Centrale Marseille

Pôle de l'Étoile
Technopôle de Château-Gombert
38, rue Frédéric Joliot-Curie
13013 Marseille – Cedex 13

Tél. +33 (0)4 91 05 45 45
www.centrale-marseille.fr

RAPPORT DE PROJET C'SPACE 2022

Nom du projet : Quetzal

Club : Centrale Marseille Aero

Matricule : MF46

Membres du projet :

Eliott Crancée - Jérémie Bron - Henri Chanzy

CENTRALE MARSEILLE AÉRO

FABLAB MARSEILLE

Table des matières

REMERCIEMENTS

Tout d'abord, nous tenons à remercier tout particulièrement et à témoigner toute notre reconnaissance aux personnes suivantes, pour leur dévouement et leur soutien dans la concrétisation de ce projet ingénieur :

- L'école Centrale de Marseille qui permet aux étudiants de réaliser ce genre de projet annexe au cursus.

- Le FabLab de Centrale Marseille, pour avoir financé le projet et mis à disposition les locaux et le matériel nécessaire pour la réalisation du projet.

- Le camp militaire du 1er RHP de Ger pour avoir chaleureusement accueilli le C'Space 2022.

- Le CNES, Planète Science et l'ensemble des bénévoles pour leur coopération professionnelle tout au long de cette expérience et pour avoir partagé avec nous, une partie de leurs savoir-faire et de leurs expériences professionnelles. Merci à eux pour leur temps et pour nous avoir transmis leur passion.

- Merci tout particulièrement à Jules Ferrand, camarade de promo sans lequel nous n'aurions pas mis au point un système électronique fonctionnel. Merci pour son intérêt porté au projet et pour ses conseils.

INTRODUCTION

Notre association s'appelle Centrale Marseille Aero, et a été créée à la rentrée des classes de septembre 2021. Son but est de promouvoir l'aéronautique et l'aérospatiale au sein de l'école d'ingénieur Centrale de Marseille par le biais de conférences, la création d'un simulateur de vol ou encore un projet de modélisme. Elle est rattachée à l'association FabLab Marseille de Centrale Marseille. Le FabLab nous offre une variété de machines numériques de hautes technologies (imprimante 3D, découpeuse laser, etc...), d'outils d'usinage, de composants électroniques et de surface de travail.

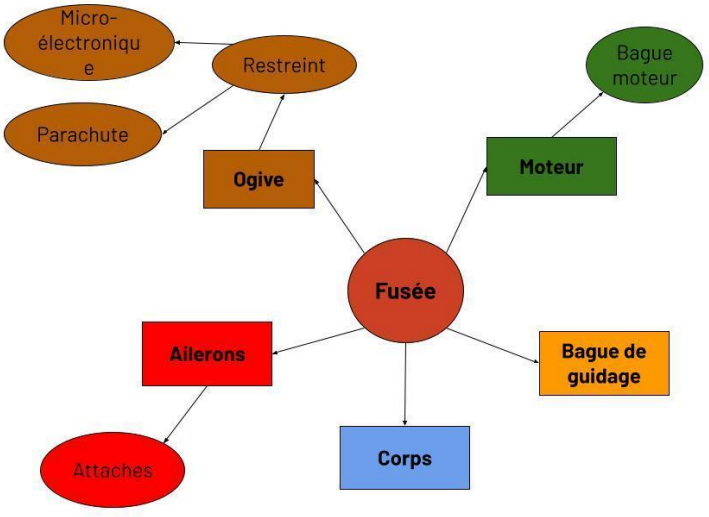
Le projet Quetzal a été de construire une minifusée et de la lancer durant la campagne nationale de lancement du C'Space de 2022. Le but était de se familiariser avec l'astromodélisme car aucun d'entre nous n'en avait fait auparavant. Nous avons tout appris cette année et avons entièrement conçu la fusée.

PLANIFICATION DU PROJET

Organisation

La première étape a été de recruter des membres pour le projet. Cela s'est fait dans les groupes Messenger d'intégration de notre école. Ainsi nous avons créé la discussion « Centrale Marseille Aero » ainsi qu'une autre « CMA pôle modélisme » dédiée uniquement au modélisme et dans notre cas, à l'unique projet de modélisme, le projet Quetzal, celui du C'Space 2022. En Janvier 2022, c'est finalement 6 membres qui ont été retenu : Eliott Crancée, Jérémie Bron, Henri Chanzy, Aymeric Bolot, Inès Capra et Marie-Sophie Boubon. Eliott a été nommé responsable modélisme au bureau restreint de CMA ainsi que chef de projet pour le C'Space 2022.

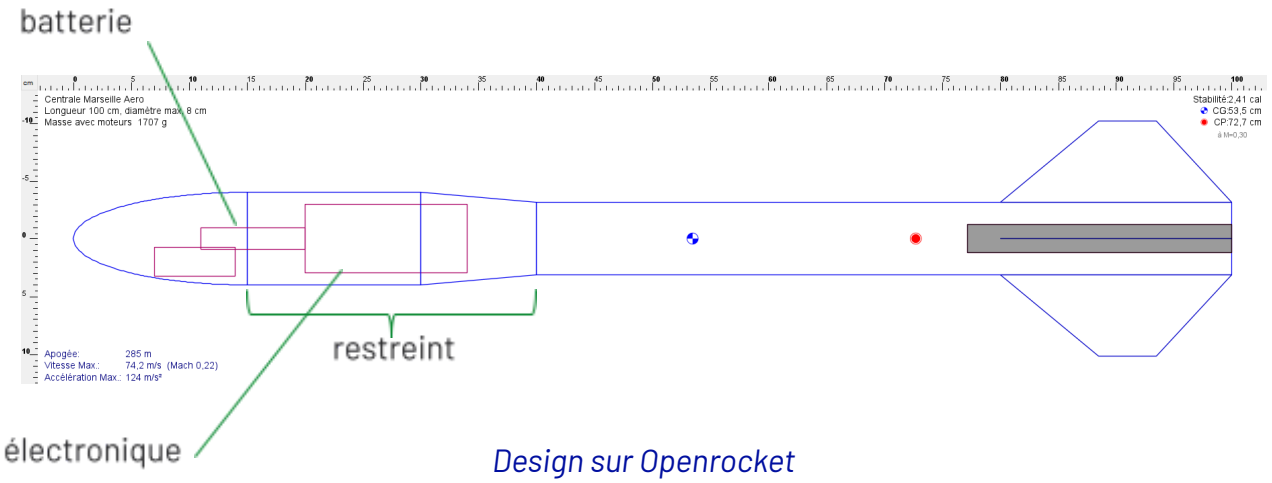
Le projet a commencé par des réunions de cadrage et d'organisation entre les membres du projet. Nous avons partagé les documents via un google drive. Nous nous sommes renseignés sur ce qui était attendu au niveau du cahier des charges et avons imaginé une première version de la fusée. Après avoir réalisé une mind map grossière des différentes parties de la minifusée, nous en avons tiré différents pôles et nous y sommes répartis.



