

Revue de projet et d'équipe - Projet Abyssé

Projet technique de l'ingénieur

Groupe 83 : E. Wolff, S. Khazari, A. Polette (relecteur), P. Gonot, T. Bruyère, G. Pauliat

Encadrant : François Le Pennec,

Destinataires : François Le Pennec, Bernard Abiven, Magalie Le Gall

Version 2.0
IMT Atlantique - FISE A1

27 Mars 2025



Sommaire

1. Présentation du projet

- 1.1 Planification et écarts
- 1.2 Tableau des livrables

2. Avancement du projet

- 2.1 Mécanique et CAO
- 2.2 Électronique embarquée
- 2.3 Site web : développement logiciel

3. Bilan

- 3.1 Fonctionnement de l'équipe
- 3.2 Conclusion et perspectives

1 - Présentation du projet



FIGURE 1: Sous marin DIY en Lego avec ajustement mécanique de l'assiette [1]



FIGURE 2: Projet *Kosmos* : caméra sous marine [2]

Notre cahier des charges succinct

Nos objectifs et exigences :

1. Piloter le sous marin à distance et le contrôler (profondeur et déplacement) de manière précise.
2. Transmettre une vidéo en direct à l'utilisateur pour observer les fonds marins

Sous systèmes du sous-marin :

- **Alimentation** : pack batterie Li-ion 18650 et BMS
- **Ballast** : gestion de la profondeur avec remplissage de seringues
- **Moteur et gouvernes** : déplacement et propulsion du sous-marin
- **Communication sans fil** : récepteur et émetteur radiocommande

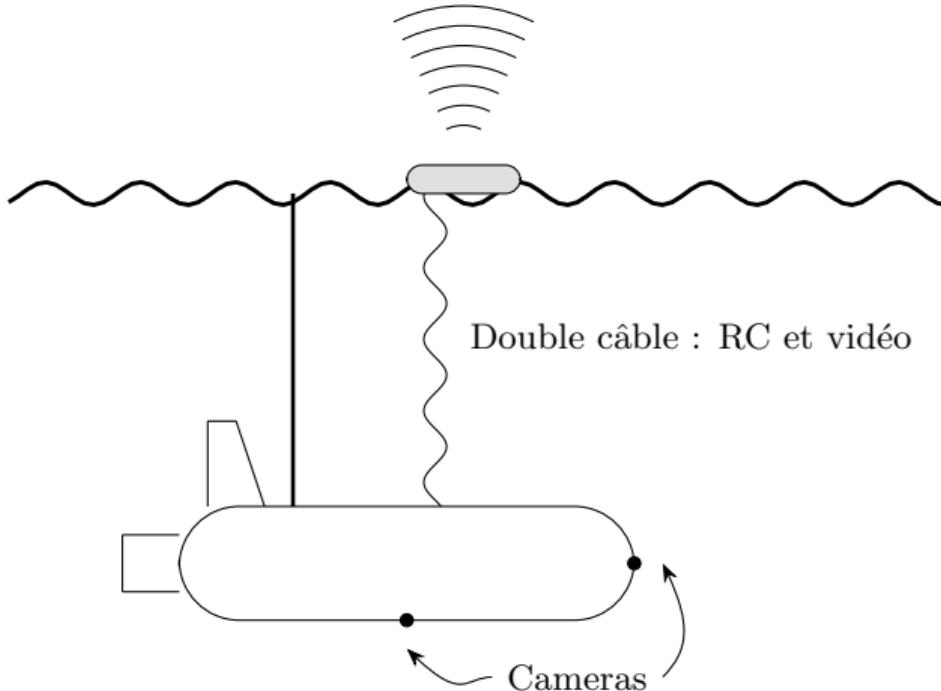


FIGURE 3: Schéma de principe

Liens avec le sous marin :

- Ligne de vie
- Câble coaxial permettant la transmission de signal de commande (2,4GHz) et signal vidéo (5,8 GHz).

↑ *t*

Transmission vidéo et RC		P2
Programmation de l'Arduino et adaptations		P2
Montage du sous-marin avec composants	Usinage des pièces mécaniques	P2
CAO complète	Création des circuits électroniques	P2
Phase de cadrage	Commande des pièces	P1

