



Projet Cobra

Rapport de vol



Figure 1: Photographie de l'équipe accompagnée de membres de l'association avant le lancement

Table des Matières

1	Introduction	7
1.1	Équipe	8
1.2	Acteurs du projet	9
2	Description Mécanique	10
2.1	Caractéristiques générales	10
2.2	Fabrication	11
2.2.1	Tube	12
2.2.2	Ogive	12
2.2.3	Bague de changement de diamètre	12
2.2.4	Bague de reprise de poussée	12
2.2.5	Ailerons	12
2.2.6	Covering	12
2.2.7	Système de récupération	12
2.3	Propulseur	12
3	Description Avionique	14
3.1	Aperçu général	14
3.2	Ordinateur de bord	15
3.3	Ordinateur d'acquisition de données	16
3.3.1	Fonctionnement	16
3.3.2	Accéléromètre	17
3.3.3	Capteur de force	17
3.3.4	Capteurs de vitesse	17
3.4	Alimentation électrique	17
3.5	Enregistrement des données	18
4	Expérience	20
4.1	Objectif de l'expérience	20
4.2	Résultats attendus de l'expérience	20
4.3	Paramètres mesurés lors du vol	21
4.3.1	Accélération	21
4.3.2	Poussée	21
4.3.3	Vitesse	21
4.4	Étalonnage	22
5	Chronologie	23
6	Déroulement du vol	24
7	Analyse du vol	25
7.1	Données des capteurs	25
7.1.1	Capteurs de vitesse	25
7.1.2	Capteur de force	26
7.1.3	Accéléromètre	26
7.2	Résultats de l'expérience	27
7.3	Interprétation des données	28
7.4	Axes d'amélioration	28

8	Conclusion	29
9	Annexes	30
9.1	Références des composants électroniques	30
9.2	Code des logiciels embarqués	31
9.3	Plan de câblage du shield AOP de l'ordinateur d'acquisition de données	35
9.4	Plage du niveau d'intensité sonore du buzzer <i>Sonitron SMA-24L</i>	36

Table des Figures

1	Photographie de l'équipe accompagnée de membres de l'association avant le lancement	1
2	L'ensemble de l'équipe à la sortie de la RCE3 2023. Alexandre à gauche, Pierre à droite	8
3	Dimensions des ailerons de la fusée	10
4	Trajectoire théorique de la fusée	10
5	CAO de la fusée	11
6	Atelier du Pôle Léonard de Vinci pas vraiment rangé	11
7	Caractéristiques de poussée du Pro54 5G C	13
8	Caractéristiques de poussée du Pro54 5G WT	13
9	Rack électronique de Cobra	14
10	Schéma électronique de l'ordinateur de bord	15
11	Schéma électronique de l'ordinateur d'acquisition des données	16
12	Étude de linéarité durant l'étalonnage du <i>MPX2010DP</i>	22
13	Cobra en rampe avant le lancement	24
14	Mesure de la vitesse de vol via les capteurs <i>MPX2010DP</i> et <i>MPX2050DP</i>	25
15	Mesure de la compression de la fusée	26
16	Caractéristiques techniques du moteur	26
17	Mesure de l'accélération de la fusée	27
18	Mesure du coefficient de traînée de Cobra en fonction de la vitesse	27
19	L'équipe accompagné d'un bénévole après la récupération de Cobra	29
20	Plan de câblage du shield AOP de l'ordinateur d'acquisition de données	35
21	Niveau d'intensité sonore par rapport à la tension d'entrée du buzzer <i>Sonitron SMA-24L</i>	36

Table des Tableaux

1	Autonomie des systèmes embarqués de Cobra	18
2	Structuration de l'enregistrement des données	18
3	Chronologie de préparation au lancement	23
4	Références des composants électroniques	30

Table des Scripts

1	Code Arduino de l'ordinateur de bord	31
2	Code Arduino de l'ordinateur d'acquisition de données	32

1 Introduction

Pour sa sixième année consécutive, notre association a participé à la campagne de lancement C'Space 2024, un projet de construction et de lancement d'une fusée expérimentale proposé par le CNES (le Centre National d'Etudes Spatiales) et Planète Sciences. Ce projet consiste à construire intégralement une fusée (de type Fusex) respectant un cahier des charges préalablement défini. A son bord, la fusée doit abriter une expérience.

Cette année, nous avons décidé de travailler sur la mesure du coefficient de traînée C_x de notre fusée.

1.1 Équipe

Considérant la multitude de disciplines concernées par le projet, nous nous devons de réunir une équipe de vétérans : Alexandre Slama et Pierre Debaisieux.



Figure 2: L'ensemble de l'équipe à la sortie de la RCE3 2023. Alexandre à gauche, Pierre à droite

1.2 Acteurs du projet

- LéoFly

LéoFly est une association d'aéronautique et d'aérospatiale fondée en novembre 2015. Elle a pour objectif de rassembler les étudiants du Pôle Léonard de Vinci (à Paris La Défense) autour de leur passion commune. Pour ce faire, les membres du bureau proposent de nombreux projets techniques (réalisation d'un cockpit de simulation à taille réelle, d'une aile volante autonome, etc.) dont certains sont en collaboration avec le CNES ainsi que des conférences et des visites dans le domaine de l'aéronautique ou du spatial.



- Centre National d'Etudes Spatiales (CNES)

Le Centre National d'Etudes Spatiales (CNES) est un établissement public à caractère industriel et commercial chargé d'élaborer et de proposer au gouvernement le programme spatial français et de le mettre en œuvre.



- Planète Sciences

Planète Sciences est une association à but non lucratif proposant aux jeunes passionnés des activités scientifiques et techniques expérimentales, grâce à différents projets durant leurs études. Cette année, Planète Sciences nous a encadré grâce à 3 réunions techniques et de contrôle, ce sont les «Rencontre Club Espace» dites «RCE». Les bénévoles sont également présents sur le Camp de Ger (lieu de la campagne C'Space pour l'édition 2024) pour nous apporter des conseils et pour effectuer les vérifications techniques imposées par le cahier des charges avant de procéder au décollage.



2 Description Mécanique

2.1 Caractéristiques générales

La structure de Cobra est dite en peau porteuse, c'est-à-dire que tous les éléments sont fixés sur le corps. Un changement de diamètre sépare la fusée en deux sections. La partie supérieure est de diamètre externe 100mm et la partie inférieure 80mm. Les deux sections font 2mm d'épaisseur. La longueur totale de la fusée est de 1.52m avec le changement de diamètre placé à 40cm en dessous de l'ogive.

