

Rapport de projet : Minifusée NAÉROBI



Club : Eclipse

Établissement : Lycée LA TRINITÉ

Campagne : C'SPACE 2024

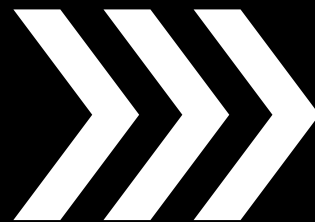
Année : 2023-2024

Matricule : MF44



SOMMAIRE :

- 1- Le club Eclipse
- 2- Présentation du projet Naérobi
- 3- La mécanique du projet Naérobi
- 4- L'électronique du projet Naérobi
- 5- Le C'SPACE 2024
- 6- Le vol
- 7- L'analyse des données & résultats
- 8- Conclusion
- 9- Remerciements



I – Le club Eclipse

Fondé en 2022 par un petit groupe de lycéens, le club Eclipse a pour objectif principal de réaliser des projets scientifiques comme la fabrication de minifusées et de ballons sondes. Le club a eu la chance de participer au C'Space 2023 et de faire voler deux minifusées (Apogée et Δ'Ship).

Cette année, le club comptabilise 21 participants, de la seconde à la terminale, répartis sur les différents projets et encadrés par deux professeurs référents.

Nous avons pu constituer un budget à partir des fonds qui nous ont été alloués par l'association des parents d'élèves du lycée mais aussi au niveau national. Nous l'avons principalement dépensé dans l'achat d'une imprimante 3D, d'une Dremel et de l'outillage en tout genre car notre lycée ne possédait pas le matériel nécessaire.

Liste des membres Eclipse 2023-2024 :

Bourbon Marie-Angéline
Boudet Maximilien
Casares Mathis
D'Antoni Sarah
Duchesne Simon
Dupuy-Rosso Emma
Durand Elie
Gabanyi Bence
Georges Edouard
Gibbins Zack
Guiraud Jérémie
Kerridge Poppy (chef de projet – Naérobi)
Larroche Marien
Lunel Arthur
Mayer Emma
Molina Annette
Pasturel Romain (chef de projet – Arès)
Quemeneur Juliette
Soares Bastien
Vicaire Enzo
Zorzi Simon

II – Présentation du projet Naérobi

Le projet Naérobi est l'une de nos deux minifusées réalisées cette année. Ce projet est similaire en plusieurs points à notre seconde minifusée Arès, qui sera plusieurs fois évoquée dans ce rapport.

Le C'Space 2023 nous a beaucoup enrichi et nous a permis d'avoir une première expérience dans la conception de minifusées. Nos deux vols se sont soldés par des vols balistiques (une torche pour Apogée et une ouverture prématurée de la trappe contenant le parachute due à un faux-contact au niveau du jack pour Δ Ship). Nous avons donc à cœur de réaliser cette année au moins un vol nominal. Les projets Arès et Naérobi sont donc dans la continuité de nos précédents projets.

Notre minifusée Naérobi possède un intérêt expérimental. En effet, nous avons ajouté un accéléromètre afin d'obtenir l'accélération de la fusée ainsi qu'une caméra embarquée pour filmer le vol. De plus, cette minifusée embarque elle aussi un puissant buzzer. Celui-ci émet un son à différente fréquence en fonction de l'accélération : plus l'accélération est importante, plus l'intervalle de temps entre deux sons est faible. Cette expérience sonore a pour but d'illustrer de façon originale l'évolution de l'accélération de la minifusée.

III – La mécanique du projet Naérobi

A) Matériaux

Grâce à un budget plus conséquent que l'an dernier, nous avons eu plus de choix en termes de matériaux. Nous recherchions des matériaux légers, résistants et simples à façonner.

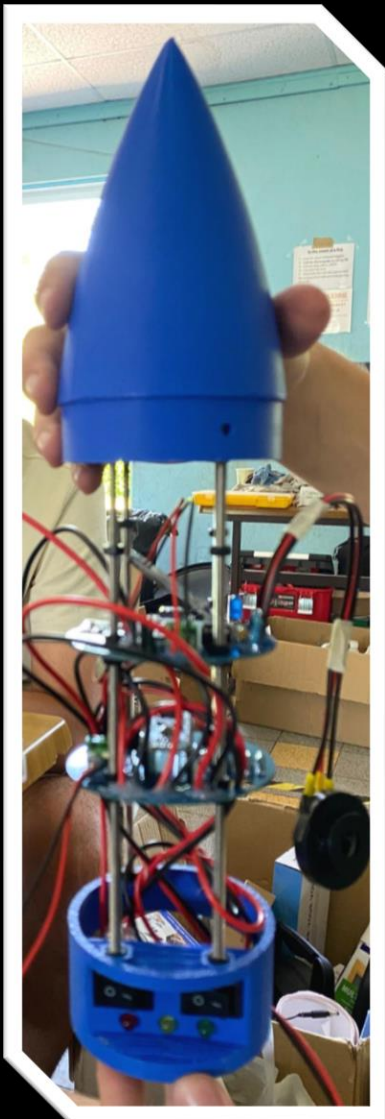
Nous avons choisi pour le corps de la fusée un tube de PVC de diamètre 80mm et d'épaisseur 1,5mm. Plusieurs éléments circulaires comme les bagues de retenue moteur, la commande d'allumage et l'épaulement de la coiffe permettent au tube de garder une forme circulaire car il avait tendance à se déformer à cause de sa faible épaisseur.

Pour les ailerons, nous avons choisi du contreplaqué 5mm. Nous avons découpé les ailerons à l'aide d'une scie à onglet.

Concernant les autres pièces de la fusée que sont la coiffe, la commande d'allumage, la trappe et sa porte ainsi que le support jack et les bagues moteurs, nous les avons conçues à l'aide d'un logiciel de CAO et nous les avons usinées à l'aide de notre imprimante 3D. Nous les avons tout d'abord prototypées en PLA puis nous avons opté pour du PETG pour les pièces finales, un filament qui résiste mieux à la température et qui est plus résistant.

B) CAO et design de la fusée

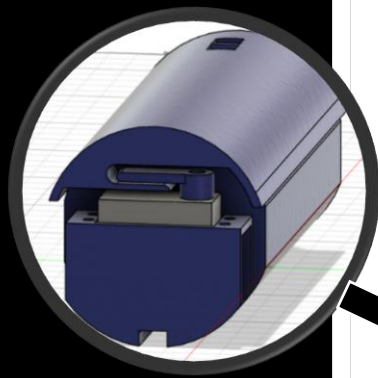
Pour la conception mécanique de la minifusée, nous avons réutilisés des méthodes que nous avons employés l'an dernier. Le tube de la fusée est une peau porteuse et les éléments sont fixés à l'aide d'époxy (sauf pour la coiffe et la commande d'allumage, reliées entre elles par des tiges filetées qui servent aussi à maintenir l'électronique en place dans la fusée)



Le bloc électronique est composé de deux PCB fixés grâce à des tiges filetées et des écrous. Les tiges sont fixées dans la coiffe ainsi que sur la commande d'allumage.