Eléments de base de la fusée

Premier design

Nous avons défini que la fusée serait composée d'un circuit séquentiel permettant l'ouverture du parachute par une trappe latérale située dans l'ogive. Elle contiendrait également un circuit expérimental intégrant un accéléromètre, permettant de mesurer sa vitesse. Enfin une ouverture et une caméra permettraient de filmer le vol depuis la fusée. Le logiciel Openrocket nous a permis de concevoir le premier design. Nous avons choisi une tête de fusée plus large pour pouvoir intégrer plus facilement l'électronique.



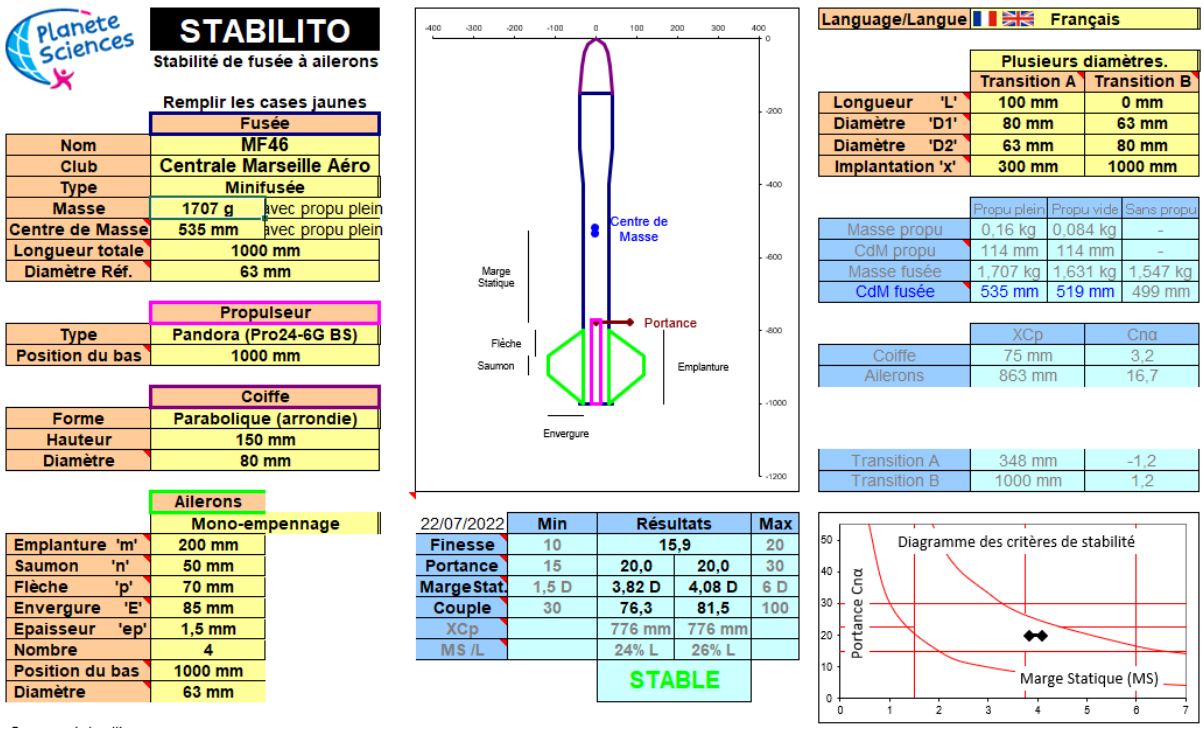
Design sur Openrocket

On peut retrouver ici les principaux éléments qui composent la fusée, ils sont simplifiés en rectangles. De gauche à droite : l'ogive en bleue qui est arrondie selon une formule mathématique permettant d'optimiser la percée de la fusée dans l'air, le parachute en rouge qui est collée sur la paroi de l'ogive, la batterie en rouge, un cylindre bleu qui rétrécit par la suite (forme conique) : on appelle les deux formes bleues correspondantes le restreint. Enfin, nous trouvons les composants électroniques en rouge dans le restreint, le tube de corps en bleu, le moteur en gris et enfin les 4 ailerons (trapézoïdaux).

Premier Stabtraj

Le Stabtraj est un tableur de calcul qui permet de vérifier la stabilité de la fusée et qu'elle respecte les obligations du cahier des charges. Il nous a permis de déterminer la forme des ailerons en fonction de la répartition de poids et de la forme globale de la fusée. Il nous a aussi permis de déterminer le dimensionnement du parachute. L'utilisation d'Openrocket au préalable était nécessaire car le Stabtraj a besoin du poids de la fusée et du centre de masse, qui ont été calculés sur Openrocket en fonction du design, de l'emplacement des composants et des types de matériaux.

Les critères de stabilité reposent sur 4 éléments : la finesse de la fusée qui est le rapport de sa longueur sur son diamètre le plus gros, la portance qui dépend de la surface des ailerons, la marge statique qui est l'écart entre le centre de portance et le centre de poids, et le couple qui mesure la force de rotation que fournissent les ailerons pour garder la fusée stable. Ces éléments doivent être compris dans des intervalles définis. Ici, nous avons fait en sorte que la forme des ailerons soit telle que la fusée respecte les critères de stabilité.



Stabilito du Stabtraj du 3 avril 2022

Ces calculs permettent de déduire finalement la trajectoire de la fusée. Celle-ci doit respecter un critère : la portée balistique (si le parachute ne se déploie pas) de la fusée lancée à 80° doit être inférieur à 200m. Cela implique que la fusée ne doit pas être trop légère sinon elle ira trop loin horizontalement et sera dangereuse. De plus, nous pouvons indiquer la surface du parachute et en déduire la vitesse de descente qui doit être comprise entre 5 et 15 m/s^2.

TRAJECTO
Trajectographie de fusée

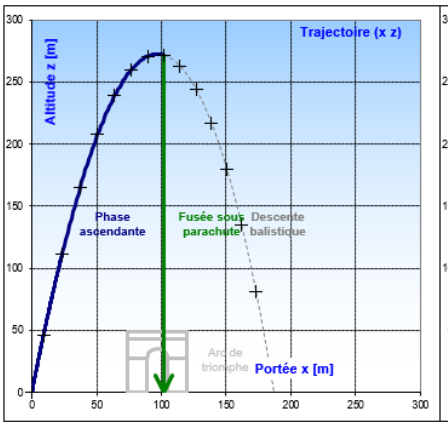
Remplir les cases jaunes

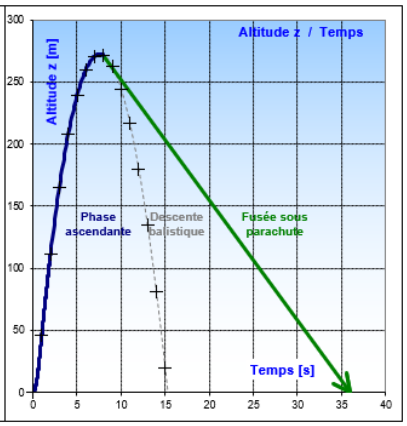
	Fusée
Nom	MF46
Club	Centrale Marseille Aéro
Masse totale	1,707 kg
Propulseur	Pandora (Pro24-6G BS)