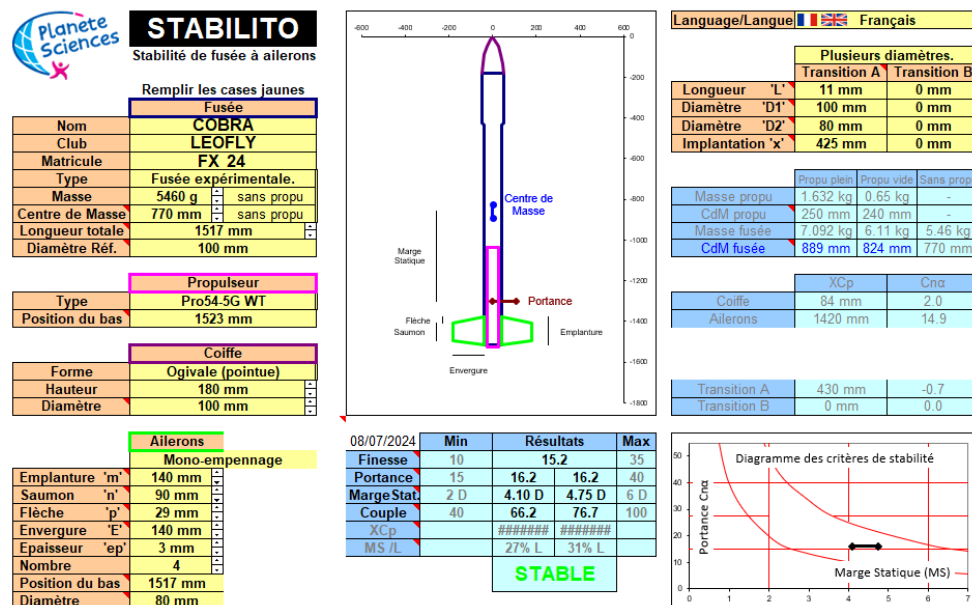


Figure 3: Dimensions des ailerons de la fusée

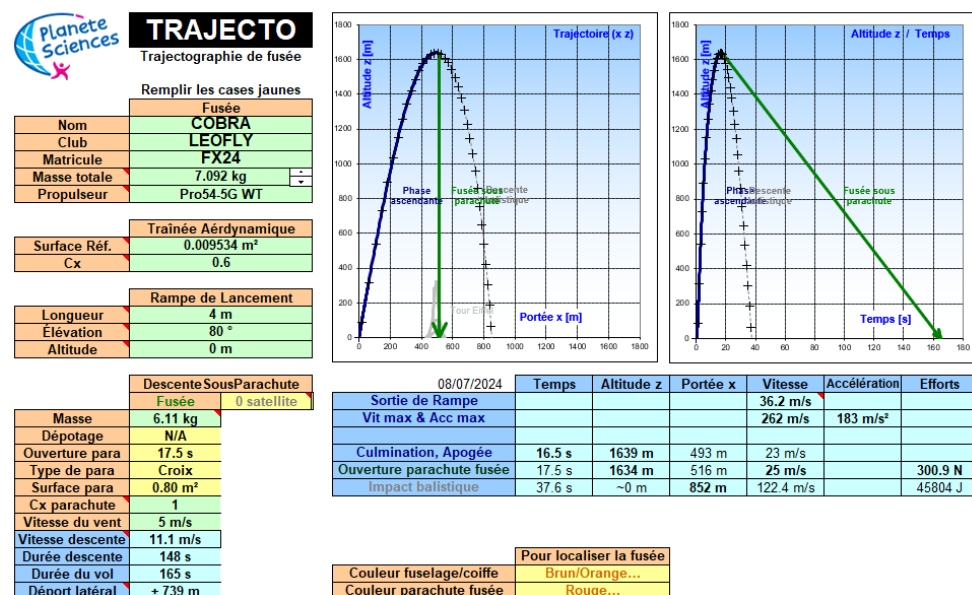


Figure 4: Trajectoire théorique de la fusée

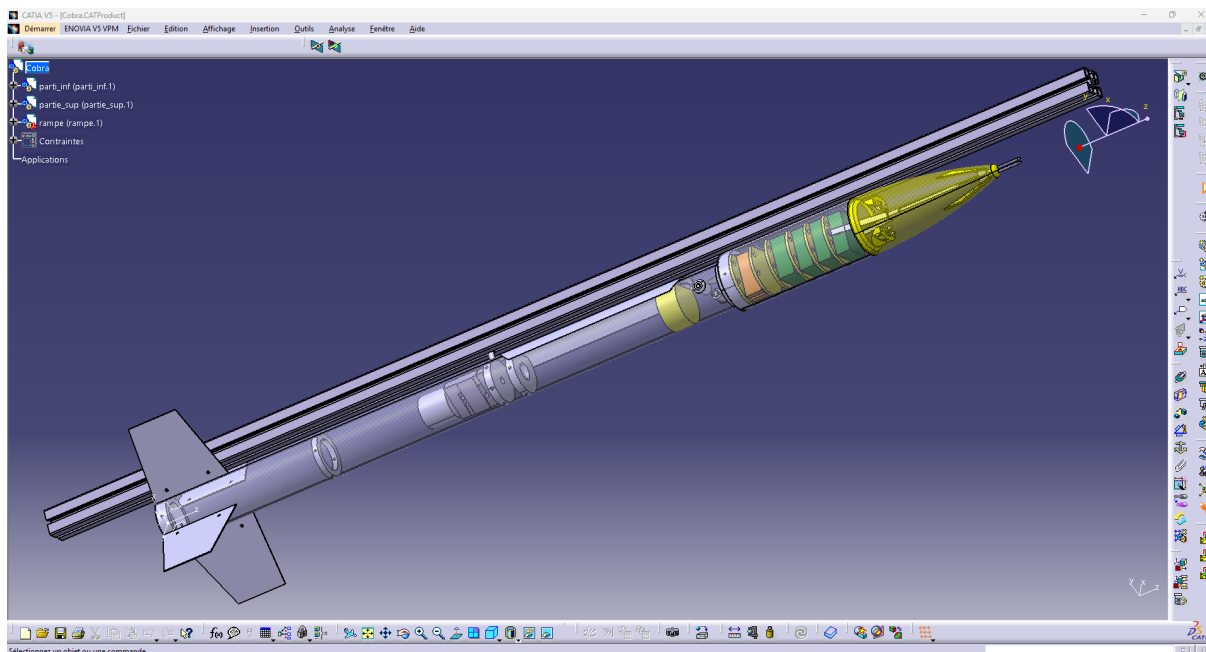


Figure 5: CAO de la fusée

2.2 Fabrication

L'ensemble de nos pièces est réalisé grâce au centre d'usinage de notre école. L'intégralité des bagues sont fabriquées à partir de 2 bruts cylindrique en aluminium (100mm et 80mm de diamètre). Plusieurs machines à outils sont à disposition tels qu'une scie à ruban pour sectionner, un tour manuel pour différentes opérations : surfacages, alésages, contournages, gorges (etc..) ainsi qu'une fraiseuse 3-axes manuelle. Cette dernière permet d'usiner des formes complexes sans démontage de la pièce. Les opérations sont effectuées partir d'un G-Code généré par les opérations d'usinage définis dans la FAO (Fabrication Assistée par Ordinateur) des pièces sur CATIA V5.



Figure 6: Atelier du Pôle Léonard de Vinci pas vraiment rangé

2.2.1 Tube

La fusée est composée de deux tubes en aluminium AlMgSi 0,5. Le tube de la partie supérieure est de diamètre extérieur de 100mm avec une épaisseur de 2mm. Le tube de la partie inférieure est de 80mm de diamètre extérieur et 2mm d'épaisseur. Les tubes sont achetés via le fournisseur LANGLITZ Metalle sur Amazon.

2.2.2 Ogive

L'ogive est de forme parabolique. Elle est imprimée en PLA et possède une longueur totale de 180cm pour un diamètre de 10cm. En haut de l'ogive se situent deux trous espacés de 2.2mm permettant d'y insérer les deux sondes Pitot. A l'intérieur de l'ogive se trouvent des fixations vissables pour les deux capteurs des sondes Pitot.

2.2.3 Bague de changement de diamètre

La bague de changement de diamètre s'insère de 20mm sous la peau du tube de 100mm de diamètre, et vient se coulisser sur la partie extérieure du tube de 80mm. Elle est réalisée en aluminium. Au total, quatre vis de type M5 en interne et quatre vis de type M5 en externe permettent de fixer la bague aux deux tubes. Le changement de diamètre est effectué avec un angle de 45°.

2.2.4 Bague de reprise de poussée

La bague de reprise de poussée est fixée au tube par six vis de type M5 et a une épaisseur de 15mm. Elle est réalisée en aluminium. Cette bague possède une double fonction car elle permet d'y fixer le capteur de force. Une autre pièce en aluminium est placée sous le capteur de force pour pouvoir épouser le haut du propulseur, ce qui permet d'être au plus proche de la force exercée par le propulseur.

2.2.5 Ailerons

Les ailerons sont découpés dans des plaques de carbone à l'aide d'une découpeuse jet d'eau. Les plaques sont achetées sur Amazon via le fournisseur Kalolary. Ce sont des plaques au format 300x300x3mm. La conception des ailerons permet de faire rentrer les quatre ailerons dans une seule plaque.

2.2.6 Covering

Le covering de la fusée est commandé sur MPA Pro dans la rubrique autocollant rectangle. Le supplément pelliculé est prit pour mieux conserver le covering notamment durant les phases de manipulations de la fusée. Le covering est resté intact après le vol malgré la retombée dans une zone marécageuse.

2.2.7 Système de récupération

Le récupération est effectuée par l'éjection d'une trappe latérale dans la section de 80mm de diamètre. L'électro-aimant Intertec ITS-PE3529-12VDC libère la trappe à l'apogée. Une minuterie active un relais mécanique lui permettant de s'alimenter sur une batterie LiPo 3S. Cette batterie est uniquement dédiée à l'électro-aimant.

2.3 Propulseur

Cobra étant initialement prévu pour un lancement à la campagne 2023, la fusée a été conçue pour un moteur Cesaroni Pro54 5G C. Un changement de propulseur a été effectué pour la campagne 2024,

le moteur proposé pour les lancements était un Pro54 5G WT. Les caractéristiques dimensionnelles de ces deux propulseurs sont identiques, aucune modification mécanique n'a du être apportée pour la campagne 2024. Cependant, ces deux moteurs ont des poussées différentes. Le Pro54 5G WT pousse deux fois moins longtemps que le 5G C, en contrepartie, le 5G WT pousse deux fois plus fort. Cette multiplication par deux de la poussée ne correspondait plus aux hypothèses que nous avons fixées durant le choix de nos capteurs. Nous avons décidé de ne pas modifier nos capteurs malgré le changement de propulseur. Ce choix va apporter des saturations des mesures durant le vol. Nous reviendrons sur ce point dans la parti d'analyse des résultats de vol 16.

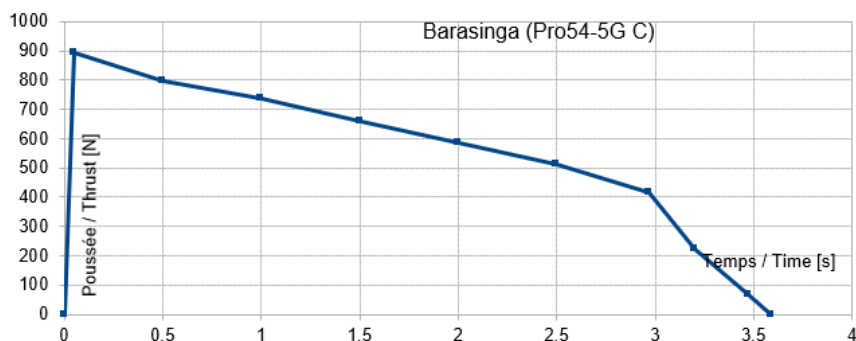


Figure 7: Caractéristiques de poussée du Pro54 5G C

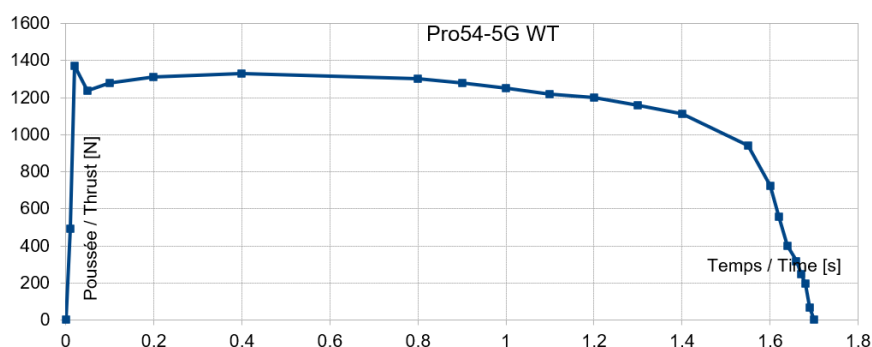


Figure 8: Caractéristiques de poussée du Pro54 5G WT

3 Description Avionique

3.1 Aperçu général

L'avionique de Cobra est située dans la partie haute de la fusée, sous la coiffe. Elle peut être décomposée en deux systèmes distincts :

- Un ordinateur de bord (ou séquenceur)
- Un ordinateur d'acquisition de données (ou expérience)

L'avionique est composée d'un rack électronique de trois étages avec des connexions électriques vers différents capteurs et actionneurs. Le premier étage permet de stocker physiquement toutes les sources d'alimentation (piles alcalines 9V et batterie Lipo 3S). Au deuxième étage se trouve la carte de l'ordinateur de bord. Enfin, on retrouve au troisième étage la carte de l'ordinateur d'acquisition de données avec ses deux shields.

