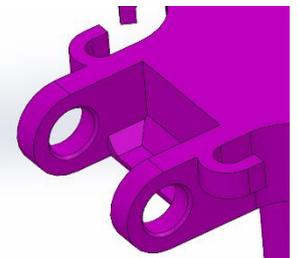


Figure 21 - Crochets élastique rotor

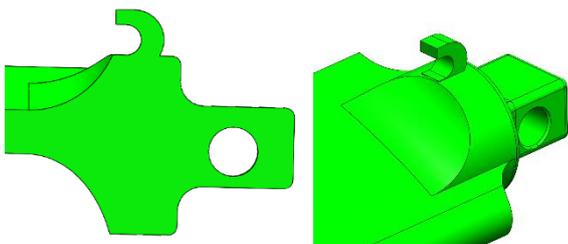


Figure 22 - crochet élastique extérieur (1ère version)

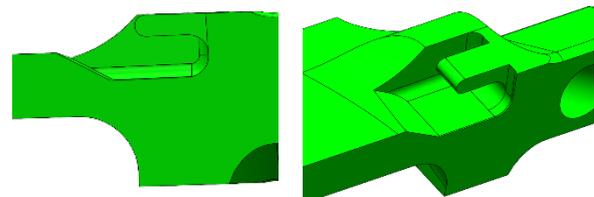


Figure 23 - crochet élastique intérieur (2ème version)

3.2.3.5 Stabilité et centre de gravité

Pendant la phase de conception, nous avons effectué de nombreux tests, notamment en lançant le CanSat depuis le haut d'un viaduc de 30 mètres de hauteur. Au cours de ces tests, nous avons observé que le CanSat avait du mal à se stabiliser, ce qui rendait l'entraînement des pâles par autorotation plus compliqué et entraînait des difficultés à ralentir. Cette instabilité est due au centre de gravité relativement élevé du CanSat. En effet, la majeure partie du poids se trouve dans le système d'autorotation (roulements, rotor, pâles), tandis que la partie inférieure du CanSat ne contient pas de poids significatif (PCB, batterie, ...).

Pour remédier à ce problème d'instabilité, nous avons ajouté une extension au bas du CanSat, reliée au rack électrique et équipée d'un réservoir fermable dans lequel nous pouvons ajouter du poids. Avant l'ajout de cette extension, le CanSat pesait environ 350 g. Grâce à l'extension, nous avons pu ajouter 200 g dans le réservoir, portant le poids total à 550 g. En plus de faire descendre le centre de gravité et donc d'augmenter la stabilité, l'ajout de cette masse permet au CanSat de descendre plus rapidement, à environ 12 m/s, ce qui reste conforme aux exigences de vitesse de descente du cahier des charges du C'Space (comprises entre 5 et 15 m/s). Étant donné que le CanSat Athos doit suivre Porthos, il est essentiel que ce dernier tombe plus rapidement, ce qui est pratiquement assuré avec une vitesse de descente de 12 m/s.