	Traînée Aérdynamique
Surface Réf.	0,003627 m²
Cx	0,5

	Rampe de Lancement
Longueur	2,5 m
Élévation	80 °
Altitude	0 m

	DescenteSousParachute	
	Fusée	0 satellite
Masse	1,6314 kg	
Dépotage	N/A	
Ouverture para	8 s	
Surface para	0,28 m²	
Cx parachute	1	
Vitesse du vent	5 m/s	
Vitesse descente	9,7 m/s	
Durée descente	28 s	
Durée du vol	36 s	
Déport latéral	± 140 m	





22/07/2022	Temps	Altitude z	Portée x	Vitesse	Accélération	Efforts
Sortie de Rampe				22,5 m/s		
Vit max & Acc max				74 m/s	131 m/s²	
Culmination, Apogée	7,6 s	272 m	97 m	13 m/s		
Ouverture parachute fusée	8,0 s	272 m	102 m	13 m/s		29,1 N
Impact balistique	15,3 s	~0 m	187 m	67,6 m/s		3732 J

	Pour localiser la fusée
Couleur fuselage/coiffe	Blanc
Couleur parachute fusée	Bleu

Trajecto du Stabtraj du 3 avril 2022

Jalons

Après avoir défini les différents éléments de travail de la fusée, nous avons dû déterminer des jalons d'avancement pour pouvoir tenir les échéances imposées par le C'Space.

Les échéances immuables du C'Space :

3 Avril : donner une version définitive du StabTraj et donc définir les paramètres généraux de la fusée. En réalité, celui-ci serait modifié lors des contrôles de fusée durant la semaine de lancement.

5 Juin : rendre un rapport de projet pré-vol (nous ne l'avons pas fait).

11-12 Juin : rencontre des Clubs Espace 3 (RCE 3) ou nous devons présenter une version aboutie de la fusée qui est autorisée ou non à participer à la semaine finale de lancement.

16-23 Juillet : la semaine finale de lancement durant laquelle la fusée doit subir les contrôles et être qualifiée, si elle ne l'est pas nous devons la modifier jusqu'à ce qu'elle le soit.

Jalons du pôle microélectronique :

10 Mars : donner le poids des composants, cela permet de finaliser le fichier Openrocket vers une répartition de poids plus précise.

27 Mars : donner la liste et les emplacements des composants, indispensable pour finaliser le Stabtraj (avant le 3 Avril) et démarrer la modélisation 3D du restreint en conséquence.

24 Avril : terminer le circuit séquentiel.

15 Mai : terminer le circuit expérimental.

Jalons du pôle 3D :

27 Mars : modéliser l'ogive.

24 Avril : modéliser le restreint.

15 Mai : avoir tout imprimé.

Jalons du tube de corps :

27 Mars : déterminer la taille finale du tube de corps.

24 Avril : acheter le tube de corps, élaborer les bagues moteur.

15 Mai : découper le tube de corps, fixer les bagues moteur et les ailerons.

Jalons du parachute :

20 Février : déterminer les dimensions exactes du parachute.

15 Mai : finir la construction du parachute.

Jalons des ailerons :

3 Avril : déterminer la forme prototype des ailerons.

15 Mai : déterminer la forme finale des ailerons.

15 Mai - 22 Mai : Assemblage

22 Mai - 10 Juin : Tests

Budget prévisionnel

La construction de la fusée a un prix que nous avons dû essayer de prévoir. Nous avons négocié avec le FabLab, qui nous finançait, le budget prévisionnel suivant :

Microélectronique : 200 €
Mécanique : 195 €
Parachute : 15 €
Peinture et autocollants : 15 €

Total : 425 €
Budget alloué : 450 €

Répartition du travail

Nous nous sommes répartis le travail de la manière suivante :

Eliott : chef de projet
Jérémie et Aymeric : pôle microélectronique
Henri : pôle 3D
Inès et Marie-Sophie : pôle tube de corps, ailerons et parachute (pôle mécanique)

CONSTRUCTION DE LA FUSÉE

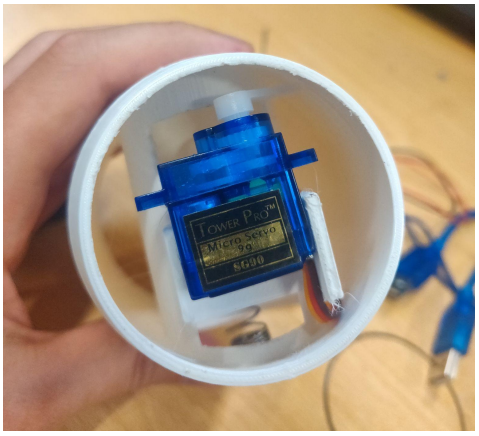
Parachute

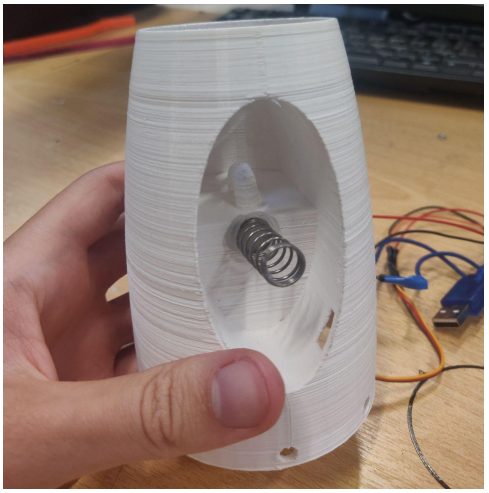
Le parachute a été construit en toile de Spi bleue, il est de forme hémisphérique. Nous sommes partis d'un carré, qui a été découpé en octogone puis découpé en étoile de façon à recoudre pour donner la forme hémisphérique. Il est constitué de 8 points d'attaches avec œillets, et d'une attache au centre pour la porte de la trappe du parachute. Le parachute ne rentrait initialement pas dans la trappe, c'est pourquoi sa taille a été réduite au cours du projet. Les suspentes sont reliées à un émerillon qui est relié à la corde principale attachée dans l'ogive.

Le parachute

Ogive, restreint et impression 3D

La conception de la fusée a été faite à partir de Openrocket. Les plans détaillés ont été faits sur papier (en annexe). Nous avons fait le choix de placer le parachute dans l'ogive à l'intérieur d'une trappe ouverte pas un servomoteur. Le parachute est éjecté par la force d'un ressort. Les composants micro-électroniques sont situés juste en dessous, dans le restreint. Ils sont placés dans des cases adaptées à leurs dimensions. Le restreint et l'ogive ont été imprimés en 3D avec du PLA, ce qui nous a posé problème puisque la fusée a été tordue par la suite à cause de la chaleur.