La commande d'allumage est composée de deux interrupteurs (séquenceur et système expérience) et de trois LED témoin d'état : la LED verte signifie que le séquenceur est allumé et est en attente du décollage, la LED rouge signifie que le jack est débranché et que la minuterie du séquenceur est enclenchée et la LED jaune signifie que le servomoteur a libéré la porte de la trappe.

Coiffe : Elle permet de stocker les deux piles 9V qui alimentent le séquenceur et la carte expérience. Cela permet également de rehausser le centre de masse de la fusée.

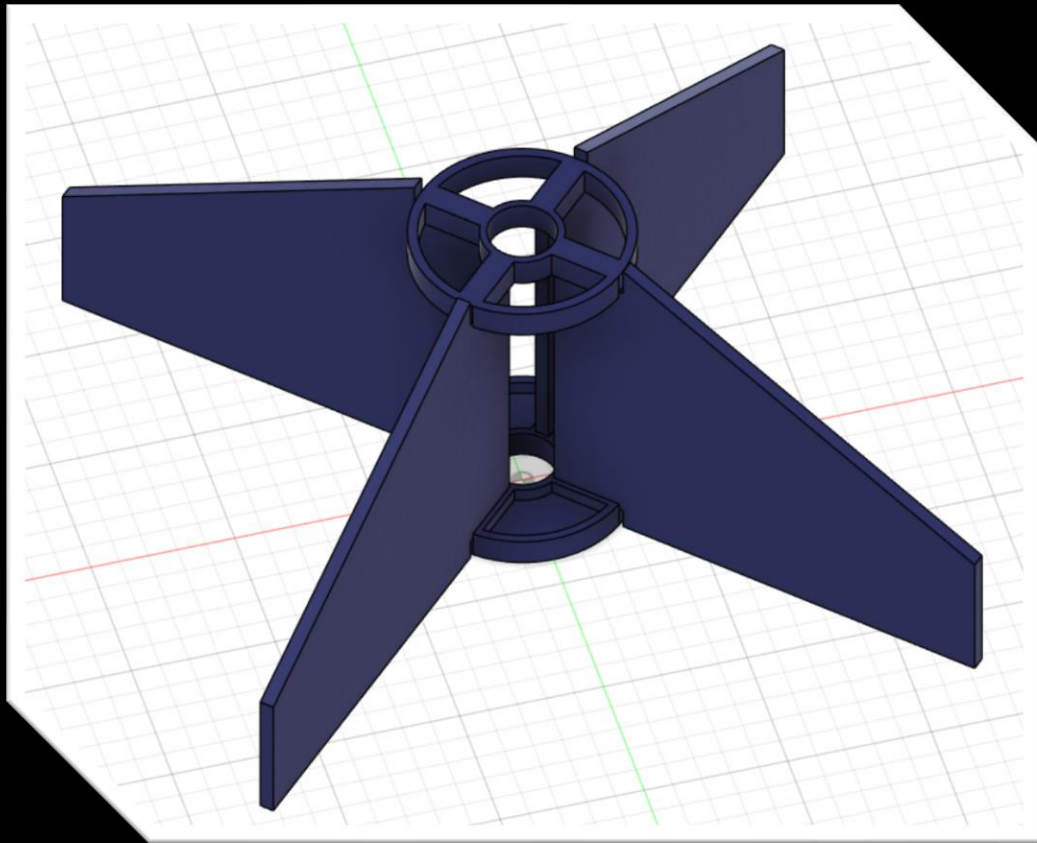


Trappe parachute : ce bloc est constitué de plusieurs pièces dont l'attache parachute (barre visible sur la bulle de droite). Un ressort (non visible ici) fait pression sur la porte de la trappe afin que l'éjection soit franche. Tout ceci est retenu par un servo moteur fixé à la trappe.



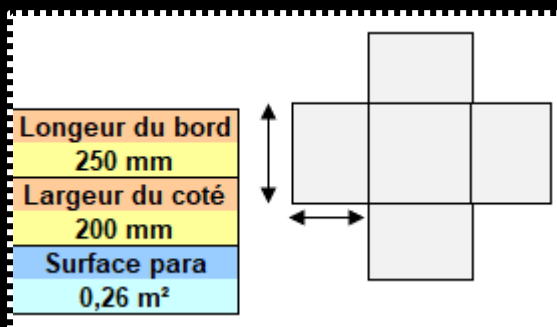
Support Jack et support caméra : ces pièces collées au corps de la fusée permettent de maintenir en place la partie femelle du jack ainsi que la caméra embarquée (fixée de l'autre côté).





Les ailerons sont encastrés dans le corps de la fusée et alignés grâce à des fentes dans les bagues moteurs. L'ensemble est collé avec de l'époxy.

C) Parachute



Le modèle de parachute que nous avons choisi est le parachute en forme de croix pour faciliter sa réalisation. Nous avons utilisé de la toile de parachute et du fil de couture résistant pour le confectionner.

8 cordes reliées à un émerillon forment les suspentes du parachute. L'émerillon fait ensuite le lien à la fusée à l'aide d'une corde du même type que celle des suspentes.



IV – L'électronique du projet NAEROBI

Le système électrique de NAEROBI se divise en 2 parties :

le système « primaire » et le système « secondaire ».

1. Le système primaire

Le système primaire a pour objectif principal d'assurer l'ouverture de la trappe du parachute, mais également l'activation des LEDs et buzzers qui communiquent son état actuel, afin que nous ou les pyrotechniciens puissions intervenir en cas de mauvais fonctionnement avant le décollage.

Le fonctionnement de la trappe :

Pour contrôler l'ouverture et la fermeture de la trappe du parachute, nous avons mis en place un système de loquet attaché à un servomoteur, pouvant ainsi bloquer ou débloquent la trappe selon sa position.

Déclenchement de l'ouverture avec le Jack :

Une fois le code du servomoteur étant fonctionnel, nous avons rajouté un système de Jack modifié, qui agirait comme un capteur déclenchant un compte à rebours pour l'ouverture de la trappe une fois débranché.

Lors du décollage de la fusée, une extrémité du Jack reste sur la fusée, tandis que l'autre reste attaché à la cage de lancement, le débranchant.



➔ *Vue de près sur Arès avec jack branché*



Déroulement de l'ouverture (capteurs et signaux) :

La totalité du système primaire peut être allumé avec un interrupteur intégré sur le côté de la fusée.

1. Une fois le système allumé, il est autonome et attend le signal du Jack.
 - Pendant ce temps, une **LED verte** clignote doucement ainsi qu'un buzzer pour nous indiquer que le système est sous tension en attente du débranchement du jack.
2. Lors du décollage le Jack se débranche, le compte à rebours commence alors.
(la durée celui-ci est calculée en fonction de l'apogée de la fusée).
 - Une **LED jaune** clignotera maintenant rapidement et le buzzer s'accélère pour nous indiquer que la trappe va s'ouvrir.

