

# Projet Arduino : AmicaPlant

## (serre connectée)

### Rapport séance n°3 – Jessica Kahungu

#### Structure de la serre

Durant cette séance, j'ai pu passer à la découpe laser de la structure de la serre, en bois d'épaisseur 5mm pour qu'elle soit la plus solide possible. Cette partie a pris plus de temps que prévu, les deux premières pièces, découpées à partir de la même planche de bois, se sont parfaitement bien découpées, il n'y a pas eu de problème pour retirer les pièces de la planche avec les encoches intactes. Pour toutes les pièces d'après, malgré parfois jusqu'à trois passages au laser, il a fallu suivre tous les contours au cutter, pour pouvoir retirer la pièce de la planche de bois. Cela a pris beaucoup de temps, notamment pour découper la pièce de façon précise (sans abîmer les encoches). Finalement, j'ai pu terminer les pièces et monter la serre pour avoir une idée de son aspect. Une arête en particulier s'emboîte avec les faces adjacentes mais a du mal à tenir, il faudra donc bien coller les pièces et éventuellement combler les petits s'il en reste. Nous passerons ensuite du papier sablé pour bien lisser toutes les faces de la serre, pour ensuite peindre la serre. En revanche, les ouvertures sont très convenables, surtout celles correspondant au ventilateur (face de derrière) et celle correspondant à l'écran LCD sur la face avant.

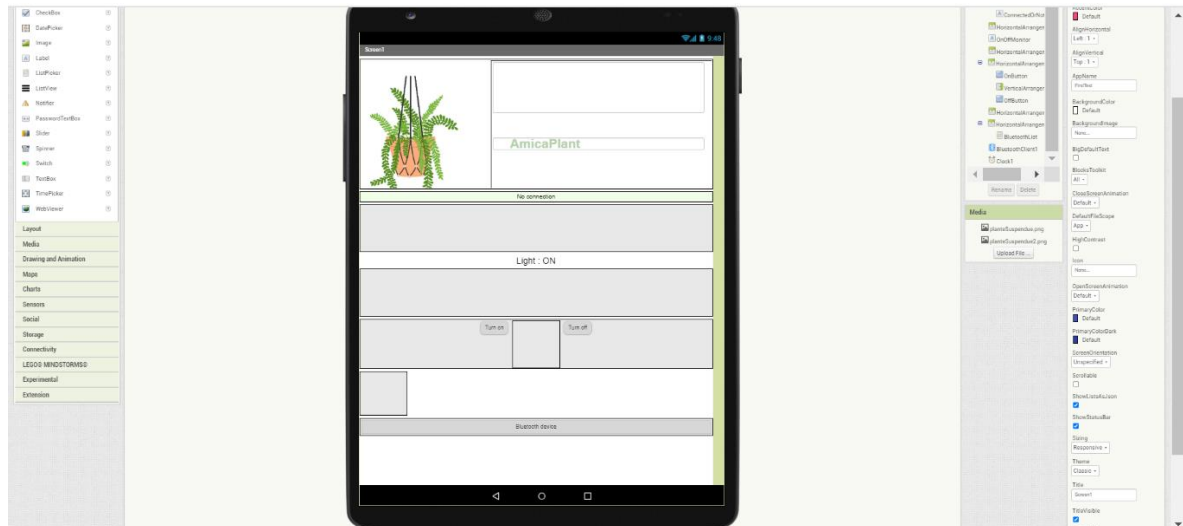


#### Application

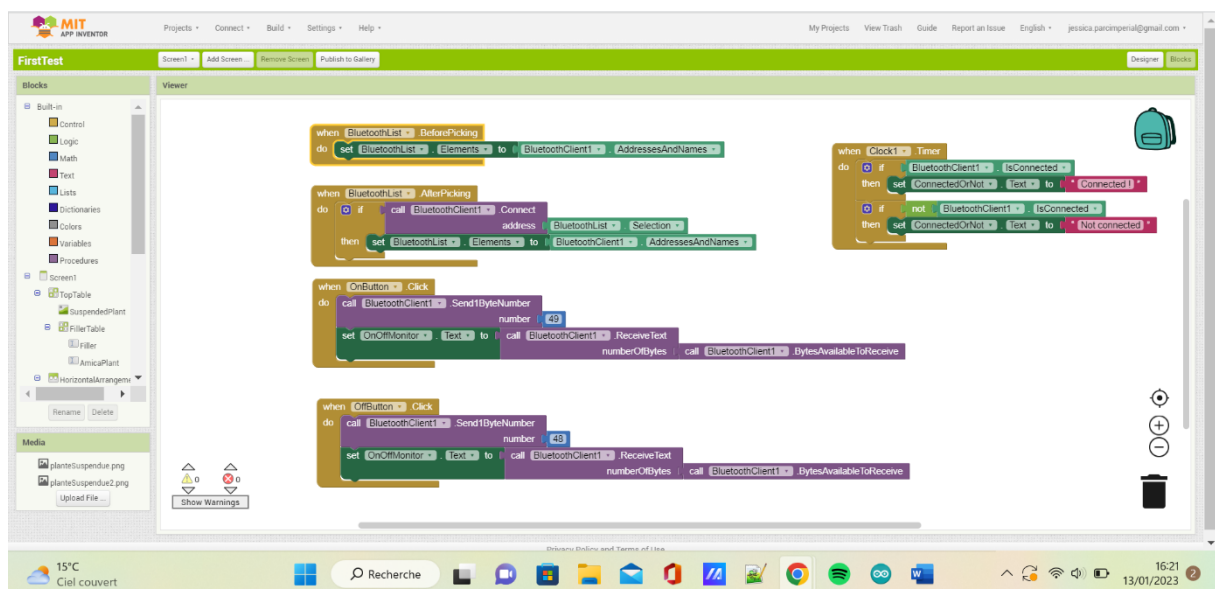
Grâce au MIT App Inventor, j'ai pu faire une petite application très rapidement, l'interface graphique ainsi que les blocs de « code » qui permettent de faire fonctionner l'application et la connexion Bluetooth avec le module HC-06. Avant de transférer la dernière version de l'application sur mon

téléphone (cette version est un test pour simplement allumer une LED), je peux tester l'application grâce à un QR Code que me donne le site.

