

# Rapport de projet

Séance : 10 janvier 2022

## Objectif de séance :

Avec mon binôme nous voulions consacrer la séance à établir la connexion Bluetooth avec nos composants. De plus la séance a été interrompue par notre passage à l'oral afin de présenter notre projet.

## Déroulement de la séance :

### 1. App Inventor

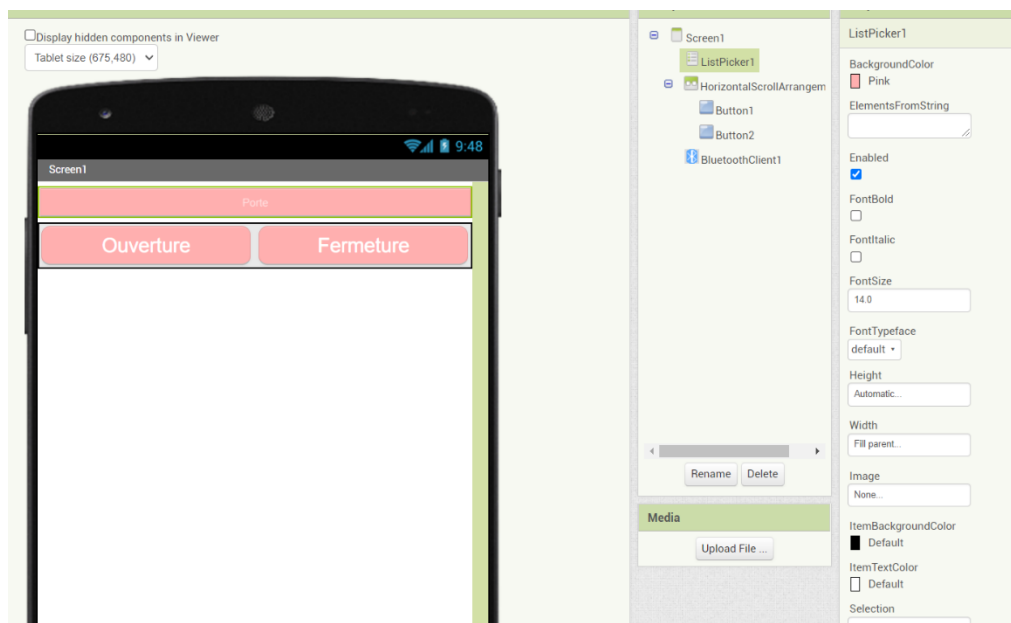
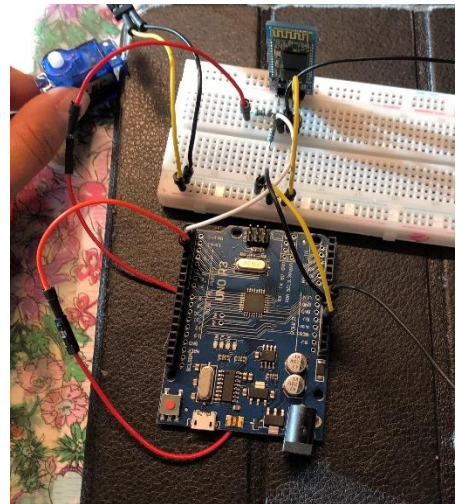
En amont du projet nous avons choisi le logiciel MIT APP INVENTOR afin d'établir la connexion Bluetooth.