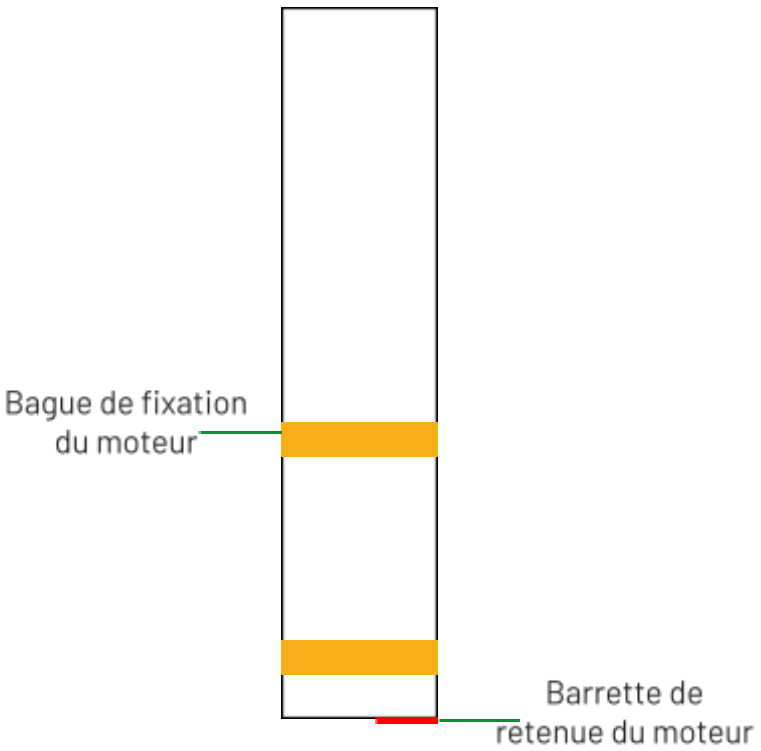


L'ouverture de la trappe nous a posé beaucoup de problème : parachute qui reste coincé, câbles qui se prennent dans le ressort, bras du servomoteur qui se tord sous la pression. Nous avons dû faire face à ces problèmes et trouver des solutions. La forme de la trappe qui posait beaucoup de problème a été sculpté avec du plastique et un fer à souder pour offrir une grande liberté dans sa forme. Le design final offrait finalement une grande répétabilité.

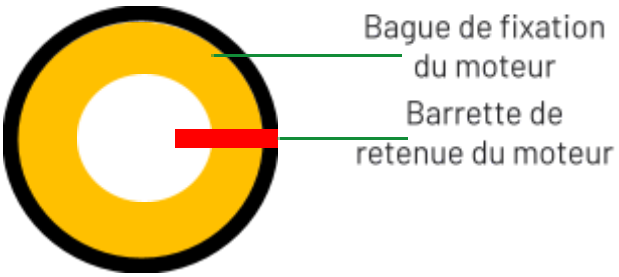
Corps et ailerons

Pour cette partie, nous avons acheté un tube en PVC de 63 mm de diamètre pour le découper aux bonnes dimensions. Nous avons ensuite découpé les ailerons selon la forme escomptée dans une plaque d'aluminium de 2 mm d'épaisseur. Nous avons ensuite conçu des équerres pour les fixer au tube, que le designer 3D de notre équipe, Henri Chanzy, a modélisé et imprimé via SolidWorks. Nous avons par la suite réalisé une deuxième version d'équerres, plus rigides, car nous nous sommes rendus compte qu'elles étaient trop fragiles à la RCE3. Une fois les ailerons fixés, nous avons réalisé des bagues en bois pour

maintenir le moteur dans le tube de corps, ainsi qu'une petite barrette en aluminium à l'extrémité basse du tube (fixée à l'aide d'un simple clou)



Corps vu de profil



Corps vu d'en-dessous

Microélectronique

N'ayant aucune connaissance préalable en électronique et particulièrement en programmation, exceptées celles enseignées en classes préparatoires, nous sommes partis sur un système électronique Arduino car nous savions qu'il remplirait les fonctions dont nous avions besoin.

Après s'être familiarisé avec le langage de programmation et les composants classiques (leds, résistances, boutons poussoirs...), nous avions deux circuits à réaliser : le circuit séquentiel (pour le déploiement du parachute) et le circuit expérimental (pour la mesure de la vitesse de la fusée). Comme le circuit séquentiel était le plus important, c'est par celui-ci que nous avons débuté. Il consiste à installer un câble jack mâle sur la rampe de lancement et de brancher la partie femelle au circuit pour mesurer la différence de potentiel à ses bornes. Une fois le câble débranché (décollage de la fusée), cette ddp devient non nulle et on lance alors un tempo qui, au bout du temps souhaité (temps de vol

de la fusée calculé dans le StabTraj), fait tourner le servomoteur qui retient la trappe du parachute, le libérant. Ci-dessous le schéma du circuit :

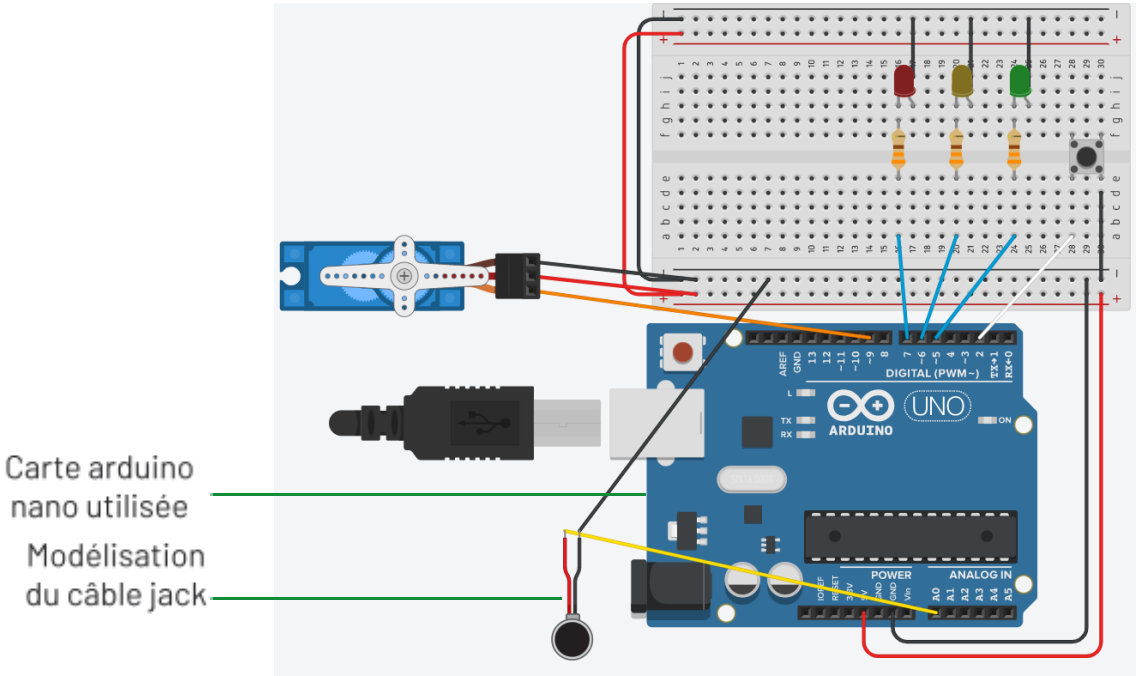


Schéma du circuit séquentiel

Avant de souder le circuit définitif avec des fils multibrins, nous avons testé le programme que nous écrivions au fur et à mesure sur un circuit test avec une carte Arduino Uno et une breadboard (nous utiliserons pour le circuit réel un shield avec horloge RTC et port SD). Cela nous a permis d'aboutir à un programme fonctionnel à la fin février, qui sera modifié par la suite sur des détails. Les programmes définitifs sont disponibles en annexe. Nous avons ensuite soudé les composants, et le circuit séquentiel était prêt début mars.