FIGURE 4: Étagement du projet

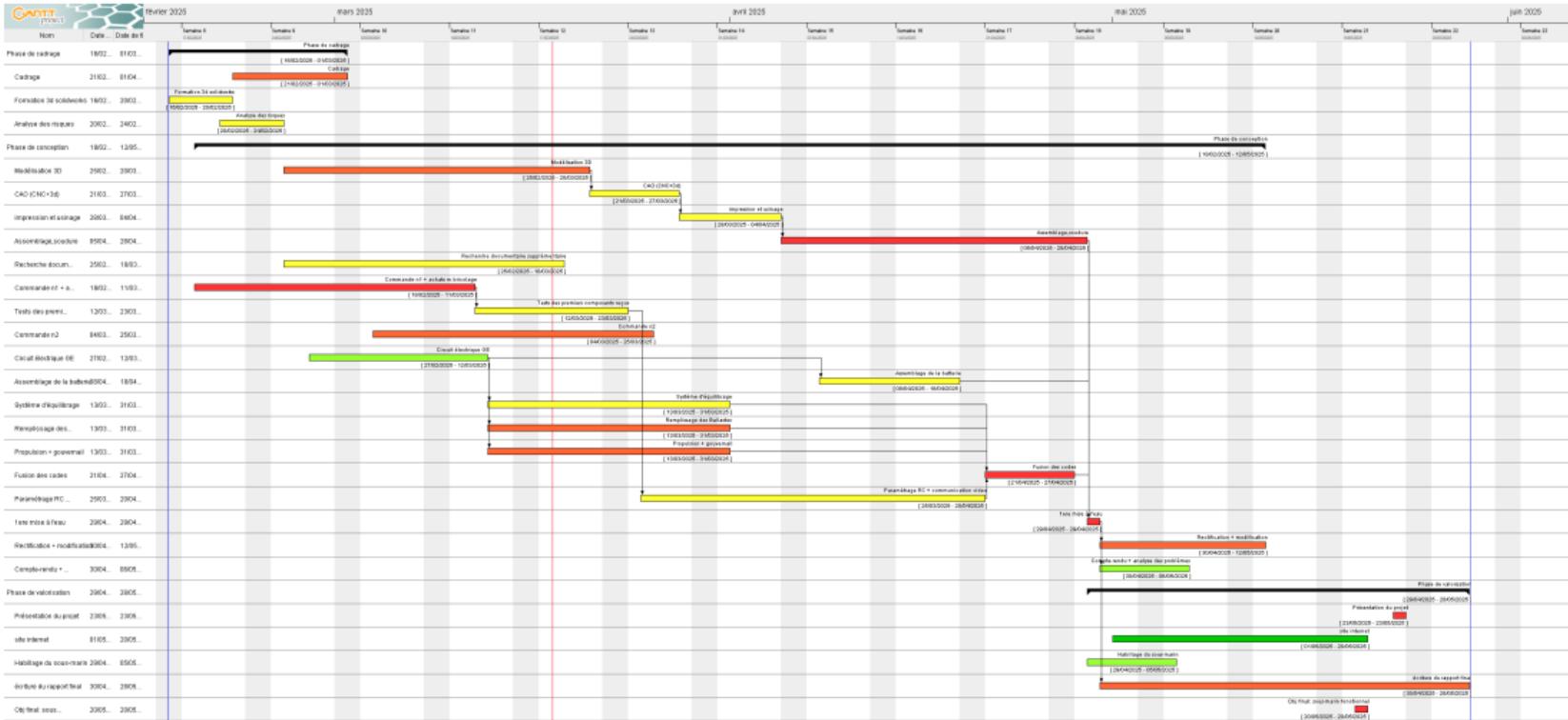
Organisation du groupe Pronto 83 :

Gr.1 Saad et Emilien : pôle mécanique et conception

Gr.2 Antonin et Pierre : pôle électronique/programmation sur la propulsion

Gr.3 Thomas et Gautier : pôle électronique/programmation et étude des ballasts

1.1 - Planification et écarts



Remise à niveau des risques

Intitulé	Solution	Précédent	Revalorisation
Problématiques non anticipées	Réévaluation des jalons		
Utilisation de Solidworks	Retour sur Fusion 360		
Conflit interne	Réunion ou temps à part		
Répartition des tâches	Réunion/ réattribution		

Table 1: Tableau des risques

1.2 - Tableau des livrables

Livrable	Date
Document de cadrage	27 février
Revue de projet et d'équipe	27 mars
Rapport technique	Environ 1 semaine avant le 23 mai
Livrable, jalon Forum et présentation du projet	23 mai

Table 2: Livrables et leurs dates

2.1 - Avancement du projet : partie mécanique

Ce qui a été fait jusqu'à présent :

- Identification des besoins en composants et matériaux + **commandes pièces**
- Modélisation 3D cinématique paramétrée **complète** du sous marin ($\approx 10\text{h}$ de travail)
- Calcul du lest nécessaire pour assurer l'équilibre statique → adaptation du modèle
- Début des **usinages**