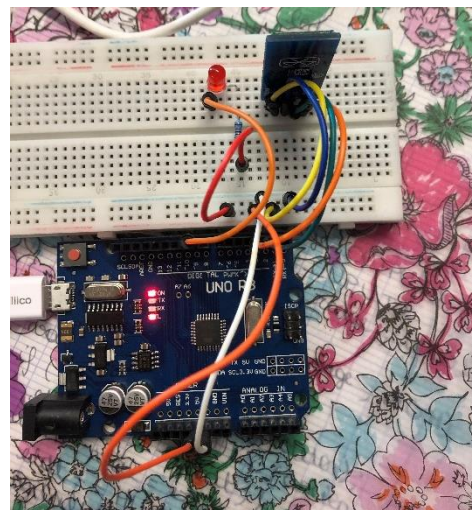
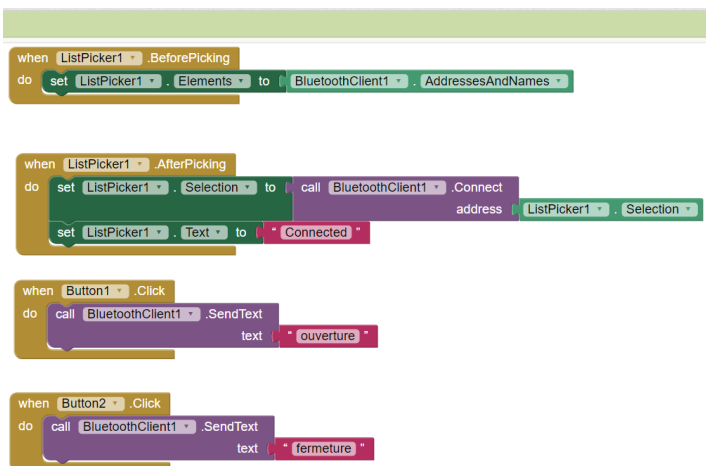
J'ai réalisé un premier montage avec un servomoteur qui était censé ouvrir la porte et la fermer à distance. Cependant je ne suis pas parvenue à créer l'application sur le logiciel.

Donc j'ai repris les recherches et je suis tombée sur une vidéo qui expliquait comment allumer une LED à distance grâce au logiciel. Voici la vidéo en question.

<https://www.youtube.com/watch?v=aQcJ4uHdQEA>

J'ai reproduit les manipulations de la vidéo, c'est-à-dire le montage et la création de l'application sur mon logiciel. J'ai essayé d'adapter le logiciel à mon problème : gérer le pilotage d'une porte tout en conservant un montage avec une LED.



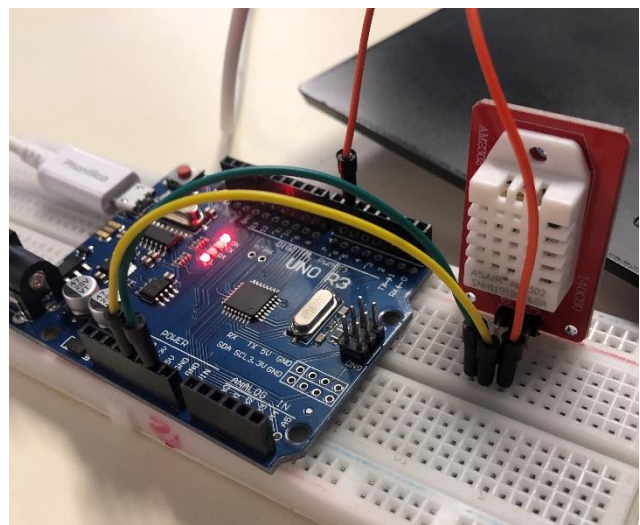
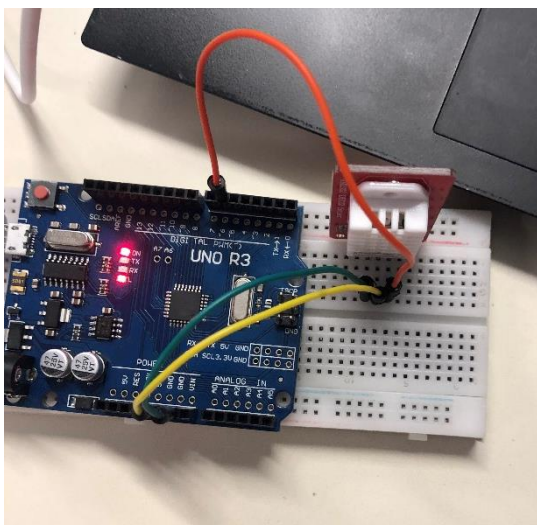


Cependant l'utilisation du logiciel reste encore trop floue pour aboutir à un résultat concret. J'ai donc continué les recherches afin d'obtenir plus d'explications. Pour éviter de perdre du temps sur la séance j'ai décidé de m'occuper du recensement de la température et de l'humidité, et finir mes recherches chez moi.