En ce qui concerne le circuit expérimental, sa réalisation prit plus de temps, à cause de certaines difficultés rencontrées. Afin de mesurer la vitesse, nous avons utilisé un gyroscope Arduino MPU6050 pour sa compatibilité avec Arduino. Le travail consista à récupérer la mesure de l'accélération (en bits) que donne le composant, de la convertir et de l'intégrer pour accéder à la vitesse. Malgré certaines difficultés et grâce à l'aide de Jules Ferrand en particulier, nous avons abouti à une version de programme qui permet d'obtenir la vitesse de la fusée. Nous avons parallèlement remarqué que malgré la marque de l'accéléromètre, la compatibilité n'est pas idéale et les valeurs obtenues en perdent de la fiabilité.

Toutefois le deuxième circuit soudé, le système électronique de la minifusée fut prêt à être intégré dans le restreint, alors imprimé.

LE LANCEMENT

Le C'Space de 2022 s'est déroulé du 16 au 23 Juillet 2022. Jérémie et Eliott sont les deux membres à s'être rendus sur place. Les groupes disposent d'un atelier de travail pour continuer la finalisation de la fusée. En effet, chaque fusée est soumise à une qualification, notamment à un vol simulé. Si la fusée n'est pas qualifiée, le club doit réaliser des modifications.

Pour notre projet, nous sommes arrivés avec une fusée tordue au C'Space à cause de la chaleur sur le PLA. Nous avons détordu la fusée à l'aide d'un décapeur thermique, manœuvre très délicate. Nous avons dû limer les bagues moteurs pour qu'elles s'adaptent bien au moteur. Nous avons dû rajouter des extensions sur les ailerons à cause d'une erreur de calcul dans le Stabtraj qui fait que les ailerons étaient trop courts. Nous avons remodelé la trappe qui avait aussi été déformé par la chaleur. Après tout cela, la fusée à enfin été qualifiée, le lundi soir, pour un vol le mardi matin.

Le lancement de la fusée à eu lieu le 19 Juillet 2022 à 10h47. Le vol a été nominal, c'est une grande réussite. Le parachute s'est bien éjecté, malgré avoir pris un peu de temps pour se déployer car trop comprimé. La fusée est retombée trop vite à cause d'un sous dimensionnement du parachute. Elle s'est donc cassée en deux au niveau du restreint lors de l'impact au sol. La caméra embarquée n'avait plus de batterie donc elle n'a pas filmé le lancement.

Les pistes d'améliorations sont les suivantes : agrandir considérablement la trappe du parachute et la taille du parachute pour réduire la force de l'impact au sol, concevoir une trappe plus fiable sans avoir besoin de modeler à la main, rendre le système électronique plus facilement accessible, réussir à calibrer l'accéléromètre correctement pour avoir des bonnes mesures, ne pas oublier de vérifier que la caméra ait de la batterie, réussir à faire fonctionner le système électronique sur pile et non sur batterie externe, oublier le PLA et utiliser des matériaux plus résistants, mieux concevoir la stabilité pour avoir plus de marge sur les ailerons, être plus précis dans la découpe et dans le perçage des différents trous de la fusées qui étaient loins d'être parfait, faire une fusée avec un seul diamètre.

ANNEXE

Décodage couleurs des LEDS

Chaque étape est séparée par une pression sur le bouton du circuit concerné.

Circuit séquentiel

État initial *(vert, orange et rouge)* : servo en position ouverte, système prêt à recevoir le parachute et à refermer la trappe.

En attente *(vert)* **:** la trappe est verrouillée, le système est désarmé et prêt à être armé (en attente du débranchement du câble jack). La LED verte clignote trois fois pour passer de l'état initial à l'état en attente.

Armé *(vert et orange)* : le système est armé, en attente du débranchement du jack.

Ascension *(orange)* : le décompte est en cours, le parachute se déploiera à la fin quoi qu'il se passe.

Ouverture *(rouge)* : le servomoteur libère la trappe et le parachute se déploie. Revient à l'état initial après quatre secondes.

Circuit expérimental

Enregistrement *(bleu)* : en cours d'enregistrement. La LED bleu clignote trois fois avant de démarrer l'enregistrement.

En attente *(éteint)* : pas d'enregistrement, système expérimental en attente

Programme Arduino

Programme du circuit séquentiel

```
#include <Servo.h>
Servo servo;

int bouton = 3;
int ledverte = 5;
int ledjaune = 7;
int ledrouge = 9;
int pinservo = 11;
int etatbouton = 0;
int GO = 0;
int EnMarche = 0;

void setup() {


  servo.attach(pinservo);
  pinMode (bouton, INPUT_PULLUP);
  pinMode(A0, INPUT_PULLUP);
  pinMode(ledverte, OUTPUT);
  pinMode(ledjaune, OUTPUT);
  pinMode(ledrouge, OUTPUT);
  digitalWrite(ledverte, HIGH);
  digitalWrite(ledjaune, HIGH);
  digitalWrite(ledrouge, HIGH);

}

void loop() {

  etatbouton = digitalRead(bouton);

  if (EnMarche == 0) {
    if (etatbouton == LOW) {
      EnMarche = 1;
      servo.write(135);
      digitalWrite(ledjaune, LOW);
      digitalWrite(ledrouge, LOW);
      for (int n = 0; n < 2; n++) {
        delay(750);
        digitalWrite(ledverte, LOW);
        delay(750);
        digitalWrite(ledverte, HIGH);
      }
```

```
  }
 }
 else {

  if (etatbouton == LOW) {
   digitalWrite(ledjaune, HIGH);
   GO = 1;
  }


  float decollage = digitalRead(A0);

  if (EnMarche == 1 and GO == 1 and decollage > 0.00) {
   digitalWrite(ledverte, LOW);

   delay(8000);

   servo.write(60);
   digitalWrite(ledjaune, LOW);
   digitalWrite(ledrouge, HIGH);

   delay(4000);

   digitalWrite(ledverte, HIGH);
   digitalWrite(ledjaune, HIGH);
   EnMarche = 0;
   GO = 0;

  }


 }

 delay(100);
}
```