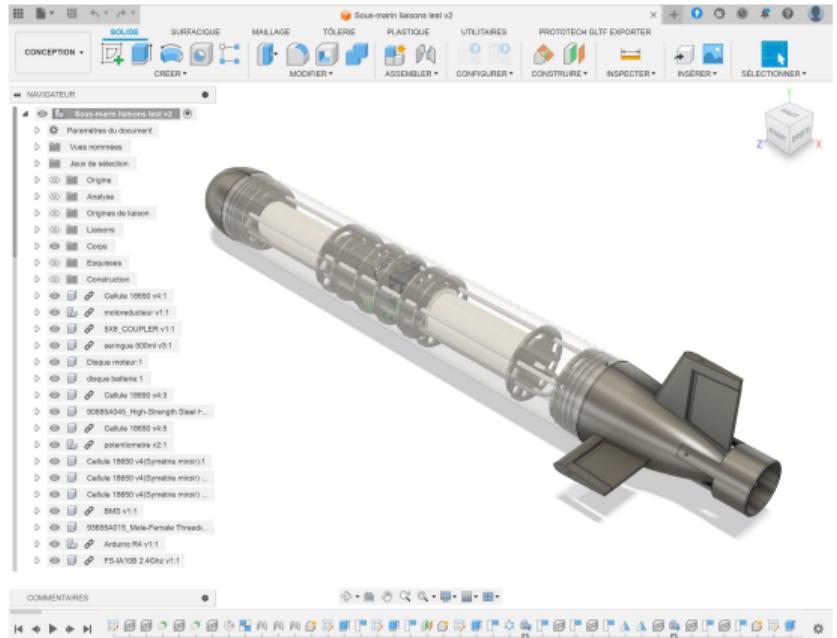
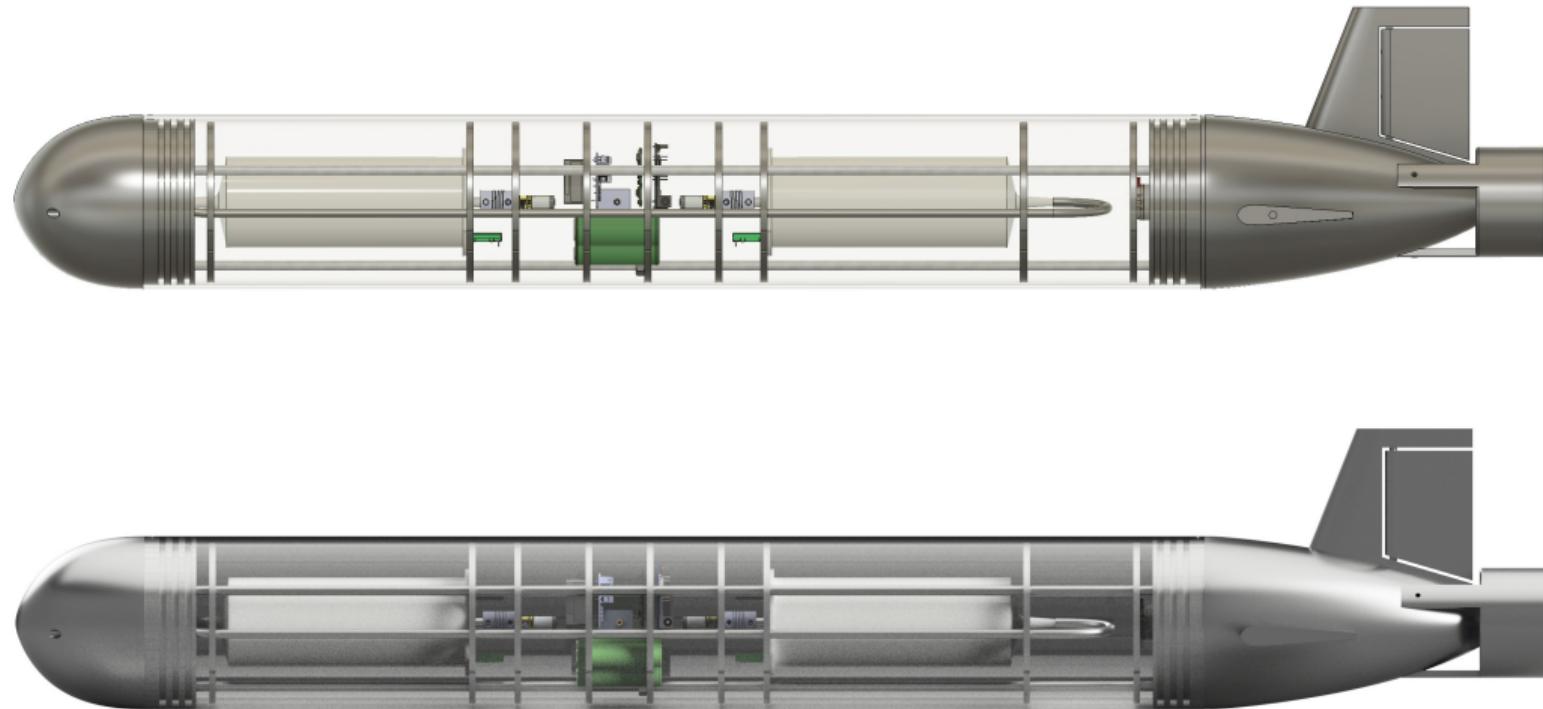


