

PROJET ELECTRONIQUE ARDUINO - PEIP2

Année scolaire 2018-2019

" Arduinoglisseur "

Étudiants : Océane MOFID - Pierre POLETTE

Encadrants : Pascal MASSON

Industriel : SoFab

REMERCIEMENTS

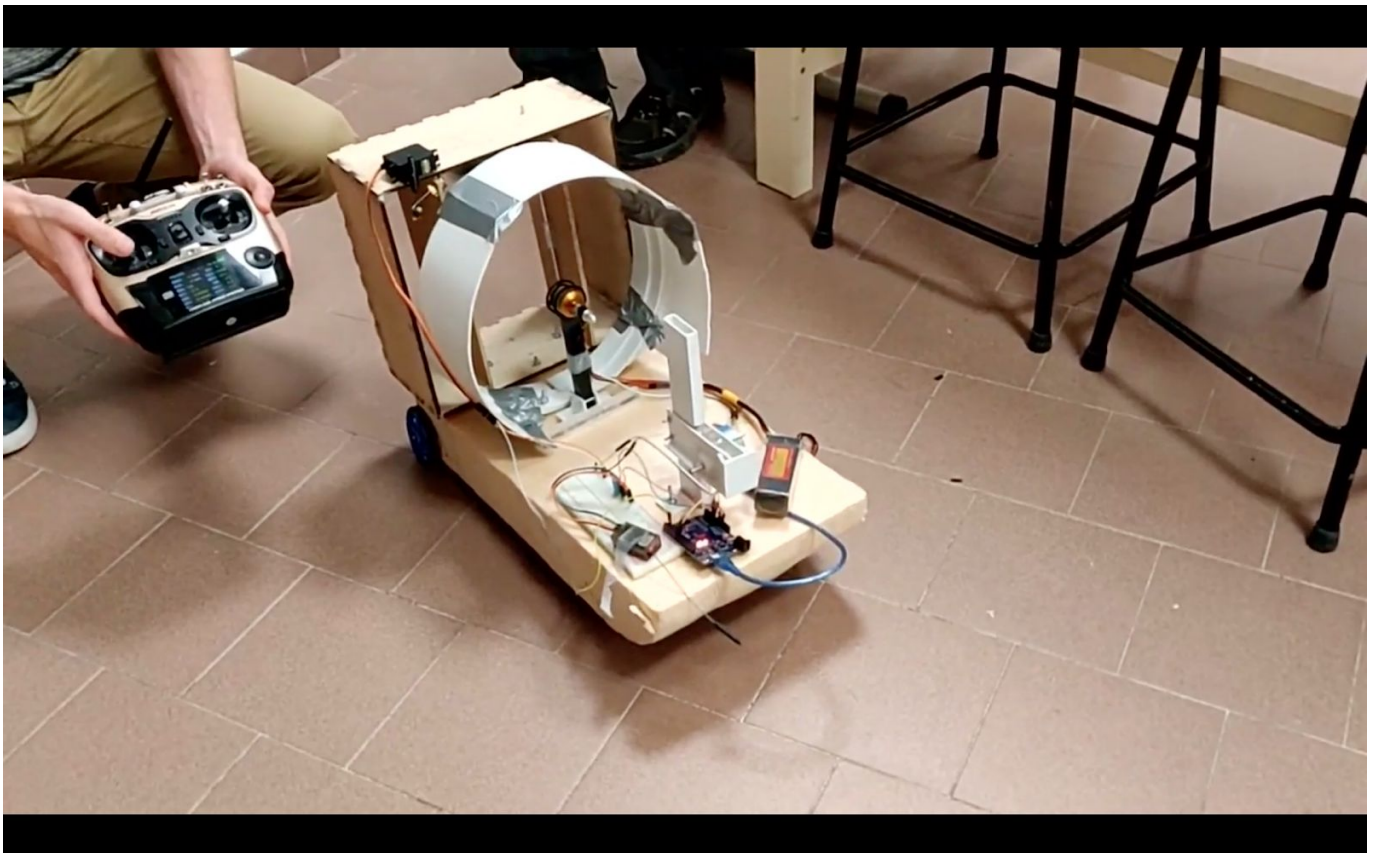
Nous tenons à remercier les personnes qui ont contribué à notre travail. Nous remercions évidemment Mr Masson, pour sa disponibilité, pour ses conseils ainsi que pour son aide précieuse; Mr Abderrahmane pour nous avoir accompagné pendant notre projet, ainsi que Mr Forner pour nous avoir donné l'accès à des appareils d'industrie.