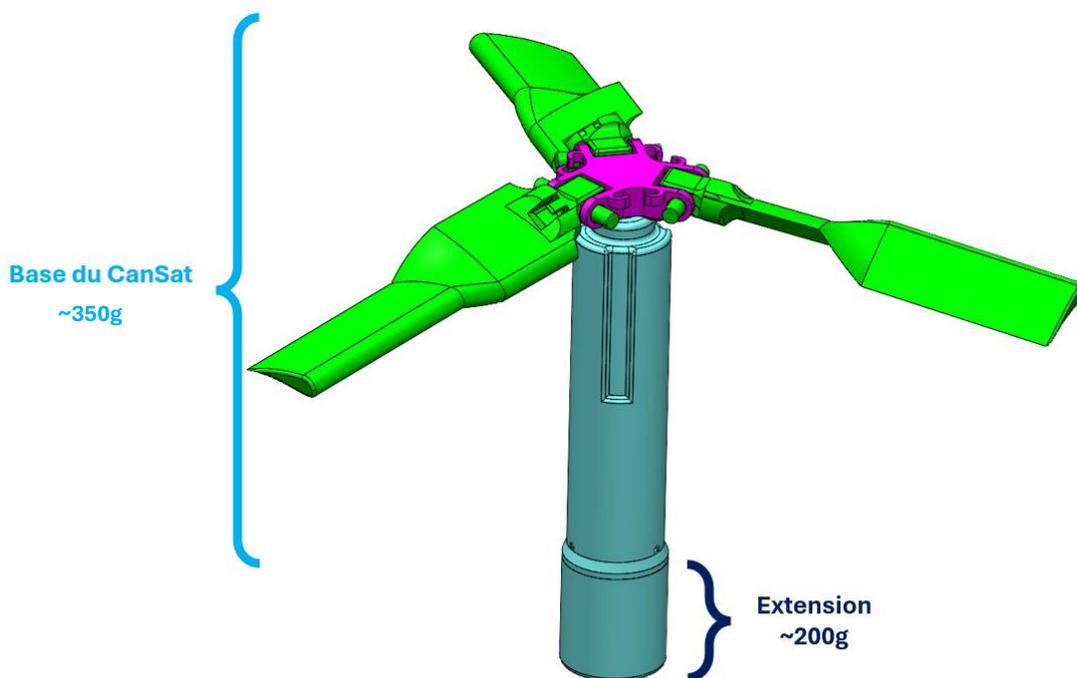


Figure 24 - Répartition de la masse

3.2.3.6 Mesure de la vitesse de rotation du CanSat

Afin d'approfondir et de compléter l'expérience du CanSat autorotatif, nous avons décidé de mesurer la vitesse de rotation des pales ainsi que du CanSat lui-même. Ces mesures nous permettront de juger et mesurer l'impact de la rotation des pâles sur l'inertie du CanSat. Pour mesurer la vitesse de rotation du CanSat, nous utilisons une centrale inertielle MPU 9250 à 9 degrés de liberté, embarquant un accéléromètre 3 axes, un magnétomètre 3 axes, mais surtout un gyromètre 3 axes, contrôlé grâce à un microcontrôleur ESP32.

3.2.3.7 Mesure de la vitesse de rotation du rotor

Comme dit précédemment, nous cherchons aussi à mesurer la vitesse de rotation du rotor. Pour ce faire, nous embarquons un capteur à effet Hall sur le CanSat, ainsi que des aimants installés sur le rotor. Le nombre d'aimants reste à déterminer mais l'objectif est d'en intégrer le maximum afin d'affiner les mesures. Le capteur à effet Hall fonctionne de façon à ce que la tension aux bornes soit nulle lorsqu'un champ magnétique est détecté, et non nulle dans le cas inverse. Ainsi nous pouvons mesurer à l'aide de l'ESP32 le nombre de fois qu'un champ magnétique est détecté durant un laps de temps défini, afin d'en extraire la vitesse de rotation du rotor.



Figure 25 - Capteur à effet Hall

Le programme intégré à l'ESP32 est écrit en micropython. Lors de l'écriture du programme, nous avons utilisé la fonctionnalité des interruptions. L'interruption permet de suspendre temporairement l'exécution du programme principal pour traiter un événement spécifique, tel qu'une entrée utilisateur ou une condition particulière. Les interruptions offrent une manière efficace de gérer des tâches asynchrones, améliorant ainsi la réactivité du système en permettant le traitement immédiat d'événements sans être gêné par le temps d'exécution de la boucle principale du programme.

Dans notre cas, cette fonctionnalité nous est particulièrement utile puisque la détection du passage d'aimant est cruciale si l'on veut déterminer de manière précise la vitesse du rotor. On ne peut donc pas se permettre de rater un passage d'aimant.

3.2.3.8 Test au C'Space

La version finale de Porthos utilisé au C'Space a été imprimé en 2 couleurs majoritaires : noir pour la structure et le rotor, orange pour le support élec. Cependant les pâles ont été imprimé en 3 couleurs différentes (orange, jaune et blanc) dans le but de simuler l'image d'une "cible" avec la vitesse de rotation. Dans le but de conserver des pâles résistantes et légère, nous avons préféré imprimer les pâles en une seule pièce en utilisant une imprimante Bambulab P1P équipé d'un AMS (automatical material system) permettant d'imprimer un seul élément en plusieurs couleurs.

Réaliser ces pâles multicolores en plusieurs pièces aurait pu les rendre plus fragile, plus lourde ou même moins efficace.



Figure 26 - Porthos Version finale



Figure 27 - Largage de Porthos au C'Space

Le vol de la fusée n'ayant pas eu lieu, Porthos a pu être testé grâce au drone de largage du concours CanSat étudiant. Il a été installé dans un cylindre accroché sous le drone, qui est ensuite monté à 150m de hauteur avant de s'ouvrir pour laisser tomber Porthos.

Les pâles se sont immédiatement déployées et ont commencé à tourner assez vite pour ralentir Porthos, au point qu'il s'est déséquilibré. Il a commencé à partir en vrille, avant de se stabiliser de nouveau et de ralentir juste assez avant de toucher le sol.

La chute a duré environ 12s pour une vitesse moyenne de 12.5 m/s, cependant le capteur à effet hall a enregistré un maximum de 209 rpm. Nous supposons qu'il a rencontré un problème et qu'il n'a pas réussi à enregistrer les infos au-dessus de cette valeur. En nous basant sur les tests en soufflerie, nous pouvons déterminer que les pâles ont atteint une vitesse de rotation d'environ 650 rpm.