Etant donné la criticité de l'ordinateur de bord pour la réussite du vol, les deux systèmes sont électroniquement indépendants.

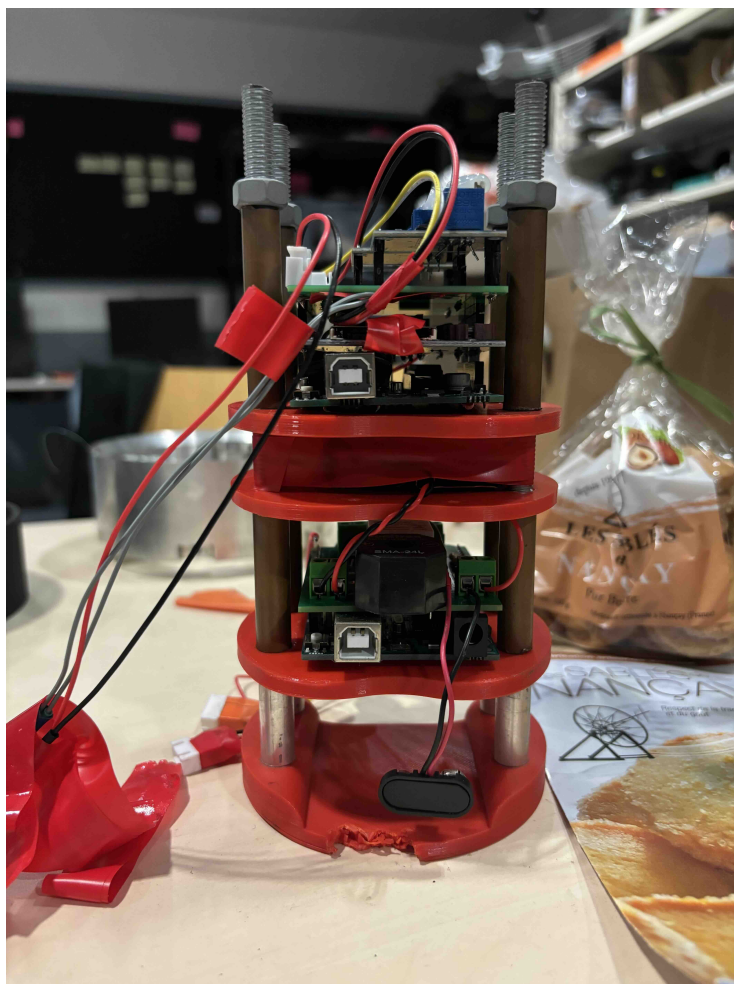


Figure 9: Rack électronique de Cobra

3.2 Ordinateur de bord

L'ordinateur de bord pilote le système de récupération de Cobra. Son but principal est le déploiement du parachute lorsque Cobra atteint son apogée, estimée à 1639 mètres d'après la simulation de la trajectoire de Cobra (voir figure 4). Comme explicité précédemment, ce système est très critique car son bon fonctionnement est impératif pour un vol nominal.

Lors de son activation par un interrupteur à clé, l'ordinateur de bord démarre. Le microcontrôleur utilisé est un *ATMEGA328-PU*, utilisé avec une carte *Arduino UNO R3*. Il est interfacé avec plusieurs composants électronique :

- Un buzzer *Sonitron SMA-24L*
- Un électro-aimant *Intertec ITS-PE3529-12VDC*
- Deux relais mécaniques de puissance *Finder 36 Series*

Un buzzer émet un son toutes les quatre secondes pour acquitter son état en marche. Le niveau sonore du buzzer est de *95dB* mais est légèrement atténué étant donné qu'il est situé à l'intérieur de la fusée. Le buzzer est alimenté par l'intermédiaire d'un relais mécanique.

Un câble de type jack est connecté à l'ordinateur de bord. Lors du décollage, le microcontrôleur détecte le changement d'état du GPIO concerné et déclenche une minuterie de 16.5 secondes, étant le temps nécessaire pour que Cobra atteigne son apogée d'après la simulation de la trajectoire de Cobra (voir figure 4).

Lorsque Cobra atteint son apogée - donc lorsque la minuterie s'est écoulée - l'ordinateur de bord alimente le relais mécanique connecté à l'électro-aimant qui retient la trappe de la fusée. Ce dernier est de type permanent. Son alimentation vient désactiver le champ magnétique afin de libérer la trappe. Par la suite, le buzzer émet un son toutes les dix secondes pour aider à la récupération de la fusée.

La partie électronique de l'ordinateur de bord est composé d'une carte *Arduino UNO R3* sur laquelle est ajouté un shield custom contenant tous les autres composants électroniques.

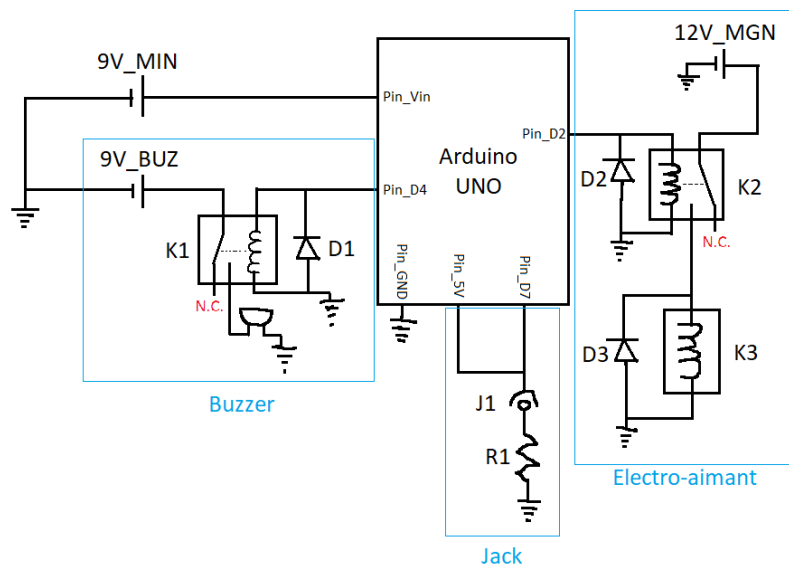


Figure 10: Schéma électronique de l'ordinateur de bord

3.3 Ordinateur d'acquisition de données

3.3.1 Fonctionnement

L'ordinateur d'acquisition de données pilote les capteurs d'acquisition permettant de mener à bien l'expérience de Cobra. Son bon fonctionnement est donc crucial afin d'obtenir des données exploitables.

Lors de son activation par un interrupteur à clé, l'ordinateur de bord démarre. Le microcontrôleur utilisé est un *ATMEGA328-PU*, utilisé avec une carte *Arduino UNO R3*. Il est interfacé avec plusieurs capteurs et composants électroniques :

- Un accéléromètre *MPU-6050*
- Un capteur de force *Tedea Huntleigh*
- Deux capteurs de vitesse *MPX2010DP* et *MPX2050DP*
- Trois modules d'amplificateur opérationnel *AD620*
- Un COTS¹ de carte SD en shield *AZ-DELIVERY Data Logger Shield*

Par analogie avec l'ordinateur de bord, un câble de type jack est connecté à l'ordinateur d'acquisition de données. Lors du décollage, le microcontrôleur détecte le changement d'état du GPIO concerné et débute l'enregistrement des données provenant des capteurs d'acquisition.

La partie électronique de l'ordinateur d'acquisition de données est composé d'une carte *Arduino UNO R3* sur laquelle est ajouté deux shield : un premier shield permet de connecter tous les modules d'amplificateur opérationnel *AD620* (voir figure 20), un deuxième shield étant le *AZ-DELIVERY Data Logger Shield*, le *MPU-6050* ainsi que la connection au jack est soudé sur ce shield.

