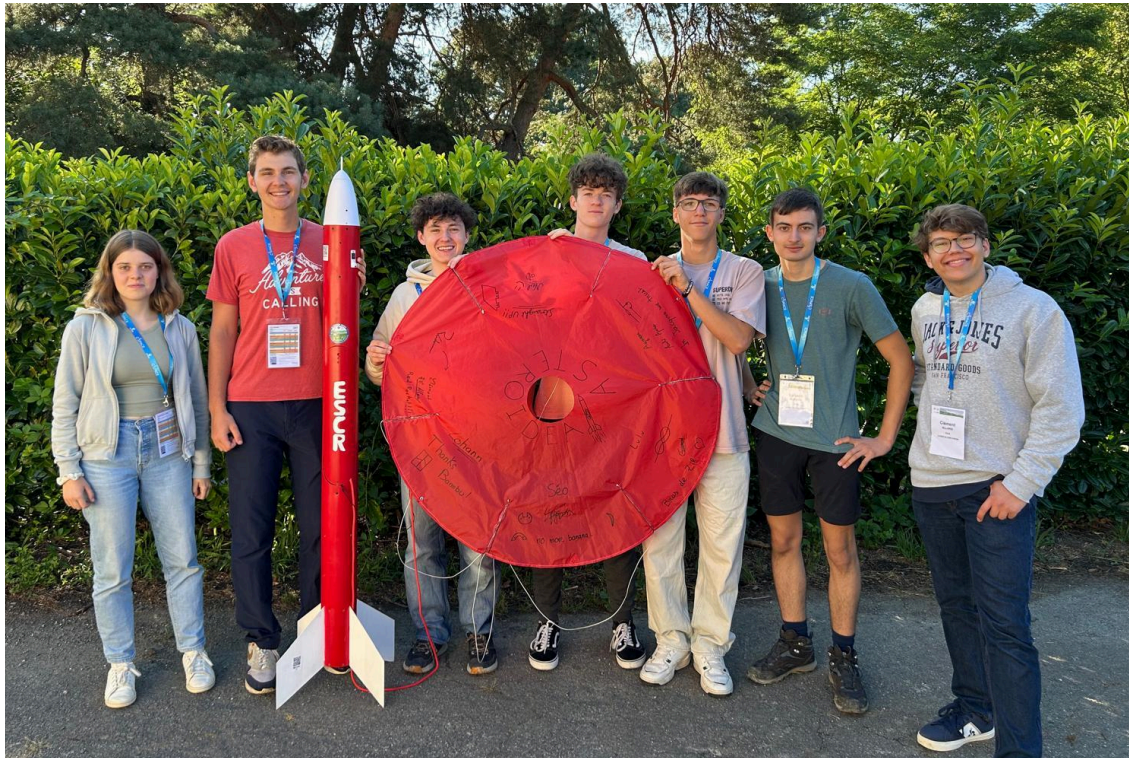




Asteroidea – Rapport de Projet

ESCR

Année 2024-2025



Sommaire

I. Le club.....	2
1. Histoire du club.....	2
2. L'équipe.....	2
 II. Objectifs	3
1. Le projet.....	3
2. Les expériences.....	3
 III. Conception mécanique.....	4
1. Caractéristiques générales.....	4
2. Le moteur.....	6
3. Partie supérieure de la fusée.....	7
 IV. Conception électronique.....	10
1. Expériences.....	10
2. Récupération des données.....	12
 V. Semaine du C'space.....	13
1. Problèmes et solutions.....	13
2. Déroulé du vol.....	14
 VI. Résultats des expériences et des instruments de bord.....	15
1. Détermination de l'altitude en fonction du temps.....	15
2. Tube de Pitot.....	17
3. Capteur de gaz nocifs.....	20
4. Caméra.....	22
 VII. Remerciements.....	22

I. Le club

1. Histoire du club

Le club **ESCR Astromodelisme** est un club créé en **2021**, avec 5 tirs au total depuis sa création répartis sur 4 campagnes de lancement, trois Rocketry Challenge (2023, 2024 et 2025) et une Minifusée au C'Space (2024). Nous sommes une équipe de 12 lycéens soudés, passionnés d'ingénierie et d'aérospatiale, et avons pour la plupart d'entre nous l'ambition d'intégrer une école d'ingénieurs. Nous sommes répartis en 2 équipes distinctes, une équipe est chargée d'un projet Rocketry et une autre réalise le projet présenté dans ce document, une fusée expérimentale.



2. L'équipe

L'équipe est composée de **7 membres** :

Prénoms et NOMS	Rôles
Agnese ACERBI	Conception électronique
Séo LAPOUGE--WERNER	Conception électronique
Lucas TRESILLARD	Expérience
Neil LE MEUNIER	Calcul de la trajectoire et aérodynamisme
Robin VIDONNE	Fabrication de la chaîne de récupération
Clément ALLARD	Intégration et impression 3D
Lohann BUISSON	Intégration et impression 3D

II. Objectifs

1. Le projet

Dans la suite de 2 **Rocketry challenges** et une **Minifusée**, cette fusée expérimentale vise à mettre à profit les connaissances que nous avons acquises au cours de ces 3 campagnes de lancement et nous ouvrir des opportunités de rencontres dans le monde de l'ingénierie, ainsi que développer de nouvelles compétences durant la réalisation de ce projet.

2. Les expériences

Notre expérience principale est la mesure de la vitesse de la fusée à l'aide d'un tube pitot. Celui-ci nous permet de calculer la différence entre la pression totale et la pression statique, ce qui nous permet d'en déduire la vitesse.

Notre seconde expérience est la mesure de la qualité de l'air en fonction de l'altitude. Pour ce faire, notre fusée embarquera un capteur de gaz nocifs, ainsi qu'un capteur de pression et de température afin de déterminer l'altitude.

Enfin, la fusée embarquera également une caméra afin d'obtenir des images du vol.

Toutes les données seront stockées à bord d'une carte SD, nous n'utilisons pas de télémétrie.

III. Conception mécanique

1. Caractéristiques générales

La fusée :

- Taille : 1,750 m
- Diamètre : 125 mm
- Masse : 9.5 kg au décollage
- Matériau : Tube PVC CR8
- Moteur : Pro 54

Les ailerons :

- Matériau : Aluminium
- Epaisseur : 2 mm
- Dimensions : Voir Schéma ci-contre

La coiffe :

- Dimension (longueur) : 210 mm
- Matériau : ABS

Système de récupération :

- Parachute :
 - Dimensions : Voir Stabtraj (Fig 1.4)
 - Matériau : Toile de parapente
- Système d'ouverture :
 - Ejection par une trappe latérale à l'aide d'une minuterie
 - Code : Voir image (3)

Charge utile (Expérience) :

- Un baromètre et thermomètre
- Un capteur de gaz nocifs
- Caméra



Fig 1.1 - Schéma global 3D

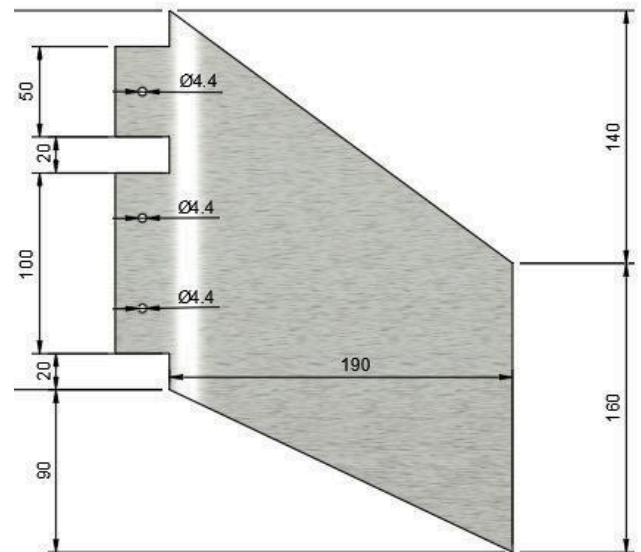
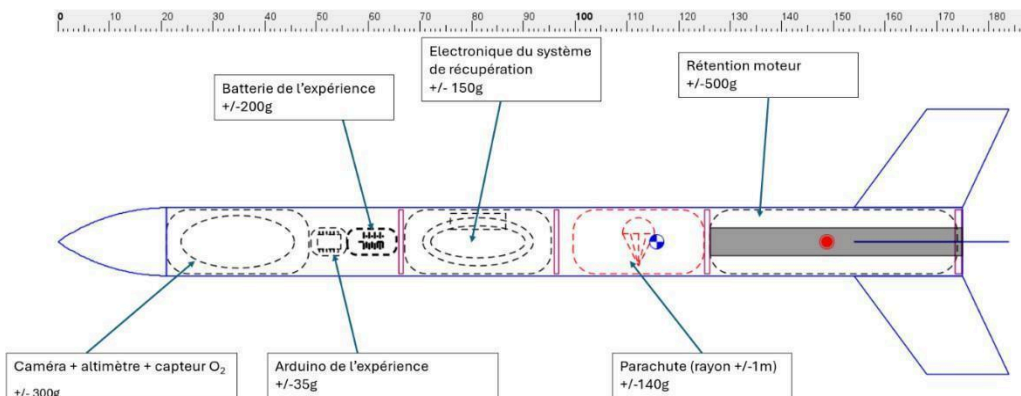