Après avoir récupéré Porthos, nous avons repéré que les seuls éléments qui se sont cassés étaient les jointures des pâles avec l'axe qui les relie au rotor. Cette faiblesse avait déjà été repéré mais jugé comme acceptable, la cassure n'arrivant qu'à l'atterrissage.

En conclusion, malgré le fait que Porthos n'est pas été éjecté depuis la fusée, le vol s'est déroulé comme prévu et l'autorotation a parfaitement fonctionné. Il reste cependant des points d'améliorations, notamment sur l'équilibrage, le capteur à effet hall, et la résistance des pâles.

3.2.4 Circuit générique

3.2.4.1 GPS

Notre projet est composé de la fusée et des 3 CanSats qui porteront nos expériences. Pour connaître leur localisation pendant le vol ou après l'atterrissage, chacune de ces parties possédera un module GNSS qui nous servira à avoir leur emplacement. Ces modules récupèrent diverses informations sous forme de trames NMEA. Il existe en tout 6 trames différentes que nous recevons grâce au GPS mais nous ne gardons de notre côté que les trames GGA et RMC qui nous suffisent à obtenir toutes les données que nous voulons.

Exemple des trames GGA et RMC :

```
$GPGGA,123519.30,4807.038,N,01131.324,E,1,08,0.9,545.4,M,46.9,M, , *42
```

Cette trame nous permet de connaître:

- L'heure UTC, ici 12h35min19s et 3 dixième de seconde
- La position avec :
 - la latitude : 48° et 7,038 minutes d'angle au nord de l'équateur
 - la longitude: 11° et 31,324 minutes d'angle à l'est du méridien de greenwich
- L'altitude, ici 545.4m au-dessus du niveau moyen des océans.

```
$GPRMC,225446.40,A,4916.45,N,12311.12,W,0.0,5.054.7,191194,020.3,E*68
```

Cette trame nous permet de connaître:

- l'heure UTC et la position de la même manière que sur la trame GGA
- La position
- la vitesse, ici 0.5 noeud
- le cap, ici 54.7° par rapport au Nord

Même si les 2 trames sont redondantes sur la position et l'heure, elles fournissent également des données uniques qui nous intéressent. Sachant que nous avons besoin à la fois de l'heure, la position, l'altitude (donné uniquement par la trame GGA), la vitesse et son cap (donné uniquement par la trame RMC), nous devons garder les 2 trames.

Au début d'année, nous avons décidé d'utiliser les modules NEO-6M de chez u-blox. Ces modules fonctionnaient très bien et nous étions satisfaits mais pour une autre raison que nous verrons plus tard, nous avons du changer de composant. Nous avons finalement utilisé l'ESP32-S3FN8 wireless tracker qui utilise comme puce GNSS une UC6580 de chez Unicore Communications.

i) NEO-6M

Les GPS NEO-6M sont configurables de beaucoup de manières différentes. Dans notre cas, nous n'avons modifié que 3 paramètres par rapport à la configuration de base: la fréquence de réception des trames NMEA que nous avons fixée à 10 Hz c'est à dire une nouvelle trame reçue toutes les 0.1 secondes, le baudrate mis à 115200 bps, qui est plus important que le débit de données par défaut pour recevoir plus rapidement les trames entre le GPS et le microcontrôleur. Par défaut les GPS sont configurés en mode piéton pour avoir une précision maximum. Ce mode a une limite de vitesse et d'accélération très basse, à cause de ça, beaucoup de fusées au C'Space perdent la position GPS pendant la montée. Nous avons donc réglé le GPS dans le mode qui permet le plus de vitesse et d'accélération : Airborne <4g.

Comme dit juste au-dessus, nos GPS sont normalement capables de fonctionner jusqu'à une accélération de 4g. Cependant nous savons que lors du décollage de Triplex, la fusée atteindra une accélération de 5.5g puis ne descendra sous la barre des 4g qu'au bout d'environ 1.5s. Comme nous ne savions pas comment réagissent les GPS une fois la limite dépassée, nous avons décidé de faire un test pour vérifier si nous ne perdions que momentanément le signal ou s'il s'arrêterait de fonctionner sans redémarrer. Nous voulions savoir comment fonctionne le GPS dans une telle situation pour pouvoir anticiper au mieux le vrai vol en juillet.

Ainsi est né Miniplex, une mini-fusée dont le but était d'embarquer un GPS et de dépasser la limite de 4g d'accélération.