FIGURE 5: CAO paramétrique *Fusion 360*



Les parties qui ont nécessitées une attention particulière ...

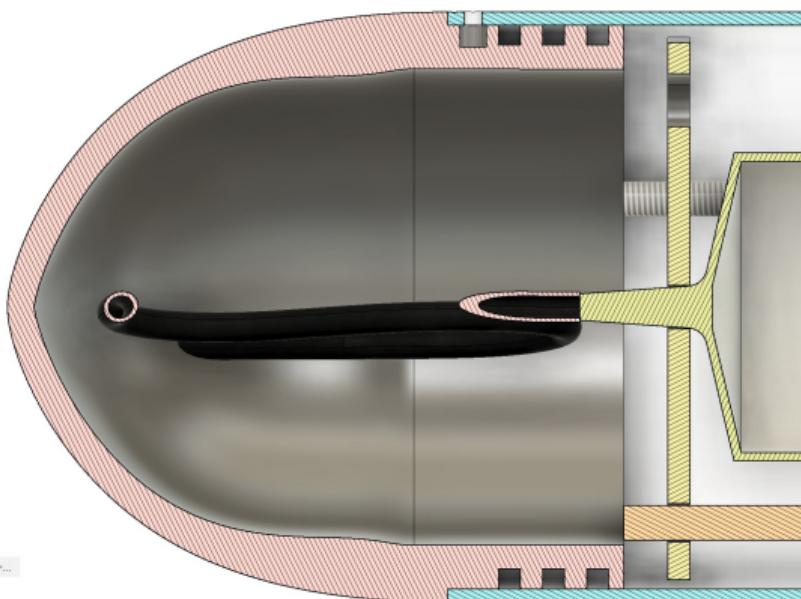


FIGURE 6: Solution de jointage étanche choisie [3]

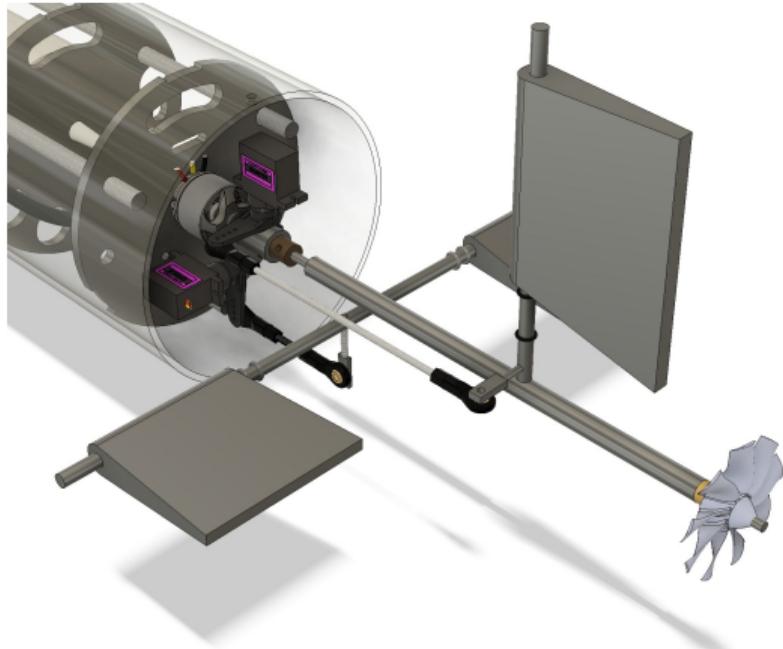


FIGURE 7: Solution technique pour les gouvernes

Dimensionnement du lest

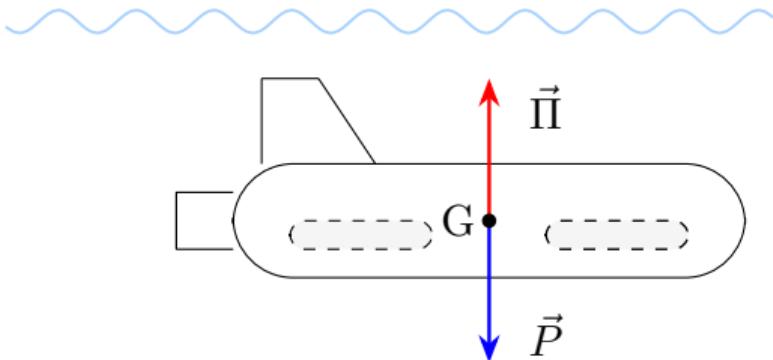


FIGURE 8: Schéma de l'étude

Objectif : Déterminer la masse m_{lest} à ajouter au sous marin pour garantir l'équilibre

