

Rapport de projet

Séance : 31 janvier 2022

Objectif de séance :

Au cours de cette séance j'avais comme but d'établir la connexion Bluetooth entre mes différents composants et mon appareil Android. En approfondissant mes recherches sur MIT APP INVENTOR durant les vacances, je me suis rendu compte que l'application Bluetooth Electronics était plus simple pour établir la transmission. Je voulais également réaliser l'ouverture et la fermeture du portail.

Déroulement de la séance :

1. Révisions

Afin de réaliser cette connexion, j'ai commencé par reprendre les cours sur le Bluetooth et les codes effectués durant le semestre 3. Les codes étudiés permettaient :

- De configurer un module HC-06 ;
- D'allumer une LED progressivement grâce à un appareil Bluetooth.

Voici les codes utilisés et le montage réalisé.



```
bluetooth_nom_et_pin | Arduino 1.8.19 (Windows Store 1.8.57.0)
Fichier Édition Croquis Outils Aide

bluetooth_nom_et_pin
//modifie le nom et le code pin de l'appareil e bluetotth
#include<SoftwareSerial.h>
#define RX 10
#define TX 11
SoftwareSerial BlueT (RX,TX);

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  delay(500);
  Serial.println("Bonjour -Pret pour les commandes AT");
  BlueT.begin(9600);
  delay(500);
}

void loop() {
  // put your main code here, to run repeatedly:
  while(BlueT.available()){
    Serial.println(char(BlueT.read()));
  }

  while (Serial.available()){
    BlueT.write(char(Serial.read()));
  }
}

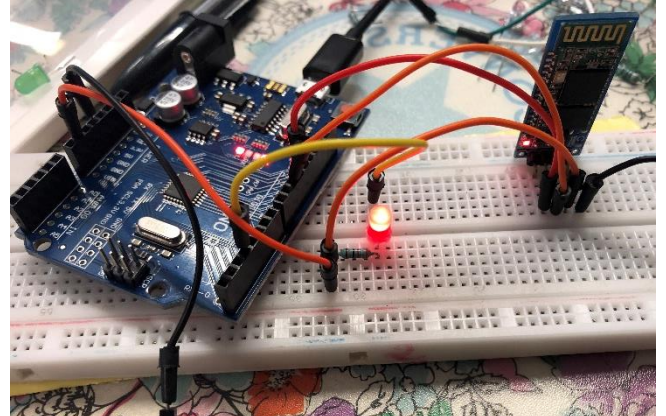
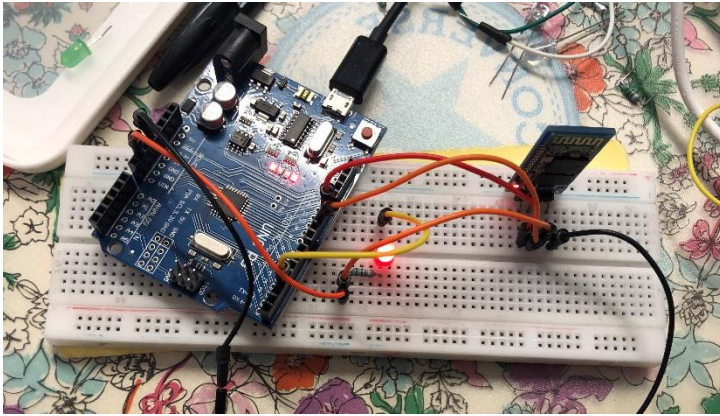
bluetooth_led | Arduino 1.8.19 (Windows Store 1.8.57.0)
Fichier Édition Croquis Outils Aide

bluetooth_led
#include<SoftwareSerial.h>
#define RX 10
#define TX 11
#define LED 3
SoftwareSerial BlueT (RX,TX);
int PWM=128;
char Data;

void setup() {
  Serial.begin(9600);
  BlueT.begin(9600);
  pinMode(LED,OUTPUT);
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
  if (BlueT.available()){
    Data=BlueT.read();
    if (Data=='A'){
      PWM=BlueT.parseInt();
      Serial.println(PWM);
      analogWrite(LED,PWM);
    }
  }
}
```