Fig 1.2 - Dimensions ailerons



Masse totale 9,432kg
Fig 1.3 - Schéma global



STABILITO

Stabilité de fusée à ailerons

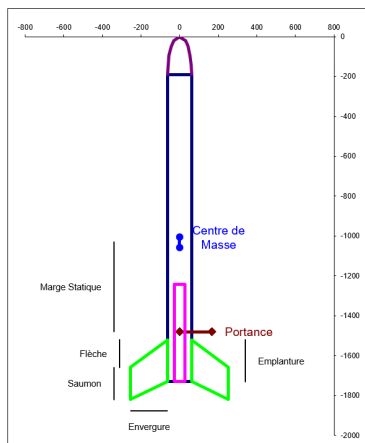
Remplir les cases jaunes

Fusée	
Nom	Asteroïdea
Club	ESCR
Matricule	FX 29
Type	Fusée expérimentale.
Masse	7800 g sans propu
Centre de Masse	965 mm sans propu
Longueur totale	1730 mm

Propulseur	
Type	Pro54-5G WT
Position du bas	1730 mm

Coiffe	
Forme	Parabolique (arrondie)
Hauteur	190 mm
Diamètre	125 mm

Ailerons	
Mono-empennage	
Emplanture 'm'	210 mm
Saumon 'n'	160 mm
Flèche 'p'	139 mm
Envergure 'E'	191 mm
Epaisseur 'ep'	2 mm
Nombre	4
Position du bas	1730 mm



07/07/2025	Min	Résultats	Max
Finesse	10	13.8	35
Portance	15	20.2	40
MargeStat.	2 D	3.39 D	3.80 D
Couple	40	68.4	76.7
XCp		1480 mm	1480 mm
MS / L		25% L	27% L

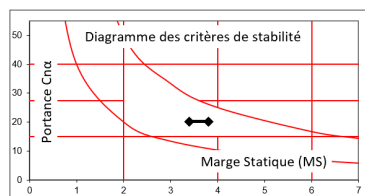
STABLE

Language/Langue  Français

Fusée mono-diamètre,

	Propu plein	Propu vide	Sans propu
Masse propu	1.632 kg	0.65 kg	-
CdM propu	250 mm	240 mm	-
Masse fusée	9.432 kg	8.45 kg	7.8 kg
CdM fusée	1056 mm	1005 mm	965 mm

	XCp	Cna
Coiffe	95 mm	2.0
Ailerons	1633 mm	18.2



Commentaire libre :

Checksum : propu OK

v3.4.5



TRAJECTO

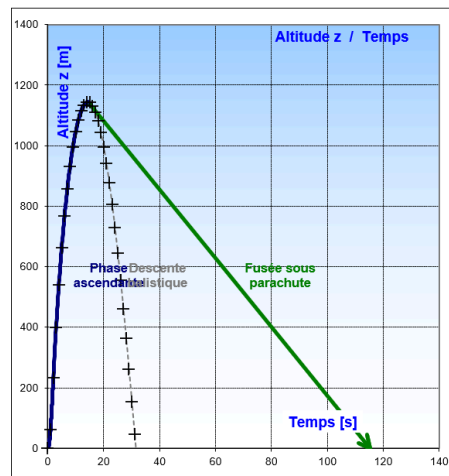
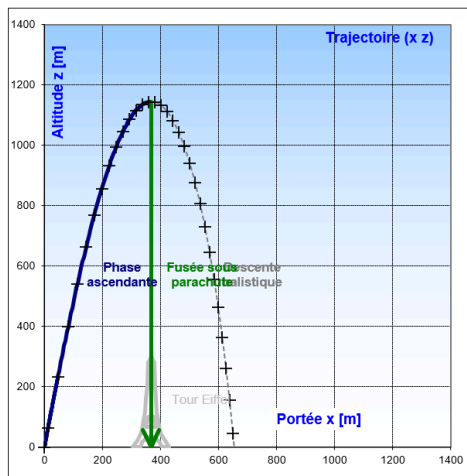
Trajectographie de fusée

Remplir les cases jaunes

Fusée	
Nom	Asteroïdea
Club	ESCR
Matricule	FX29
Masse totale	9.432 kg
Propulseur	Pro54-5G WT

Traînée Aérodynamique	
Surface Réf.	0.013800 m²
Cx	0.6

Rampe de Lancement	
Longueur	4 m
Élévation	80 °
Altitude	0 m



DescenteSousParachute	
Fusée	0 satellite
Masse	8.45 kg
Dépotage	N/A
Ouverture para	14.4 s
Type de para	Rond
Surface para	1.06 m²
Cx parachute	1
Vitesse du vent	5 m/s
Vitesse descente	11.3 m/s
Durée descente	101 s
Durée du vol	115 s
Déport latéral	± 505 m

07/07/2025	Temps	Altitude z	Portée x	Vitesse	Accélération	Efforts
Sortie de Rampe				31.9 m/s		
Vit max & Acc max				193 m/s	134 m/s²	
Culmination, Apogée	14.4 s	1144 m	367 m	21 m/s		
Ouverture parachute fusée	14.4 s	1144 m	367 m	21 m/s		292.1 N
Impact balistique	31.5 s	~0 m	654 m	112.0 m/s		52971 J

Pour localiser la fusée	
Couleur fuselage/coiffe	Brun/Orange...
Couleur parachute fusée	Rouge...

Commentaire libre :

propu OK
v3.4.5

Fig 1.4 - Stabtraj

La fusée a été réalisée avec un système de modules qui viennent s'insérer et se verrouiller directement dans le tube PVC (voir schéma ci-dessous).

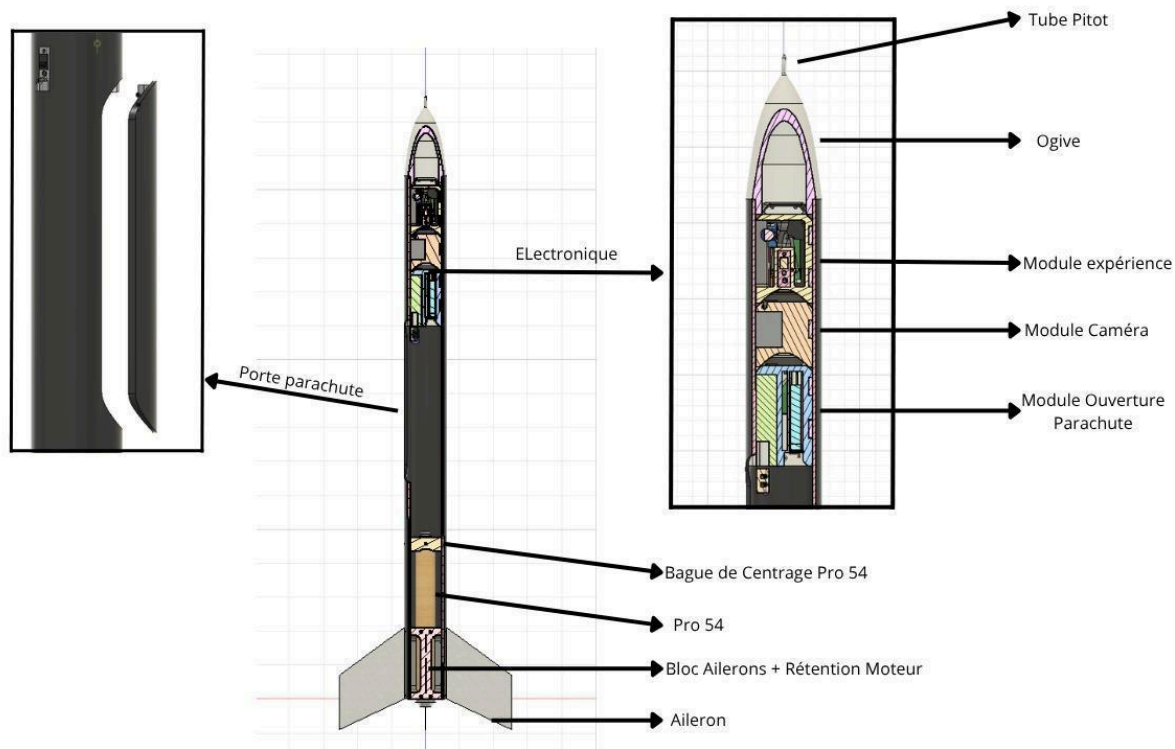


Fig 1.5 - Plan intégral de la fusée

2. Le moteur

Le moteur (Pro 54 WT) est maintenu par deux pièces : la rétention moteur en bas du tube et la bague de centrage en haut du moteur(voir schéma ci-après).