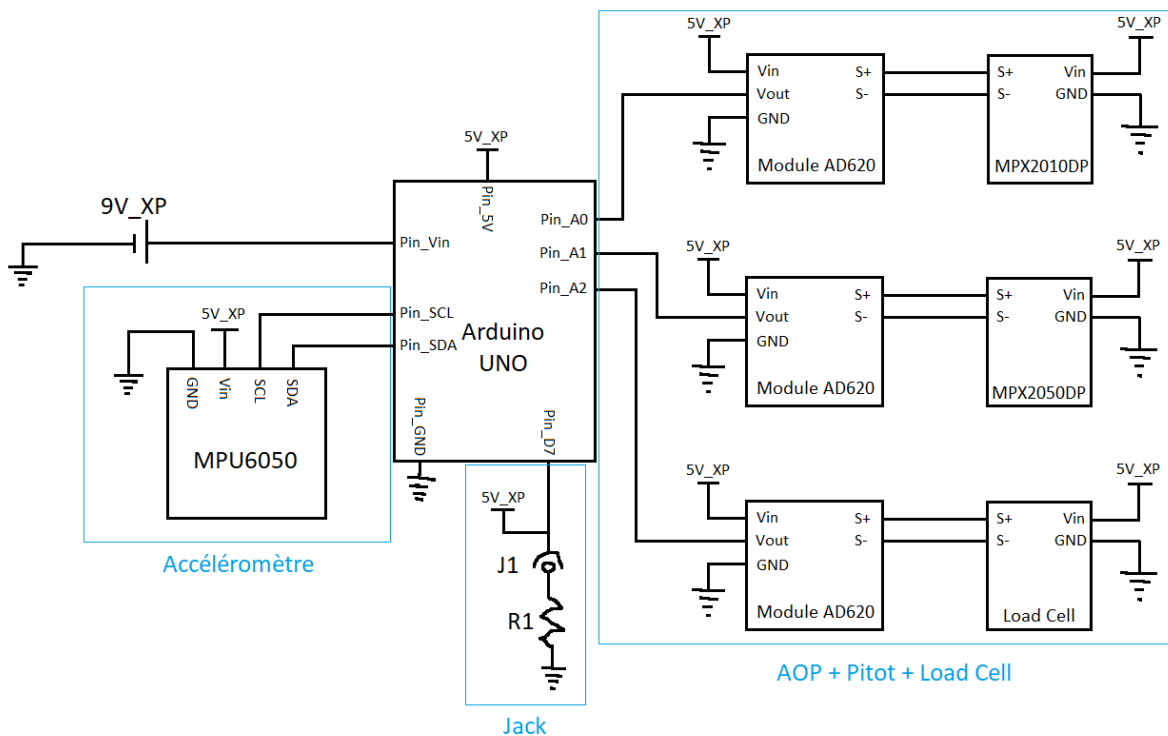


Figure 11: Schéma électronique de l'ordinateur d'acquisition des données

¹Customer Of The Shell

3.3.2 Accéléromètre

Le *MPU-6050* est un module bas de gamme constitué d'un capteur de température, un gyroscope et un accéléromètre. Seul l'accéléromètre est utilisé dans l'expérience (Les données du capteur de température et du gyroscope sont aussi enregistrées mais ne sont pas exploitées).

L'accéléromètre du *MPU-6050* est capable de mesurer une accélération de $+/- 16g$ sur 16 bits. En cas de saturation, le capteur reste à sa valeur la plus proche. Il communique ses données via un bus de communication I2C, ce qui facilite son intégration avec la carte *Arduino UNO R3* étant donné que le microcontrôleur *ATMEGA328-PU* embarque déjà un bus de communication I2C.

3.3.3 Capteur de force

Le capteur de force de poussée est une cellule de charge *Tedea Huntleigh* par compression pouvant mesurer jusqu'à $100kg$. Sa valeur de non élasticité est de $150kg$ et il peut résister jusqu'à $300kg$.

Ce capteur est capable de mesurer la force de poussée exercée par le propulseur sur la fusée. Il demande une tension d'excitation pour fonctionner, il va renvoyer une tension qui sera proportionnelle à la force qui lui est exercée. La tension de retour est lue par un GPIO digital du microcontrôleur *ATMEGA328-PU*. Ce microcontrôleur possède des convertisseurs internes travaillant sur 10 bits non signé. La tension de sortie étant trop faible pour le microcontrôleur, un module d'amplificateur opérationnel *AD620* est placé entre ce dernier et la sortie du capteur. Le capteur est alimenté via la sortie 5V de la carte *Arduino UNO R3*.

3.3.4 Capteurs de vitesse

Les deux capteurs de vitesse permettent de mesurer la vitesse de la fusée via la pression différentielle. Chacun des deux capteurs est relié à une sonde Pitot. Une première sonde est équipée d'un capteur *MPX2010DP*. Ce capteur peut mesurer une tension entre $0kPa$ et $10kPa$. La deuxième sonde Pitot est équipée de la version *MPX2050DP*. Ce capteur peut mesurer une tension de $0kPa$ à $50kPa$.

Par analogie avec le capteur de force, un GPIO du *ATMEGA328-PU* est attribué à la tension de sortie pour chacun des deux capteurs de vitesse, la tension de sortie étant proportionnelle à la pression. Un module d'amplificateur opérationnel *AD620* est présent pour chaque capteur. Les MPX sont alimentés via la sortie 5V de la carte *Arduino UNO R3*.

3.4 Alimentation électrique

L'ordinateur de bord est alimenté par une pile alcaline 9V. Le courant passe directement dans la pin *Vin* de la carte *Arduino UNO R3*. Cette pin est reliée à un convertisseur de tension qui vient mettre à disposition au *ATMEGA328-PU* une tension propre de 5V. Une deuxième pile alcaline 9V est dédiée uniquement au buzzer. On peut apercevoir sur la figure 21 qu'une tension de 9V permet d'obtenir un niveau d'intensité sonore d'environ $95dB$. Enfin, une batterie Lipo 3S est dédiée uniquement à l'électro-aimant. Etant donné sa criticité, le choix d'une source d'alimentation unique a été privilégié. La pile alcaline 9V du buzzer ainsi que la batterie Lipo 3S de l'électro-aimant sont chacune d'entre elles pilotées par un relais, interfacé avec le microcontrôleur *ATMEGA328-PU*.

Concernant l'ordinateur d'acquisition de données, une seule pile alcaline 9V est utilisée pour l'intégralité des composants (microcontrôleur, capteurs, etc...). Par analogie avec l'ordinateur de bord, le courant passe directement dans la pin *Vin* de la carte *Arduino UNO R3* qui est reliée à un convertisseur de tension afin de mettre à disposition au *ATMEGA328-PU* une tension propre de 5V.

Etant donné la faible consommation de chacun des systèmes, l'autonomie des deux ordinateurs dépassent largement les trois heures requises par le cahier des charges FusEx (voir table 1).

Système embarqué	Durée d'autonomie
Ordinateur de bord	> 3 heures
Ordinateur d'acquisition de données	> 3 heures

Table 1: Autonomie des systèmes embarqués de Cobra

3.5 Enregistrement des données

L'enregistrement des données se fait uniquement au sein de l'ordinateur d'acquisition de données. Un COTS de carte SD en shield *AZ-DELIVERY Data Logger Shield* est utilisé pour piloter la fonctionnalité d'enregistrement de données. Le shield se fixe au dessus de la carte *Arduino UNO R3*. Il utilise le bus de communication SPI, ce qui facilite son intégration étant donné que le microcontrôleur *ATMEGA328-PU* embarque déjà un bus de communication SPI. Une horloge interne au shield permet d'horodater automatiquement les enregistrements.

L'enregistrement débute dès que l'ordinateur d'acquisition de données détecte le décollage, c'est-à-dire lorsque le câble de type jack se déconnecte. La fréquence d'échantillonnage des données est d'environ 125Hz , ce qui permet d'acquérir des données avec une précision suffisante pour l'exploitation post-vol. Aucune limitation de fréquence d'échantillonnage n'est imposé à la carte *Arduino*, seul les performances de la carte fixe la limite. A chaque boucle, plusieurs données sont directement écrites dans la carte SD. Cela permet de ne pas perdre le début des données dans le cas où la carte SD rencontre un problème durant le vol. La structure de chaque enregistrement est décrite dans le tableau 2.

Index	Données	Unité	Type	Taille
0	Temps écoulé depuis le décollage	ms	entier	64 bits
1	Accélération Axe X	m/s^2	flottant	32 bits
2	Accélération Axe Y	m/s^2	flottant	32 bits
3	Accélération Axe Z	m/s^2	flottant	32 bits
4	Gyroscope Axe X	rad/s	flottant	32 bits
5	Gyroscope Axe Y	rad/s	flottant	32 bits
6	Gyroscope Axe Z	rad/s	flottant	32 bits
7	Tension Capteur de force	mV	entier	32 bits
8	Tension MPX2010DP	mV	entier	32 bits
9	Tension MPX2050DP	mV	entier	32 bits
10	Température	$Celsius$	flottant	32 bits

Table 2: Structuration de l'enregistrement des données

Parmi toutes les données enregistrées, seules le temps écoulé, l'accélération sur l'axe z, et les trois tensions de sorties sont exploitées. Les données du gyroscope et du capteur de température sont aussi enregistrées car elles étaient mise à disposition, mais n'interviennent pas dans l'expérience de la mesure du C_x . Par ailleurs, malgré une précision de 10 bits sur les tensions de sortie des capteurs de pression différentielle ainsi que du capteur de force, les tensions sont stockées sur des données de 32 bits.

L'équipe a dû faire face à une problématique d'adressage des composants utilisant le bus de communication I2C. En effet, deux composants utilisent ce bus de communication (le *AZ-DELIVERY Data Logger Shield* et le *MPU-6050*), ce qui entraîne un multiplexing sur le bus I2C. Pour que le

microcontrôleur sache à qui communiquer, les composants d'un même bus doivent avoir des adresses différentes. Cependant, ces deux composants avaient la même adresse. Il a donc été décidé de faire changer l'adresse du *MPU-6050*. C'est la pin AD0 qui gère cette fonctionnalité : l'état de la pin AD0 change le dernier bit significatif (LSB) de l'adresse. Ainsi, en alimentant la pin AD0, l'adresse passe de 0x68 à 0x69.

4 Expérience

4.1 Objectif de l'expérience

L'objectif du vol de Cobra est d'étudier la faisabilité de la mesure du C_x . Pour cela, nous avons équipé notre fusée de plusieurs capteurs, ce qui nous permet de retrouver les différents paramètres pour le calcul du coefficient de traînée.

4.2 Résultats attendus de l'expérience

Nous nous attendons à retrouver la traînée de la fusée en effectuant un bilan des forces. Le bilan des forces nous donne :

- le poids de la fusée noté P
- la force de poussée noté F_p
- la force de traînée noté F_t

La deuxième loi de Newton nous donne :

$$m \cdot \vec{a} + \frac{dm}{dt} \cdot \vec{v} = \vec{P} + \vec{F}_p + \vec{F}_t$$

où m est la masse, v la vitesse, et a l'accélération de la fusée.

