

Pour cette séance nous avons pris un rendez-vous au FabLab avec Louis. Nous aurons donc un rapport de séance similaire car ça nous a pris la majorité du temps de la séance.

FabLab :

Nous avons rendez-vous au Fab Lab pour nous occuper du module stockant et distribuant les capsules de café. Nous avons donc rencontré la personne s'occupant de l'atelier. Nous lui avons expliqué ce que nous recherchions à faire et quelles solutions nous avons trouvées. Il nous a conseillé de modifier quelques aspects du module stockant les capsules pour plus de praticité. Après s'être mis d'accord, il nous a fourni un site où on pouvait concevoir notre boîte qui stockera les capsules au-dessus de la machine à café. On devra donc y retourner avec le fichier 3D pour le créer avec la découpeuse laser.

Aperçu avec « MakerCase » de la boîte

Grâce à cette boîte, nous stockerons les capsules verticalement et nous ajouterons un système en bas de la boîte pour libérer une seule capsule à la fois. Comme par exemple à l'aide d'un solénoïde ou d'un servo moteur.

De plus, nous avons réfléchi à un moyen de positionner tous nos modules pour faire fonctionner la cafetière mais sans fixer quoique ce soit sur la cafetière pour qu'on puisse la réutiliser à la fin de notre projet. Nous avons donc pensé à utiliser trois planches de bois, une fera office de socle où nous poserons la cafetière. Une autre fixée verticalement sur le socle qui se trouvera à droite de la cafetière où se fait l'ouverture de la cafetière grâce au servo-moteur. Et une dernière qui fera un toit au-dessus de la cafetière pour fixer les solénoïdes, le réservoir à capsules et poser la carte Arduino et tous les branchements.

Module solénoïdes avec Bluetooth :

Pour le reste de la séance, après être retournés en cours et passé l'oral, je me suis occupé d'ajouter la connexion Bluetooth à mon montage et code de la séance dernière. J'ai donc relu les cours sur le Bluetooth et revu les programmes déjà réalisés en début d'année puis je l'ai adapté à mes besoins.

J'ai tout d'abord configuré mon module Bluetooth HC-06 grâce à un programme et des branchements simples pour effectuer les commandes AT. Le HC-06 est composé de 3 entrées et 1 sortie, Vcc et GND sont les broches d'alimentation du module donc des entrées. RX est une broche pour la réception des données donc une entrée également. Et enfin, TX est la dernière broche pour la transmission des données donc une sortie.



HC-06

Module Bluetooth

```
//Réglage bluetooth

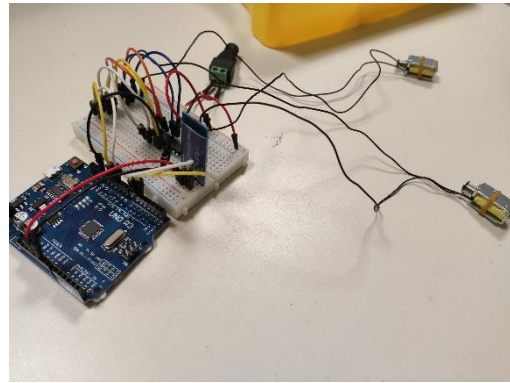
#include<SoftwareSerial.h>
#define RX 6
#define TX 5
SoftwareSerial BlueT(RX,TX);

void setup(){
  Serial.begin(9600);
  delay(500);
  Serial.println("Bonjour -Prêt pour les commandes AT");
  BlueT.begin(9600);
  delay(500);
}

void loop(){
  while (BlueT.available()) {
    Serial.print(char(BlueT.read()));
  }
  while (Serial.available()) {
    BlueT.write(char(Serial.read()));
  }
}
```

Programme commandes AT

J'ai donc continué par effectuer le montage en rajoutant le module Bluetooth et en connectant les bonnes entrées aux entrées de la carte Arduino. J'ai vérifié mon montage en réalisant les commandes AT. J'ai donc utilisé la sorties V et GND de la carte Arduino UNO et les deux sorties 5 et 6 pour l'entrée et la sortie RX et TX.