Le module ailerons-moteur, fait en ABS (matériaux résistant aux effets de la chaleur au-delà de 100°C), est renforcé avec une bague en aluminium de 2 mm vissée et protégé par une couche de liège pour éviter le contact direct avec le moteur. Il est maintenu par 16 vis M4 au tube et les 4 ailerons sont maintenus à ce module par 3 vis M4 chacun lui conférant ainsi une grande résistance au travail vertical. La bague de centrage, également en ABS, est maintenue par 4 vis M4 (voir schéma ci-dessous) et sa configuration conique côté moteur permet un centrage rapide de ce dernier.

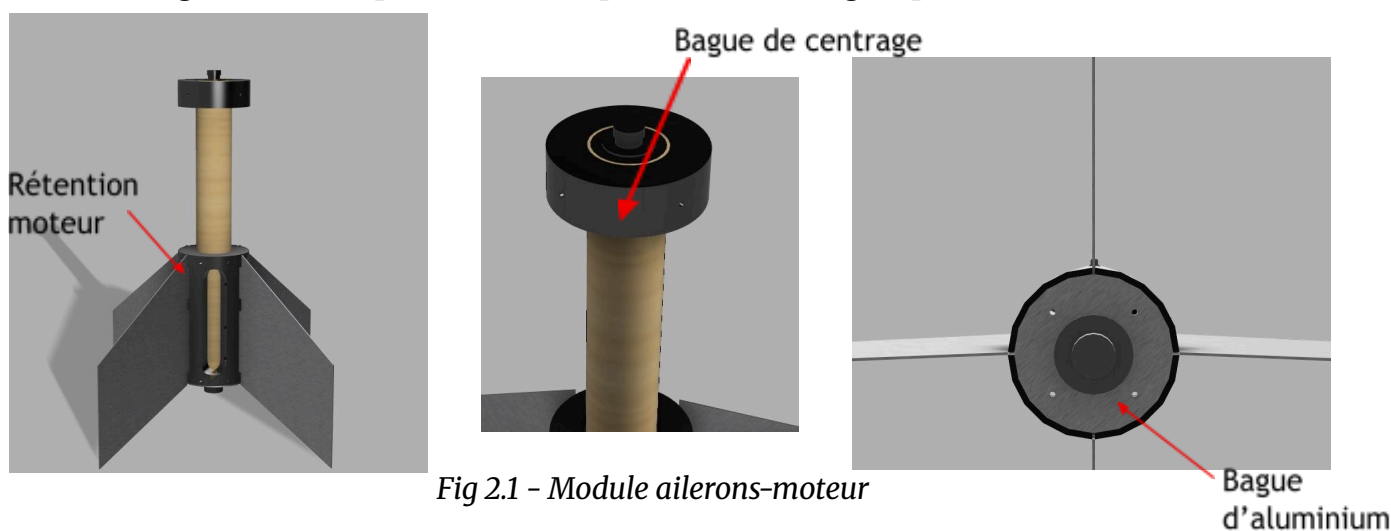


Fig 2.1 - Module ailerons-moteur

3. Partie supérieure de la fusée

Bague de rétention du parachute :

La partie cylindrique (1) du compartiment dans laquelle passe la sangle du parachute est renforcée par une bague en aluminium de 2 mm d'épaisseur, là où viendra le nœud d'arrêt de la sangle, et est fixée au tube par 8 vis M4. Le demi-cylindre du compartiment (2) est lui traversé par une plaque en aluminium de la même épaisseur et fixé au PVC par 6 vis M5

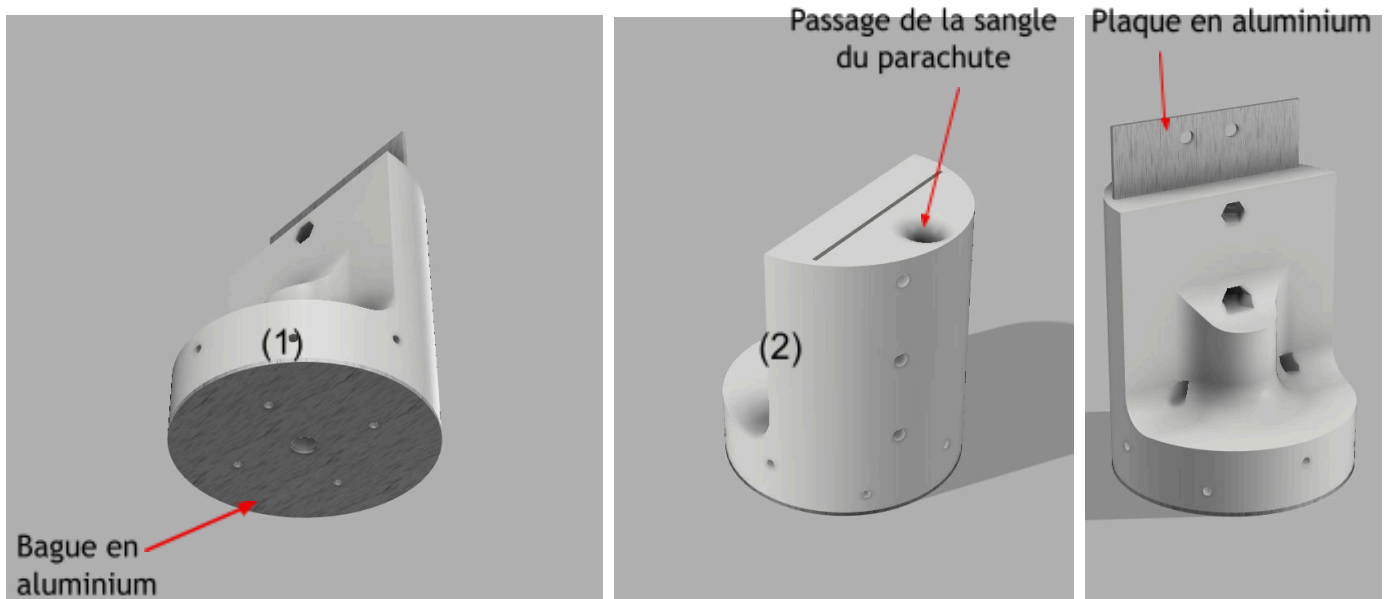


Fig 3.1 - Bague rétention parachute

A l'autre bout de la sangle parachute se trouve un émerillon qui relie l'ancrage aux 6 suspentes du parachute sur lesquelles une bague anti torche en ABS est placée.



Fig 3.2 - Bague anti-torque

Système d'ouverture de la trappe :

La trappe est retenue en bas par une languette de métal et en haut par un système de loquet mobile. Pour verrouiller la porte, le servomoteur à rotation continue entraîne un engrenage, permettant de descendre le loquet mobile et de l'emboîter avec la partie fixée sur la porte. Un bouton servant à détecter la position du loquet mobile permet de connaître l'état de la porte, ainsi à l'allumage de la carte, le servomoteur est au repos si la porte est déjà fermée, ou s'active pour fermer automatiquement la porte.