Ceci est important car si on entend ce changement de vitesse lorsque la fusée est au sol, cela nous indique qu'il y a un problème et nous ne devons pas lancer la fusée.

3. Enfin, une fois le compte à rebours fini, la trappe s'ouvre et le parachute se déploie.
 - Une **LED rouge** s'allume alors et le buzzer émet un son continu.

2. Le système secondaire

Le système secondaire est le système contenant les expériences.

Nous avons décidé de mettre un accéléromètre afin d'avoir une idée sur les accélérations subies par notre fusée sur 3 axes. Les données de celui-ci étaient traitées par le microcontrôleur Nano Arduino et stockées dans une carte SD.

Nous avons également mis à bord un Buzzer puissant directement lié à l'accéléromètre, qui émettrait des sons « bips » rythmés en fonction de l'accélération. Plus l'accélération augmente, plus le clignotement du buzzer est rapide. Tout ceci était accompagné d'un micro au niveau de la tente pour enregistrer les résultats du Buzzer à bord de la fusée.

De plus nous avons installé une caméra dans la fusée (indépendante du système électronique) qui a filmé le vol (visuel et audio), ainsi qu'une caméra au sol pour obtenir un deuxième angle du décollage.

Tout ceci nous a permis d'obtenir : non seulement les vidéos du vol, mais les fichiers audios qui étaient très intéressants car en plus du Buzzer puissant, nous pouvons remarquer les clignotements prévus du buzzer appartenant au système primaire.

3. Les schémas et codes

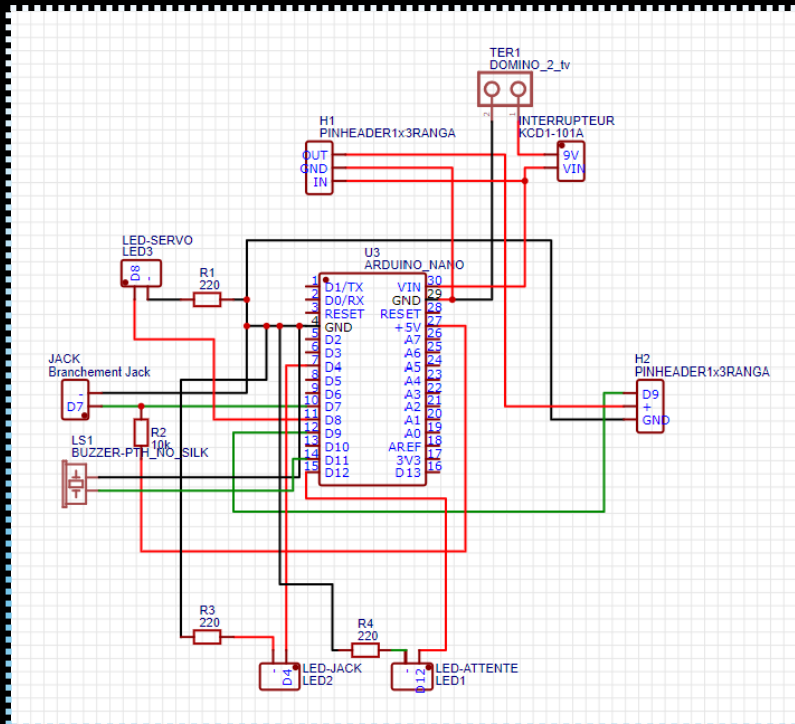
Nous avons préparé de nombreux codes au cours de ce projet jusqu'à arriver aux codes finaux de système primaire (identique pour les deux fusées) et systèmes secondaires indépendants. Ainsi que des codes pour tester les LEDs, les différents buzzers et le bon fonctionnement du Jack.

- Voici le code (Arduino) du système primaire ainsi que son schéma correspondant :

```
Systeme_primaire_NAEROBI_check3

1 //-----INFO LED-----
2 //LED rouge - jack débranché - 12
3 //LED jaune - SERVO P2 - 4
4 //LED verte - ATTENTE - 8
5
6 //
7 //CODE WITH EFFICIENT LED BLINKING + BUZZER + SERVO + JACK --> all good
8 //
9
10
11 // !!! ADD REAL DELAY TIME !!!
12 // change LEDs
13
14
15 //-----SERVO-----
16
17 /*
18 Servo Motor Control using the Arduino Servo Library
19 by Dejes, https://hackemchatemage.com
20 */
21
22 #include <Servo.h>
23
24 Servo myservo; // create servo object to control a servo
25
26
27 //-----REAL PARACHUTE DELAY-----
28 int delayApogee = 6400; // temps pour arriver à l'Apogée en ms
29
30
31 //-----SERVO ANGLES-----
32 int ServoP1 = 50; // angle servo en position 1
33 int ServoP2 = 150; // angle servo en position 2
34
35 int JACK = 7;
36
37
38 //LED + BUZZER + DELAY variables
39 unsigned long delayStartTime;
40 unsigned long blinkStartTime;
41 unsigned long buzzerStartTime;
42 bool delayActive = false;
43 bool ledState = false;
44 bool buzzerState = false;
45 const unsigned long delayDuration = delayApogee; // delay entre lancement et parachute (!!!!!) à changer selon les prédictions
46 const unsigned long blinkInterval = 100; // delay de clignotement
47 const unsigned long buzzerInterval1 = 500; // delay de buzzer clignot SLOW
48 const unsigned long buzzerInterval2 = 100; // delay de buzzer clignot FAST
49
50 //-----
51
52 void setup() {
53
54 //-----SERVO SETUP-----
55 myservo.attach(9, 600, 2300); // (pin, min, max)
56 myservo.write(ServoP1); // Position P1 SERVO trappe fermée
57
58 //-----LED SETUP-----
59 pinMode(4, OUTPUT); // SET LED as output
60 pinMode(12, OUTPUT); // SET LED as output
61 pinMode(8, OUTPUT); // SET LED as output
62
63 digitalWrite(12, LOW); // LED n°1 - ATTENTE (system is ready and waiting)
64 digitalWrite(4, LOW); // LED n°2 - JACK (off if JACK in, blink cycle if JACK out) - rouge clignote
65 digitalWrite(8, HIGH); // LED n°3 - SERVO (to be added) - verte
66
67 //-----JACK SETUP-----
68 pinMode(JACK, INPUT);
69 Serial.begin(9600);
70
71 //-----BUZZER SETUP-----
72 pinMode(11, OUTPUT); // SET BUZZER as output
73
74 }
75
76
77 //-----Handle delay and Action-----
78 if (delayActive) { //----- timer has started
79
80 if (currentMillis - delayStartTime >= delayDuration) { // function for when delay is passed
81 digitalWrite(4, HIGH);
82 // delayActive = false; //---> timer has ended
83
84 //-----SERVO activation-----
85 myservo.write(ServoP2); // Position P2 SERVO trappe ouverte
86 //-----LED 1 stuck for end-----
87 digitalWrite(8, HIGH); //LED n°1 for JACK Timer stays on, because timer is over
88 //-----BUZZER stuck for end-----
89 digitalWrite(11, HIGH); //Buzzer stays on, because timer is over
90 }
91
92 else { //-----while we're still in timer-----
93 //-----LED Blinking-----
94 if (currentMillis - blinkStartTime >= blinkInterval) { //check toggle timer for when over interval
95 blinkStartTime = currentMillis; // LED 'last toggle' timer resets
96 ledState = !ledState; //toggle LED state
97 digitalWrite(12, ledState ? HIGH : LOW); // Set LED based on ledState
98
99 //-----BUZZER FAST BLINK-----
100 if (currentMillis - buzzerStartTime >= buzzerInterval2) { //check toggle timer for when over interval
101 buzzerStartTime = currentMillis; // buzzer 'last toggle' timer resets
102 buzzerState = !buzzerState; //toggle buzzer state
103 digitalWrite(11, buzzerState ? HIGH : LOW); // Set buzzer based on buzzerState
104 }
105 }
106 }
107 }
108 }
109
110 //-----BUZZER SLOW BLINK-----
111 if (currentMillis - buzzerStartTime >= buzzerInterval1) { //check toggle timer for when over interval
112 buzzerStartTime = currentMillis; // buzzer 'last toggle' timer resets
113 buzzerState = !buzzerState; //toggle buzzer state
114 digitalWrite(11, buzzerState ? HIGH : LOW); // Set buzzer based on buzzerState
115 }
116
117 //-----JACK CONDITION-----
118 int JACKState = digitalRead(JACK); // prep JACK condition
119 if (JACKState == HIGH) { // JACK condition detected
120
121 if (!delayActive) { // Only set delayStartTime once when delayActive is false
122 delayStartTime = currentMillis; // Record start time of delay
123 delayActive = true; // The timer has started
124 }
125
126 else {
127 delayActive = false;
128 }
129 }
130 }
```