Programme du circuit expérimental

```
//LIBRARIES

//MPU
#include "MPU6050.h"
MPU6050 accelgyro;

//SD card
#include <OneWire.h>
#include <SPI.h>
#include <SD.h>
#include "Wire.h"
#include "RTClib.h"
```

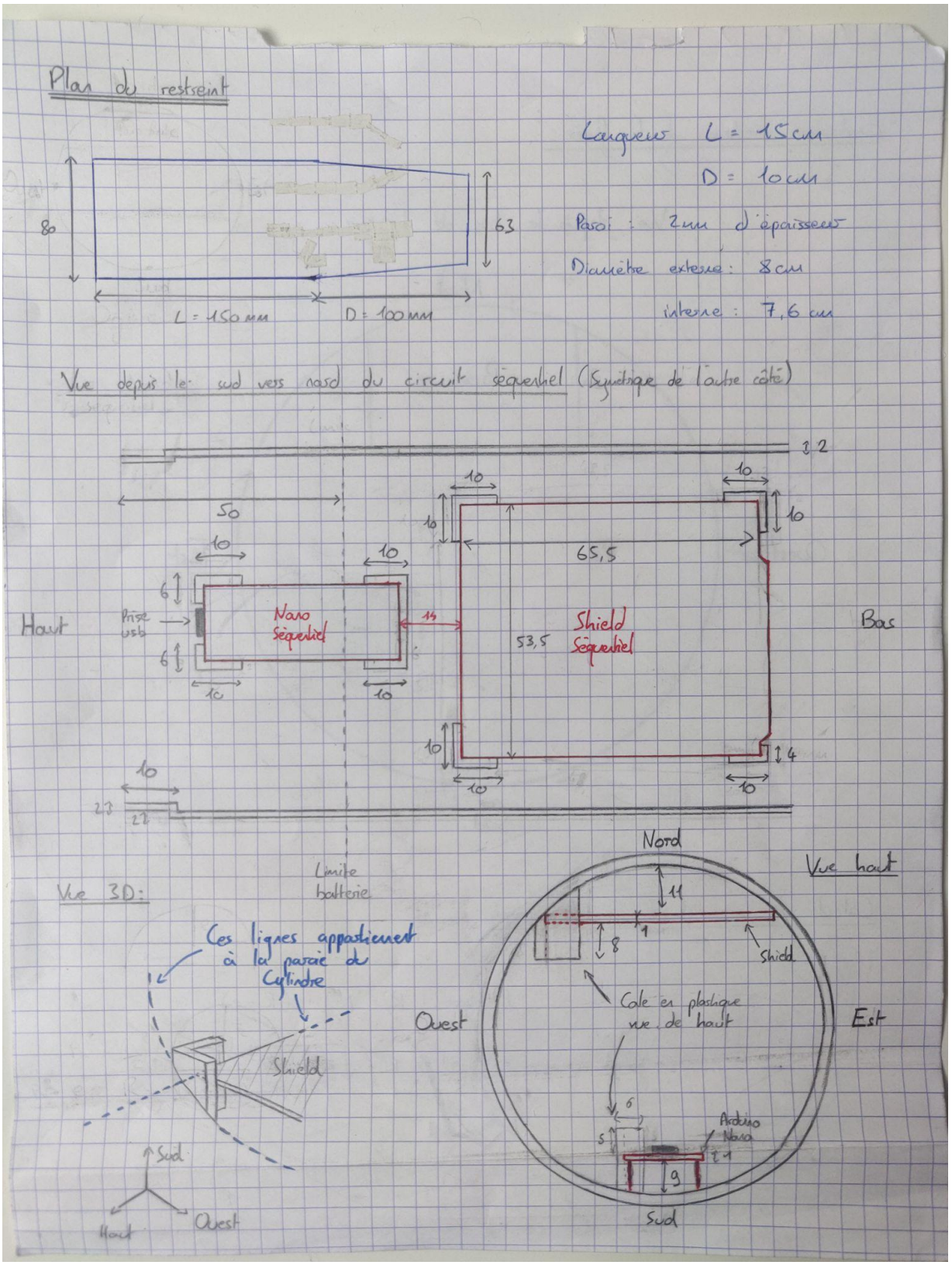
```
RTC_DS1307 RTC;
const int chipSelect = 10; //cs or the save select pin from the sd shield is connected to 10.
File dataFile;
DateTime now;

// variables de temps
float t0 = 0;              // temps de début des mesures
float t_old = 0;           // ancienne valeur de temps
float t_new = 0;           // nouvelle valeur de temps
float dt = 0;              // temps écoulé entre les deux plus récentes mesures

// PARTIE MESURE

const int MPU_addr = 0x68;
int16_t AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ;

int minVal = 265;
int maxVal = 402;

double x;
double y;
double z;

const uint16_t AccelScaleFactor = 2048;
float v_old = 0;
float v_new = 0;
float Ac[3];
float A=0;
float Ac0[3];
float g=0;
float AcY0=0;        // résultats aberrants si AcY0 déclaré en integer

int first = 1;

// FIN PARTIE MESURE

// PARTIE DIALOGUE

int led = 3;
int bouton = 5;
int etatbouton = 0;
int GO_mesure = 0;
int NOGO=0;

// FIN PARTIE DIALOGUE


void setup() {
```

```
  MPU6050 mpu;
  Serial.println(mpu.getFullScaleGyroRange());
  Serial.println(mpu.getFullScaleAccelRange());
  mpu.setFullScaleGyroRange(MPU6050_GYRO_FS_2000);
  mpu.setFullScaleAccelRange(MPU6050_ACCEL_FS_16);
  Serial.println(mpu.getFullScaleGyroRange());
  Serial.println(mpu.getFullScaleAccelRange());


  Wire.begin();
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  Serial.begin(9600);
  //DIALOGUE
  pinMode (bouton, INPUT_PULLUP);

  // SD card
  if (!SD.begin(chipSelect)) {
    Serial.println("Card failed, or not present");
    NOGO=1;
    // don't do anything more:
    return;
  }

  Serial.println("card initialized.");
}

void loop() {

  while (NOGO==1); {
    digitalWrite(led, LOW);
  }

  // ATTENTE GO_MESURE

  while (GO_mesure == 0) {
    etatbouton = digitalRead(bouton);
    if (etatbouton == LOW) {
      GO_mesure = 1;
      first = 1;
      for (int n = 0; n < 3; n++) {
        digitalWrite(led, HIGH);
        delay(750);
        digitalWrite(led, LOW);
        delay(750);
      }
    }
  }
```