Montage solénoïde avec Bluetooth

Ensuite, j'ai modifié mon programme faisant fonctionner les solénoïdes pour pouvoir utiliser le Bluetooth. Mon but est d'activer un solénoïde ou l'autre en fonction du choix de l'utilisateur pour la taille du café. J'ai donc ajouté des boucles IF avec pour condition la réception d'une lettre (P ou G) pour activer l'un ou l'autre. Cependant j'ai rencontré des problèmes, en premier je ne parvenais pas à connecter mon module Bluetooth avec l'application « Bluetooth Electronics » alors que les commandes AT fonctionnaient. J'ai donc changé de module est le problème s'est réglé. Mon deuxième problème est toujours présent, après avoir connecté mon module et réalisé mon panneau sur l'application avec deux boutons simple « petit café » et « grand café » envoyant la lettre P et G respectivement, aucun solénoïde s'activait.

Solenoid | Arduino 1.8.9 (Windows Store 1.8.21.0)

Fichier Édition Croquis Outils Aide

Solenoid

```
//Solénoïdes

#include<SoftwareSerial.h>
#define RX 5
#define TX 6
SoftwareSerial BT(RX,TX);
char DATA;

int solenoidePetit = 10;
int solenoideGrand = 11;

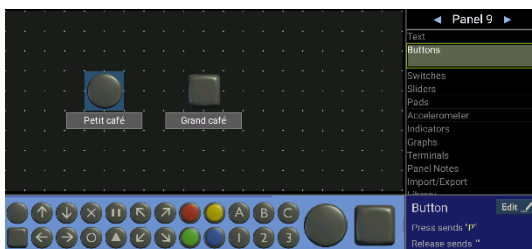
void setup() {

  pinMode(solenoidePetit, OUTPUT);
  pinMode(solenoideGrand, OUTPUT);
  Serial.begin(9600);
  BT.begin(9600);
}

void loop() {
  if (BT.available()) {
    DATA=BT.read();
    if (DATA=='P'){
      digitalWrite(solenoidePetit, HIGH); // Active le solenoide
      delay(500);
      digitalWrite(solenoidePetit, LOW); // Relache
      delay(1000);
    }

    if (DATA=='G'){
      digitalWrite(solenoideGrand, HIGH); // Active le solenoide
      delay(500);
      digitalWrite(solenoideGrand, LOW); // Relache
      delay(1000);
    }
  }
}
```

Programme avec Bluetooth



Panneau sur « Bluetooth Electronics »

Enfin pour tenter de comprendre d'où venait l'erreur et la corriger j'ai ajouté dans le programme des lignes de vérification envoyant « OK_testX » sur le moniteur série. Une première juste avec le premier IF ensuite une affichant la variable DATA et deux autres justes après les deux autres boucles IF. Je me suis donc rendu compte que le programme ne rentrait même pas dans le premier IF ce qui expliquait que rien ne se passait peut importe sur lequel des boutons j'appuyais. Cependant je ne sais pas pourquoi la première condition n'est pas vérifiée alors que tout est connecté correctement et fonctionne avec le programme pour les commandes AT.

```
void loop() {
  if (BT.available()) {
    Serial.println("OK_test1");
    DATA=BT.read();
    Serial.println(DATA);
    if (DATA=='P'){
      Serial.println("OK_test2");
      digitalWrite(solenoidePetit, HIGH);
      delay(500);
      digitalWrite(solenoidePetit, LOW);
      delay(1000);
    }

    if (DATA=='G'){
      Serial.println("OK_test3");
      digitalWrite(solenoideGrand, HIGH);
      delay(500);
      digitalWrite(solenoideGrand, LOW);
      delay(1000);
    }
  }
}
```

Programme avec vérifications