Le code est un peu long car nous devons créer des conditions par rapport au débranchement du jack, et rajouter diverses boucles pour les différents clignotements des LEDs et du Buzzer.

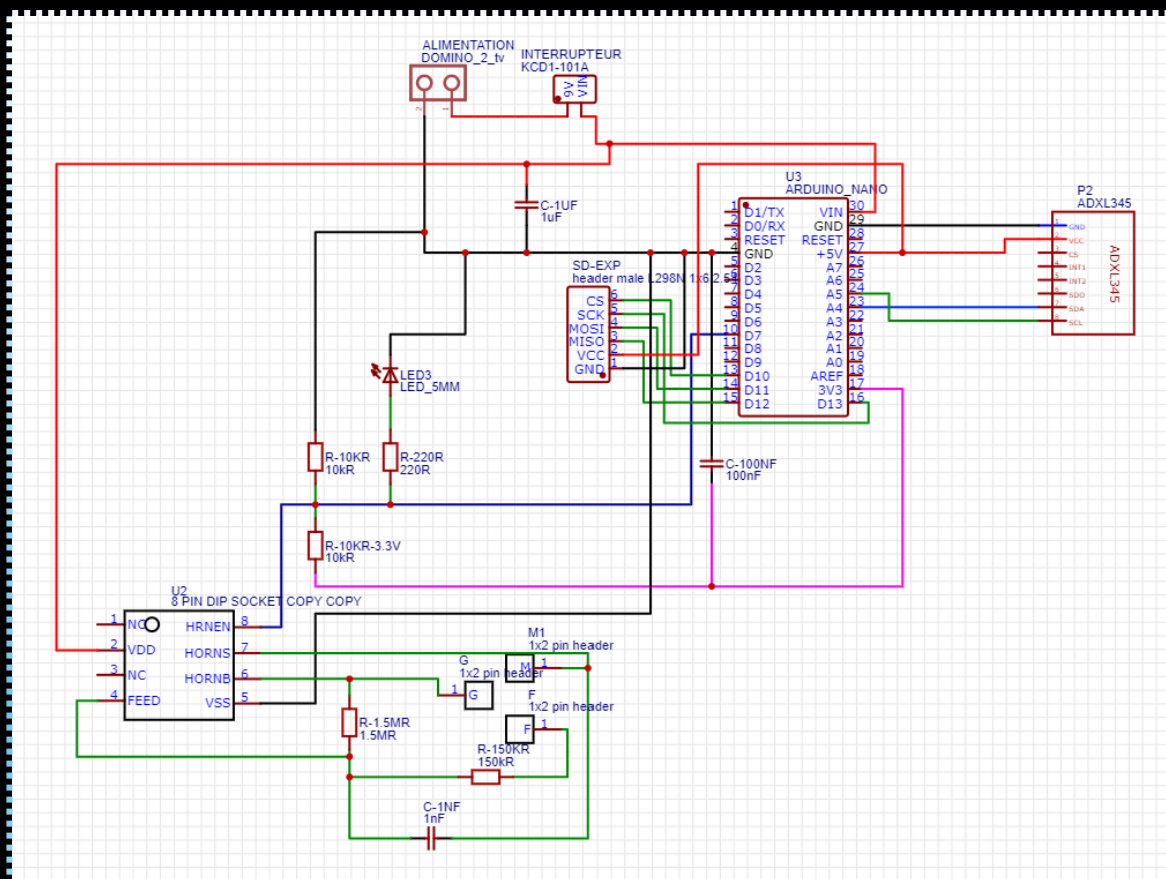


Le schéma a été créé sur EasyEDA.

La seule modification que nous avons créée sur le PCB était au niveau du transistor que nous avons remplacé par un composant emprunté, qui a nécessité la coupure d'une piste pour créer une route ultérieure avec des fils.

Sinon nous sommes restés fidèles au schéma.

- Voici le schéma du système secondaire :



Comme pour le système primaire, le schéma a été fait sur EasyEDA, mais celui-ci n'a pas été altéré, nous l'avons utilisé tel qu'il est.

Le code du système secondaire de Naérobi avait donc pour but de récupérer les données d'accélération sur les 3 axes, en calculer une norme, puis faire fonctionner le buzzer (activé lors d'une grande accélération au décollage) avec un délai inversement proportionnel à l'accélération.

Le buzzer aurait ainsi un délai plus petit, plus l'accélération augmente, accélérant donc la fréquence d'émission de son du buzzer.

Le code permettait également le stockage des données de l'accéléromètre dans la carte SD prévue à cet effet.