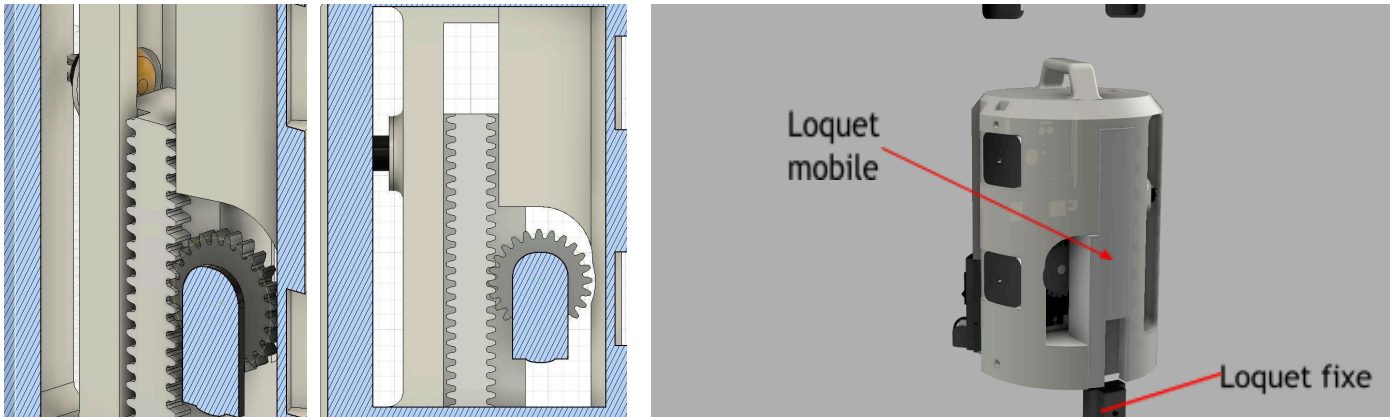


Fig 3.3 - Coupe 3D et plan du système de verrouillage de la porte

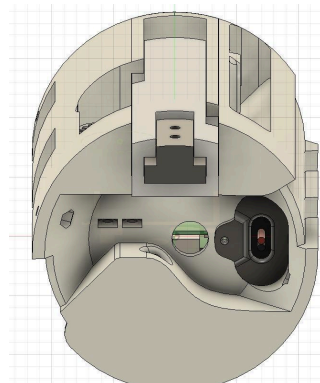


Fig 3.4 - Compartiment derrière trappe dédié au rangement de la batterie

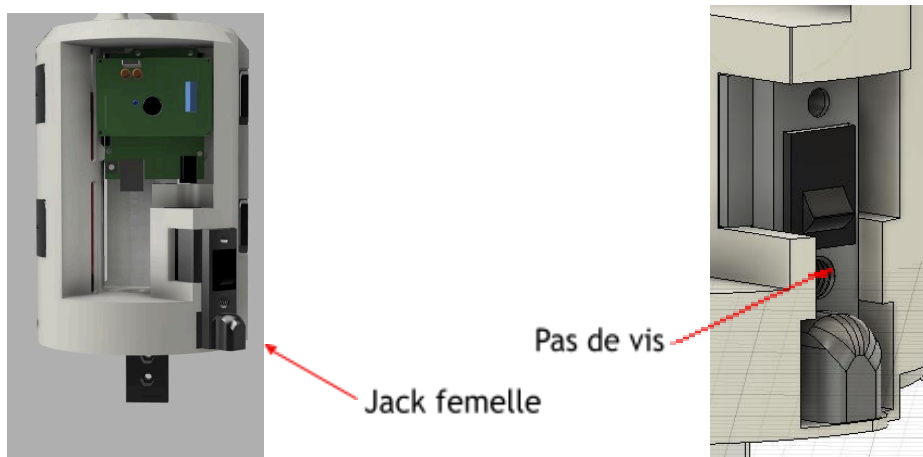


Fig 3.5 - Système externe mobile

La figure 3.5 montre l'autre face du module de la porte. En vissant une vis prévue à cet effet, l'interrupteur et le jack sortent du tube PVC. La présence de "pattes" vont permettre de garantir que le jack n'abime pas le module au décollage. Ce système nous permet d'avoir accès au bouton, LED et Jack, tout en ayant un module facilement amovible.

Système de verrouillage des modules :

Nous utilisons dans l'entièreté de la partie supérieure de la fusée un système de baïonnette qui consiste à glisser nos modules à l'emplacement dédié puis à les verrouiller à leurs emplacements en tournant la pièce sur elle-même, ainsi, les plaquettes de verrouillage sont placées dans leur cavité associée. De plus, pour éviter une rotation des modules nous mettons 4 vis M4 par module. Pour plus de clarté une animation est disponible avec le lien suivant : <https://youtu.be/rGr013mxcTE> .

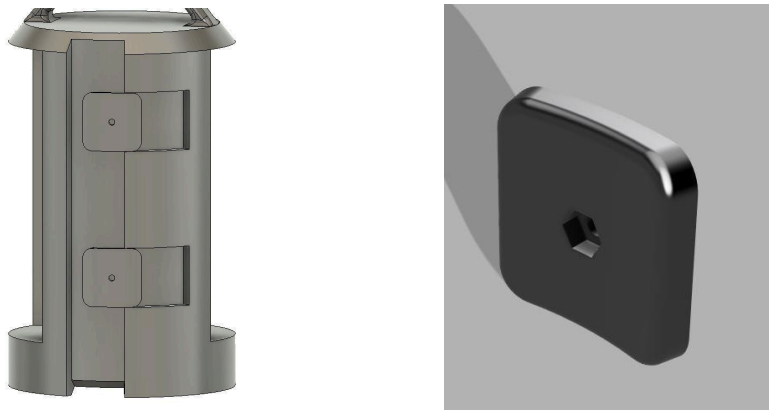


Fig 3.6 – Système baïonnette

IV. Conception électronique

1. Expériences

Le module central contient la caméra, une GoPro HERO3+ qui a pour but de filmer le vol. La caméra sera logée dans le module de telle sorte que l'objectif soit en face du trou. D'autres trous ont également été usinés afin de rendre accessibles les commandes de l'appareil.

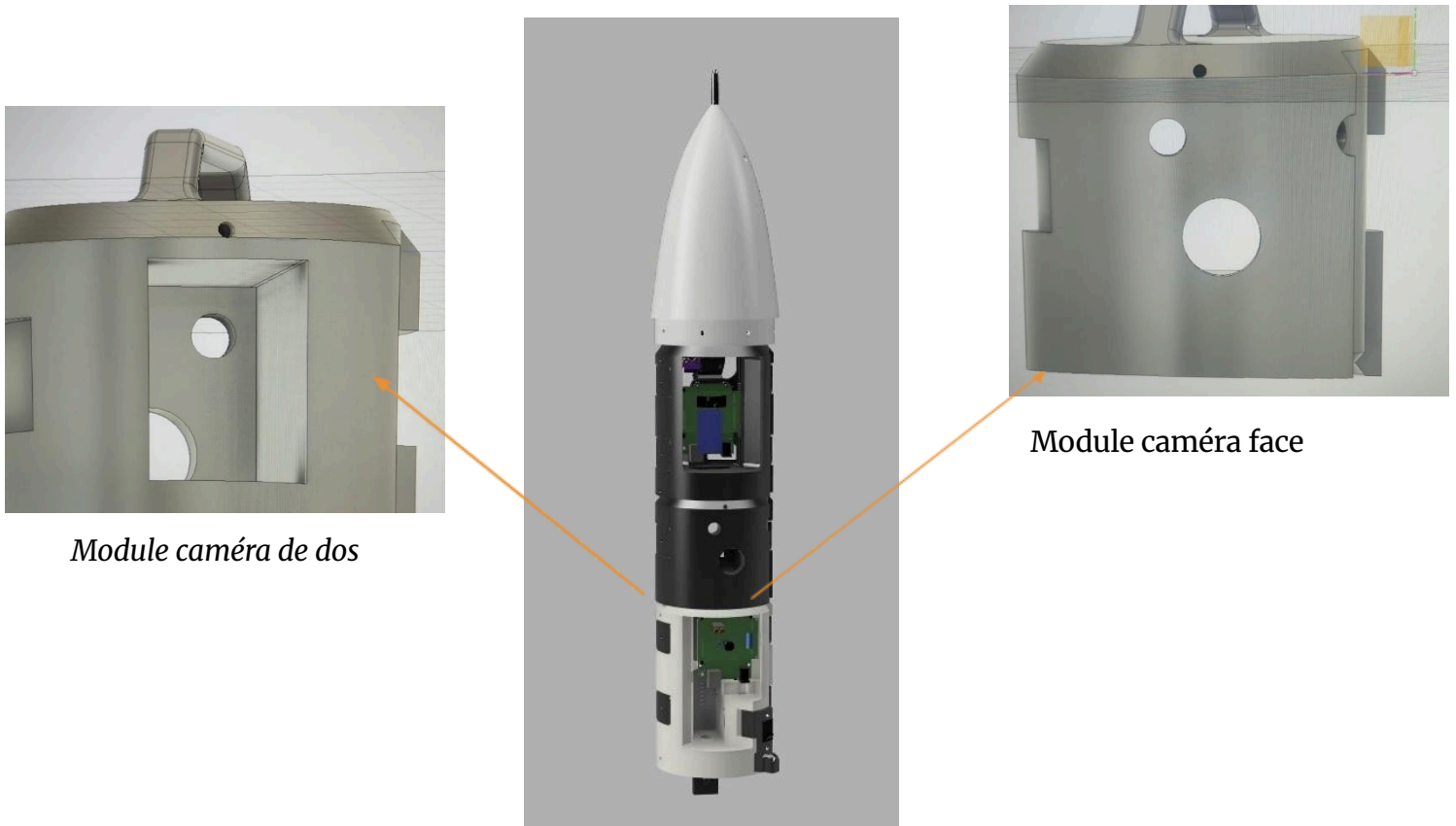
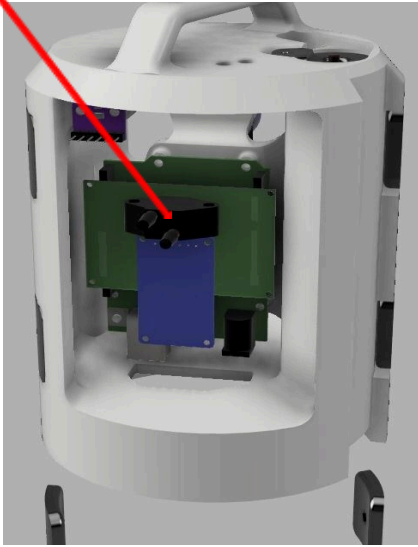