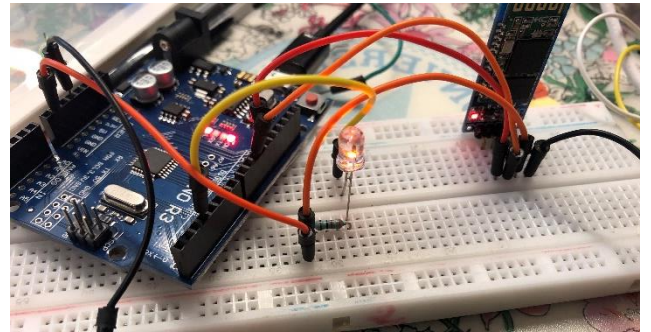
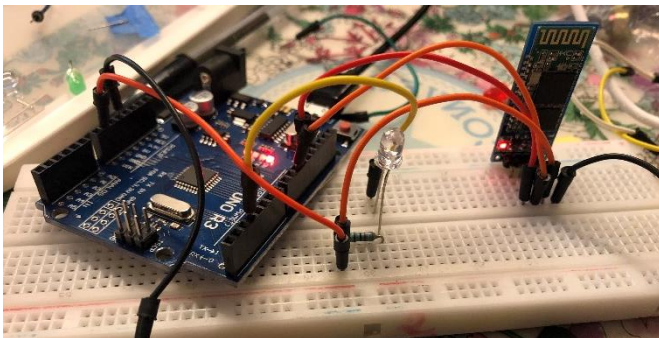
Voici le lien de la vidéo du fonctionnement <https://youtu.be/JcD-LIXcs2c>



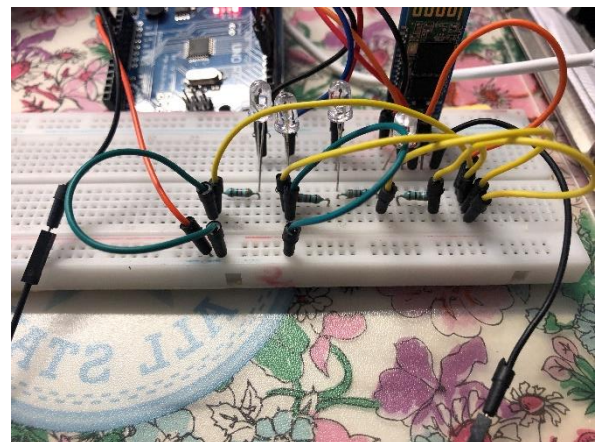
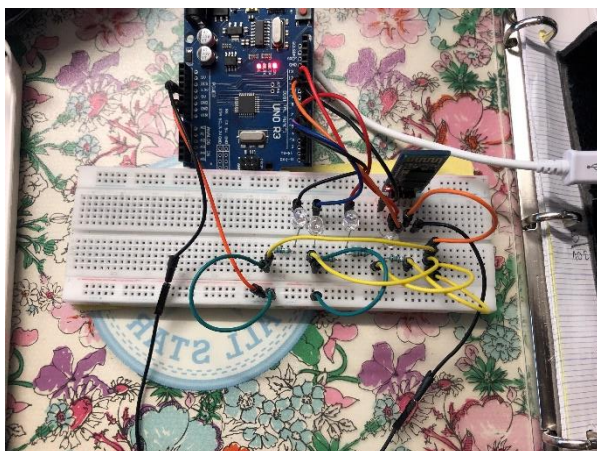
Je souhaitais reprendre tout le cours sur le Bluetooth afin d'avoir toutes les notions en tête et être efficace durant cette séance.

2. Approfondissement

Une fois que j'avais bien assimiler les notions de base sur le Bluetooth, j'ai proposé à mon binôme de réaliser sa connexion Bluetooth qui était d'allumer une LED à distance. J'ai commencé par la vérification du bon fonctionnement de chacune des LED, en les remplaçant une par une dans le montage précédant avec la LED déjà présente. Je me suis aperçue que l'une parmi les quatre que nous disposions était grillée.



Pour finir j'ai pris toutes les LED, j'ai complété le montage et le code précédant.