Introduisons l'axe x comme l'axe de symétrie de la fusée. Développons les différentes forces :

$$\vec{P} = m \vec{g}$$

$$\vec{F}_t \cdot \vec{u}_x = \frac{1}{2} \rho S C_x (\vec{v} \cdot \vec{u}_x)^2$$

où ρ la masse volumique de l'air, S la surface projetée de la fusée sur le plan de norme $\vec{u}_x$. En projetant sur l'axe x , cela donne :

$$m(\vec{a} \cdot \vec{u}_x) + \frac{dm}{dt}(\vec{v} \cdot \vec{u}_x) = m(\vec{g} \cdot \vec{u}_x) + \vec{F}_p \cdot \vec{u}_x + \frac{1}{2} \rho S C_x (\vec{v} \cdot \vec{u}_x)^2$$

Considérons le cas après la phase de poussée. Durant cette phase, seule la force de traînée et le poids sont appliqués sur la fusée. La masse ne change pas au cours du temps. Nous avons donc :

$$m(\vec{a} \cdot \vec{u}_x) = m(\vec{g} \cdot \vec{u}_x) + \frac{1}{2} \rho S C_x (\vec{v} \cdot \vec{u}_x)^2$$

Soit :

$$C_x = 2m \frac{|a_x - g \cos(\theta)|}{\rho S v_x^2}$$

où θ est l'inclinaison de la fusée par rapport au sol. Durant cette phase, a_x , v_x vont être mesurés à l'aide de capteurs. Nous allons considérer que θ est connu en nous basant sur le fait que la fusée va suivre une trajectoire parabolique. La masse volumique de l'air ρ sera déterminée via les conditions météo le jour du lancement. Les valeurs S et m seront trouvées à l'aide de mesure sur la fusée avant le lancement. L'attraction gravitationnelle sera considérée comme connue et constante au cours du vol. Le cas pendant la poussée implique de prendre en considération le propulseur. Nous allons considérer que la variation de masse du propulseur est linéaire, que sa masse et sa poussée sont identiques à celles fournies dans sa datasheet.

$$C_x = \frac{2}{\rho S v_x^2} (m a_x + \frac{dm}{dt} v_x - m g_x - F_p)$$

Un capteur de force sera placé en haut du propulseur pour vérifier F_p en début de vol afin synchroniser la poussée théorique avec l'enregistrement.

4.3 Paramètres mesurés lors du vol

4.3.1 Accélération

L'accélération au cours du temps est mesurée par l'accéléromètre du *MPU-6050*. Ce capteur possède une plage d'acquisition de $+/- 16g$ avec une précision de 16 bits.

Cette précision est satisfaisante. Cependant, la borne positive de la plage d'acquisition est problématique car elle est inférieure à celle estimée par la trajectoire théorique de la fusée (voir figure 4). En effet, l'accélération maximale estimée est $183m/s^2$, ce qui est égal à environ $18,7g$, soit $2,7g$ supplémentaire par rapport à ce que permet le *MPU-6050*. Malgré cette contrainte, cet accéléromètre a tout de même été sélectionné car il est facilement interfaçable avec le microcontrôleur de l'ordinateur d'acquisition de données. De plus, ce capteur a souvent été utilisé par les membres de l'association et possède plusieurs documentations sur Internet.

Lors de sa phase de saturation, le capteur communiquera au microcontrôleur la valeur la plus proche, c'est-à-dire la borne positive de sa plage d'acquisition ($16g$). Cette phase de saturation intervient pendant la phase de poussée.

4.3.2 Poussée

La poussée au cours du temps est mesurée par le capteur de force *Tedea Huntleigh Model 00614-100K-C3-01X*. Ce capteur fonctionne est conçu pour mesurer une force jusqu'à $100kg$ avec une précision de 10 bits pour le microcontrôleur de la carte *Arduino UNO R3*. Sa valeur de non élasticité est de $150kg$ et il peut résister jusqu'à $300kg$.

La précision proposée par le microcontrôleur est relativement faible mais acceptable. Cependant, sa limite de mesure de $100kg$ est problématique car elle est inférieure à la poussée maximale du moteur de la campagne 2024 (voir figure 16). En effet, les documentations constructeurs montrent une poussée aux alentours de $1,2kN$ sur presque toute la durée de la phase de poussée (jusqu'à 1.2 secondes soit 70% de la phase de poussée). Nous avons décidé de ne pas modifier ce capteur malgré le changement de moteur intervenu entre 2023 et 2024. Le fait de garder le même capteur nous laisse supposer que la relation entre tension de sortie et force appliquée sur le capteur est identique au delà de $100kg$. Un capteur de la même gamme est cependant disponible pour une valeur de mesure maximale de $150kg$.

4.3.3 Vitesse

La vitesse au cours du temps est mesurée par deux sondes Pitot, chacune étant reliée à un capteur de pression différentielle (*MPX2010DP* et *MPX2050DP*). Le capteur *MPX2010DP* couvre la plage de $0kPa$ à $10kPa$. Quant au capteur *MPX2050DP*, il couvre la plage de $0kPa$ à $50kPa$.

La plage $0kPa$ à $10kPa$ est donc couverte par les deux capteurs, avec le *MPX2010DP* travaillant uniquement sur cette plage, ce qui peut réduire les incertitudes de mesures. La vitesse maximale estimée par la trajectoire théorique de la fusée (voir figure 4) est $262m/s$, soit une pression différentielle d'environ $43kPa$, pression par ailleurs couverte par le capteur *MPX2050DP*.

Pendant les phases où la vitesse sera comprise entre $0m/s$ et $125m/s$ environ (pression différentielle de $10kPa$), les deux capteurs communiquent la vitesse mesurée au microcontrôleur, avec une réduction d'incertitude potentielle pour le *MPX2010DP*.

Lors de la phase où la vitesse sera supérieure à environ $125m/s$, seul le *MPX2050DP* fournira la vitesse. Le *MPX2010DP* sera saturé et fournira la valeur de sa borne maximale étant $125m/s$.

4.4 Étalonnage

Pour l'accéléromètre du *MPU-6050*, plusieurs analyses en statique sont réalisées afin de vérifier la bonne orientation des axes par rapport à la fusée.

Le capteur de force *Tedea Huntleigh* est étalonné via une série de poids entre 0 et 8 kg dans une configuration désassemblée de la fusée. Les deux potentiomètres du module d'amplificateur opérationnel *AD620* ainsi qu'une lecture en directe de la tension en entrée d'ADC de la carte *Arduino UNO R3* permettent un étalonnage assez simple. Le potentiomètre par défaut gérant le gain est remplacé par un potentiomètre de plus faible valeur afin d'augmenter la précision dans la bonne plage de données. La valeur des deux potentiomètres sont étalonnées pour avoir *10LSB* à *0kg* et *710LSB* à *200kg*. Une saturation est observée sur la sortie de l'AOP à $\frac{3}{4}$ de V_{in} . L'alimentation de l'AOP en *5V* permet une lecture du microcontrôleur jusqu'à environ *770LSB*.

Les deux capteurs de vitesse sont étalonnés en utilisant la malle pression mise à disposition durant la campagne C'Space. L'étalonnage est similaire à celui du capteur de force. Pour le *MPX2010DP*, la valeur de sortie est fixée à *10LSB* pour *0mbar* et *700LSB* pour *100mbar*. De même pour le *MPX2050DP*, la valeur de sortie est fixée à *10LSB* pour *0mbar* et *630LSB* pour *350mbar*. Les estimations (voir figure 4) donnent une pression maximale légèrement inférieure de *350mbar*.

Des tests de linéarités ont été réalisés sur les deux capteurs de vitesse ainsi que le capteur de force avec des résultats très satisfaisant.

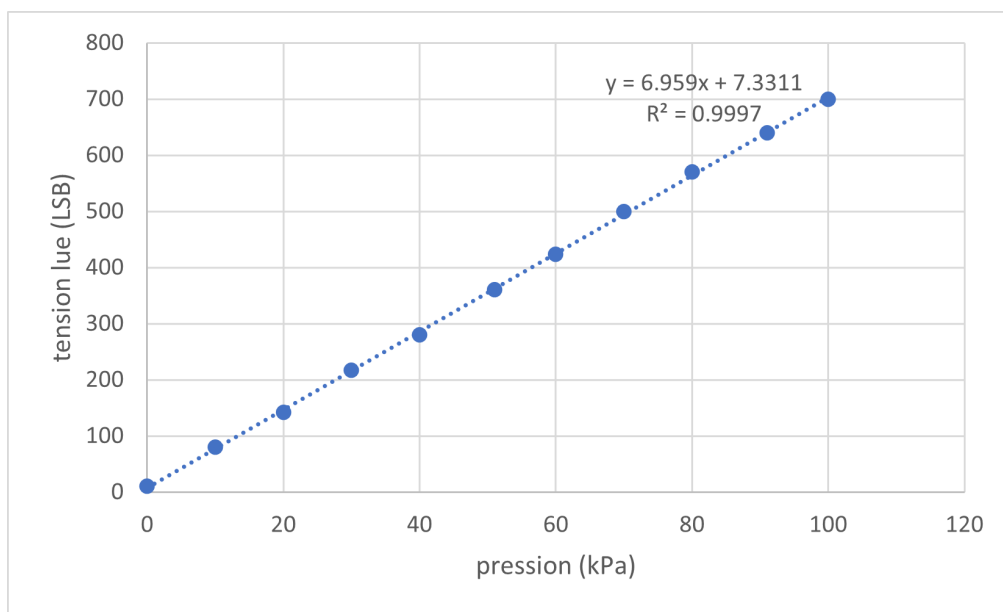


Figure 12: Étude de linéarité durant l'étalonnage du *MPX2010DP*

5 Chronologie

Zone	Action	Qui
base vie	Charger la lipo 3S à 12.8V et prendre 3 piles 9v neuves.	Alex / Pierre
base vie	Scotcher la carte SD.	Alex / Pierre
base vie	Intégrer le parachute, fermeture de la trappe.	Alex / Pierre
base vie	Placer les deux clés dans les deux interrupteurs en position off.	Alex / Pierre
base vie	Intégrer la lipo et les piles 9v dans le rack elec.	Alex / Pierre
base vie	Intégrer le rack elec dans la fusée.	Alex / Pierre
base vie	Fixer la protection pitot avec un morceau de scotch.	Alex / Pierre
base vie	Prendre en rampe : la clé allen 4 la fusée le jack la chrono le scotch orange les sacs poubelle la pince coupante un sac à dos de transport	Alex / Pierre
en rampe	Mettre la fusée en rampe	Alex
en rampe	Fixer le jack à la rampe, utiliser pince coupante si trop long.	Pierre
en rampe	Laisser du jeu.	Alex
en rampe	Fixer le jack à la fusée.	Alex
en rampe	Retirer la clé MIN, on s'attend à 1 premier bip, puis bip bip toutes les 4s.	Alex
en rampe	Retirer la clé XP.	Alex
en rampe	Retirer la protection pitot.	Alex
en rampe	Redresser la rampe.	Alex
en rampe	Laisser la clé allen de 4 en évidence avec le scotch sur la rampe.	Alex
en rampe	Laisser un peu de jeu dans la fixation propu.	Alex
en rampe	Fin des opérations en rampe, retour au pupitre.	Alex / Pierre

Table 3: Chronologie de préparation au lancement

6 Déroulement du vol

La fusée a décollé le 9 juillet 2024 à 15:21. Le ciel était couvert, Cobra a été perdue de vue durant le vol. Le parachute a été déployé à l'apogée.