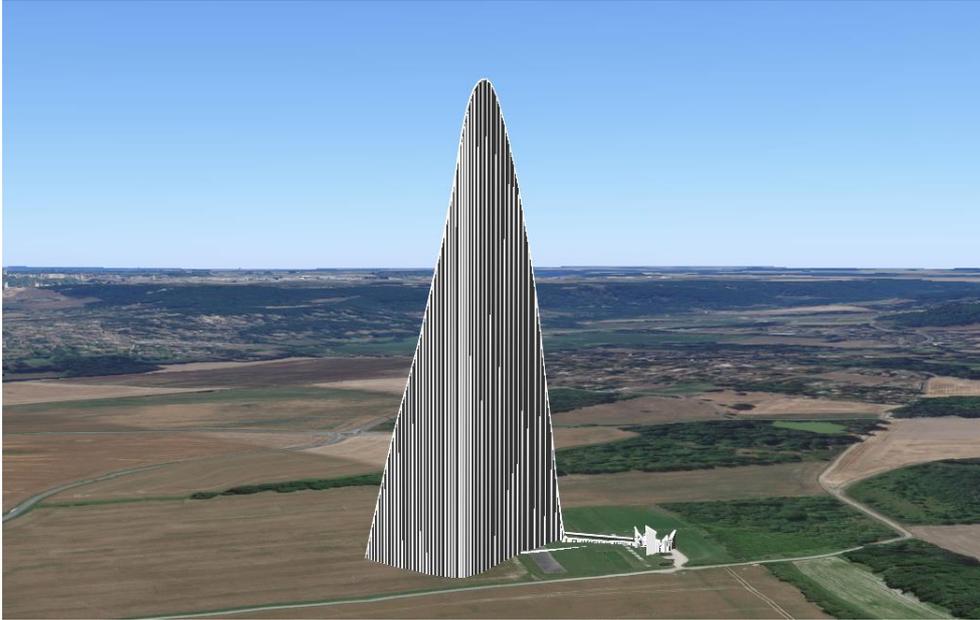


Figure 4 - Trajectoire GPS

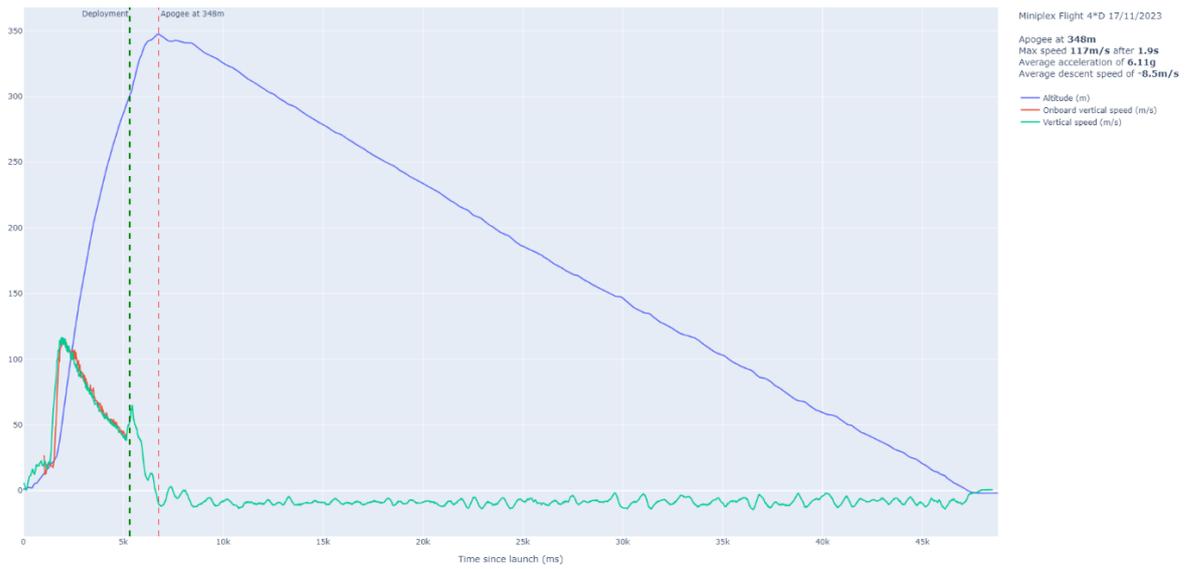


Figure 5 - Altitude et vitesse ascensionnelle en fonction du temps



Figure 6 - Accélération subie en g par Miniplex en fonction du temps (1s / division)

Il est important de noter que nous avons forcé le déploiement du parachute à 300m d'altitude pour limiter la distance de retombée, on observe que la fusée a continué sur sa lancée sur 50m d'altitude mais l'apogée s'est limitée à 350m au lieu des 500m prévus sans déclenchement forcé.