Voici le code ainsi qu'une vidéo montrant le fonctionnement : https://youtu.be/FGVSTeQ3_zw

pwm_4_leds

```
#include<SoftwareSerial.h>
#define RX 10
#define TX 11
#define LED1 3
#define LED2 5
#define LED3 6
#define LED4 9
SoftwareSerial BlueT(RX,TX);
int PWM=128;
char Data;

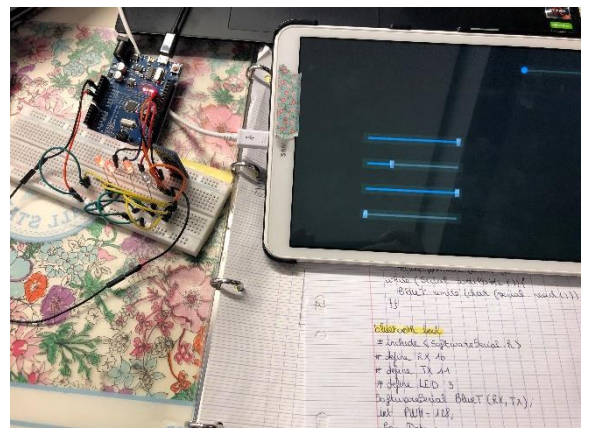
void setup() {
  Serial.begin(9600);
  BlueT.begin(9600);
  pinMode(LED1,OUTPUT);
  pinMode(LED2,OUTPUT);
  pinMode(LED3,OUTPUT);
  pinMode(LED4,OUTPUT);
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
  if (BlueT.available()){
    Data=BlueT.read();
    if(Data=='A'){
      PWM=BlueT.parseInt();
      Serial.println(PWM);
      analogWrite(LED1,PWM);}
    if(Data=='B'){
      PWM=BlueT.parseInt();
      Serial.println(PWM);
      analogWrite(LED2,PWM);}
    if(Data=='C'){
      PWM=BlueT.parseInt();
      Serial.println(PWM);
      analogWrite(LED3,PWM);}
    if(Data=='D'){
      PWM=BlueT.parseInt();
      Serial.println(PWM);
      analogWrite(LED4,PWM);}
```

Enfin j'ai séparé chaque LED mais en gardant le même montage. <https://youtu.be/AhkQyBUrNLY>

3. LED

J'ai terminé par la connexion des quatre LED par Bluetooth. Pour cela j'ai placé quatre boutons qui représentent chacun une LED. Le code est disponible dans le dossier Bluetooth de la séance 5 sur GitHub.



4_LEDS

```
#include<SoftwareSerial.h>
#define RX 10
#define TX 11
#define LED1 3
#define LED2 5
#define LED3 6
#define LED4 9
SoftwareSerial BlueT(RX,TX);
char Data;

void setup() {
  Serial.begin(9600);
  BlueT.begin(9600);
  pinMode(LED1,OUTPUT);
  pinMode(LED2,OUTPUT);
  pinMode(LED3,OUTPUT);
  pinMode(LED4,OUTPUT);
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:
  if (BlueT.available()){
    Data=BlueT.read();

    //allumage
    if(Data=='1'){
      digitalWrite(LED1,HIGH);
      Serial.println(HIGH);
    }
  }
}
```