SOMMAIRE

Introduction	8
Chapitre I : Définition du projet	9
I.1 - Idée globale	9
I.2 - Cahier des charges	9
I.3 - Planning	10
Chapitre II : Partie technique	11
II.1 - Hélice.....	11
II.2 - Gouvernail.....	12
II.3 - Crémaillère.....	13
Chapitre III : Partie électronique.....	14
III.1 - Bluetooth.....	14
III.2 - Radiofréquence.....	15
Conclusion.....	18
Bibliographie.....	19
Annexes	20

Introduction

Notre but était de réaliser un système complexe et communicant, d'imaginer et de mener à bien un projet jusqu'au prototype ainsi que de s'auto-former sans limite (à part celle du temps, de l'aspect financier ainsi que de la faisabilité). Nous nous sommes donc décidés sur un système nautique, un hydroglisseur. Une coque lisse lui permettant de glisser sur l'eau en utilisant la poussée de l'air provoquée par une hélice et un gouvernail qui le dirigera selon un moyen de télécommunication. La fonction de l'Arduinoglisseur sera de relever la température en profondeur grâce à une sonde accrochée à une crémaillère qui descendra grâce à un appel de commande.



Chapitre I : Définition du projet

I.1. Idée globale

Nous voulions réaliser un hydroglisseur à l'image des "airboats" que nous pouvons voir en particulier dans les Everglades.

Nous avons d'abord imaginé une structure en polystyrène. En effet ce matériau possède des propriétés très intéressantes, sa densité élevée permet à la fois l'étanchéité et la flottabilité de l'hydroglisseur, de plus ce matériau est léger et très peu coûteux, ce qui correspond à nos critères.

En nous renseignant sur différents projets d'hydroglisseur disponible sur les "open sources", nous avons donc pu lister le matériel dont nous avons besoin comme un moteur brushless à la place d'un moteur à courant continu (qui est moins adapté à cette utilisation) ainsi qu'un axe pour le moteur brushless.

L'hydroglisseur est donc composé de trois parties principales; l'hélice permettant l'avancée du projet, composée d'un moteur brushless, d'un axe moteur, d'une hélice et d'une enveloppe pour diriger un maximum d'air dans le gouvernail. Ce dernier est lui-même composé d'un servomoteur, de deux pales et d'une boîte pour structurer les deux éléments. La dernière partie est la crémaillère, commandée par un moteur pas à pas qui fait tourner un pignon pour faire descendre une sonde de température en profondeur.

I.2. Cahier des charges

Notre projet avait pour objectif principal de se déplacer de manière relativement simple et fluide via la communication Bluetooth ainsi que la communication RF. L'hydroglisseur devait également être capable de relever la température en fonction de la profondeur de manière autonome.

A la fin des séances dédiées au projet l'hydroglisseur se déplace simplement via la communication RF. Cependant nous n'avons pas pu réaliser des tests en milieu aquatique car l'étanchéité des composants n'a pu être effectuée par manque de temps. Nous avons tout de même tester la flottabilité, en laissant le projet immobile, les résultats ont été satisfaisants.

Le moteur pas à pas commandant la crémaillère, ainsi que la sonde n'ont pu être commandés par RF mais seulement par Bluetooth, nous n'avons pas les connaissances nécessaires combinées à un manque de temps.

I.3. Planning

SEMAINE DU PROJET :		dates des séances d'Arduino	DEC.		JANV.		FEV.			MARS
			10	17	7	14	5	13	27	6
1	Conception et planification du projet	- Définition du projet - Cahier des charges - Liste du matériel								
2	Structure de du projet	- Schéma - Coque - Etanchéité								
3	Installation des modules (gouvernail, hélice, crémaillère)	- Gouvernail - Hélice - Crémaillère								
4	Communication (Bluetooth, RF)	- Bluetooth brushless - Bluetooth servomoteur - Bluetooth pas à pas - Bluetooth sonde + total - RF								
5	Tests du projet	- Test avec roues - Test de flottabilité - Test en milieu aquatique								

Figure I.3.1 - Planning initial

Nous avons essayé de prévoir le déroulement de notre projet en découpant en cinq grandes sections composées de plusieurs objectifs.

Finalement, à la fin de notre projet nous avons pu constater que ce planning a été respecté pour trois des cinq sections. La conception et la planification du projet ont été réalisées avant la première séance afin de pouvoir directement commencer la structure du projet.