Le code se décompose en plusieurs parties :

ECLIPSE_exp_Naerobi_V3

ADXL

BUZZER

SD

```
1 /*
2  EXPERIENCE NAEROBI
3  Accelero ADXL345 -> mesure acceleration 3 axes + norme acceleration
4  Initialisation en tente club acceleration
5  Buzzer mesure effet doppler (activation si depassement de valeur seuil acceleration norme (5g)
6  */
7
8 #include <SPI.h>
9 #include <SD.h>
10 #include <Wire.h>
11
12 #define pinSD 10
13 #define ADXL345adress 0x53
14 #define buzzer 7
15
16
17 //sensor datasheet:
18 //https://www.analog.com/media/en/technical-documentation/data-sheets/adxl345.pdf
19
20 //-----DEF VARIABLES-----
21 //ADXL variables
22 //X axis variables
23 int16_t rawAccX=0;
24 float accX=0;
25 float accXerror=0;
26 //Y axis variables
27 int16_t rawAccY=0;
28 float accY=0;
29 float accYerror=0;
30 //Z axis variables
31 int16_t rawAccZ=0;
32 float accZ=0;
33 float accZerror=0;
34
35 float accNorm=0;
36 float accNormError=0;
37 //SD VARIABLES
38 File dataFile;
39 File flagFile;
40 #define flagFileName "iflag.bin"
41 #define dataFileName "Data.csv"
```

```
42 void setup() {
43   Serial.begin(9600);
44   Wire.begin();
45   pinMode(buzzer, OUTPUT); // SETS BUZZER as output
46   digitalWrite(buzzer, LOW); // SETS BUZZER OFF
47   rangeAcc(); //sets 16g range for acceleration measurements//OK
48   startAccMeas(); //sets ADXL to measure mode//OK
49   SD.begin(pinSD);
50   if (!SD.exists(flagFileName)) {
51     initialErrorADXL(); //OK
52     delay(20);
53     createSDflag(); //OK
54     buzzerBlink(100,500,3); //signal init OK
55     while(1); //doesn't proceed to the loop
56   }
57   else {
58     //buzzerBlink(2000,200,1);
59     readInitValADXL(); //OK
60     createSddataFile(); //OK
61     buzzerBlink(2000,200,1);
62     accNorm=sqrt((accX*accX)+(accY*accY)+(accZ*accZ));
63     dataFile=SD.open(dataFileName,FILE_WRITE);
64   }
65 }
66
67 void loop() {
68   flyTime=millis();
69   readADXL();
70   //get correct values for acc norm
71   accX=accX-accXerror;
72   accY=accY-accYerror;
73   accZ=accZ-accZerror;
74   accNorm=sqrt((accX*accX)+(accY*accY)+(accZ*accZ));
75   printSdbuff(flyTime,accX,accY,accZ,accNorm);
76 }
77
78 Serial.print(accThreshold);
79 Serial.print(' ');
80 Serial.print(accNorm);
81 Serial.print(' ');
82 Serial.println(buzzIgnite);
```

```
43 //UTILS
44 uint16_t initialNUM=1000; //number of measurements for initialisation
45 #define accThreshold 5 //MAX acceleration to activate buzzer and timer to flush(in g unit)
46 #define apogeeTime 7000 //in ms / to flush at apogee
47 #define descentTime 14000 //in ms / to flush before softly landing
48 #define gndTime 100000 //in ms / to flush at ground
49 #define timeOUT 100000 //in ms
50 #define secureSdFlushPeriod 5000 //ms flush each 5sec for secure flight
51 uint32_t flyTime=0;
52 uint32_t launchTime=0;
53 uint32_t previousMillis=0;
54 uint32_t triggTime=0;
55 bool buzzIgnite=false;
56 bool blinkState=LOW;
57 bool buzz=false;
58 uint8_t count=0;
59 #define maxCount 3
60 bool flush1=true;
61 bool flush2=true;
62 bool flush3=true;
63
64 //-----DEF FUNCTIONS-----
65 void rangeAcc(); //ADXL
66 int16_t readingI2Cacc(uint8_t adress0, uint16_t adress1); //ADXL
67 void readADXL(); //ADXL
68 void buzzerBlink(uint16_t noisetime, uint16_t delaytime, uint8_t iterations); //BUZZER
69 void startAccMeas(); //ADXL
70 void createSDflag(); //SD
71 void checkSDflag(); //SD
72 void initialErrorADXL(); //ADXL
73 void readInitValADXL(); //ADXL
74 void createSddataFile(); //SD
75 void printSdbuff(); //SD
76 void buzzAdaptBlink(float acc); //BUZZER
77
78
79 //-----
80
```

```
124 if(accNorm>=accThreshold){
125   count++;
126 }
127 else{
128   count=0;
129 } //avoid noise trigger
130
131 if (count>=maxCount && !buzzIgnite){
132   buzzIgnite=true;
133   triggTime=millis();
134 }
135
136 if (buzzIgnite){
137   buzzAdaptBlink(accNorm);
138   if (!buzz){
139     launchTime=millis();
140   }
141   buzz=true;
142
143   if (flush1 && (millis()-triggTime)>apogeeTime){
144     dataFile.flush();
145     flush1=false; //flushes one single time
146   }
147   if (flush2 && (millis()-triggTime)>descentTime){
148     dataFile.flush();
149     flush2=false;
150   }
151
152   if (flush3 && (millis()-triggTime)>gndTime){
153     dataFile.flush();
154     flush3=false;
155   }
156
157   if ((millis()-launchTime)>timeOUT){
158     digitalWrite(buzzer, LOW);
159   }
160 }
161
162 static unsigned long millisLastSave = 0;
163 if (millis() - millisLastSave > secureSdFlushPeriod){
164   dataFile.flush();
165   millisLastSave = millis();
166 }
167 }
```

```

1 //functions for accelero readings
2
3 void startAccMeas() {
4     //enable measurement mode
5     Wire.beginTransmission(ADXL345address); // Start communicating with the device
6     Wire.write(0x2D); // Access/ talk to POWER_CTL Register - 0x2D
7     // Enable measurement
8     Wire.write(0b00001000); // IMPORTANT (0000 1000 binary) Bit D3 High for measuring enable
9     Wire.endTransmission();
10 }
11
12 void rangeAcc() {
13     Wire.beginTransmission(ADXL345address);
14     Wire.write(0x31);
15     Wire.write(0b00000011); //select 16g range
16     Wire.endTransmission();
17     /*
18     Binary order to select range (see page 27 ADXL345 datasheet)
19     0b00000000 -> 2g range -> /256 to convert in g unit
20     0b00000001 -> 4g range -> /128 to convert in g unit
21     0b00000010 -> 8g range -> /64 to convert in g unit
22     0b00000011 -> 16g range -> /32 to convert in g unit
23     */
24 }
25
26
27 int16_t readingI2Cacc(uint8_t adress0, uint16_t adress1) {
28     uint8_t data0;
29     uint8_t data1;
30     Wire.beginTransmission(ADXL345address);
31     Wire.write(adress0);
32     Wire.endTransmission();
33     Wire.requestFrom(ADXL345address, 1); //read 1 byte
34     data0 = Wire.read();
35     Wire.beginTransmission(ADXL345address);
36     Wire.write(adress1);
37     Wire.endTransmission();
38     Wire.requestFrom(ADXL345address, 1); //read 1 byte
39     data1 = Wire.read();
40     int16_t res;
41     return res = (int16_t)((data1 << 8) | data0); //final data
42 }
43
44 void readADXL() {
45     //raw readings
46     rawAccX = readingI2Cacc(0x32, 0x33);
47     rawAccY = readingI2Cacc(0x34, 0x35);
48     rawAccZ = readingI2Cacc(0x36, 0x37);
49
50     //final readings (in g unit)
51     accX = (float)rawAccX/32;
52     accY = (float)rawAccY/32;
53     accZ = (float)rawAccZ/32;
54 }
55
56 void initialErrorADXL() {
57     for(uint16_t i=0; i<initialNUM; i++){
58         readADXL();
59         accXError = accX + accXError;
60         accYError = accY + accYError;
61         accZError = accZ + accZError;
62     }
63     accXError = accXError/initialNUM;
64     accYError = accYError/initialNUM;
65     accZError = accZError/initialNUM;
66
67     accZError = 1 - accZError; //get error to correct and have 1g on Z-axis
68 }