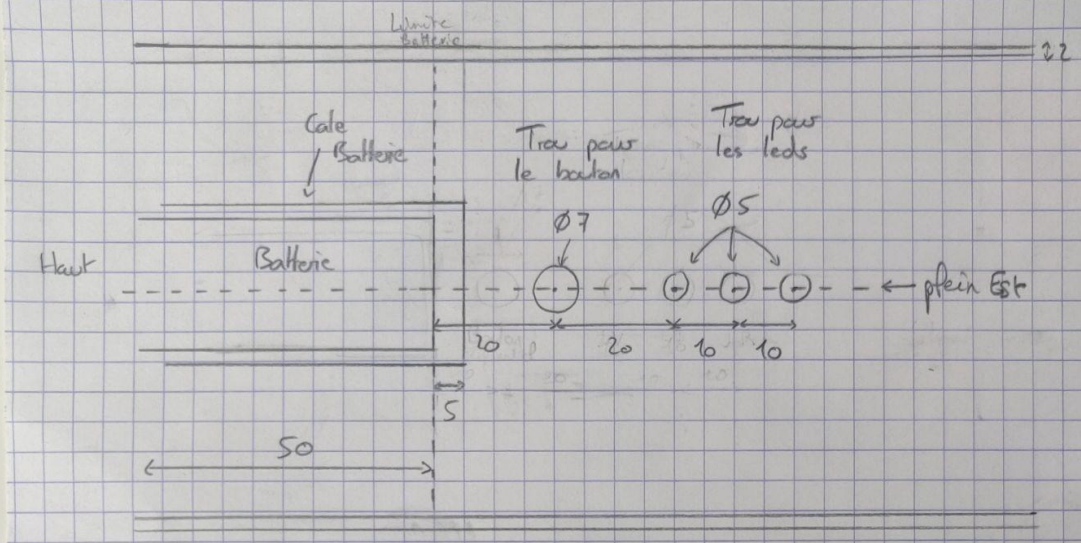
```
  digitalWrite(led, HIGH);
  t0=millis();



  // MESURE

  while (GO_mesure == 1) {

    now = RTC.now();        // indispensable sinon dataFile non créé
    dataFile = SD.open("vitesse.csv", FILE_WRITE);      //création fichier txt


    Wire.beginTransmission(MPU_addr);
    Wire.write(0x3B);
    Wire.endTransmission(false);
    Wire.requestFrom(MPU_addr, 14, true);
    AcY = Wire.read() << 8 | Wire.read();
    t_old = t_new;      // actualisation
    t_new = millis();   // nouvelle valeur de temps
    dt=t_new-t_old;

    Serial.println("AcY = "); Serial.println(AcY);




    if (first == 1) {                   // Etalonnage sur la première valeur de l'accélération mesurée
      Serial.println("Etalonnage");
      first = 0;
      AcY0=AcY;
      dt=0;       // ainsi v_new=v_old=0 car sinon dt = 7s pour la première itération (hypothèse que
rien ne se passe au début)
    }


    v_old = v_new;
    v_new = v_old - (AcY-AcY0)/AccelScaleFactor * 9.81 * dt/1000;          // dt/1000 car dt en ms ;
signe "-" : accélération par rapport au référentiel d'une chute libre + mpu avec la tête en bas
    Serial.print("VitesseY= "); Serial.println(v_new); Serial.println(t_new-t0); Serial.println(dt);

    Serial.println("----------------------------------------");

    // if the file is open :
    if (dataFile) {
       dataFile.print(v_new);          // Vitesse
       dataFile.print(";");
       dataFile.println(t_new-t0);     // temps écoulé depuis le début des mesures

       dataFile.close();
```

```
      // print to the serial port too:
      Serial.println("data stored");

    }
    // if the file isn't open, pop up an error :
    else {
      Serial.println("error opening datalog.txt");
    }

    etatbouton = digitalRead(bouton);
    if (etatbouton == LOW) {
      GO_mesure = 0;
      v_old=0;
      v_new=0;
      first=1;
      t0=0;
      t_old=0;
      t_old=0;
      dt=0;
      digitalWrite(led, LOW);

      dataFile.close();
      Serial.println("data stored");

      delay(2000 - dt);
    }

    // FIN PARTIE MESURE
  }
}
```

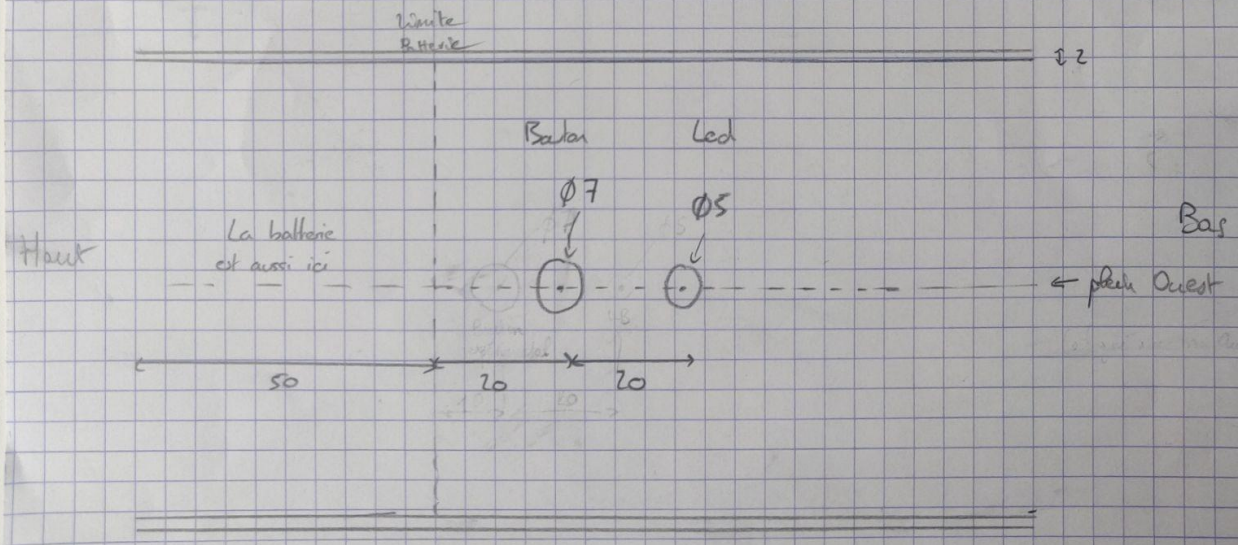
Plans papiers



Vue Est :



Vue Ouest :