4_LEDS

```
if(Data=='2'){
  digitalWrite(LED2,HIGH);
  Serial.println(HIGH);
}

if(Data=='3'){
  digitalWrite(LED3,HIGH);
  Serial.println(HIGH);
}

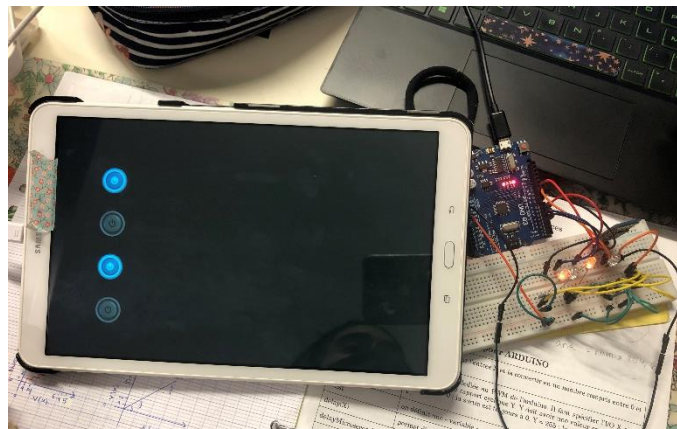
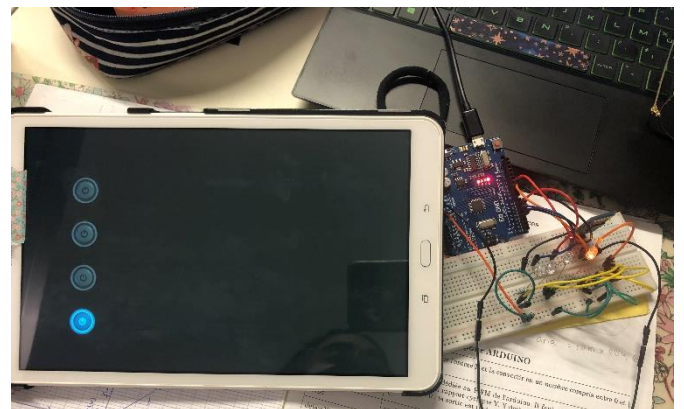
if(Data=='4'){
  digitalWrite(LED4,HIGH);
  Serial.println(HIGH);
}

if(Data=='5'){
  digitalWrite(LED1,LOW);
  Serial.println(LOW);
}

if(Data=='6'){
  digitalWrite(LED2,LOW);
  Serial.println(LOW);
}

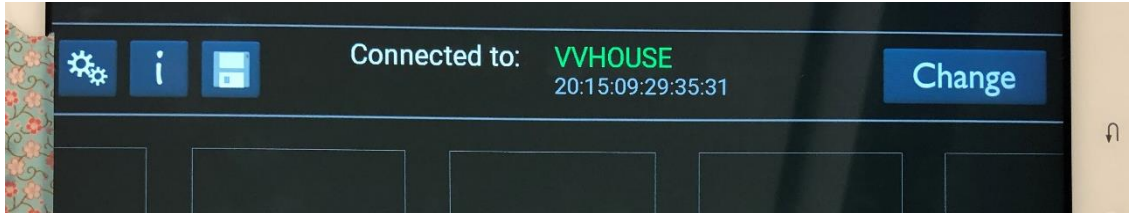
if(Data=='7'){
  digitalWrite(LED3,LOW);
  Serial.println(LOW);
}

if(Data=='8'){
  digitalWrite(LED4,LOW);
  Serial.println(LOW);
}
}
```