La fusée a atterri dans le champ de tir et a subi quelques dégâts sur l'ogive. Un militaire s'est chargé de la récupération le 11 juillet en fin de journée.



Figure 13: Cobra en rampe avant le lancement

7 Analyse du vol

Les données de vol ont été récupérées depuis la carte SD embarquée. Nous avons remarqué la présence d'un deuxième enregistrement lors de l'ouverture de la carte. Cet enregistrement, daté du 11 juillet à 23h30, a dû être la conséquence du déchargement de la source d'alimentation de l'ordinateur d'acquisition de données. Aucun mouvement de la fusée n'est détecté sur l'enregistrement.

Nous avons estimé, au vu des conditions météorologiques, que la masse volumique de l'air était de $\rho = 1.18 \text{ kg/m}^3$. Nous considérons cette masse volumique comme constante durant le vol. L'attraction gravitationnelle est fixée à $g = 9.81 \text{ m/s}^2$ et est aussi considérée comme constante. L'inclinaison de la rampe est fixée à un angle de 80° .

7.1 Données des capteurs

Aucun post-traitement n'a été ajouté aux données présentées dans ce chapitre.

7.1.1 Capteurs de vitesse

La vitesse au cours du temps est calculée avec la masse volumique de l'air en utilisant la formule suivante :

$$v = \sqrt{\frac{2P}{\rho}}$$

La mesure de vitesse semble être cohérente avec les valeurs attendues.

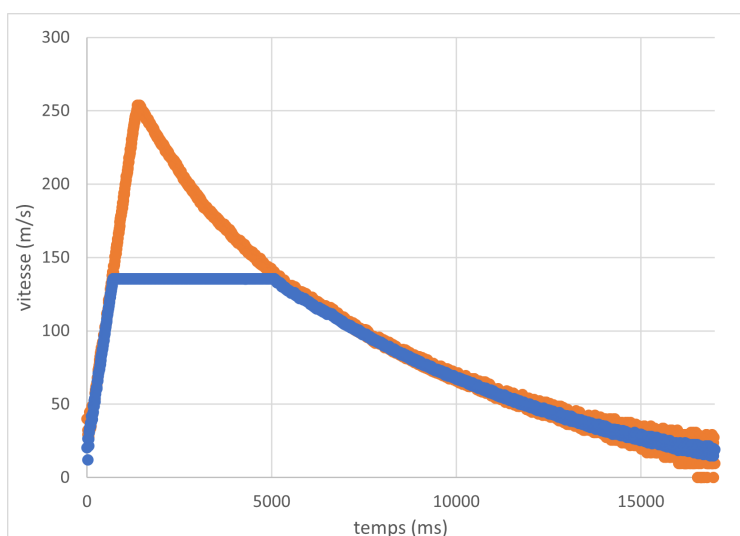


Figure 14: Mesure de la vitesse de vol via les capteurs *MPX2010DP* et *MPX2050DP*

Nous pouvons observer le seuil de saturation du *MPX2010DP* vers les 140 m/s . Un léger écart entre le *MPX2010DP* et le *MPX2050DP* est aussi présent lors de la désaturation, vers les 5 secondes de vol.

Dans l'ensemble, les deux sondes Pitot semblent avoir une estimation cohérente de la vitesse. Cette figure nous permet d'apprécier la précision du *MPX2010DP* à faible vitesse, comme ce qui était attendu.

Pour la suite, nous allons considérer que la vitesse de la fusée est fournie par le *MPX2010DP*. Durant la période de saturation, nous allons prendre les valeurs du *MPX2050DP*.

7.1.2 Capteur de force

Les mesures du capteur de force semblent malheureusement aberrantes lors de la phase de poussée. Nous nous attendions à avoir une maximale de l'ordre de $130kg$.

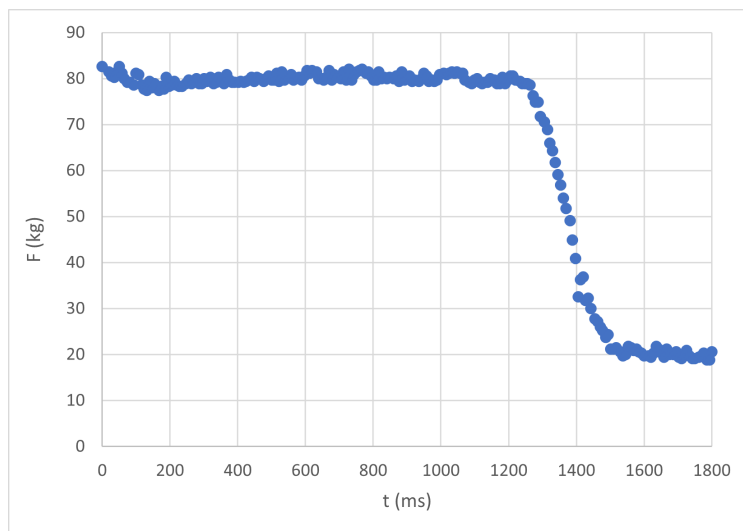


Figure 15: Mesure de la compression de la fusée

Nous pouvons cependant remarquer des similitudes d'amplitude avec les caractéristiques constructeurs du moteur.

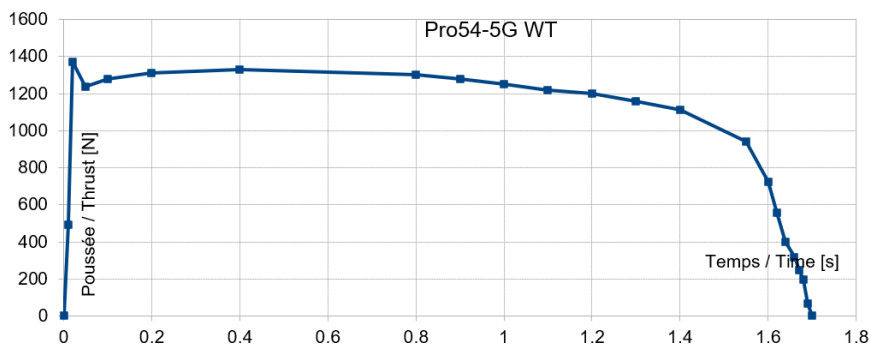


Figure 16: Caractéristiques techniques du moteur

La valeur finale de force doit correspondre à la pression exercée par la fixation du moteur. Cette valeur reste constante sur le reste de l'enregistrement.

7.1.3 Accéléromètre

Les données de l'accéléromètre permettent de détecter les différentes étapes de vols. Nous pouvons observer la phase de poussée durant les premières secondes. Malheureusement, l'accéléromètre choisi n'était pas dimensionné pour la force de poussée exercée par le propulseur utilisé. Nous pouvons donc observer une saturation du capteur à $16g$. Les capteurs proposant une borne positive supérieure à $16g$ sont moins accessibles au grand public en termes de prix et de facilité d'utilisation.

Le reste des données de vol durant la phase balistique sont exploitable pour la mesure du C_x . Nous pouvons observer l'instant de l'ouverture du parachute vers les 17 secondes de vol.

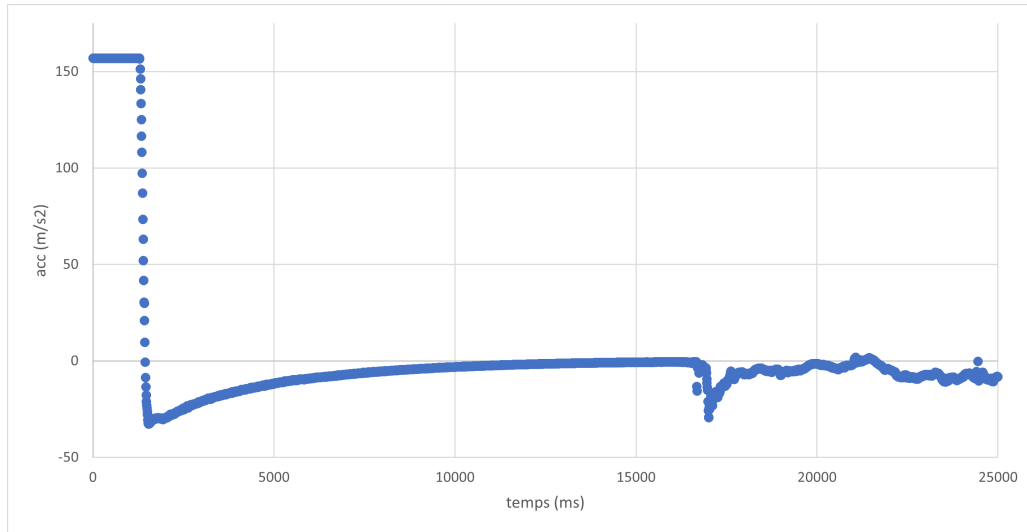


Figure 17: Mesure de l'accélération de la fusée

7.2 Résultats de l'expérience

Au vu de la saturation de l'accéléromètre durant la phase de poussée et de l'incohérence des données du capteur de force, seule la phase balistique du vol sera étudiée pour mesurer le C_x .

Reprenons la formule pour déterminer le C_x obtenu précédemment :

$$C_x = 2m \frac{|a_x - g \cos(\theta)|}{\rho S v_x^2}$$

Nous avons, $m = 6.11kg$ la masse de la fusée avec le propulseur vide, $S = 330cm^2$ la surface projetée de la fusée, $g = 9.81m/s^2$, $\rho = 1.18kg/m^3$

L'inclinaison de la rampe était de 80° . Nous considérons que $\cos(\theta)$ varie linéairement de $\cos(80^\circ)$ à 0 du décollage à l'apogée. Nous avons fixé le temps d'atteinte de l'apogée à $16.5s$.

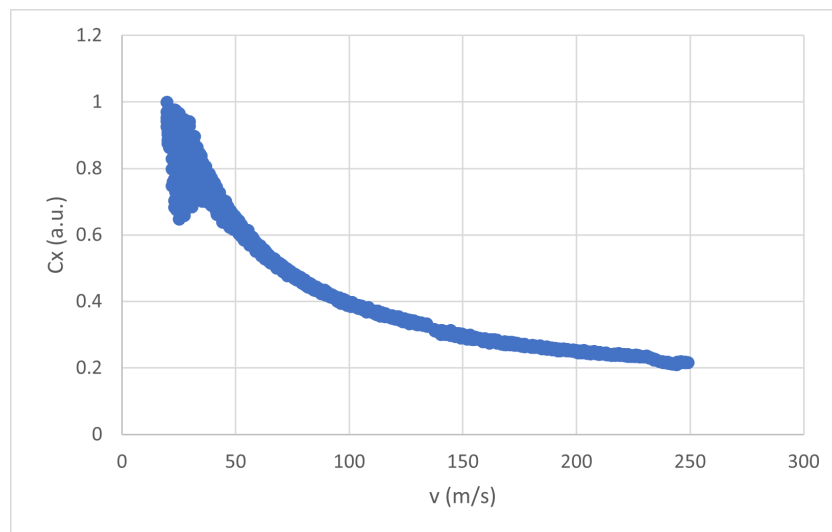


Figure 18: Mesure du coefficient de traînée de Cobra en fonction de la vitesse

7.3 Interprétation des données

Les valeurs de C_x obtenues correspondent à l'ordre de grandeur attendue. Nous pouvons observer une diminution du C_x en fonction de la vitesse. Cette diminution est là encore cohérente. Nous observons que la diminution en précision des sondes Pitot impactent grandement la mesure du C_x à faible vitesse.

7.4 Axes d'amélioration

Différents axes peuvent être améliorés au vu des résultats :

- Une simulation du C_x à l'aide de la modélisation 3D de la fusée pourra être réalisée afin de croiser les résultats avec les données de vol
- La mesure de l'inclinaison de la fusée aurait pu se faire à l'aide d'un capteur afin de confirmer l'hypothèse d'inclinaison prise dans nos calculs
- L'accéléromètre *MPU-6050* et le capteur de force *Tedea Huntleigh* n'étaient pas adaptés pour les mesures en phase de poussée. Des capteurs avec de plus grandes plages de travail sont nécessaires afin d'éviter la saturation
- L'utilisation de capteurs d'une meilleure précision ainsi qu'un microcontrôleur avec un ADC plus précis pourraient nettement améliorer la qualité des résultats
- Une étude de frottement entre le propulseur et les bagues de centrage de la fusée pourra amener à une explication de la faible compression mesurée en phase de poussée. Un étalonnage post-vol pourra permettre de vérifier le bon étalonnage du capteur
- Un enregistrement une seconde avant le décollage aurait pu être appréciable pour avoir les conditions initiales des capteurs
- L'encodage des données de mesures dans des types appropriés avec une taille au juste besoin permettrait d'optimiser le temps de traitement du microcontrôleur, ainsi que de diminuer le temps d'enregistrement sur la carte SD
- L'intégration d'une télémessure dans l'avionique de la fusée permettrait de garantir la réception des données, quelque soit la finalité du vol (nominal ou balistique)

8 Conclusion

Ce projet nous a permis de mieux comprendre les mécanismes de traînée durant le vol. Nous avons pu monter en compétences sur l'utilisation des capteurs. Nous espérons que cette expérience permettra à l'association d'améliorer ses performances lors de compétitions de visée d'altitude.

L'équipe Cobra aimerait spécialement remercier Planète Sciences, le CNES ainsi que tous les bénévoles pour l'organisation de cette édition 2024 du C'Space.

Un grand merci aux sponsors de LéoFly pour leur soutien.



Figure 19: L'équipe accompagnée d'un bénévole après la récupération de Cobra

9 Annexes

9.1 Références des composants électroniques

Nom	Type	Fournisseur	Référence
<i>Arduino UNO R3</i>	Carte d'extension de microcontrôleur	Arduino	7630049200050
<i>Sonitron SMA-24L</i>	Buzzer	RS	178-4024
<i>Intertec ITS-PE3529-12VDC</i>	Electro-aimant	Conrad	506163 - 62
<i>Finder 36 Series</i>	Relais mécanique de puissance	RS	793-3768
<i>MPU-6050</i>	Module température/accéléromètre/gyroscope	Amazon	B07Y9KL5Q1
<i>MPX2010DP</i>	Capteur de pression différentielle	RS	719-1021
<i>MPX2050DP</i>	Capteur de pression différentielle	RS	922-7311
<i>Tedea Huntleigh AD620</i>	Capteur de force	RS	185-5959
<i>AD620</i>	Module AOP	Amazon	B086YNCTF3
<i>AZ-DELIVERY Data Logger Shield</i>	Enregistreur Carte SD	Amazon	B01NATCT7N
<i>Interrupteur à clé DPST 4 A</i>	Interrupteur à clés 2 positions	RS	863-7026
<i>BMS 3S 12.6V 20A</i>	BMS 3S 12.6V 20A	Amazon	B08FT59NMG

Table 4: Références des composants électroniques

9.2 Code des logiciels embarqués

```
1 int BROCHE_BUZZER = 12;
2 int BROCHE_ELEKTROAIMANT = 11;
3 int BROCHE_LECTURE_JACK = 7;
4 int BROCHE_MASSE = 2;
5
6 int LI_ms = 150;
7 int TR_ms = 1000;
8
9 int duree_vol_ms = 16000;
10
11 void setup() {
12   pinMode(BROCHE_ELEKTROAIMANT, OUTPUT);
13   pinMode(BROCHE_LECTURE_JACK, INPUT);
14   pinMode(BROCHE_MASSE, OUTPUT);
15   pinMode(BROCHE_BUZZER, OUTPUT);
16
17   digitalWrite(BROCHE_BUZZER, LOW);
18   digitalWrite(BROCHE_ELEKTROAIMANT, LOW);
19   digitalWrite(BROCHE_MASSE, LOW);
20   delay(100);
21 }
22
23 void loop() {
24   digitalWrite(BROCHE_MASSE, LOW);
25   long last_update_ms = millis();
26   long time_ms;
27   long instant_decollage_ms;
28   int duree_pulse_ms = 1000;
29   bool etat_buzzer = false;
30   delay(1000);
31   while(digitalRead(BROCHE_LECTURE_JACK) == HIGH){
32     time_ms = millis();
33     if(etat_buzzer){
34       if(time_ms - last_update_ms > 100){
35         last_update_ms = time_ms;
36         etat_buzzer = false;
37         digitalWrite(BROCHE_BUZZER, LOW);
38       }
39     }
40     else{
41       if(time_ms - last_update_ms > duree_pulse_ms){
42         last_update_ms = time_ms;
43         etat_buzzer = true;
44         digitalWrite(BROCHE_BUZZER, HIGH);
45         if(duree_pulse_ms == 4000){
46           duree_pulse_ms = 100;
47         }
48         else{
49           duree_pulse_ms = 4000;
50         }
51       }
52     }
53
54     delay(10);
55   }
56
57   // decollage
58   instant_decollage_ms = millis();
59
```



```

60  time_ms = millis();
61  duree_pulse_ms = TR_ms;
62
63  etat_buzzer = false;
64  while(time_ms - instant_decollage_ms < duree_vol_ms){
65      time_ms = millis();
66      if(etat_buzzer){
67          if(time_ms - last_update_ms > LI_ms){
68              etat_buzzer = false;
69              digitalWrite(BROCHE_BUZZER, LOW);
70          }
71      }
72      else{
73          if(time_ms - last_update_ms > duree_pulse_ms){
74              last_update_ms = time_ms;
75              etat_buzzer = true;
76              digitalWrite(BROCHE_BUZZER, HIGH);
77              duree_pulse_ms = int((TR_ms - LI_ms)*(1 - float(time_ms -
instant_decollage_ms)/duree_vol_ms)) + LI_ms;
78          }
79      }
80
81      delay(10);
82  }
83
84  digitalWrite(BROCHE_BUZZER, LOW);
85
86  // ouverture para
87  digitalWrite(BROCHE_ELEKTROAIMANT, HIGH);
88  delay(60000);
89  digitalWrite(BROCHE_ELEKTROAIMANT, LOW);
90
91  while(1){
92      digitalWrite(BROCHE_BUZZER, HIGH);
93      delay(100);
94      digitalWrite(BROCHE_BUZZER, LOW);
95      delay(10000);
96  }
97 }

```

Listing 1: Code Arduino de l'ordinateur de bord

```

1  #include <Adafruit_MPU6050.h>
2  #include <Adafruit_Sensor.h>
3  #include <Wire.h>
4  #include <List.hpp>
5
6  #include <SPI.h>
7  #include <SD.h>
8
9  #include "RTClib.h"
10
11  Adafruit_MPU6050 mpu;
12
13  RTC_DS1307 rtc;
14
15  File myFile;
16
17  long duree_enregistrement_ms = 25000;
18
19  const int PIN_SD = 10;

```

```
20 const int BROCHE_LECTURE_JACK = 7;
21 const int BROCHE_LECTURE_CAPTEUR_FORCE = A0;
22 const int BROCHE_LECTURE_CAPTEUR_PRESSION_MIN = A1;
23 const int BROCHE_LECTURE_CAPTEUR_PRESSION_MAX = A2;
24
25 void setup() {
26   Serial.begin(9600);
27   Serial.println("initialisation");
28
29   // Try to initialize MPU6050
30   if (!mpu.begin(0x69)) {
31     Serial.println("Failed to find MPU6050 chip");
32     while (1) {
33       delay(10);
34     }
35   }
36   Serial.println("MPU6050 Found");
37
38   mpu.setAccelerometerRange(MPU6050_RANGE_16_G);
39   mpu.setGyroRange(MPU6050_RANGE_500_DEG);
40   mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);
41
42
43   // Try to initialize SD card
44   Serial.print("Initializing SD card...");
45
46   if (!SD.begin(PIN_SD)) {
47     Serial.println("initialization failed!");
48     while (1);
49   }
50   Serial.println("initialization done.");
51
52   // Try to initialize RTC
53   rtc.begin();
54
55   if (!rtc.isrunning()) {
56     Serial.println("RTC is NOT running!");
57     // following line sets the RTC to the date & time this sketch was compiled
58     rtc.adjust(DateTime(__DATE__, __TIME__));
59   } else {
60     Serial.println("RTC is running");
61   }
62
63   // create data file
64   DateTime now = rtc.now();
65
66   // open a new file and immediately close it:
67   Serial.println("creating .txt...");
68   myFile = SD.open(String(now.day()) + String(now.hour()) + String(now.minute()) + ".txt", FILE_WRITE);
69
70   // initialisation pin
71
72   pinMode(BROCHE_LECTURE_JACK, INPUT);
73   pinMode(BROCHE_LECTURE_CAPTEUR_FORCE, INPUT);
74   pinMode(BROCHE_LECTURE_CAPTEUR_PRESSION_MIN, INPUT);
75   pinMode(BROCHE_LECTURE_CAPTEUR_PRESSION_MAX, INPUT);
76 }
77
78
79
```

```
80 void loop() {
81
82     long t_decollage_ms;
83     sensors_event_t a, g, temp;
84     long t_ms;
85
86     Serial.println("attente");
87
88     while(digitalRead(BROCHE_LECTURE_JACK) == HIGH){
89         delay(10);
90     }
91
92     // decollage
93
94     Serial.println("decollage");
95
96     t_decollage_ms = millis();
97
98     do{
99         t_ms = millis();
100         mpu.getEvent(&a, &g, &temp);
101
102         myFile.print(t_ms - t_decollage_ms);
103         myFile.print(",");
104         myFile.print(a.acceleration.x);
105         myFile.print(",");
106         myFile.print(a.acceleration.y);
107         myFile.print(",");
108         myFile.print(a.acceleration.z);
109         myFile.print(",");
110         myFile.print(g.gyro.x);
111         myFile.print(",");
112         myFile.print(g.gyro.y);
113         myFile.print(",");
114         myFile.print(g.gyro.z);
115         myFile.print(",");
116         myFile.print(analogRead(BROCHE_LECTURE_CAPTEUR_FORCE));
117         myFile.print(",");
118         myFile.print(analogRead(BROCHE_LECTURE_CAPTEUR_PRESSION_MIN));
119         myFile.print(",");
120         myFile.print(analogRead(BROCHE_LECTURE_CAPTEUR_PRESSION_MAX));
121         myFile.print(",");
122         myFile.print(temp.temperature);
123         myFile.print("\n");
124     }while(millis() - t_decollage_ms < duree_enregistrement_ms);
125
126     myFile.close();
127
128     Serial.println("fin");
129
130     while(1){
131         delay(10000);
132     }
133 }
```

Listing 2: Code Arduino de l'ordinateur d'acquisition de données

9.3 Plan de câblage du shield AOP de l'ordinateur d'acquisition de données

Etant donné que les trois modules d'amplificateur opérationnel *AD620* sont assez volumineux, un shield a été spécialement dédié pour eux. Ci-dessous se trouve le schéma électronique du shield.

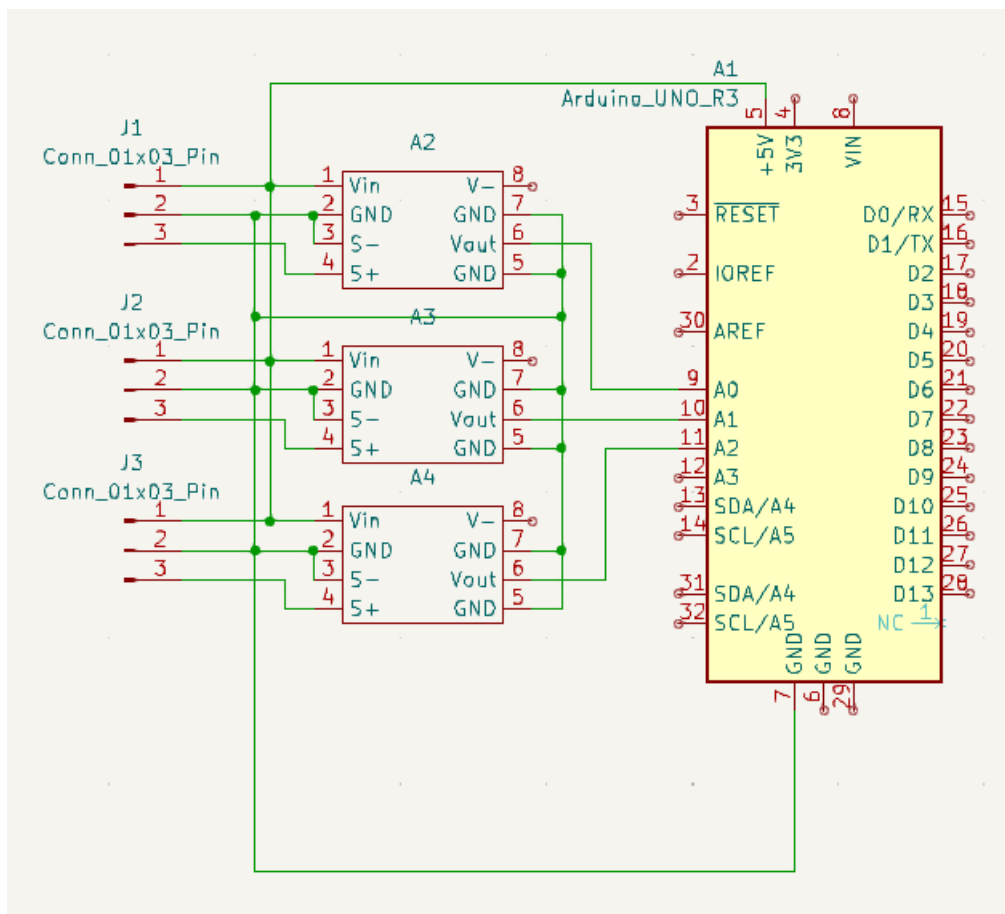


Figure 20: Plan de câblage du shield AOP de l'ordinateur d'acquisition de données

9.4 Plage du niveau d'intensité sonore du buzzer *Sonitron SMA-24L*

Le buzzer utilisé est un *Sonitron SMA-24L*. Il faut donc observer la courbe désignée par la lettre "L".
Cette courbe est issue de la datasheet du *Sonitron SMA-24L* fournie par *Sonitron*.

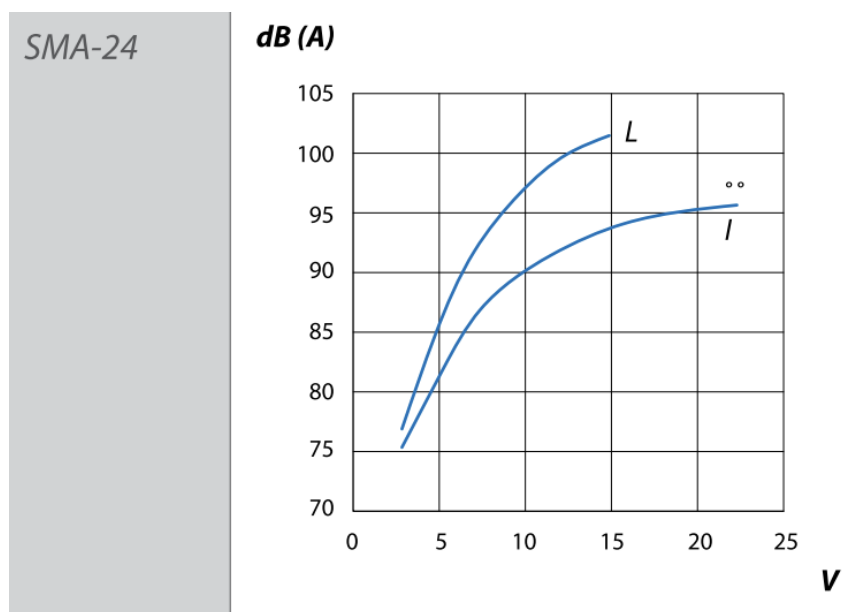


Figure 21: Niveau d'intensité sonore par rapport à la tension d'entrée du buzzer *Sonitron SMA-24L*