L'interface graphique :



Les blocs de code :



Après cette première expérience avec ce site, j'ai pu globalement en comprendre le fonctionnement et je pourrais m'en servir pour l'application finale.

L'aspect final sur téléphone :



## Connexion Bluetooth

J'ai eu beaucoup de difficultés au niveau de la connexion Bluetooth avec mon application. En effet, le module Bluetooth ne répondait pas : le programme ne compilait pas (il n'y avait pas de messages d'erreur mais ça ne fonctionnait toujours pas), puis il y avait un problème de port que je ne comprenais pas donc j'ai pris beaucoup de temps pour vérifier et revérifier le circuit et le code, et faire beaucoup de recherches sur les modules HC-05 et HC-06. J'avais beaucoup modifié le code en essayant de trouver le problème, avec l'aide de monsieur Charlon aussi mais on ne trouvait pas parce que la connexion avec mon téléphone avait marché une fois (j'ai pu changer le nom du module) puis n'a plus marché. J'ai donc décidé de repartir de zéro, j'ai redémarré Arduino et mon ordinateur, j'ai changé tous les fils et j'ai recommencé le programme en vérifiant chaque ligne.

Après beaucoup d'essais et de print, je me suis aperçue qu'une fois qu'on affiche dans le moniteur série de la valeur de `BlueT.read()` (avec `BlueT` qui est le canal de communication en provenance du module HC-06), sa valeur changera étant donné qu'on a déjà appelé la fonction `read()`. Dans le code que j'ai récupéré du cours et encore d'autres modèles de code que j'ai suivi, il y avait toujours un `print` de `BlueT.read()`, pour voir dans le moniteur série ce que reçoit le module. C'est en tâtonnant longtemps avec différents `print`, notamment voir si `BlueT.read()==49` lorsque j'envoie 49 (donc 1 en ASCII) à partir de mon application, que j'ai pu trouver ce problème. J'ai affiché deux fois la même chose à la suite dans le moniteur série :

```
Serial.println(BlueT.read());
```

```
Serial.println(BlueT.read());
```

Etonnamment, dans le premier cas j'obtenais 49 (lorsque j'appuyais sur allumer la LED sur mon téléphone), et dans le deuxième cas, en appuyant le même bouton, j'obtenais le caractère « 1 ».

J'ai donc compris que, même si auparavant j'avais déjà essayé de stocker la valeur de `BlueT.read()` dans une variable, le problème était avec dû au `print` de `BlueT.read()`, que je laissais constamment

pour pouvoir afficher les valeurs envoyées par Bluetooth sur le moniteur série. J'ai retiré cette ligne et tout a marché. Voici le programme Arduino qui fonctionne :

```
GoodLed | Arduino 1.8.19
File Edit Sketch Tools Help

GoodLed

#include<SoftwareSerial.h>
#define RX 10
#define TX 11
SoftwareSerial BlueT(RX,TX);
const int LED=13;
int mess;

void setup() {

  Serial.begin(9600);
  delay(500);
  Serial.println("Bonjour -Pret pour les commandes AT");
  BlueT.begin(9600);
  delay(500);
  pinMode(LED, OUTPUT);

}

void loop() {

  //send
  while (BlueT.available()>0) {
    mess=BlueT.read();
```

```
GoodLed | Arduino 1.8.19
File Edit Sketch Tools Help

GoodLed

}

void loop() {

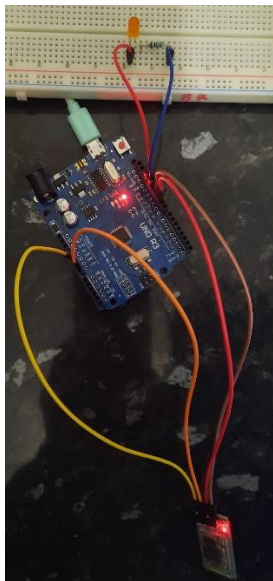
  //send
  while (BlueT.available()>0) {
    mess=BlueT.read();

    if(mess==49){
      digitalWrite(LED, HIGH);
    }
    else if(mess==48){
      digitalWrite(LED, LOW);
    }
  }

  //arduino read
  while (Serial.available()>0) {
    BlueT.write(char(Serial.read()));
  }

}
```

Le montage fait est le suivant :



Voici un lien de démonstration du fonctionnement de l'application à partir d'un QR code :

<https://www.youtube.com/watch?v=GZcwDPk3Z08>

Je ne sais pas si ce problème s'applique uniquement au module que j'ai ou si c'est la même chose avec tous les modules. Je ferai d'autres essais d'ici jeudi et je demanderai à d'autres personnes s'ils ont quelque chose de similaire, parce qu'après avoir testé l'application sur plusieurs appareils (parce

qu'une image ne s'affichait pas correctement sur un autre téléphone que le mien), au bout d'un certain temps le module Bluetooth ne se connectait plus, même après avoir tout redémarré donc je cherche à comprendre si le problème vient du module ou de moi.

L'objectif final est de pouvoir activer et désactiver la lumière, le ventilateur et l'arrosage avec une application comme celle-ci.