Fig 4.1 – Vue interne des modules expérience et de l'ogive

Le module électronique de l'expérience est représenté ci-dessous et embarque le capteur de pression différentielle, le capteur de gaz nocifs, le BMP280 (capteur de pression et température) et la carte SD. Il possède sa propre batterie NiMh de 1600mah et 9,6 V qui alimente toute la partie expérience. Ce module est positionné en dessous de l'ogive pour pouvoir connecter les tubes souples du tube Pitot vers le capteur.

Capteur de pression différentielle



Capteur de gaz

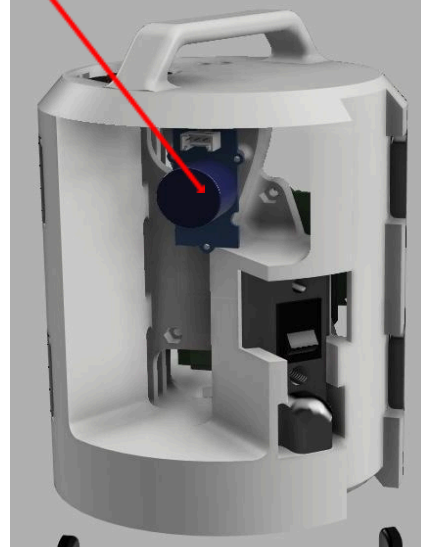


Fig 4.2 - Module expérience

L'ogive est constituée de deux parties: le tube de pitot est vissé dans la partie principale et la partie haute vient se visser sur cette dernière. Le tube de pitot lui est relié depuis l'intérieur de l'ogive au module électronique d'expérience par des tubes souples.



Tube de Pitot

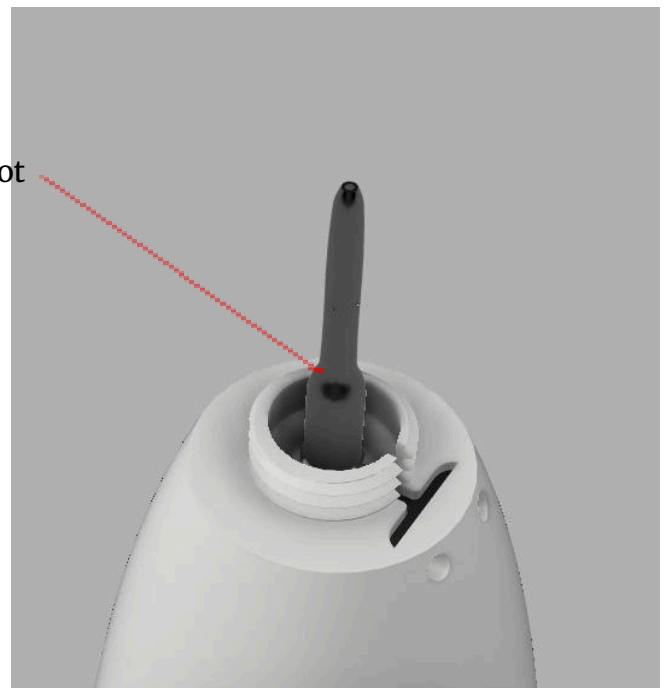


Fig 4.3 - Ogive et tube de Pitot

2. Récupération des données

Notre fusée n'embarque pas de télémétrie. Les données recueillies sont stockées sur une carte micro SD de 32 Go dans un document texte avant d'être traitées ultérieurement. Le décollage se détecte lors de l'arrachage des jacks, l'ouverture de la trappe et l'enregistrement des données étaient tous les deux gérés par une minuterie.

V. Semaine du C'space

1. Problèmes et solutions

Lors de notre premier passage aux contrôles nous avons eu 2 problèmes majeurs :

1^{er} Problème: La peau de la fusée n'est pas droite

En passant le contrôle de flèche statique, nous avons remarqué que la peau de la fusée était légèrement courbée dans un sens. Cela nous empêchait formellement d'être qualifiés pour lancer la fusée, les valeurs mesurées dépassaient le seuil limite imposé par le cahier des charges.

Causes: En optant pour un tube PVC nous avons réduit nos dépenses mais nous avons aussi grandement réduit la résistance de la fusée. Lorsque nous avons découpé la porte dans le tube, nous avons fragilisé la structure de celui-ci. Le tube a ainsi perdu sa forme d'origine et s'est légèrement courbé.

Solution: Pour pallier ce problème nous avons pensé à l'aide des bénévoles à un système placé à l'intérieur de la fusée, au niveau de la courbure. Celui-ci était composé de deux tiges filetées (M12) placées entre deux bagues en ABS (solidement renforcées par des pièces en acier). Au bout de chaque tige, nous avons placé des écrous que nous pouvons serrer afin d'écarter les deux bagues. Cela nous permet ainsi de redresser le tube

Cependant, ce système étant très lourd, notre fusée est passée de 6,2 kg à 7,8 kg à vide. Heureusement le centre de masse et la stabilité de la fusée n'ont pas beaucoup été impactés.



Fig 5.1 - Système employé pour redresser la fusée

2^{ème} Problème : Le parachute n'était pas assez solide

Lors du test de solidité du parachute, la plupart des œilletons se sont arrachés, prouvant que ce système n'était pas assez solide.

Solution : Nous avons alors opté pour un autre système de fixation des suspentes, nous les avons directement cousues à la toile, nous permettant ainsi de passer le test.



2. Déroulé du vol

Le vol est nominal !!! Le moteur s'est allumé et a eu une durée de combustion d'environ 2 secondes suivie d'une phase de 12 secondes de décélération qui amène sur l'ouverture du parachute à l'apogée (14 secondes après le décollage). Celle-ci s'est parfaitement bien passée, le servo moteur s'est déclenché au bon moment, comme prévu, ce qui a permis à la porte de s'écarter afin de laisser le parachute sortir. Le parachute s'est lui aussi bien déployé, la bague anti torche a parfaitement fonctionné et le parachute a résisté à la force du déploiement. Cependant, au début du vol nous avons pu remarquer que la fusée s'est redressé à 90 degrés (malgré un tir à 80 degrés) puis a continué son vol nominalement (*Fig 5.2*). Cela est sans doute dû à une rafale de vent qui a poussé la fusée sur les ailerons. D'après les données de notre capteur de pression, nous avons eu une apogée à 1350 m, contre les 1150 m prévus par le stabtraj. La déviation de trajectoire au décollage a donc considérablement modifié l'apogée.

A la suite du vol, nous avons pu récupérer la fusée. Celle-ci était intacte au niveau de la structure ainsi qu'au niveau de l'électronique.