On remarque que Miniplex a bien atteint une accélération de plus de 4g et pendant près de 2 secondes et que malgré cela, le GPS n'a jamais arrêté de fonctionner et aucune perte de trame NMEA n'a été détectée.

ii) UC6580

L'objectif de la configuration de notre UC6580 était de garder les mêmes paramètres que nous avons sur le précédent module car ces derniers nous satisfaisait. Nous voulions une nouvelle fois mettre la fréquence de réception des trames NMEA à 10 Hz pour recevoir une nouvelle trame toutes les 0.1s, malheureusement, cela n'était pas possible avec ce GNSS et nous avons dû diviser par deux cette fréquence en configurant à 5 Hz, nous n'avons pas eu besoin d'agir sur le baudrate entre le MCU et le GNSS car ce dernier est directement à 115200 bps ce que nous voulons pour que les trames prennent peu de temps à être communiquées au MCU. Nous avons également configuré le GNSS en mode « portable » plutôt qu'en « static » car plus résistant aux fortes accélérations.

Il est également intéressant de préciser que nous avons décidé de rajouter et d'utiliser une antenne GNSS externe plutôt que celle interne au module dans le seul but d'obtenir une meilleure et plus rapide connexion avec les satellites.



Figure 3 - connexion satellite et trame reçu avec antenne interne

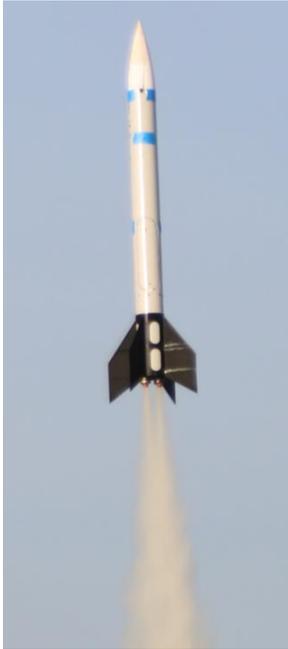


Figure 4 - connexion satellite et trame reçu avec antenne externe

La première figure nous montre ce que capte notre GNSS simplement avec notre antenne interne alors que sur le deuxième graphique, une antenne externe a été rajoutée, il s'agit de l'antenne externe du GNSS NEO-6M que nous utilisons au début d'année.

On peut voir, qu'avec l'utilisation d'une antenne GNSS externe, nous avons réussi à capter la trame L5 permettant une plus grande précision et robustesse (pour Galileo, le L5 est appelé E5b), il s'agit de la colonne en bleu foncé que nous voyons au niveau du satellite numéro 7.

On ne peut pas le remarquer sur les figures mais il est aussi important de noter que grâce à l'antenne externe, nous avons été capable de capter jusqu'à 37 satellites de différentes constellations et de récupérer un fix (une position) en 2/3 minutes seulement alors que nous n'avons que 13 satellites avec l'antenne interne et nous obtenons un fix en bien plus de temps.



Cette fois encore, nous avons décidé de réaliser des tests à hautes accélérations, toujours avec l'objectif de vérifier si notre module sera capable de résister à l'accélération de notre fusée le jour du vol. En effet, nous n'avons pas trouvé d'indication claire sur l'accélération maximum qu'est capable d'accepter l'UC6580, nous avons donc, ce semestre ci, réaliser de nouveaux lancements tests pour essayer divers éléments du projet et notamment le comportement de notre module face aux fortes accélérations.

Nous avons donc lancer une nouvelle mini-fusée, PTBL80, qui a atteint une accélération de plus de 5g. Bien que nous soyons tout de même assez loin des 9g prévu pour dans quelques mois. Les résultats du test ont été concluants car même si le GNSS n'a pas réussi à supporter une telle valeur instantanée, il a réussi à se remettre en bon fonctionnement très rapidement dès la diminution de l'accélération.

3.2.4.2 LoRa

Dans le cadre de notre projet, la communication entre les CanSats et la fusée est essentielle que ce soit pour transmettre les données que nous voulons récupérer au CanSat relai ou pour partager la position du CanSat autorotatif au CanSat parafoil. Pour répondre à cette demande nous avons décidé d'utiliser la technologie LoRa (Long Range). La raison pour laquelle nous avons choisi cette technologie est qu'elle permet d'émettre jusqu'à une distance de 10km tout en gardant une puissance d'émission légale, ce qui est amplement suffisant dans notre cas.