On applique le *Principe Fondamental de la dynamique* sur \vec{z}

$$0 = \Pi - P = \rho_{\text{fluide}} V g - mg$$

$$N = \frac{4(\rho_{\text{fluide}} V - m_{\text{sans}})}{\rho_{\text{fer}} \pi D^2 L}$$

On peut donc connaître le nombre de barres $\varnothing 20\text{mm}$ à insérer dans le tube.

Mécanique : les problèmes et sacrifices !



FIGURE 9: Disques imprimés en 3D

Grand challenge du projet : faire une conception en partant d'une feuille blanche

- Découpe laser de PMMA ratée → **impression 3D** et inserts filetés
- Penser la séquence d'assemblage en **amont** (commande des gouvernes, ...)
- Utilisation d'un bloc moteur étanche MAIS **trop cher** → penser une nouvelle solution technique atteignable avec notre budget

2.2 - Avancement du projet : partie électronique

Pour chaque composant:

- Première manipulation
- Prise en main du fonctionnement
- Test de fonctions

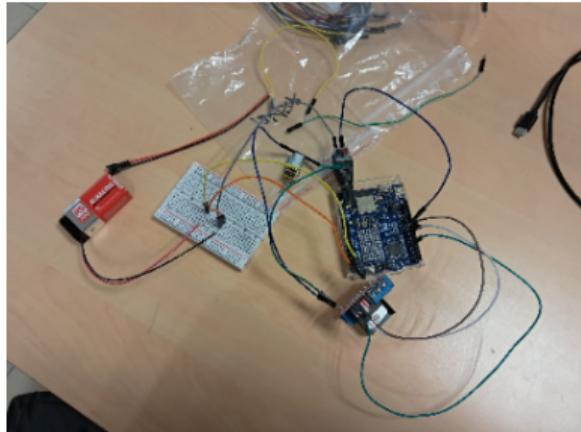


FIGURE 10: Montage de test

Composants testés:

- (ballasts) Mini-moteurs
- (ballasts) Driver mini-moteurs
- (direction) Servomoteurs
- (ballasts) Potentiomètres linéaires
- (propulsion) Moteur brushless
- (propulsion) Driver moteur brushless
- (équilibrage) Centrale inertielles

Ce qui a été fait jusqu'à présent :

- Circuit électrique de remplissage des ballasts et de l'asservissement afin d'assurer le maintien de l'assiette
- Code pour le remplissage des ballasts (cf. ANNEXE 4)
- Base du code pour l'asservissement de l'assiette (paramètres à modifier expérimentalement) (cf. ANNEXE 5)