Grâce à cette avance nous avons pu effectuer le schéma global de l'hydroglisseur, ainsi que réaliser la coque et commencer la réalisation des modules.

Cette progression s'est révélée bénéfique car pour les modules et la communication, nous avons rencontré des difficultés qui nous ont demandé énormément de temps.

Nous n'avons tout de même pas réussi à totalement remplir nos objectifs, le test du déplacement de l'hydroglisseur n'a pu être effectué étant donné que l'étanchéité n'a pu être créée.

En ce qui concerne la répartition du travail, nous avons dans un premier temps établi qu'Océane Mofid s'occuperait majoritairement de la partie mécanique et que Pierre Polette aurait à sa charge la partie électronique. Cependant, il s'est avéré que cette répartition n'a pas du tout été respectée, en effet nous avons réalisé les tâches ensemble. La répartition du travail a donc été effectuée en sous-tâches.

Cette répartition s'est produite naturellement car nous avons chacun besoin des connaissances de l'autre.

Chapitre II : Partie technique

II.1. Hélice

Le module “Hélice” a, en élément principal, un moteur brushless de 1000KV.

Un peu de recherche s’imposait face à ce type de moteur ; n’ayant étudié que les moteurs à courant continu et les servomoteurs.

Un moteur brushless (ou moteur sans balais en français) est réellement un moteur à courant continu mais qui ne possède aucun collecteur tournant, ce qui permet la liaison entre le stator et le rotor, améliorant relativement la commutation.

Un moteur brushless se mesure en KV, la caractéristique physique d’un moteur qui indique la qualité d’un moteur par rapport à ses aimants, au nombre de tours de son bobinage et à sa géométrie (plus on met de tours au bobinage ou des aimants puissants plus le KV sera faible). Le KV est déterminé par le nombre de spires et le diamètre du fil du cuivre utilisé pour le bobinage des dents et correspond à la vitesse de rotation d’un moteur pour 1 volt.

Il indique donc le nombre de tour par minute par volt du moteur à vide et est donc une valeur théorique.

Le moteur avait été livré avec une hélice de type drone qui donnait clairement assez de puissance donc nous avons avancé avec cela. Nous nous sommes quand même renseignés sur la physionomie d’une hélice car nous trouvions cela intéressant et utile à savoir.

Ces informations nous ont aidé à placer l’hélice dans le bon sens pour avoir la meilleure traction possible.

Nous avons fixé le moteur brushless à un axe que l’on peut retrouver dans les squelettes de drone, dans la perspective de le faire rentrer dans la coque du bateau de manière vertical.

Un ESC est également nécessaire pour ce type de moteur ; ESC veut dire *Electronic Speed Control* et contrôle les transistors présents dans le moteur lui permettant de varier sa vitesse.



Figure II.1.1 - Moteur brushless, hélice et ESC

II.2. Gouvernail

Le gouvernail constitué de pales pilotées par un servomoteur a pour but de contrôler la direction de l'hydroglisseur.

La version initiale a été réalisée en plexiglass, cependant le gouvernail était trop petit, le volume d'air dirigé n'était donc pas assez important et ne permettait donc pas l'orientation de l'hydroglisseur.

Le gouvernail a donc été refait en bois. La construction est simple, une boîte découpée au SoFab à la découpeuse laser, nous avons ensuite percé deux trous pour les tiges qui serviront d'axes aux pales. Les pales ont donc été fixées aux axes, puis reliées entre elles par un fil de fer, ainsi si une pivote l'autre aussi, cela évite d'utiliser deux servomoteurs coordonnés. Le servomoteur est lui attaché par un système de crochet à une pale et est fixé sur la boîte.

Le gouvernail était fixé sur une extension en plexiglas de l'hydroglisseur, cependant cette dernière s'étant brisée, nous avons donc installé le gouvernail directement sur l'hydroglisseur, les composants sont donc un peu plus serrés.

Voici le gouvernail initial, ainsi que sa version finale :

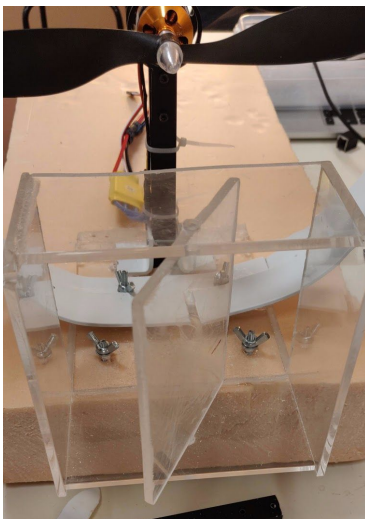


Figure II.2.1 - Première version pales



Figure II.2.2 - Version finale

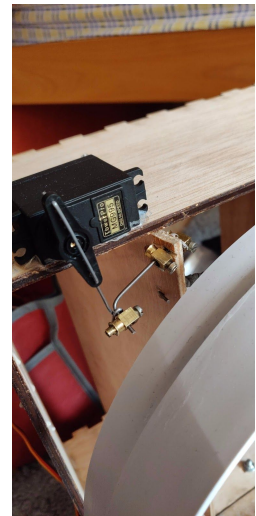


Figure II.2.3 - Liaison servomoteur -

Lors du Play Azur Festival du 9 février 2019, où nous avons exposé notre projet (qui était en cours de réalisation), nous avons rencontré un spécialiste en hydro conception qui nous a donné plusieurs conseils ainsi que des propositions d'améliorations. Une de ses idées concernait les axes fixés aux pales du gouvernail. Ces axes transpercent actuellement la face supérieure de la boîte, et se pose dans une encoche réalisée en tant que gravure sur la face inférieure de la boîte. L'idée consistait à transpercer également la face inférieure en réalisant une encoche sur l'axe, comme ci-dessous, cependant cette idée n'était pas réalisable par nos moyens car nous n'avons pas accès à des appareils d'usinage.



Figure II.2.4 - Schéma axe avec encoches

II.3. Crémaillère

Afin de relever la température en profondeur nous avons besoin d'un mécanisme qui permettait de faire descendre la sonde. Nous avons eu plusieurs idées mais l'une d'entre elles fut gardé grâce au gain de temps et de manipulation que celle ci nous amenait. Nous avons donc construit une crémaillère à laquelle serait accroché la sonde, la crémaillère et son pignon seront encastrés dans leur guide respectif pour plus de stabilité.

Nous avons ensuite défini la longueur ainsi que le rayon du pignon pour pouvoir aller au SoFab et à partir de là les difficultés se sont enchaînées, la longueur voulue ne pouvant pas être découpée à la découpeuse nous avons dû assembler deux parties ce qui amenait une certaine faiblesse à ce niveau.

Le pignon devait posséder un trou de la même forme que l'axe du moteur pas à pas afin que le moteur ne tourne pas dans le vide donc nous avons refait le pignon à plusieurs reprises.

Le modèle finale présente une crémaillère et un pignon en Plexiglass ; ce qui réduit le coefficient de frottements entre les deux, un guide pour la crémaillère en PVC planté dans la coque du bateau pour plus de stabilité ainsi qu'une extension également en PVC pour le moteur afin de le fixer au guide principal.

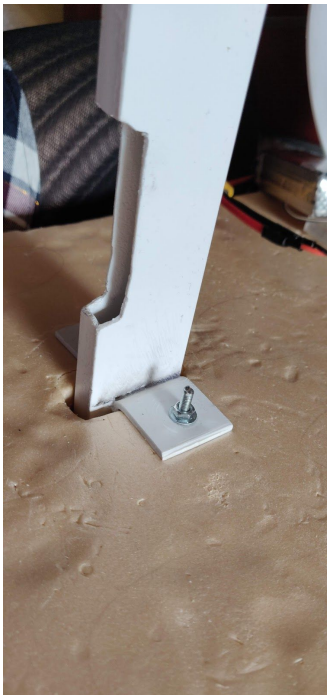


Figure II.3.1 - Guide en PVC de la crémaillère

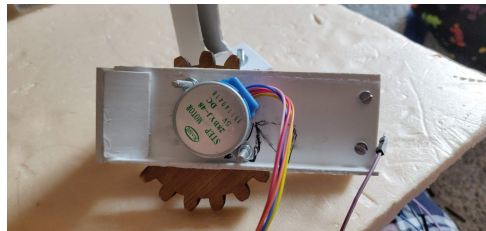


Figure II.3.2 - Extension en PVC du moteur



Figure II.3.3 - Crémaillère et pignon en Plexiglass

Chapitre III : Partie électronique

III.1. Bluetooth

Le Bluetooth est un mode de communication accessible depuis n'importe quel smartphone, en utilisant l'application gratuite disponible sur Play Store "*Bluetooth Electronics*".

Voici le montage du module Bluetooth :

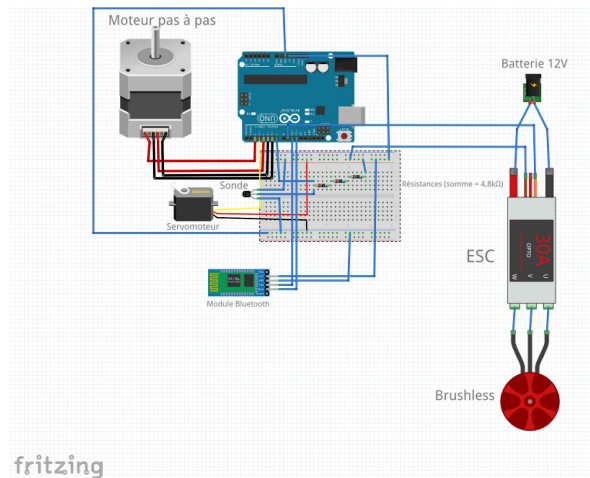


Figure III.1.1 - Montage avec module bluetooth

La télécommande de ce module étant totalement personnalisable, nous avons fait le choix d'utiliser :

- 2 boutons pour le moteur pas à pas, un pour la rotation dans le sens horaire pour faire remonter la crémaillère et l'autre pour une rotation dans le sens antihoraire afin de la faire descendre.
- 2 sliders permettant de contrôler la vitesse de rotation du moteur brushless, ainsi que l'angle du servomoteur, ces sliders servent donc de télécommande pour diriger l'hydroglisseur.
- 1 bouton avec 1 "indicator", le bouton permet de commencer à relever la température par la sonde et l'indicator affiche donc cette valeur sur le smartphone.

Voici notre télécommande Bluetooth :



Figure III.1.2 - Installation des commandes bluetooth

Nous allons maintenant nous intéresser à l'algorithme du programme avec la connexion Bluetooth.

En appuyant sur un bouton ou un slider, on envoie un caractère spécifique. L'Arduino va ensuite lire le premier caractère afin d'identifier le module que l'on souhaite utiliser. Si cela correspond à un bouton du moteur pas à pas, l'Arduino va simplement demander au moteur d'effectuer un tour et quatorze dix-septième d'un autre dans le sens voulu, ce nombre correspond à la rotation nécessaire pour que la crémaillère puisse descendre suffisamment en laissant une marge pour que la crémaillère ne se détache

pas du pignon. Si c'est le bouton correspondant à la sonde l'Arduino va envoyer un message au capteur pour demander la température, puis une fois cette valeur reçue, l'Arduino envoie cette valeur associée à un caractère spécifique pour que l'indicateur sache qu'il est concerné. Enfin, si c'est l'un des sliders l'Arduino va lire la valeur qui suit le caractère et l'envoyer au moteur concerné.

Nous avons ensuite voulu passer à une communication RF pour plusieurs raisons. En effet la connexion Bluetooth ne permet pas de commander l'hydroglisseur à plus de 10 mètres, ce qui constitue une contrainte trop importante. De plus, les autres modules Bluetooth utilisés en même temps et dans la même salle provoquent des interférences, les modules ne reçoivent pas toujours les caractères ou avec un certain délai, ou des valeurs différentes. Le dernier désavantage est d'ordre pratique, car pour utiliser notre hydroglisseur avec la communication Bluetooth, il faut d'abord paramétrer les sliders et les boutons à l'identique.

(Les programmes sont disponibles en annexe I, page 21)

III.2. Radiofréquence

Une télécommande RF nous a été fournie (une RadioLink AT9S et sa carte radio R9DS) et la notice a pu être trouvée en ligne ce qui nous a aidé à mieux comprendre le montage des voies de la carte radio.

Nous voulions contrôler le servomoteur et le moteur brushless donc 2 voies nous étaient nécessaires, en regardant les paramètres de la télécommande nous avons pu voir que :

- le joystick gauche gardait sa position actuelle lorsque nous effectuons une translation haut/bas, ce qui serait utile pour gérer la vitesse du bateau (car nous avons voulu garder une vitesse de croisière)
- le joystick droit revenait à sa position initiale lorsque nous effectuons une translation gauche/droit, ce qui serait utile pour gérer la direction du bateau (car nous voulions que le déplacement initial serait "tout droit")

Ces deux déplacements de joystick correspondent respectivement aux voies 3 et 1, et grâce à la carte Arduino nous récupérons l'impulsion en microsecondes envoyée par la télécommande et nous transformons cela en valeur acceptable par l'ESC ainsi que par le servomoteur. Cela nous a permis d'initialiser la télécommande (placer les bonnes valeurs initiales) pour ne pas faire surchauffer le moteur brushless ou de bloquer le servomoteur.

Il nous a été impossible de programmer le moteur pas à pas ainsi que la sonde par radiofréquence par manque de temps mais nous avons envisagé de les contrôler grâce à l'un des nombreux boutons que proposait la télécommande.



Figure III.2.1 - Télécommande AT9S et son module R9DS

Conclusion

L'Arduinoglisseur est un projet satisfaisant car il correspond à nos attentes même si nous avons manqué de temps pour l'étanchéité.

Nous avons dans un premier temps travailler sur un projet communiquant grâce au Bluetooth car nous avions des connaissance dû aux travaux pratiques d'électronique. Une fois que nous connaissions sur le bout des doigts comment fonctionnent tous les modules de notre projet nous avons pu faire des recherches et découvrir la communication RF.

À la fin des séances dédiées au projet Arduino, nous avons réfléchi aux différentes perspectives à l'Arduinoglisseur.

Nous pourrions dans un premier temps programmer le moteur pas à pas contrôlant la crémaillère, ainsi que la sonde sur une communication RF. Puis réaliser l'étanchéité de tous les composants.

Une amélioration possible serait de vouloir mesurer la température en fonction de la profondeur, pour cela il faudrait soit augmenter la taille de la crémaillère et donc, de tout l'hydroglisseur pour l'équilibre ,soit utiliser un autre système comme un cable avec une masse au bout et qui remonte grâce à un enrouleur.

Ces améliorations sont complètement envisageables si le projet de l'Arduinoglisseur est repris par nos soins ou d'autres élèves.

Bibliographie

Nous avons utilisé beaucoup de ressources sur des supports différents afin de réaliser notre projet, voici les sources des principaux :

- ❑ https://woodgears.ca/gear_cutting/template.html (générateur d'engrenages pour la crémaillère)
- ❑ <https://www.youtube.com/watch?v=sHO7jplkNTc&t=1s> (hovercraft)
- ❑ <https://www.youtube.com/watch?v=srwmY5v0KYo> (hovercraft)
- ❑ <https://pinshape.com/items/38955-3d-printed-rc-airboat-35kmh-v10> (modèle 3D d'un gouvernail)
- ❑ <https://www.instructables.com/id/How-to-run-an-ESC-with-Arduino/> (notice moteur brushless)
- ❑ <https://www.youtube.com/watch?v=gV4WDjs6udc> (explication des esc)
- ❑ <https://inkscape.org> (logiciel pour le découpeuse laser)
- ❑ <http://hmf.enseeiht.fr/travaux/CD0910/tpld/mfi/anciens/Web/pages/Hydrodyn/Hydrodynamique.htm#t2> (cours de physique sur les frottements)
- ❑ <http://users.polytech.unice.fr/~pmasson/index.html> (cours d'électronique)
- ❑ Rapport de projet 2017-2018; Bee Connect, La ruche connectée; Dihia Bitam, Qingna Xue, Axel Le Bourhis

Annexes

Annexe I : Programme du projet	21
I.1 - Programme Bluetooth.....	21
I.2 - Programme RF	23
Annexe 2 : Liens utiles	25
II.1 - Github.....	25
II.2 - Chaîne Youtube.....	25

Annexe 1 : Programmes du projet

Le programme a été réalisé directement sur le logiciel Arduino, il se décompose donc en trois parties, la première où sont les librairies, ainsi que les variables sont définies ; la seconde le setup, où chaque module est initialisé, le programme contenu à l'intérieur n'est effectué qu'une seule fois ; et enfin le loop dans lequel on implémente le programme principal qui est répété en boucle.