```

```

1 //SD functions
2
3 void createSDflag() {
4     flagFile = SD.open(flagFileName, FILE_WRITE);
5     SD.open(flagFileName, FILE_WRITE);
6     flagFile.write((char*)&accXError, 4);
7     flagFile.write((char*)&accYError, 4);
8     flagFile.write((char*)&accZError, 4);
9     flagFile.flush();
10    flagFile.close();
11 }
12
13
14 void createSDdataFile() {
15     dataFile = SD.open(dataFileName, FILE_WRITE);
16     dataFile.write("time_ms, accX_(g) noPr, accY_(g) noPr, accZ_(g) noPr, accNorm_(g) Pr\n");
17     dataFile.close();
18 }
19
20 void readInitValADXL() {
21     flagFile = SD.open(flagFileName, FILE_READ);
22     flagFile.readBytes((char*)&accXError, 4); //buffer "&nom" adresse variable
23     flagFile.readBytes((char*)&accYError, 4);
24     flagFile.readBytes((char*)&accZError, 4); //stocke lecture dans variable
25     flagFile.close();
26 }
27
28 void printSDbuff(uint32_t flyTime, float accX, float accY, float accZ, float accNorm) {
29     dataFile.print(flyTime);
30     dataFile.print(',');
31     dataFile.print(accX);
32     dataFile.print(',');
33     dataFile.print(accY);
34     dataFile.print(',');
35     dataFile.print(accZ);
36     dataFile.print(','); //single char
37     dataFile.print(accNorm);
38     dataFile.print("\n");
39
40     /*
41     Serial.print(flyTime);
42     Serial.print(',');
43     Serial.print(accX);
44     Serial.print(',');
45     Serial.print(accY);
46     Serial.print(',');
47     Serial.print(accZ);
48     Serial.print(','); //single char
49     Serial.println(accNorm);
50     */
51 }

```

ECLIPSE_exp_Naerobi_V3	ADXL	BUZZER	SD
------------------------	------	--------	----

```
1 //functions for buzzer noise
2
3 void buzzerBlink(uint16_t noisetime, uint16_t delaytime, uint8_t iterations){
4     for(uint8_t j=0; j<iterations; j++){
5         digitalWrite(buzzer,HIGH);
6         delay(noisetime);
7         digitalWrite(buzzer,LOW);
8         delay(delaytime);
9     }
10 }
11
12 void buzzAdaptBlink(float acc){
13     unsigned long currentMillis = millis();
14     uint16_t delayBuzz = (uint16_t)((1/acc)*1000.0);
15
16     if (currentMillis-previousMillis>=delayBuzz){
17         previousMillis=currentMillis;
18         if (blinkState){
19             blinkState=LOW;
20         }
21         else{
22             blinkState=HIGH;
23         }
24     }
25     digitalWrite(buzzer, blinkState);
26 }
```

4. Les erreurs et difficultés rencontrées avec l'électronique

Entre les difficultés rencontrées avec le code du système secondaire pour joindre l'accéléromètre et le Buzzer, et notre erreur par rapport à la tension aux bornes du servomoteur... l'électronique n'a pas été un développement linéaire.

Heureusement, nous étions bien encadrés lors du C'Space et avons pu remédier à tout cela.

(dédicace à Julien qui nous a aidé et prêté des convertisseurs de Tension jusqu'à trouver le bon !)

Finalement, le code a été réglé (merci Max), et le système primaire a pu retrouver son fonctionnement, mais cela n'a pas été sans nécessiter un changement de servo, qui a impacté la mécanique et forcé nos chefs de méca (Romain et Simon) à modifier la trappe et le bras du servomoteur.

V – Le C'SPACE 2024

Le C'SPACE est un rassemblement de passionnés d'aérospatiale, se déroulant au camp de GER à Tarbes pendant une semaine en été. Ce rassemblement permet le lancement sécurisé des fusées de chaque club (à condition que celles-ci soient qualifiées suivant les modalités du cahier des charges).

Tout d'abord, à notre arrivée nous avons été plongés dans un autre environnement sur la base militaire du 1^{er} RHP (1^{er} Régiment de Hussards Parachutistes de l'armée de terre). En effet, pour la moitié des membres du club entrer dans une base et partager la vie des militaires pendant une semaine était une première expérience. Après avoir échangé nos cartes d'identité à l'accueil contre des badges à n'absolument pas perdre, nous sommes allés poser nos affaires dans les dortoirs du bâtiment réservé aux participants mineurs (bâtiment Verdun).

Cependant, cet environnement assez spartiate permet réellement de se dédier uniquement à la finalisation de projets, en l'occurrence, nos deux fusées : ARÈS et NAÉROBI.

Ainsi, durant le C'Space, la grande majorité de notre temps a été consacrée à finaliser nos fusées en vue de pouvoir les qualifier. Grâce à un atelier bien équipé ouvert de 7 h à 0 h, une cafétéria de qualité et les dortoirs à proximité, nous avons pu travailler dans des conditions exceptionnelles. Et même si de telles journées peuvent paraître excessivement longues, nous ne nous rendions pas compte du temps qui passait et le travail était tout sauf fastidieux, car nous travaillions sur quelque chose qui nous tiens à cœur : nos fusées. Puis nous souhaitons les voir voler après tant d'efforts ! Ainsi une journée type commençait vers 8 h et se finissait aux environs de 0 h/1 h (avec pour uniques pauses celles pour manger...).

Même si l'objectif est propre à chaque club, dans le but de qualifier sa fusée, l'espoir de la lancer et qu'elle fasse un vol nominal, chaque équipe s'entraide en se prêtant du matériel comme un fer à souder lorsqu'une équipe a un problème d'électronique ou en apportant des conseils lors de difficultés.