Fig 5.2 - Décollage de la fusée



Fig 5.3 - Récupération de la fusée

VI. Résultats des expériences et des instruments de bord

1. Détermination de l'altitude en fonction du temps

Le module expérience (voir schéma fusée) est composé de capteurs qui transmettent des données à une carte arduino uno qui les enregistre sous forme d'un fichier texte sur une carte SD dont sont extraites les données après vol. Des données sont enregistrées toutes les 200 millisecondes et chaque prise de données correspond à une ligne dans le document avec les différentes données comme le temps, la pression différentielle, la pression atmosphérique, etc, séparées par un point virgule. Ainsi, à chaque temps correspondent des données.

```

1 13;968.89;41.26;0.205;1229.44;1486.25;776.92;240.76;115.65
2 244;967.50;41.23;0.298;1229.44;1486.07;776.92;195.56;115.69
3 468;965.80;41.24;0.430;1229.44;1486.13;776.92;195.56;115.68
4 693;963.61;41.24;0.616;1229.44;1486.13;776.92;195.56;115.68
5 917;959.77;41.20;0.846;1229.44;1485.89;776.92;195.56;115.73
6 1141;956.11;41.14;1.090;1333.21;1610.92;842.50;156.25;92.53
7 1365;950.81;41.07;1.354;1333.21;1610.59;842.50;195.56;92.58
8 1591;946.63;41.02;1.300;1333.21;1610.14;842.50;156.25;92.66
9 1815;942.37;40.97;1.241;1333.21;1484.51;842.50;195.56;116.03
10 2041;937.20;40.93;1.183;1333.21;1609.55;842.50;156.25;92.75
11 2265;933.31;40.90;1.119;1333.21;1484.09;776.92;195.56;116.12
12 2491;929.52;40.87;1.061;1333.21;1609.17;842.50;156.25;92.81
13 2715;924.55;40.84;1.022;1333.21;1483.74;776.92;195.56;116.20

```

Fig 6.1 – Extrait de données récoltées

La fusée Asteroidea est équipée d'un capteur de pression et de température intégré dans le module expérience. Celui-ci peut capter la pression statique extérieure grâce à un trou percé dans la partie haute de la fusée, assez éloigné du moteur, pour mesurer la température de l'air extérieur. La formule suivante (Code Python) permet de calculer l'altitude de la fusée en fonction du temps. Cela nous permet d'obtenir le graphique ci-contre et de déduire l'altitude maximum de 1350m.

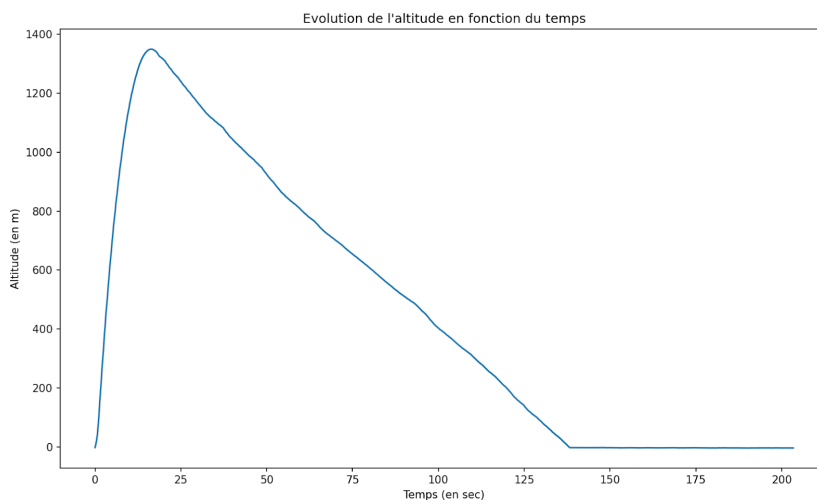


Fig 6.2 – Evolution de l'altitude en fonction du temps

```

1 from math import *
2 import os
3 import os.path
4 import matplotlib.pyplot as plt
5
6
7 températures = []
8 pressions = [] # création de listes
9 voltage = []
10 temps_en_secondes = []
11 altitudes = []
12
13
14 fichier = "pression.txt" # le nom du fichier à lire
15 chemin_absolu = os.path.join(os.path.dirname(
16     os.path.abspath(__file__)), fichier) # chemin absolu du fichier
17 # le code lira lignes par lignes le document
18 with open(chemin_absolu, "r", encoding="utf-8") as fichier:
19     for ligne in fichier:
20         temps, pression_statique, température, Voltage, R0, R0corrigé, résistance, PPM, PPMcorrigé = ligne.strip().split(";")
21         # le document doit être de la forme: pression; température(retour à la ligne et ainsi de suite)
22         # modification de la liste en intégrant la valeur recueillie
23         temps_en_secondes.append(float(temps)/1000) # conversion en secondes
24         pressions.append(float(pression_statique))
25         températures.append(float(température))
26         voltage.append(float(Voltage))
27
28 for i in range(len(pressions)):
29     P = pressions[i]
30     T = températures[i]
31     h = (((((1018/P)**(1/5.257)-1)*(0.01*T+273.15))/0.0065))-400
32     altitudes.append(h)
33
34 plt.plot(temps_en_secondes, altitudes)
35 plt.title("Evolution de l'altitude en fonction du temps")
36 plt.xlabel("Temps (en sec)")
37 plt.ylabel("Altitude (en m)")
38 plt.show()
39

```

Fig 6.3 - Code Python pour calculer l'altitude

En isolant la portion de courbe correspondant à l'ascension de la fusée, nous obtenons le graphique suivant, plus détaillé. On remarque que le gain d'altitude en fonction du temps est quasiment linéaire.

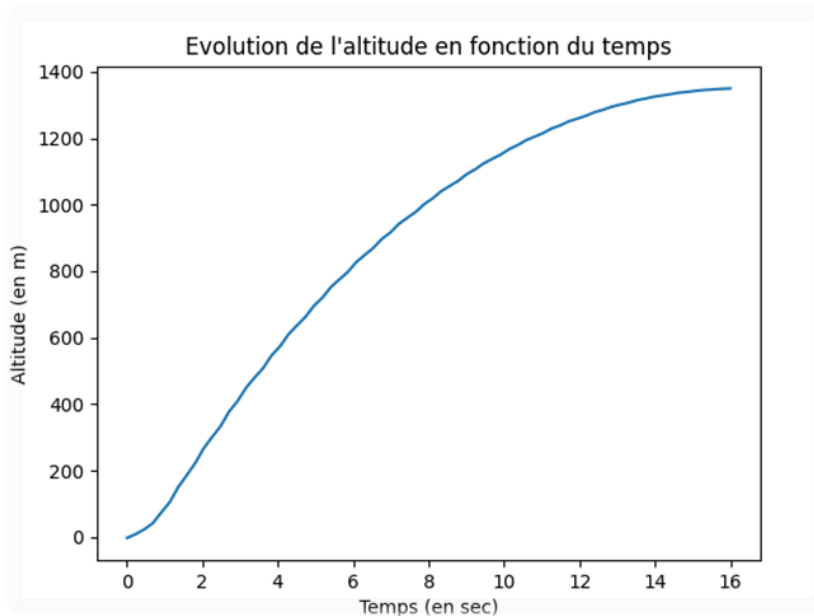


Fig 6.4 - Evolution de l'altitude en fonction du temps lors de la phase ascendante

2. Tube de Pitot

L'expérience principale de notre fusée consiste à mesurer sa vitesse au cours du vol à l'aide d'un tube de pitot. Le tube de pitot que nous avons utilisé est schématisé en figure 6.5. Il possède deux entrées d'air, une frontale et une latérale. Ces deux entrées permettent à l'air d'atteindre le capteur de pression situé au cœur du module expérience qui va enregistrer une différence de pression sur la carte SD. En utilisant la formule de Bernoulli et en négligeant la variation d'altitude, nous pouvons calculer la vitesse en-dessous de mach 0,3 grâce à la formule suivante :

$$\frac{1}{2}\rho v^2 + p_s = p_t \Rightarrow v^2 = \frac{2(p_t - p_s)}{\rho} \quad \text{Avec } P_t - P_s \text{ correspondant à la différence de pression}$$

Cependant, au-dessus de mach 0,3, l'air devient compressible et une nouvelle formule est nécessaire:

$$\frac{p_t}{p_s} = \left(1 + \frac{\gamma - 1}{2} M^2\right)^{\frac{\gamma}{\gamma - 1}}$$

M = nombre de Mach

p_t = pression totale

p_s = pression statique

γ = rapport des capacités calorifiques du fluide C_p/C_v .