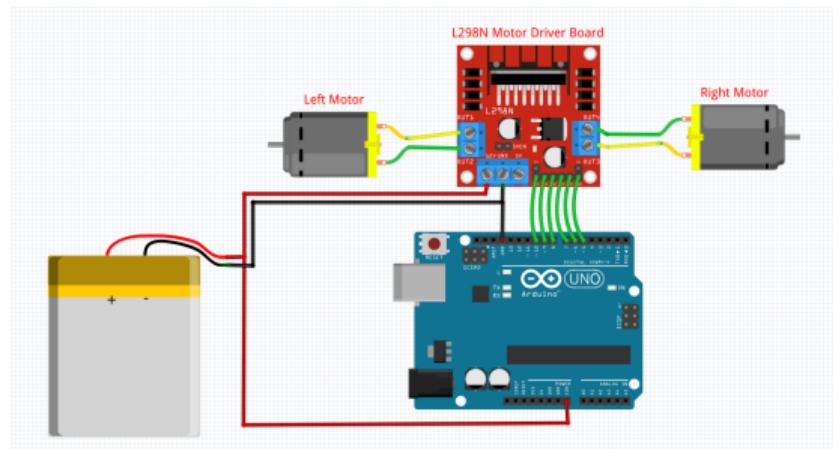


FIGURE 11: Circuit de remplissage des ballasts

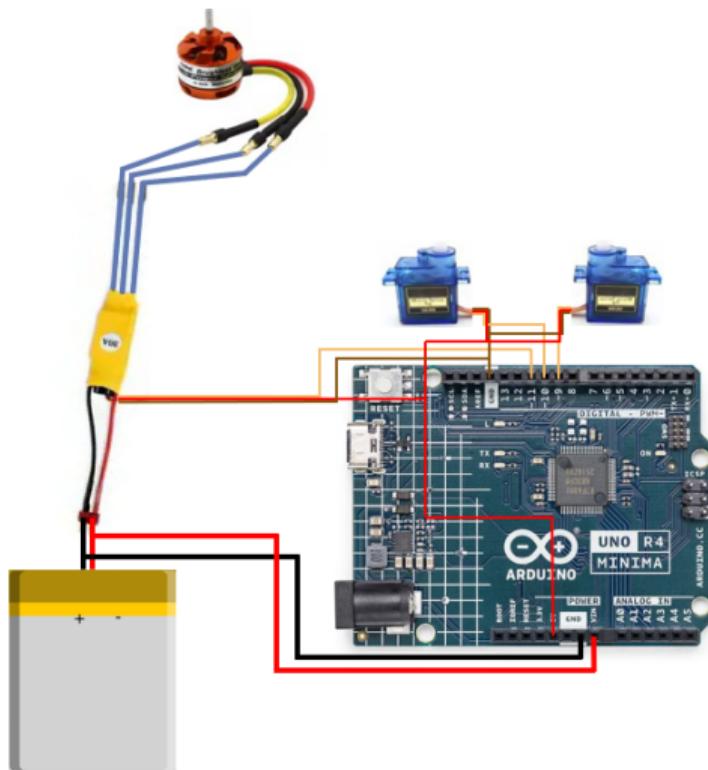


FIGURE 12: Circuit pour la propulsion et la commande des gouvernes

Problèmes rencontrés

Pilotage des servomoteurs

Pilotage des servomoteurs problématique:
contrôle en vitesse et non en angle

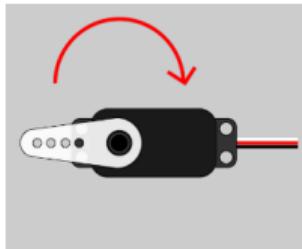


FIGURE 13: Servo

Solution :

- Faire des tests pour déterminer la position
- Emprunter des servos au fablab (même taille et contrôle en angle)

Matériel à disposition

- Contraintes de temps
- Contraintes de budget
- Contraintes de provenance

Solution :

- Commande autonome
- Blocs pas préfaits

Circuit vidéo

Plusieurs enjeux dans le choix de la caméra:

- étanchéité
- prix
- flux vidéo
- compatibilité avec câble

L'ajout de la caméra est donc en réflexion pour le moment et pourrait constituer un lot facultatif



FIGURE 14: Caméra

Solution :

Recherche de solutions plug and play abordables implantées sur les drônes FPV

2.3 - Avancement du projet : partie interface web [3]

The screenshot shows the homepage of the "Projet Abyss" website. The header includes the logo "Abysse" and navigation links for Home, About, and Contact. The main title "Projet Abyss" is displayed prominently. A message encourages users to stay connected for future updates. Below the message is a countdown timer showing 58 Days, 13 Hours, 39 Minutes, and 40 Seconds. At the bottom, there are social media icons for X, Facebook, Instagram, and LinkedIn. The background features a dark image of a submarine.

Abysse

Projet Abyss

Nous travaillons toujours sur notre site web. Restez connectés pour les prochaines mises à jour !

58 Days 13 Hours 39 Minutes 40 Seconds

X Facebook Instagram LinkedIn

Objectifs du projet Abyss :

- Concevoir un sous-marin qui puisse plonger et remonter à la surface, précisément et selon les commandes de l'opérateur
- Le sous-marin doit intégrer une caméra permettant la transmission vidéo en direct à l'utilisateur

Le présent projet ingénieur de la mission Pronto vise à l'élaboration d'un drone sous-marin. Dans le contexte maritime présent aux abords de l'école, un tel outil est justifiable pour la surveillance des fonds marins. De plus, l'élaboration d'un tel projet en groupe de 6 nécessite diverses compétences relatives à l'organisation d'un projet, les problématiques de systèmes embarqués, électronique et télécommunication, domaines fortement inscrits dans la formation de l'école d'Ingénieur IMT Atlantique.