La présence de bénévoles tout aussi passionnés et de la « Debug Team » nous a aussi grandement dépannés. L'esprit d'équipe était au rendez-vous et c'était l'occasion de créer des liens entre passionnés et de garder des contacts pour les années futures (amitiés, études, carrière...). De plus, nous sommes très reconnaissants envers les clubs expérimentés qui ont l'expérience et les astuces utiles ayant déjà participé maintes fois au C'Space. En effet c'est aussi grâce à cette aide précieuse que les jeunes clubs comme ÉCLIPSE avancent et acquièrent une expérience qui fait la différence dans la réalisation de projets.

Cette cohésion s'est même faite ressentir lors des lancements des autres équipes. Ainsi, lorsqu'une équipe s'apprêtait à lancer sa fusée, nous nous installions tous ensemble autour d'un écran et regardions leur vol tout en partageant leur stress.

Après les résultats du vol, nominal ou balistique, les équipes se félicitaient si le vol avait été nominal ou venaient poser des questions et demander des conseils s'il avait été balistique.

Le C'Space est également un excellent endroit pour apprendre de nouvelles compétences (en plus de celles acquises durant l'année) telles que la soudure de composants électroniques, la modélisation 3D, la mécanique... car il y a toujours quelqu'un qui peut enseigner ces savoirs.

C'est alors dans un environnement frénétique que nous travaillions sans relâche pour atteindre notre objectif : le lancement. En immersion dans le camp, avec toutes les infrastructures et l'aide disponible pour réussir, auprès de passionnés qui partagent le même objectif, nous avons réglé les ultimes problèmes (car nous l'avons compris : il y a toujours quelque chose qui cloche sur les projets en arrivant au C'Space... mais le but c'est de régler ça au plus vite et au mieux !) en vue des contrôles.

Le temps de la qualification représente une partie importante de la semaine car pour la plupart des fusées qui ont déjà réussi à se qualifier pour le C'Space, il reste encore à se qualifier pour le vol final qui est l'aboutissement de tant de travail. Ainsi, chaque équipe s'est donnée à fond pour régler les derniers problèmes avant d'aller aux contrôles. Mais même en ayant pensé que tout était réglé, les contrôleurs trouvaient encore et encore des couacs et certains détails n'étaient pas entièrement satisfaisants. Ainsi il n'était pas rare de voir des équipes revenir à l'atelier pour revoir certains aspects de la fusée et corriger des erreurs (ce fut le cas pour les deux fusées du club également...).

Vient alors le moment du lancer ! Tout d'abord pour se rendre sur la zone de lancement, nous avons eu le choix entre deux moyens de transport : les camions de l'armée qui sont peu confortables parce que l'on saute à la moindre bosse, mais très amusants, et les voitures personnelles qui sont plus confortables. Arrivés, le stress commence à monter car le fruit d'une année de travail s'apprête à payer et à révéler sa fiabilité en quelques secondes.

Lors de la marche vers le pas de tir, nous réfléchissions à ce qui pourrait fonctionner et ne pas fonctionner correctement pendant le vol ; mais l'excitation et la tension était à leur comble. Nous avons déroulé calmement notre chronologie et préparé au mieux la fusée en vue de l'insertion du moteur, le décompte et enfin le lancement. Tout cela sous la supervision du coordinateur de vol en tente JUPITER (poste de contrôle) qui rythme les opérations et le décompte.

Vient alors le décompte final : 10, 9, 8, 7, 6, 5, 4, 3, 2, unité, mise à feu, décollage !

Puis silence. La fusée s'élève dans le ciel avec puissance. Et revient au sol après l'apogée. Tout cela sous le regard de l'équipe et du public.

la suite sur le vol dans la partie suivante du dossier qui lui est dédiée...

Une des dernières soirées du C'Space est marquée par l'arrivée de VIP qui font partie du domaine de l'aérospatiale comme des membres du CNES, des astrophysiciens... Nous avons eu la chance de pouvoir parler avec eux sur la zone de lancement en attendant notre tour.

C'est la dernière soirée avant notre départ... On commence à prendre l'habitude de l'atmosphère, du rythme et là on se rend compte que c'est déjà la fin. C'est un mélange de fierté d'avoir rempli l'objectif posé en début d'année et de mélancolie de déjà partir qui est partagé par tous les membres. Mais c'était également pour nous cette année un moment de tension particulière, en effet nous étions dans l'attente du lancement de notre deuxième fusée, parmi celles qui clôturaient le C'SPACE 2024, bien que qualifiée deux jours avant.

S'en est suivi un rapide rangement ainsi que les au revoir.

Et nous voilà partis !

En espérant se retrouver l'année prochaine

Dimanche : après être arrivés en fin d'après-midi et avoir récupéré nos « cartes » d'accès au camp, nous avons déchargé notre matériel et mis en place ce qui sera notre atelier pour les prochains cinq jours. Nous avons également établi la liste des choses que nous avons à faire pour terminer de préparer les fusées pour les contrôles.

Lundi : nous avons passé le lundi à préparer la fusée, à savoir faire les dernières retouches de coupe du tube, poncer les pièces, finir le code, souder les derniers éléments comme les boutons de commande et les leds, et peindre la fusée.

Mardi : dans la mise en place finale de la fusée, nous nous sommes retrouvés avec un problème majeur, le servomoteur que nous utilisions, capable d'une force de 15 kg par cm² était en fait en train de tirer beaucoup trop de courant via l'Arduino et ne pouvait pas être alimenté autrement dans notre circuit. Nous avons d'abord essayé différentes manières de différer l'alimentation puis nous avons demandé les conseils des bénévoles qui nous ont donné un composant appelé BEC et qui devait nous permettre de régler les problèmes du servo. Cependant cela n'a pas marché et nous avons opté pour un changement de servo pour des petits SG90.

Mercredi : après avoir réglé le problème de consommation des servo, nous avons dû renforcer les bras des servomoteurs avec une lamelle d'aluminium ainsi qu'une vis et des rondelles de différentes tailles. Après avoir réglé ces problèmes, nous sommes allés aux contrôles et nous avons qualifié Naérobi et Arès respectivement à 17 h et 21 h.

Jeudi : nous avons pu faire voler Naérobi puis nous avons attendu tout le reste de la journée dans l'espoir de pouvoir faire décoller Arès mais ce ne fut pas le cas. Ainsi

nous avons passé la soirée à analyser les restes de Naérobi et à trouver de meilleures piles pour Arès.

Vendredi : nous avons pu faire voler Arès en fin de matinée puis après l'avoir récupérée en bon état, nous sommes partis.

VI - Le vol

Nous avons pu lancer notre première fusée, Naérobi, le jeudi à 13 h. Le temps était clair : un grand soleil et un ciel bleu sans aucun nuage.