Vue Sud
en coupe du milieu

Ogive

Passie Sud

2

2

10

22

Ouest

Est

Caisson batterie

5

10

63mm

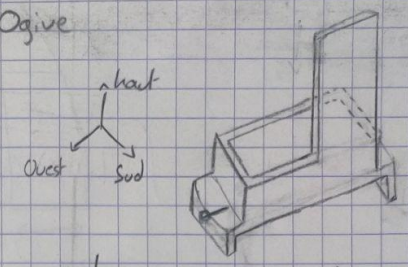
2

Limite batterie

Restreint

63
91
22

Vue 3D :



limite ogive restreint

Bas

10

Vue Est

Caisson

batterie

Nord

Sud

5

22mm

Nord

Ouest

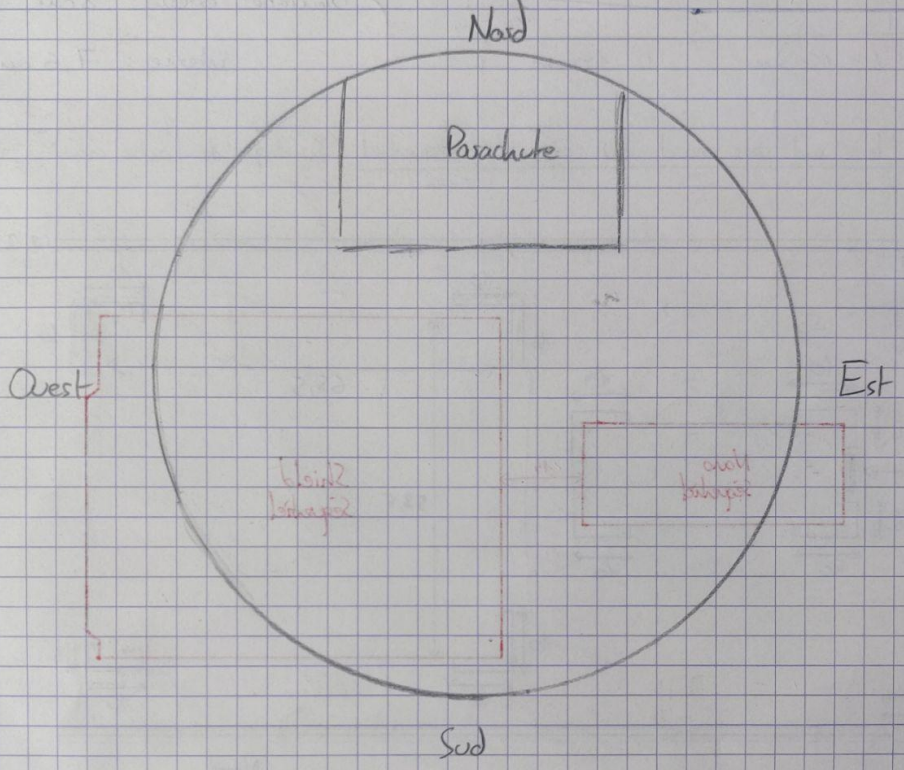
Vis

Vis Est

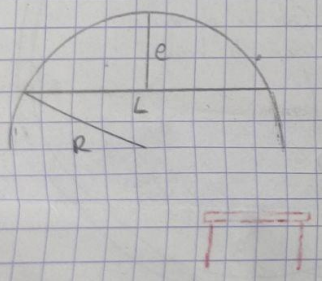
Sud

Explication orientation de la fusée :

Vue depuis le bas de l'ogive :

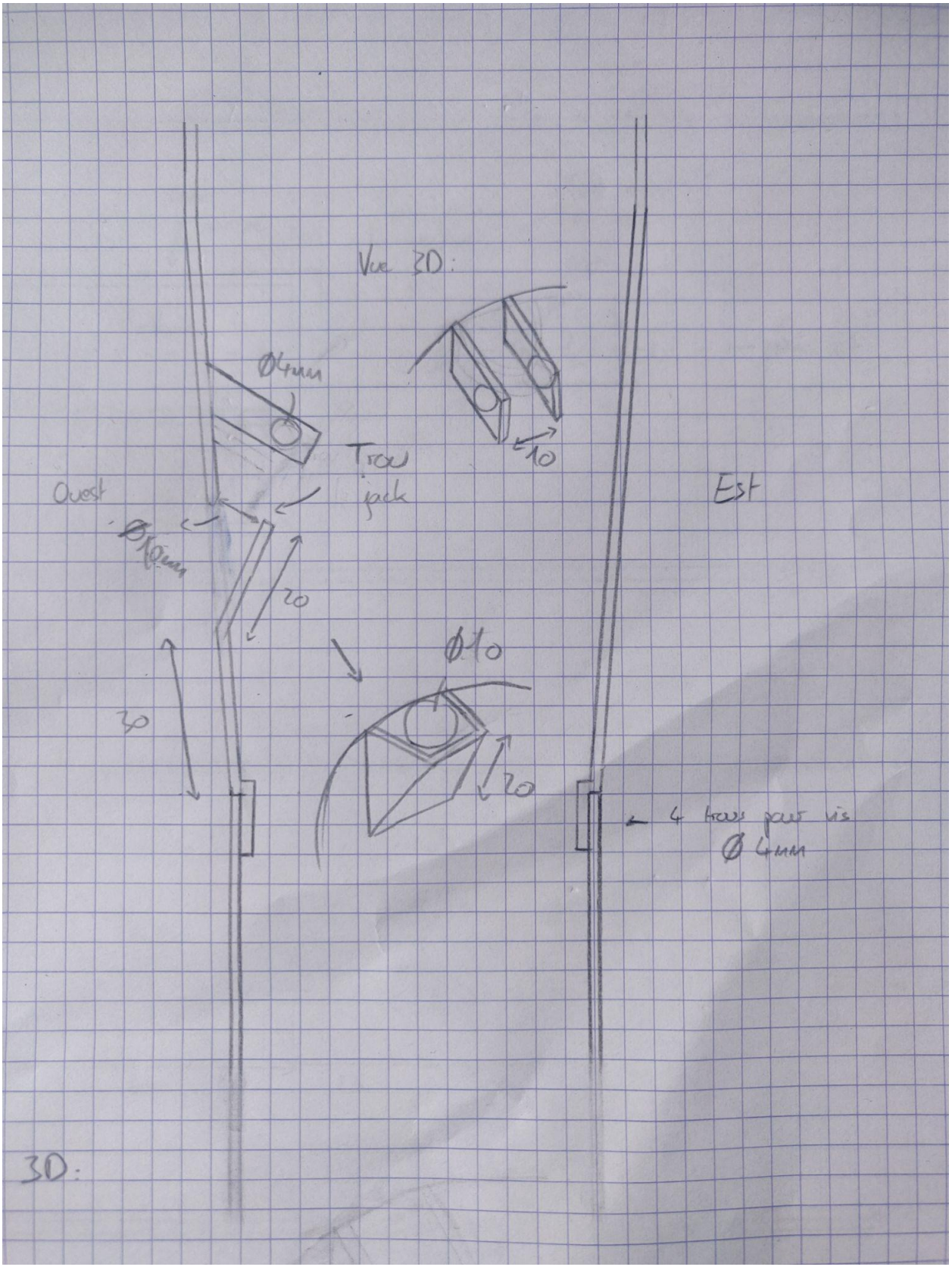


Calcul de la distance à la paroi



Pythagore : $R^2 = \left(\frac{L}{2}\right)^2 + (R-e)^2$

$\Rightarrow e = R + \sqrt{R^2 - \frac{L^2}{4}}$


Vue 3D:
Ø4mm
Trou jack
Ouest
Est
Ø10mm
20
Ø10
30
20
10
4 trous pour vis Ø 4mm
3D: