

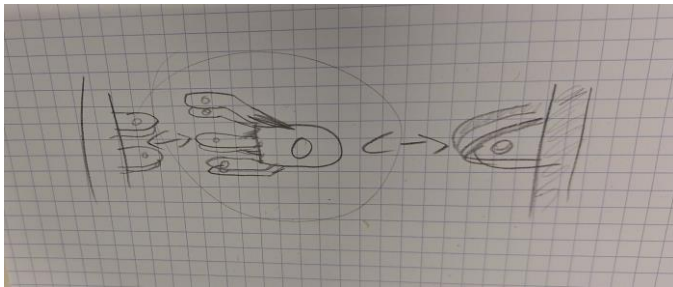
Projet Hydroptère

Séance 2

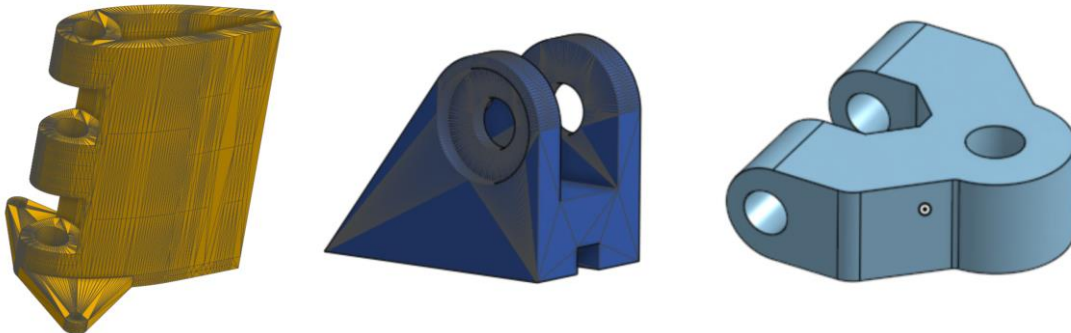
Pour cette séance 2, Je me suis consacré majoritairement à la conception 3D, à l'impression des pièces et j'ai commencé à programmer l'envoi et la réception des commandes via wifi.

- Conception 3D :

J'ai commencé par créer les pièces manquantes, dont j'ai parlé dans le premier rapport. Ces pièces permettent de faire tourner le foil arrière de l'hydroptère en hauteur et sur les côtés. Voici le dessin des pièces avant la modélisation (sans les côtes) :



J'ai donc revu les pièces de gauche et de droite, dont la base a été prise sur un modèle 3D déjà existant (voir premier compte rendu) et créé la pièce du milieu. Voici les modèles 3D (faits sur Onshape) :



Une fois que cela est fait, j'ai fait aussi les dernières modifications sur le foil arrière, puis j'ai pu commencer à imprimer les premières pièces.

- Impression

J'ai eu un certain nombre de problèmes avec les imprimantes, surtout à cause des réglages des pièces au moment de les slicer (mauvaise taille de buse ce qui provoquait des erreurs sur l'imprimante, les supports qui n'apparaissaient pas...), ce qui m'a fait perdre beaucoup de temps sur la séance. Les 3 pièces au-dessus se sont finalement imprimées, l'impression de la première était parfaite, et j'ai pu retirer les supports facilement. Pour les autres cela n'a pas été aussi facile : les trous pour faire circuler les vis étant petits, les supports dedans sont très difficile à retirer et beaucoup de filaments pendent encore des pièces, rendant leur esthétique douteuse... Je les limerais à la prochaine séance pour retirer tous les fils en plus. Voici à quoi elles ressemblent



Pendant l'impression des pièces, j'ai continué à travailler sur la transmission par wifi, pour connecter une télécommande à l'hydroptère (voir premier rapport).

- Transmission wifi

Je travaille sur deux modèles NRF M 24L01 pour la transmission des données : Une en envoi de données et une en réception. Le but est d'avoir une portée assez grande pour pouvoir communiquer sans problème sur l'eau (car l'eau réduit la portée et la qualité de transmission.) J'ai donc commencé les deux codes, le but étant de relier l'antenne qui envoie les données à une manette pour piloter entièrement l'hydroptère à distance, donc contrôler le moteur mais aussi les inclinaisons du foil arrière. Le vrai problème est que pour essayer la transmission, je dois avoir deux cartes arduino pour avoir une antenne sur chaque carte arduino avec un code différent, je ne peux donc pas vraiment avancer avant la prochaine séance. Les bibliothèques utilisées par le modèle que j'ai sont assez simples d'utilisation et comportent des exemples qui sont quasiment complets pour l'utilisation qu'on en fait, ce qui simplifie grandement la tâche. Voici à quoi ressemble une partie du code comme exemple :

```

80 | if (role) {
81 |   radio.stopListening(); // put radio in TX mode
82 | } else {
83 |   radio.startListening(); // put radio in RX mode
84 | }
85 |
86 | // For debugging info
87 | // printf_begin(); // needed only once for printing details
88 | // radio.printDetails(); // (smaller) function that prints raw register values
89 | // radio.printPrettyDetails(); // (larger) function that prints human readable data
90 |
91 | } // setup
92 |
93 | void loop() {
94 |
95 |   // This device is a TX node
96 |
97 |   unsigned long start_timer = micros(); // start the timer
98 |   bool report = radio.write(&payload, sizeof(float)); // transmit & save the report
99 |   unsigned long end_timer = micros(); // end the timer
100 |
101 |   if (report) {
102 |     Serial.print(F("Transmission successful! ")); // payload was delivered
103 |     Serial.print(F("Time to transmit = "));
104 |     Serial.print(end_timer - start_timer); // print the timer result
105 |     Serial.print(F(" us. Sent: "));
106 |     Serial.println(payload); // print payload sent
107 |     payload += 0.01; // increment float payload
108 |   } else {
109 |     Serial.println(F("Transmission failed or timed out")); // payload was not delivered
110 |   }
111 |
112 |   // to make this example readable in the serial monitor
113 |   delay(1000); // slow transmissions down by 1 second
114 | } // loop
115 |
116 |

```