Après avoir déroulé correctement notre chronologie en tente club, nous avons pu installer la fusée en rampe : révision de la chronologie, vérification des fixations...

Naérobi étant la première des deux fusées à voler pour ce C'Space, nous n'avions pas de correction de dernière minute à faire. Notre plus grande crainte était que la trappe vienne à se détacher lors du décollage car elle ne nous semblait pas très bien tenir au vu des modifications que nous avons dû apporter au bras du servo mais ce fut plutôt le contraire.

Pour pallier le détachement de la trappe à cause du flux d'air au décollage, nous avons mis une feuille de papier découpée pour rediriger l'air.

Alors, nous avons enfilé des gilets de signalisation, et nous nous sommes dirigés vers le pas de tir en équipe.

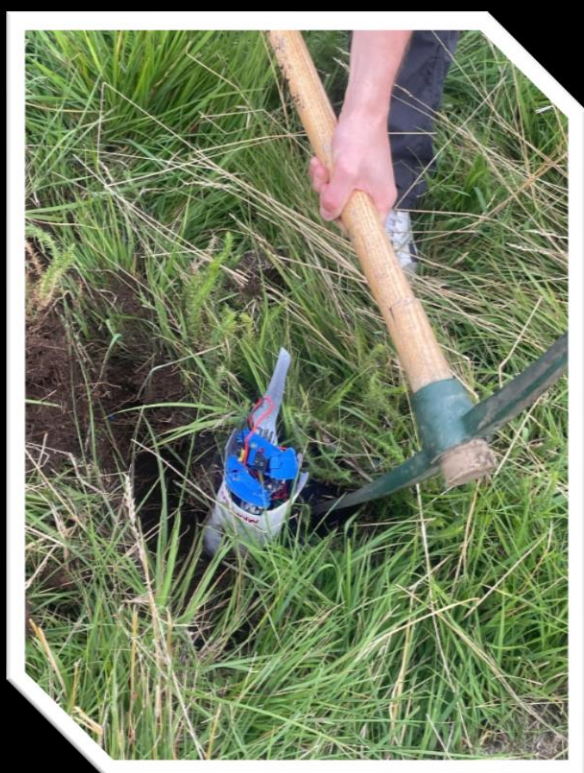
Malheureusement, après plusieurs faux départs et retours en tente car nous avons été appelés en même temps qu'une autre équipe, les nombreuses activations du système primaire pour la mise en place de la trappe ont légèrement vidé les piles.



Il était impératif pour le fonctionnement du système que les piles soient à 9V ou plus. Comme notre design nous empêchait de changer les piles ou vérifier les tensions facilement, lors du vol la tension était trop basse et la trappe n'a pas pu s'ouvrir.

Ceci à causer le vol balistique de notre première minifusée Naérobi.

Les dégâts étaient très importants, la fusée s'étant coupée en deux lors de l'impact, et nous avons dû récupérer la coiffe enfoncée dans le sol à l'aide d'une pioche.



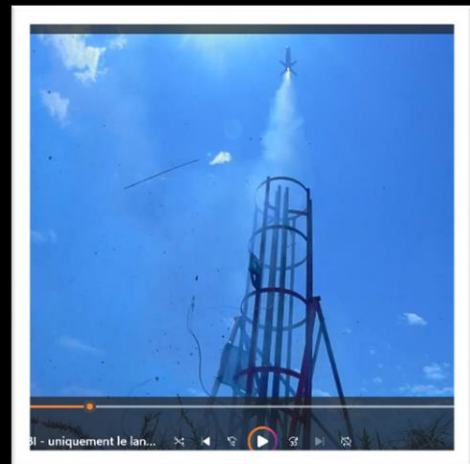
VII - L'analyse des données & résultats

Malheureusement, dû au vol balistique de Naérobi, les données de l'accéléromètre n'ont pas pu être récupérées.

Et tout comme pour Arès, le vent fort nous a empêché de pouvoir analyser le fichier audio de l'enregistrement du Buzzer. Nous pouvons tout de même remarquer l'évolution de la vitesse du clignot du buzzer à l'oreille.

D'autre part, la caméra au sol nous a offert une belle vue du décollage de Naérobi.

Et la caméra embarquée nous a donné une vidéo surprenante, non-seulement des rotations de la fusée sur elle-même, mais du moment de l'impact de la minifusée avec le sol.



Naérobi s'étant cassée en deux, la partie inférieure comprenant la caméra s'est tournée d'une telle manière que nous pouvons voir le reste de la fusée s'éclater au sol et les dégâts immédiats.

Voici les résultats de l'image la plus claire pendant l'impact :



VIII – Conclusion

En fin de compte, le club ÉCLIPSE a atteint son objectif !

En se rassemblant, nous avons su créer et mettre en place notre projet de minifusées au lycée. Ainsi c'est avec passion et persévérance que nous avons fait vivre ce projet tout au long de l'année jusqu'à son aboutissement lors du C'Space 2024.

Nous avons beaucoup appris, dans plein de domaines (CAO, mécanique, électronique, codage...) et cette expérience enrichissante sera utile pour tous les membres qui y ont participé !

Les retombées des deux vols (et des étapes pour leur préparation qui ont été partagées sur les réseaux sociaux et au sein du lycée) de cette année sont nombreuses sur le plan des acquis techniques pour les membres et de la communication avec l'extérieur. Nous espérons avoir pu susciter l'intérêt des jeunes pour ce type de projets scientifiques et nous recevons déjà des candidatures pour l'année qui arrive.

En effet, nous comptons bien maintenir les activités du club pour les années à venir avec une relève motivée !

Cela permet de continuer à partager cette précieuse expérience et de continuer à rassembler autour de projets scientifiques intéressants. De plus cela amène à faire un premier pas dans le domaine spatial et à dialoguer avec des étudiants et acteurs de ce secteur.



IX – Remerciements

Pour finir ce rapport, nous souhaiterions remercier l'ensemble de personnes et entités qui ont permis l'ensemble des opérations du club ÉCLIPSE sur cette campagne de 2023-2024 :

MERCI à toute l'équipe d'ÉCLIPSE 2023/2024 : les 21 membres qui se sont impliqués sur la conception et la fabrication de nos deux minifusées.

MERCI également aux professeures encadrant M^{me} RAFFAELLI et M^{me} MOULIN pour leur aide et leur soutien sur l'année.

MERCI aux 4 membres (parmi les 21 membres du club) qui ont été au C'SPACE pour leur finalisation et lancement : Poppy Kerridge, Romain Pasturel, Simon Zorzi et Maximilien Boudet.

MERCI à M^{me} RAFFAELLI de nous avoir accompagnés dans cet évènement.

MERCI à l'APEL du lycée la TRINITÉ d'avoir financé notre projet et d'avoir fait confiance à nos équipes cette année pour sa réalisation.

MERCI au CVL du lycée la TRINITÉ d'avoir soutenu notre projet cette année.