↑

Design de la page de présentation

The screenshot shows the Figma application interface with a dark-themed design. On the left, the sidebar displays the project structure, including 'Assets' and 'Page 1'. The main canvas features a grid-based layout for a presentation slide. The slide content includes:

- Project Name:** Text at the top.
- Our Mission:** A section with text and a small icon.
- Architecture:** A section with text and a small icon.
- Explanations:** A section with several text blocks and icons, including:
 - A purple icon with text: "Imagine bien, car c'est là que l'on démarre".
 - A blue icon with text: "Notre logo ressemble à un scorpion avec un fond bleu derrière le logo".
 - A green icon with text: "Notre logo ressemble à un scorpion avec un fond bleu derrière le logo".
 - A yellow icon with text: "On est des moustiques de l'eau".
 - A red icon with text: "Faut pas oublier la boussole derrière je suppose".
 - A pink icon with text: "Si j'en ai un, je laisse sans impression dans entre les deux animations".
 - A brown icon with text: "Sous-marin au niveau de l'eau (pas plongé en corps avec horde au dessus)".
 - A grey icon with text: "Faut pas oublier la boussole derrière je suppose".
 - A white icon with text: "Sous-marin qui plonge et qui remonte et qui plonge et qui remonte quand on bouche une des 3 portes dans le bateau et porte trois au dessus et porte trois qui décollent".
 - A light blue icon with text: "Membres de l'équipe".
 - A teal icon with text: "Comment faire l'opposition de l'opposition ?".
 - A light green icon with text: "Expérience de contact et de non-contact avec l'eau (c'est ce que le bateau a fait ou ce qu'il va faire) et l'opposition de l'opposition".

The right side of the interface shows the 'Design' tab with various settings and color swatches for the selected blue color.

3.1 - Fonctionnement de l'équipe

Les caractéristiques de la team :

- Équipe soudée et totalement engagée
- Organisation rigoureuse
- Utilisation d'outils collaboratifs
- Distribution des rôles en début de séance/phase

Actualisation des risques :

- Équipe rassurée par l'avancement du projet
- Parallélisation des tâches efficace



3.2 - Conclusion et perspective

Succès

- Bonne avancée
- Efficacité de l'équipe

Perspectives à court terme

- Montage mécanique et montage électronique
- Tests d'étanchéité
- Site web
- Programmation



Merci pour votre écoute !

Membres du groupe 83 :

Antonin Polette, Gautier Pauliat, Pierre Gonot
Thomas Bruyère, Saad-Eddine Khazari, Emilien Wolff

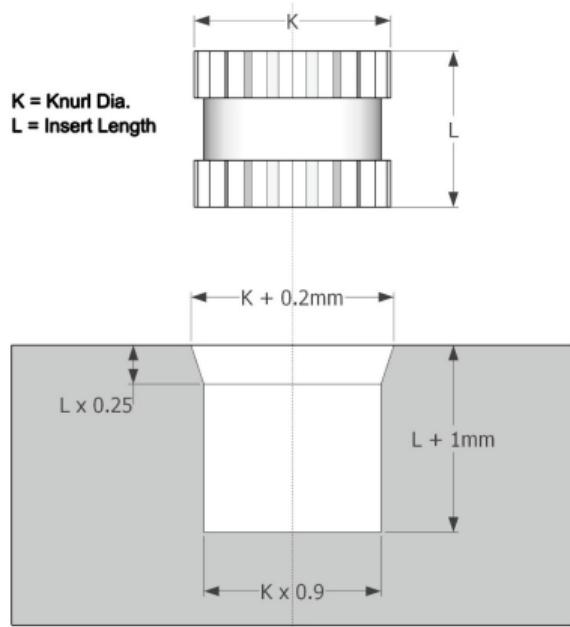
Annexes

1 - Références du projet

- [1] Brick experiment Channel, *Building a Lego-powered Submarine 4.0 - automatic depth control*, Youtube, 16 juil. 2022.
<https://www.youtube.com/watch?v=KLEH8RJsYgI>
- [2] Antoine Géju, *Projet Kosmos*, 2025. <https://kosmos.fish/index.php/faq/>
- [3] Saad-Eddine Khazari, *Site Web du projet Abyssé*, 2025.
<https://abyssé.netlify.app/>
- [4] Diy Perks, *Building a DIY submarine - DIY Perk*, 2025.
<https://www.youtube.com/watch?v=pUba126uzvU>
- [5] Norelem, *Informations techniques pour le joint torique*, 2025.
<https://urlr.me/eW8s36>

2 - Utilisation d'inserts filetés

Pour assurer le bon maintien des composants, on a utilisé des inserts filetés



Utilisation du PLA (polylactic acid) :

- Facilité d'impression : faible retrait thermique, bonne adhérence au plateau
- Écologique et biodégradable : fabriqué à partir d'amidon de maïs ou de canne à sucre
- Étanche (testé en conditions extrêmes)

FIGURE 15: Préconisation de modélisation

3 - Test étanchéités



FIGURE 16: Test d'étanchéité du filament
d'impression 3D

Test durant 24h au Dellec

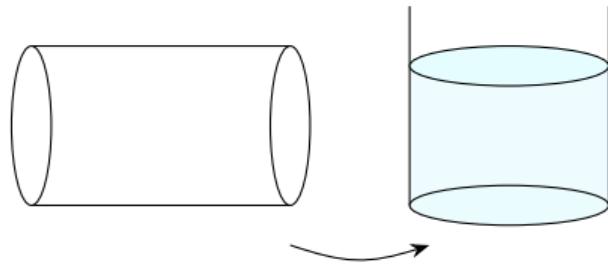


FIGURE 17: Schéma du montage

4 - Code Arduino ballast

asservissement.ino asservissement.h ●

```
1 #include <Wire.h>
2
3
4
5 void read_angles();
6
7 void fill_ballast(bool front, float filling_percentage);
8
9 float ideal_percentage(int angle);
10
11 float read_ballast(bool front);
12
13 void correct_angle();
```

FIGURE 18: Fichier header du code Arduino pour les ballasts

```
asservissement.ino asservissement.h
1 #include <Wire.h>
2 #include "asservissement.h"
3
4
5 // Adress I2C du MPU6050
6 const int MPU_addr = 0x68;
7 int16_t AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ;
8 int minVal = 265;
9 int maxVal = 402;
10
11
12 void setup() {
13     // Initialisation de la communication I2C
14     Wire.begin();
15     Wire.beginTransmission(MPU_addr);
16     Wire.write(0x6B); // Registre de configuration du MPU6050
17     Wire.write(0); // Réveille le MPU6050
18     Wire.endTransmission(true);
19
20     // Initialisation de la communication série pour l'affichage des résultats
21     Serial.begin(9600);
22 }
23
24
25 int read_angles(){
26     // Demande de lecture des données des capteurs du MPU6050
27     Wire.beginTransmission(MPU_addr);
28     Wire.write(0x3B); // Registre de début des données de capteur
29     Wire.endTransmission(false);
```

FIGURE 19: 1ere partie du code Ballast

5 - Code Arduino ballast (suite)

```
asservissement.ino asservissement.h
29 // Lecture des 14 registres
30 Wire.endTransmission(false);
31 Wire.requestFrom(MPU_addr, 14, true); // Lecture des 14 registres
32
33 // Lecture des valeurs de l'accéléromètre et du gyroscope
34 AcX = Wire.read() << 8 | Wire.read();
35 AcY = Wire.read() << 8 | Wire.read();
36 AcZ = Wire.read() << 8 | Wire.read();
37 Tmp = Wire.read() << 8 | Wire.read();
38 GyX = Wire.read() << 8 | Wire.read();
39 GyY = Wire.read() << 8 | Wire.read();
40 GyZ = Wire.read() << 8 | Wire.read();
41
42 // Calcul des angles de rotation
43 int xAng = map(AcX, minValue, maxValue, -90, 90);
44 int yAng = map(AcY, minValue, maxValue, -90, 90);
45 int zAng = map(AcZ, minValue, maxValue, -90, 90);
46
47 return yAng;
48 }
49 float ideal_percentage(int angle){
50 //Renvoie le pourcentage idéal de remplissage du ballast pour corriger l'angle
51 //Changer la formule théorique
52 return float(abs(angle)) / 100;
53 }
54
55 float read_ballast(front = true){
56 //Returns the filling percentage of selected ballast (front or back)
```

FIGURE 20: 2eme partie du code Ballast

```
asservissement.ino asservissement.h
53 }
54
55 float read_ballast(front = true){
56 //Returns the filling percentage of selected ballast (front or back)
57 //by reading the corresponding potentiometer value
58 return 0. // NOT COMPLETE
59 }
60
61
62 void correct_angle(){
63 //Check if the filling percentages of the ballast need adjustment
64 int angle = read_angles();
65
66 int threshold = 10; //minimum angle
67 float current_percentage = read_ballast(front = true);
68 if (angle > threshold){ //We need to fill the front ballast (or back one idk)
69 float filling_percentage = current_percentage + ideal_percentage(angle);
70 fill_ballast(front = true, filling_percentage = filling_percentage);
71 }
72 else if (angle < threshold){
73 angle = read_ballast(front = false);
74 float filling_percentage = current_percentage + ideal_percentage(angle);
75 fill_ballast(front = false, filling_percentage = filling_percentage);
76 }
77 // l'angle est compris entre -angle_min et angle_min, pas besoin de faire quoi que ce soit
78 }
79 }
```

FIGURE 21: 3eme partie du code Ballast