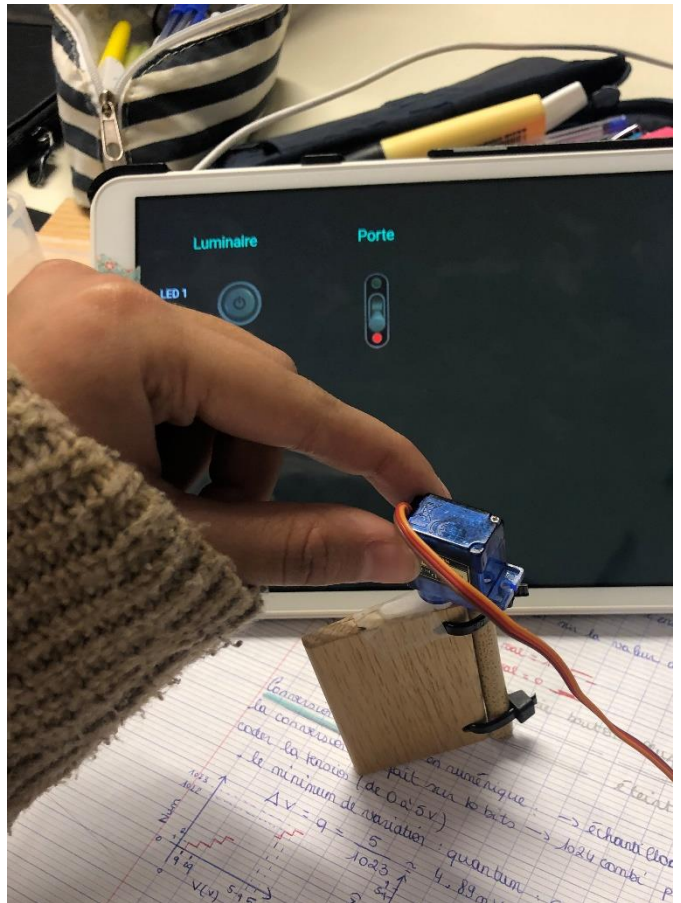
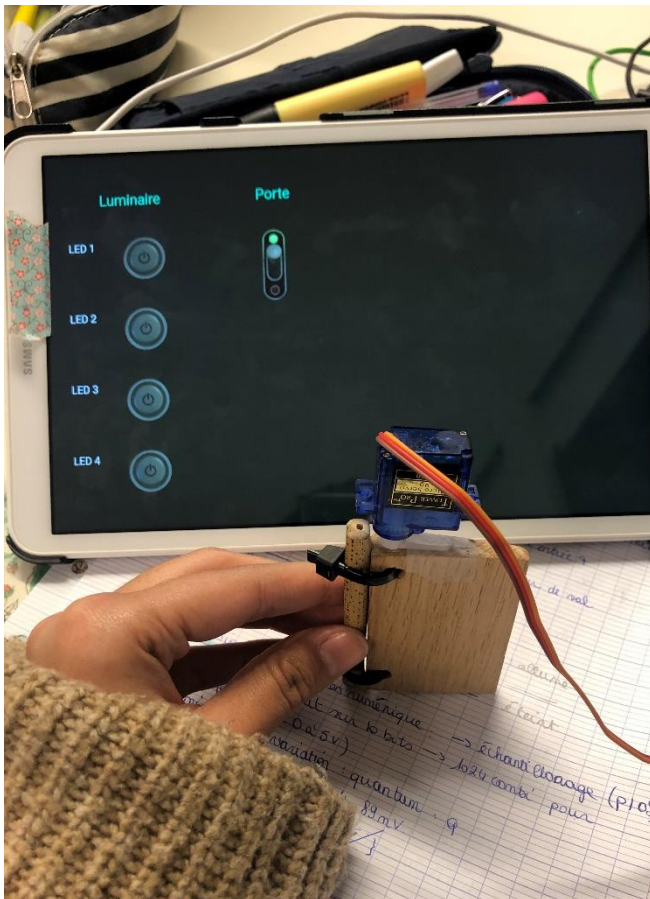


4. Configuration module et connexion du servomoteur

J'ai configuré le module HC-06 en lui attribuant un nouveau nom et un nouveau code pin.



À la suite de cela, j'ai établi la connexion entre le servomoteur et la tablette. Voici le montage, le code et une vidéo montrant le fonctionnement. Le code est disponible dans le dossier Bluetooth de la séance 5 sur GitHub. <https://youtu.be/HG8tyFyf5BY>



```

ouverture_et_fermeture

#include<SoftwareSerial.h>
#define RX 10
#define TX 11
SoftwareSerial BlueT(RX,TX);
char Data;

#include <Servo.h> // on inclut la bibliothèque pour piloter un servomoteur
Servo monServo; // on crée l'objet monServo

void setup() {
  Serial.begin(9600);
  BlueT.begin(9600);

  monServo.attach(9);
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
  if (BlueT.available()){
    Data=BlueT.read();

    //ouverture
    if(Data=='O'){
      monServo.write(35); //ouvre la porte
      Serial.println("la porte s'ouvre");}

    //fermeture
    if(Data=='F'){
      monServo.write(180); //ferme la porte //ouvre la porte
      Serial.println("la porte se ferme");}
  }
}

```

5. Interface et portail

Afin de dissocier quel est le bouton de chaque élément électronique, j'ai organisé l'interface de l'application en regroupant par catégories.

Enfin pour conclure cette séance j'ai réalisé la motorisation du portail à l'aide d'un servomoteur mg995. J'ai réalisé l'ouverture et la fermeture à l'aide de delay, cependant je devrai ajuster chaque delay afin d'ouvrir et fermer le portail coulissant au dimension prévue.

Le code est disponible dans le dossier Bluetooth de la séance 5 sur GitHub.

Voici la vidéo de démonstration du portail.

<https://youtu.be/wPajN5d6Sa0>