Nous avons d'abord décidé d'utiliser le module E22900T-30D, connecté à un micro-contrôleur, une RP2040 Zero, puis nous sommes passés sur l'ESP32 comme dit précédemment, ce dernier module possède une puce SX1262, la même qui est sensée être présente chez l'E22900T-30D.

i) E22900T-30D

L'un des avantages du module E22-900T30D est sa modularité, nous pouvons modifier la fréquence de transmission utilisée, l'adresse de l'émetteur et du récepteur, le débit de données, la taille des paquets envoyés ou encore la puissance de transmission. Dans notre cas, nous avons décidé d'utiliser la fréquence à 874.125 MHz qui autorise une puissance d'émission de 500 mW ce qui est suffisant pour notre application. La localisation de nos différents systèmes étant critique, afin de pallier un éventuel problème de relai 4G, nous allons également récupérer directement au sol les données émises par LoRa. Or nous serons bien plus loin des CanSats, nous avons donc besoin d'une portée bien plus grande que

si les CanSats devaient uniquement communiquer entre eux. C'est pour cette raison que nous avons choisi cette fréquence qui nous permet d'émettre à 500 mW. Le débit de données a été choisi à 2.4 kbps. La modulation LoRa permet d'améliorer la portée en diminuant le débit de données, nous avons donc choisi ce débit relativement faible pour maximiser la portée tout en étant suffisant pour notre application.

Pour trouver la configuration qui nous convenait le mieux, nous sommes allés faire des tests pour en essayer plusieurs. Nous sommes allés dans des champs pour garder une ligne de vue directe malgré le fait que nous soyons au sol. Nous avons effectué des essais à 1.2 puis 2.3, 3.2 et pour finir 4.5 km. Ces tests nous ont permis de trouver la configuration présentée plus tôt car même à 4.5 km au sol, il n'y avait que très peu de pertes de paquet, moins de 10 %.

Portée (km)	Bande passante (bps)	Puissance d'émission (dBm)	Pertes de paquets (%)
1.2	2 400	21	10%
1.2	2 400	24	0%
1.2	192 000	21	10%
1.2	384 000	21	40%
2.3	2 400	27	0%
2.3	300	24	0%
3.2	2 400	27	0%
3.2	2 400	24	0%
3.2	2 400	21	20%
3.2	2 400	21	20%
4.5	2 400	27	10%
4.5	1 200	24	80%
4.5	300	24	0%

Figure 7 - Résultats du test de portée

Les pertes de paquets représentent le taux d'informations perdu entre l'émission et la réception des données.

On ne peut pas espérer que les ondes transportant nos données aillent tout droit vers le récepteur sans se disperser. Ce phénomène est nommé les zones de Fresnel. Elle nous permet de déterminer une zone ellipsoïdale à l'intérieur de laquelle on ne doit rien trouver qui puisse perturber notre émission de signal.

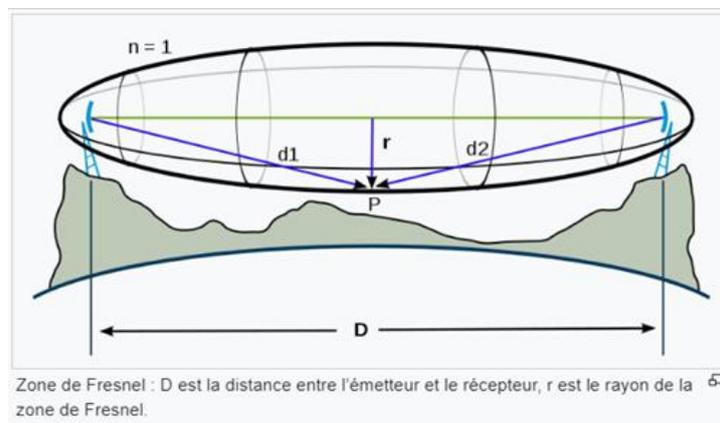


Figure 8 - Zone de Fresnel

Rayon de la zone de Fresnel pour les différentes portées que nous avons essayé :

$$r = \frac{1}{2} \sqrt{\lambda \cdot D}$$

avec $\lambda = c/f$ avec c , célérité de l'onde et f , fréquence utilisée

- 1.2km, rayon de 10.14 m
- 2.3km, rayon de 14.04 m
- 3.2km, rayon de 16.57 m
- 4.5km, rayon de 19.64 m

C'est-à-dire que sur un terrain parfaitement plat, à 4.5 km de distance, pour avoir une réception similaire à celle qu'on aurait avec la fusée en vol, il faut qu'une des antenne soit à 40m de hauteur, ou alors que les 2 antennes soient à 20m de hauteur (comme sur le schéma).

Nous avons choisi ce module LoRa pour deux raisons majeures, le microcontrôleur intégré qui simplifie l'émission de données et le paramétrage, et la possibilité d'émettre à 500mW grâce à un amplificateur. Le module comporte donc : MCU (Microcontrôleur) -> SX1262 (LoRa) -> Amplificateur.

Cependant, malgré des tests réussis avec les E22900T-30D présentées plus tôt nous avons été obligés de changer le composant de communication. En effet, peu après notre soutenance de mi-semestre, nous avons eu l'occasion d'utiliser un SDR (Software Defined Radio) pour observer le spectre émis par l'E22900T-30D. Grâce à cet outil nous avons notamment pu regarder la largeur de la bande d'émission de donnée et nous avons eu la mauvaise surprise de nous retrouver avec une largeur de 500 kHz, bien au-delà des 200 kHz autorisés. Cette donnée n'était présente dans aucune des datasheets disponibles ce qui nous a donné l'idée de vérifier par nous-même.

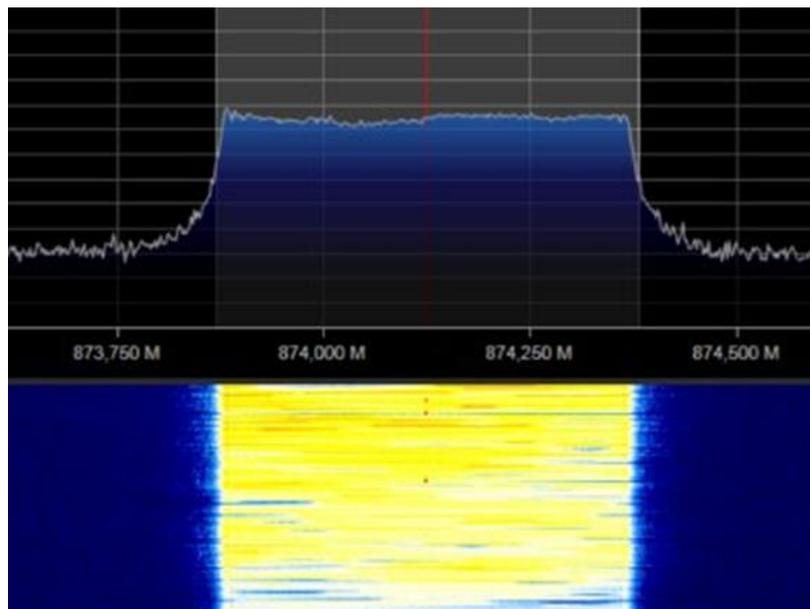


Figure 2- spectre émis par la E22 compris entre 873.875 et 874.375 MHz

De plus, avec ce module, nous avons des problèmes d'interférences électromagnétique avec notre microcontrôleur qui faisait planter la RP2040 dû à la puissance d'émission très importante. Nous avons trouvé une alternative à cela en utilisant un microcontrôleur blindé mais cela était moins pratique.

À la suite de la nouvelle et en pensant aux différents problèmes que nous avons eu avec ce module LoRa, nous n'avons pas eu d'autres choix que de trouver un remplaçant à l'E22. De plus, après avoir abandonné les E22, nous en avons démonté certaines qui se sont avéré être des contrefaçons, le chip LoRa utilisé n'étant pas le SX1262 comme prévu mais le LLCC68 qui est moins sensible.

Le remplaçant que nous avons trouvé est l'ESP32-S3FN8 wireless tracker, un composant qui s'est finalement retrouvé bien pratique pour une bonne raison. Ce module est en fait un microcontrôleur mais qui, en plus de cela, possède une partie émetteur/récepteur LoRa ainsi qu'un module GNSS intégré. L'absence d'amplificateur limite la puissance d'émission à 140 mW mais nous avons fait des tests qui ont montré que c'est suffisant.

ii) ESP32-S3FN8 wireless tracker

Sur le chip SX1262, nous pouvons modifier, la fréquence utilisée, la largeur de la bande d'émission, le coding rate ainsi que le spreading factor.

La fréquence que nous avons décidée d'utiliser est la 869.5 MHz, une fréquence auquel nous n'avons pas accès avec le précédent module. Nous avons choisi cette fréquence car elle nous permet légalement d'utiliser le maximum de la puissance d'émission.

En effet, la localisation de nos différents systèmes étant critique, afin de pallier un éventuel problème de relai 4G, nous allons également récupérer directement au sol les données émises par LoRa. Or nous seront bien plus loin des CanSats, nous avons donc besoin d'une portée bien plus grande que si les CanSats devaient uniquement communiquer entre eux. La fréquence 869.5 MHz nous permettant d'émettre jusqu'à une puissance de 500 mW, nous avons choisi cette fréquence pour notre mission.

Pour la largeur de la bande d'émission, nous ne pouvons pas choisir n'importe quelle valeur, nous avons le choix entre 250 kHz, 125 kHz ou encore moins. La limite légale étant à 200kHz, nous avons décidé de garder la plus haute valeur possible car synonyme de plus de données transmises en un instant. C'est-à-dire que nous avons pris une largeur de bande d'émission de 125 kHz.

Enfin, il nous était possible de choisir un spreading factor entre 9 et 12 et un coding rate entre 6 et 8.

Le spreading factor représente le degré d'étalement des signaux transmis sur un plus grand temps que nécessaire pour la transmission de l'information. Augmenter le spreading factor, c'est augmenter nos chances de réceptionner nos données mais c'est également diminuer la quantité d'informations envoyée. Grâce au spreading factor, la modulation LoRa peut capter des informations même avec un rapport signal/bruit inférieur à 1.

Le coding rate, lui, représente le rapport entre le nombre de bits total transmis et le nombre de bits d'information envoyé. Un coding rate élevé signifie un message plus robuste car avec des données redondantes, cependant, cela veut aussi dire qu'on y trouve moins de donnée utile.

Il est donc important de faire attention aux différents cas possibles.

Ainsi, pour trouver la configuration qui nous convenait le mieux, nous sommes allés faire des tests pour en essayer plusieurs. Nous sommes allés dans des champs pour garder une ligne de vue directe malgré le fait que nous soyons au sol. Nous avons effectué des essais à différentes distances tout en faisant varier le coding rate mais surtout le spreading factor.

Nous avons testé à 2, puis à 3 et enfin à 4 km. Nous nous sommes rapidement rendu compte qu'un spreading factor de 9 entraînait trop de perte et qu'un coding rate de 6 n'était pas suffisant. Nous avons donc assez rapidement fixé un coding rate à 8 puis nous avons vu que, que le spreading factor soit à 10, à 11 ou à 12, les pertes étaient acceptables.

Finalement, nous avons choisi de garder un spreading factor de 10 avec un coding rate de 8 car le pourcentage de perte de paquet était faible (seulement 10% environ) même à 4 km et qu'il s'agissait de la configuration de paramètre qui nous permettrait d'envoyer le plus de données possibles.

iii) Multiplexage temporel

Comme dit précédemment, l'objectif de la communication par LoRa et que les CanSats et la fusée puissent transmettre leurs données au CanSat relai, sachant que le CanSat relai ne possède qu'un seul récepteur il faut faire en sorte que les émetteurs envoient leurs trames à des moments différents pour ne pas se retrouver avec des pertes de paquets.

Pour faire cela, nous utilisons les GNSS qui se trouvent dans chaque CanSats/fusée et plus particulièrement les trames NMEA GGA ou RMC reçues par ces derniers. Elles nous permettent d'obtenir l'heure UTC qui sera la même pour tous les GNSS. Cette heure est précise à deux dixièmes de secondes près. Grâce à cette heure, nous pouvons diviser le temps d'émission en 3 parties.

Malgré ce temps très précis que nous pouvons obtenir, nous avons simplement décidé de diviser notre temps par rapport aux secondes. Pour entrer un peu plus en détail sur le fonctionnement de notre TDM (Time Division Multiplexing), le récepteur que ce soit celui au sol pour la redondance ou juste celui dans le CanSat relai, devra acquérir les données venant de 3 émetteurs différents. Ainsi, chaque émetteur aura une ou deux secondes, selon ce que l'on choisit, pour envoyer toutes ses données par LoRa avant que ce soit le tour du prochain émetteur.

Le temps accordé à chaque émetteur pour envoyer dépendra de la quantité d'informations à transmettre, 1 seconde pouvant de pas être assez.

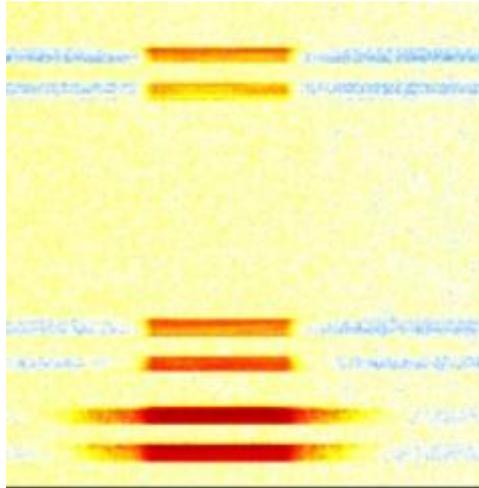


Figure - Apparition d'un deuxième émetteur

Sur la partie « haute » de l'image, on peut voir l'envoi de deux messages distincts. Malgré ces deux messages, il s'agit bien de la transmission de donnée d'un seul émetteur.

Sur la partie basse de l'image, on peut voir deux nouveaux messages qui sont apparus à la suite des premiers, il s'agit de l'ajout d'un émetteur qui transmet à son tour ses données.

Lors du vol en juillet, nous n'aurons pas 2 mais bien 3 émetteurs mais le principe restera le même.

3.2.4.3 Station sol (Hugo T.)