AI.1. Programme Bluetooth

```
3 // Importation des différentes librairies
4 #include <Servo.h> // Permet de contrôler le servomoteur en utilisant des angles.
5 #include <SoftwareSerial.h> // Permet l'utilisation du module bluetooth.
6 #include <Stepper.h> // Permet de contrôler le moteur pas à pas.
7 #include <Wire.h>
8 #include <OneWire.h> // Permettent la lecture des capteurs de température.
9 #include <DallasTemperature.h> // Permet de traduire l'information donnée par les capteurs dans l'unité de notre choix.
10
11 //-----
```

Figure AI.1.1 - Définition des librairies

```
// Définition des différents modules
#define USE_DS18B20 // Nom de la sonde.
#define RX 10
#define TX 11
SoftwareSerial BlueT(RX,TX); // Transmetteur et récepteur Bluetooth.

char Data;

Servo monServo;
int positionduServo = 0; // Définition du servomoteur et de sa position initiale.

Servo esc;
int val = 0; // Définition de l'esc du moteur brushless et de sa valeur initiale.

Stepper monMoteur(nombrePas, 2, 4, 6, 5); // Définition du moteur pas à pas, de ses sorties : IN1 , IN3 , IN2 , IN4.
const int nombrePas = 32*64; // Nombre de pas pour effectuer un tour.

#ifdef USE_DS18B20
#define ONE_WIRE_BUS 7
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature temp_sensor(&oneWire);
#endif // Définition de la sortie de la sonde ainsi que des fonctions.

double temp;

//-----
```

Figure AI.1.2 - Définition des variables

```

void setup() {
    // Initialisation des modules, chaque ligne est effectuée une seule fois.
    // put your setup code here, to run once:
    esc.attach(9); // On attache l'ESC au port numérique 9 (port PWM obligatoire).
    delay(15);

    Serial.begin(115200); // Définition de la vitesse de communication entre la sonde et l'arduino.
    delay(15);

    Serial.println("Salut, je suis prêt pour l'AT !"); // Affiche sur le moniteur série que le module Bluetooth est prêt.
    BlueT.begin(9600); // Définition de la vitesse de communication entre le module Bluetooth et l'Arduino.
    delay(500);

    Serial.println("Interfacé Arduino with ESP32"); // affiche sur le moniteur série que la sonde est prête pour l'initialisation.
    delay(15);

    monMoteur.setSpeed(10); // Donne la vitesse de rotation du moteur pas à pas.
    delay(500);

    monServo.attach(3); // Initialisation de la sortie du servomoteur.
    monServo.write(positionduServo); // Initialisation de la position du servomoteur.
    delay(500);

    #ifdef USE_DS18B20
        temp_sensor.begin(); // Démarre le processus de lecture des données des capteurs.
    #endif
    delay(500);
}

```

Figure AI.1.3 - Setup, Initialisation des modules

```

void loop() {
    // Exécute le programme ci-dessous en boucle.
    #ifdef USE_DS18B20
        temp_sensor.requestTemperatures(); // Demande la température aux capteurs.
        temp = temp_sensor.getTempCByIndex(0); // Récupère la température en degré Celsius du capteur n°0,
    #endif
    String SerialData;
    SerialData = String(temp,1); // Convertit la valeur de la température en chaîne de caractères.

    Data=BlueT.read(); // Récupère le premier caractère envoyé par le bluetooth (depuis notre smartphone avec l'application).
}

```

Figure AI.1.4 - Début du Loop (lecture données)

```

// ----SERVOMOTEUR
if (Data=='A'){
    // change la position du servomoteur à l'aide d'un slider le slider est programmé de 25 à 135, ainsi la position "0" correspond à un angle de 81
    positionduServo=BlueT.parseInt(); // Récupère la valeur suivant le caractère "A".
    Serial.println(positionduServo); // Affiche sur le moniteur série cette valeur.
    monServo.write(positionduServo); // Le servomoteur prend la position de cette valeur.
    delay(15);
    Data=BlueT.read();
}

// ----MOTEUR PAS A PAS
if (Data == 'D') { // Si le premier caractère est "D" qui signifie DOWN.

    monMoteur.step(-3734); // Le moteur pas à pas effectue une rotation antihoraire d'un tour et de 14/17 d'un autre.
    delay(500);
    Data=BlueT.read();
}

if (Data == 'U') { // Si le premier caractère est "U" qui signifie UP.
    monMoteur.step(3734); // Le moteur pas à pas effectue une rotation dans le sens horaire d'un tour et de 14/17 d'un autre.
    delay(500);
    Data=BlueT.read();
}

// ----BRUSHLESS
if (Data=='B'){
    esc.attach(9);
    val = BlueT.parseInt(); // Récupère la valeur suivant le caractère "B".
    Serial.println(val); // Affiche cette valeur sur le moniteur série.
    esc.write(val); // Le moteur brushless prend la vitesse correspondant à cette valeur.
    delay(15);
    Data=BlueT.read();
}

// ----SONDE
if (Data == 'T') {
    Serial.println("<Temperature : " + SerialData + ">"); // Affiche sur le moniteur série la température en degré Celsius.
    BlueT.print("*D"+String(SerialData)+"*"); // Affiche sur le smartphone la valeur de la température, "*D" et "*" sont des clés pour afficher sur le bon indicateur.
    Data=BlueT.read();
}
} // Fin loop, recommence la fonction loop.

```

Figure AI.1.5 - Fin du Loop (utilisation des modules)

AI.2. Programme RF

```
1 //Fait fonctionner le brushless et le servomoteur par RF//
2
3 #include <Servo.h>
4 Servo esc; // Création de l'objet permettant le contrôle de l'ESC
5 Servo monServo;
6 float ESC = 0;
7 float positionduServo = 0;
8
9
10 int V_chanel_1 = 0, V_chanel_3 = 0;
11 int chanel_1 = 14, chanel_3 = 15; // connecter les pin des voies 1 et 3 de la carte radio aux entrées analogiques 14 et 15
12 int lecture= 0;
13
```

Figure AI.2.1 - Définition des variables et librairies

```
43 void loop() {
44 // si on dépasse le temps d'attente 3ms, PulseIn renvoie la valeur 0 qu'il ne faut pas prendre en compte
45
46 lecture = pulseIn(chanel_1, HIGH, 3000); // lecture de l'impulsion envoyée par la voie 1
47 if (lecture >0) {
48     V_chanel_1=lecture;
49 }
50 delay(10);
51 Serial.print(" Chanel 1 = ");
52 Serial.print(V_chanel_1);
53
54 positionduServo = (-0.106*V_chanel_1)+215; //équation permettant de transformer l'impulsion envoyée en degrés (selon l'intervall
55 Serial.print(" positionduServo = ");
56 Serial.println(positionduServo);
57 monServo.write(positionduServo);
58
59 //*****
60 lecture = pulseIn(chanel_3, HIGH, 3000); // lecture de l'impulsion envoyée par la voie 3
61 if (lecture >0) {
62     V_chanel_3=lecture;
63 }
64 delay(10);
65 Serial.print(" Chanel 3 = ");
66 Serial.print(V_chanel_3);
67
68 ESC = (-0.081*V_chanel_3)+180;
69 Serial.print(" ESC = ");
70 Serial.println(ESC);
71
72
73
74 if ((ESC >40)&(ESC<110)){ // mesure de sécurité : on envoie à l'esc que des valeurs comprises entre 40 et 110 pur éviter la surc
75     esc.write(ESC); }
76
```

Figure AI.2.2 - Setup, Initialisation des modules

```

14 void setup() {
15     // put your setup code here, to run once:
16     Serial.begin(115200);
17     pinMode(chanel_1, INPUT);
18     pinMode(chanel_3, INPUT);
19     esc.attach(9); // On attache l'ESC au port numérique 9 (port PWM obligatoire)
20     monServo.attach(3);
21     monServo.write(positionduServo);
22     delay(15);
23
24
25     // Initialisation de l'ESC
26     // (certains ESC ont besoin d'une "procédure d'initialisation"
27     // pour devenir opérationnels - voir notice)
28     Serial.println(" Initialisation ESC ");
29     esc.write(0);
30     delay(1000);
31     esc.write(180);
32     delay(1000);
33     esc.write(0);
34     delay(2000);
35     Serial.println(" ESC = 40");
36     esc.write(40); // nécessaire au démarrage du brushless
37     delay(4000);
38     Serial.println(" ESC = 45");
39     esc.write(45); // on impose une vitesse initiale pour les tests mais non nécessaire
40     delay(2000);
41 }

```

Figure A1.2.3 - *Loop, coeur du programme*

Annexe 2 : Liens utiles

A2.1. Site Github

Lors de notre projet nous avons dû créer notre site Github pour communiquer nos rapports de séances, ainsi que nos programmes au corps enseignant.

Ce site est donc rempli de ressources tels que le cahier des charges, le planning, les rapports, ainsi que les différents programme.

<https://github.com/MofidPolette/Hydroglisseur-sonde-ARDUINO->

A2.1. Chaîne Youtube

Nous avons également créé une chaîne Youtube afin d'alimenter nos rapports de séances par des démonstrations. Cette chaîne nous permet également de palier aux aléas des présentations en directes en diffusant les vidéos des tests du projet.

<https://www.youtube.com/channel/UCjTA350zIQjDU40OQkE3isw/videos>