Formule que l'on peut modifier pour isoler la vitesse :

$$V = \sqrt{\frac{2 \left(\frac{P_t}{P_s} \right)^{\frac{\gamma-1}{\gamma}} - 2}{\gamma - 1}} \times 344$$

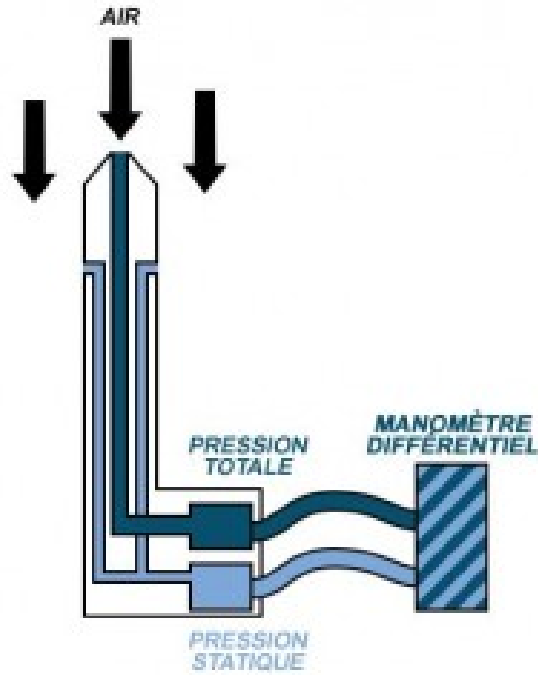


Fig 6.5 – Schéma Tube de Pitot

A l'aide du code Python ci-dessous et des données recueillies nous pouvons tracer le graphique suivant:

```

15 températures = []
16 pressions = [] # création de listes
17 voltage = []
18 temps_en_secondes = []
19 pression_différentielle = []
20 vitesses = []
21
22 fichier = "pression.txt" # le nom du fichier à lire
23 chemin_absolu = os.path.join(os.path.dirname(
24     os.path.abspath(__file__)), fichier) # chemin absolu du fichier
25 # le code lira lignes par lignes le document
26 with open(chemin_absolu, "r", encoding="utf-8") as fichier:
27     for ligne in fichier:
28         temps, pression_statique, température, Voltage, R0, R0corrigé, résistance, PPM, PPMcorrigé = ligne.strip().split(";")
29         # le document doit être de la forme: pression; température(retour à la ligne et ainsi de suite)
30         # modification de la liste en intégrant la valeur recueillie
31         temps_en_secondes.append(float(temps)/1000) # conversion en secondes
32         pressions.append(float(pression_statique))
33         températures.append(float(température))
34         voltage.append(float(Voltage))
35
36 c = len(voltage)
37 for i in range(c):
38     Pd = ((voltage[i]*0.196)/0.009148)*101.3
39     pression_différentielle.append(Pd)
40
41
42 M = 0.3*344 # Mac 0,3: limite à partir de laquelle la première formule n'est plus valable
43 v = 0
44 for i in range(len(pressions)):
45     p = 1.293
46     P = pressions[i] # la masse volumique de l'air au lieu lancement
47     Press_diff = pression_différentielle[i]
48     v2 = abs((2*(Press_diff)/p))
49     v = sqrt(v2)
50     if v <= M:
51         vitesses.append(v)
52     else:
53         V = 344*(sqrt((((Press_diff+P/P)**(1/3.5))-1)*2)/0.3997)
54         vitesses.append(V)
55
56 plt.plot(temps_en_secondes, vitesses)
57 plt.title("Evolution de la vitesse en fonction du temps ")
58 plt.xlabel("Temps (en sec)")
59 plt.ylabel("Vitesse (en m\dot{s}^{(-1)})")
60 plt.show()

```

Fig 6.6 – Code Python pour calculer la vitesse

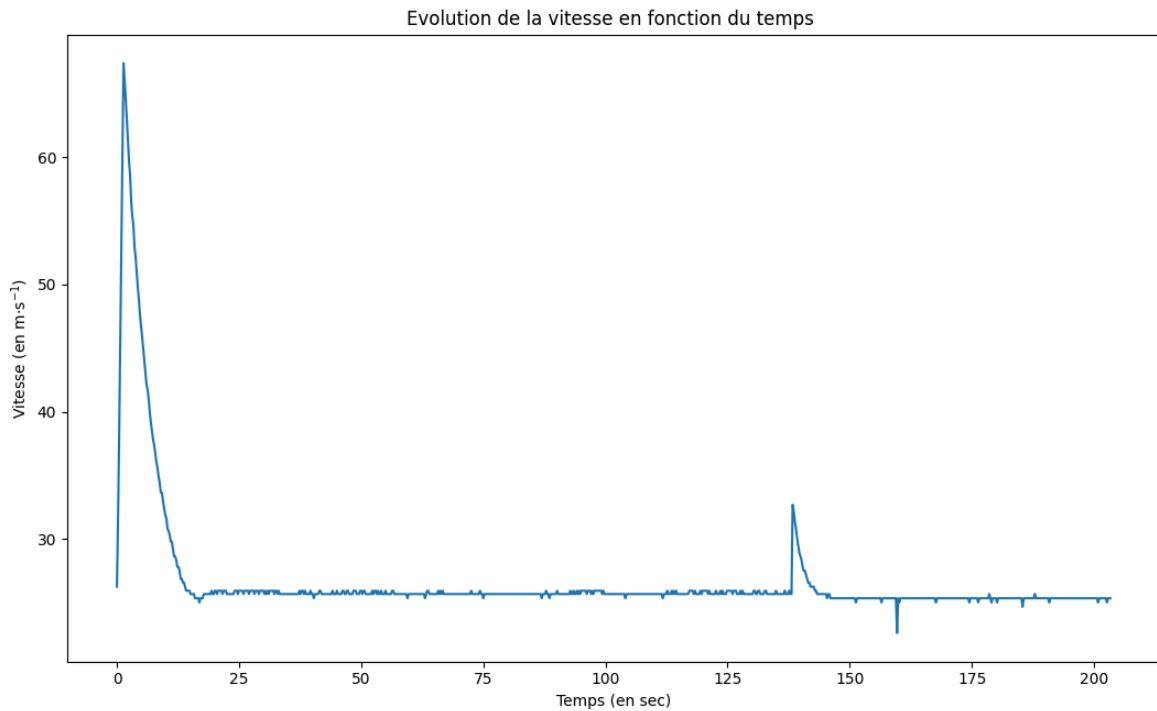


Fig 6.7 - Evolution de la vitesse en fonction du temps

Nous pouvons observer sur le graphique une vitesse maximum de 67,5 m/s ce qui est bien inférieur à la valeur estimée par le stabtraj qui est de 193 m/s. Nous pensons donc que le capteur a pu enregistrer des données erronées en raison d'une entrée d'air trop petite pour la pression totale et donc la pression différentielle. Une autre piste d'erreur possible est le mauvais alignement du tube de pitot par rapport à la fusée ce qui a pu conduire à une pression totale plus faible.

Afin de tout de même calculer la vitesse approximative de la fusée, nous avons considéré que celle-ci ne se déplaçait que verticalement afin de calculer sa vitesse avec sa variation d'altitude. Nous obtenons le graphique suivant

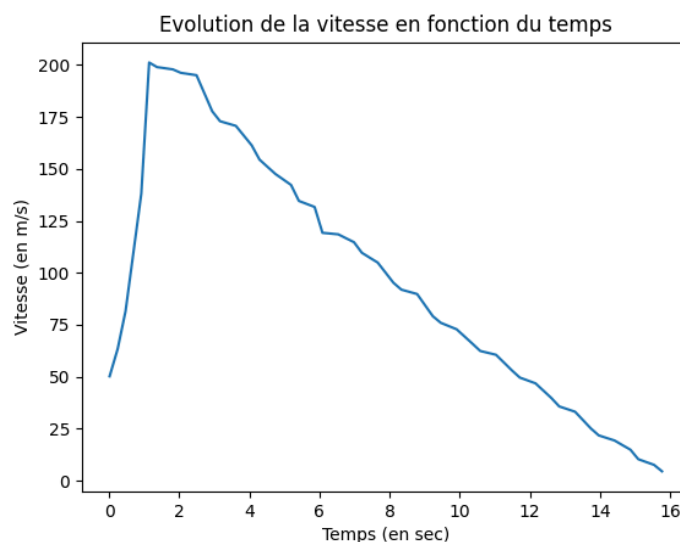


Fig 6.8 - Evolution de la vitesse en fonction du temps lors de la phase ascendante

Nous observons ici une vitesse maximale proche de 200 m/s, ce qui s'approche des valeurs du stabtraj. Nous pouvons conclure que cette expérience n'a que partiellement fonctionné car le capteur a bien mesuré des valeurs, mais celles-ci ne correspondent pas aux valeurs théoriques.

3. Capteur de gaz nocifs

Cette expérience consiste à mesurer la qualité de l'air en fonction de l'altitude. Nous utilisons le capteur MQ135, sensible au CO₂, à l'alcool, au Benzène, à l'oxyde d'azote NO_x et à l'ammoniac NH₃. On peut donc appeler la mesure "qualité de l'air" car tous ces gaz sont nocifs à forte dose, mais nous ne pouvons pas différencier les gaz entre eux.

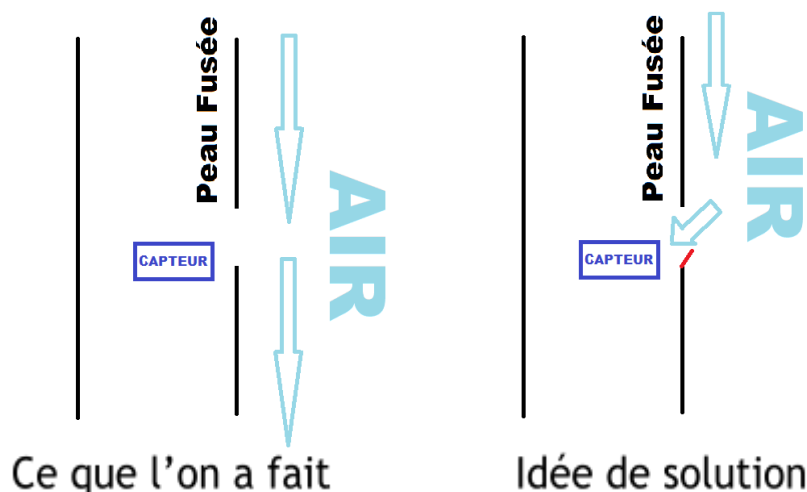
Toutefois, la concentration en CO₂ dans l'air est bien plus élevée que celle de ces autres gaz, on peut donc considérer ce capteur comme un capteur de CO₂.

Le procédé chimique qui permet la mesure est basé sur la tension entre deux électrodes séparées par le gaz ambiant. Nous pouvons noter la présence d'une résistance qui chauffe le gaz pour mesurer correctement, le capteur doit chauffer afin d'avoir des mesures précises. Nous avons donc allumé notre carte expérience dans la tente club pour laisser le temps au capteur d'atteindre la bonne température (au moins 20 min de chauffe). Lors de la première utilisation du capteur, nous avons dû le laisser chauffer environ 24h avant de pouvoir l'utiliser.

Ce capteur nous renvoie des valeurs avec comme unité le PPM . PPM signifie **P**artie **P**ar **M**illions (Exemple: 200 PPM de CO₂ correspond à 200 atomes de CO₂ pour 1 million d'atomes).

La teneur naturelle en CO₂ de l'air ambiant est d'environ 400 PPM.

A l'aide du code Python, nous avons pu obtenir un graphique représentant la teneur de CO₂ dans l'air en fonction de l'altitude. Sur ce graphique nous ne voyons pas d'évolution des ppm en fonction de l'altitude. Cela correspond à nos prévisions puisque le pourcentage d'atomes de CO₂ dans l'air reste le même peu importe l'altitude, c'est la quantité d'atome présent dans un certain volume qui diminue en fonction de l'altitude. Cependant, la valeur obtenue (environ 200 ppm) est inférieure à celle attendue, soit autour de 400 ppm. Cette différence pourrait s'expliquer par le mouvement de l'air lors du déplacement de la fusée, ou encore le trou dans la peau de la fusée était peut-être trop petit, empêchant une prise de mesures correctes.



```

1  from math import *
2  import numpy as np
3  import matplotlib.pyplot as plt
4  import os.path
5  import os
6
7  liste_temps = []
8  PPM = []
9
10 fichier = "pression.txt"
11 chemin_absolu = os.path.join(os.path.dirname(
12     | os.path.abspath(__file__)), fichier)
13
14
15 # le code lira lignes par lignes le document
16 with open(chemin_absolu, "r", encoding="utf-8") as fichier:
17     for ligne in fichier:
18         temps, pression, temperature, Vout, R0, R0corrige, resistance, PPMs, PPMcorrige = ligne.strip().split(";")
19         # le document doit être de la forme: pression; temperature(retour à la ligne et ainsi de suite)
20         # modification de la liste en intégrant la valeur recueillie
21         PPM.append(float(PPMs))
22         liste_temps.append(float(temps)/1000)
23
24 plt.plot(liste_temps, PPM)
25 plt.title("Evolution des PPM en fonction du temps ")
26 plt.xlabel("Temps (en sec)")
27 plt.ylabel("PPM")
28 plt.show()

```

Fig 6.9.1 - Code python pour tracer le graphique $PPM=f(t)$

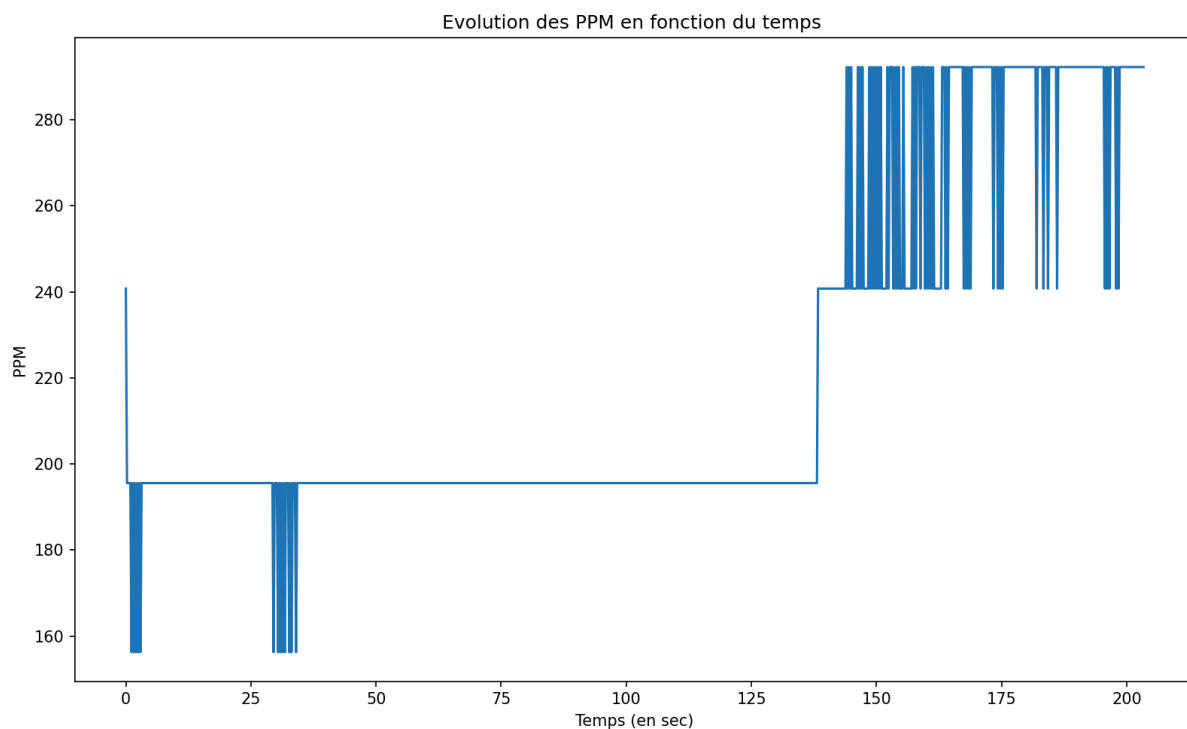


Fig 6.9.2 - Evolution des PPM en fonction du temps

4. Caméra

Notre fusée embarquait également un GoPro Hero 3. Celle-ci à parfaitement fonctionné durant tout le vol, même au décollage où nous craignons qu'elle ne s'éteigne à cause de l'accélération. Voici la superbe vidéo <https://youtube.com/shorts/zUTlao8EnbU?si=ZozKFNAAbLfm0h9G> . Cette vidéo nous permet non seulement d'avoir un très bon souvenir de cette semaine, mais aussi de vérifier que tout s'est bien passé durant le vol. On peut d'ailleurs voir que la fusée ne tourne presque pas sur elle-même durant la montée, nous entendons même le servo moteur s'actionner lors de l'ouverture de la trappe.

VII. Remerciements

Nous tenons tout d'abord à remercier l'ESCR (Ensemble Scolaire Catholique Rochois), sans qui la réalisation du projet aurait été impossible. Nous remercions également Planète Sciences et tout particulièrement nos bénévoles suiveurs pour leurs précieux conseils et leur engagement, et pour finir, tous ceux qui ont pu nous donner des conseils comme Madame Cormier et Monsieur Pugin pour leur temps et suggestions.