## 2. Module DHT22

J'ai commencé par des recherches afin d'en apprendre un maximum sur ce nouveau module. Le module DHT22 permet de capter la température et l'humidité. Contrairement au module DHT11, notre module capte des températures comprises entre  $-40^{\circ}$  et  $+80^{\circ}$  contre  $0^{\circ}$  et  $50^{\circ}$  pour le second composant. DHT22 possède une précision de  $\pm 0.5^{\circ}$  et permet un affichage au dixième. Concernant l'humidité, notre composant permet d'avoir une meilleure précision car il capte entre 0 à 100% d'humidité contre 20 à 80% pour le DHT11.

J'ai tout d'abord téléchargé et installé la librairie dht22 sur arduino. À la suite de cela j'ai réalisé le montage proposé par le site [PassionElectronique.fr](https://passionelectronique.fr/tutorial-dht22/#survol-rapide) <https://passionelectronique.fr/tutorial-dht22/#survol-rapide>.



Voici le code et le résultat.

```
#include <DHT.h>

#define brocheBranchementDHT 6 // Si la ligne de données du DHT22 est branchée
//sur la pin D6 de votre Arduino, par exemple

#define typeDeDHT DHT22 // Si votre DHT utilisé est un DHT22 (mais cela
//pourrait aussi bien être un DHT11 ou DHT12, par exemple)

DHT dht(brocheBranchementDHT, typeDeDHT);

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Serial.println("Programme de test du DHT22");
  Serial.println("=====");
  Serial.println();

  // Initialisation du DHT22;
  dht.begin();
}

void loop() {
  // put your main code here, to run repeatedly:

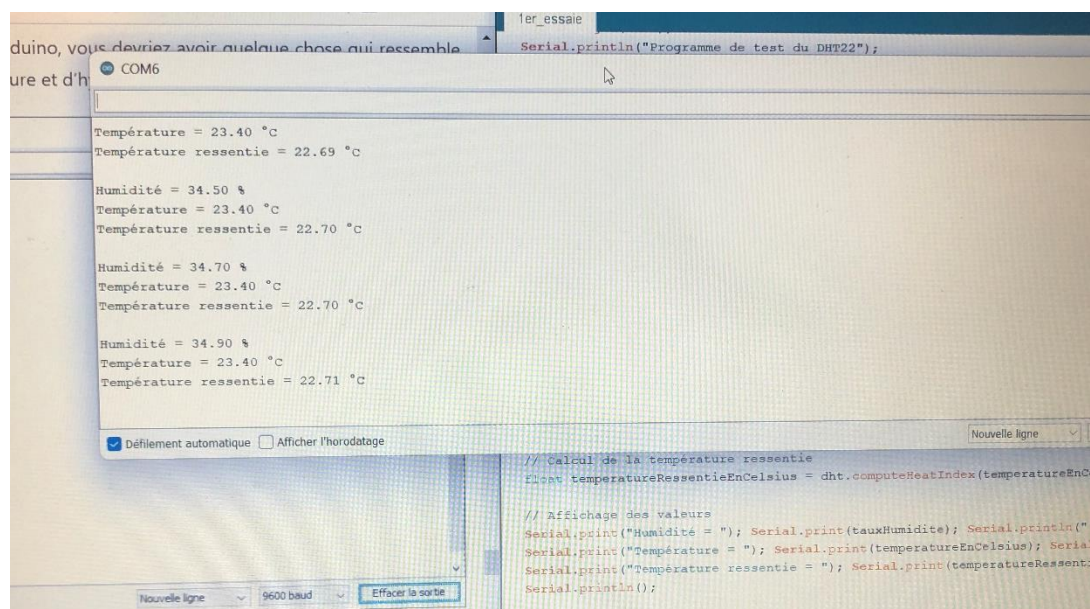
  // Lecture des données
  float tauxHumidite = dht.readHumidity(); // Lecture du taux d'humidité (en %)
  float temperatureEnCelsius = dht.readTemperature(); // Lecture de la température, exprimée en degrés Celsius

  // Vérification si données bien reçues
  if (isnan(tauxHumidite) || isnan(temperatureEnCelsius)) {
    Serial.println("Aucune valeur retournée par le DHT22. Est-il bien branché ?");
    delay(2000);
    return; // Si aucune valeur n'a été reçue par l'Arduino, on attend 2 secondes, puis on redémarre la fonction loop()
  }

  // Calcul de la température ressentie
  float temperatureResentieEnCelsius = dht.computeHeatIndex(temperatureEnCelsius, tauxHumidite, false); // Le "false" est là pour dire qu'on travaille en °C, et non en °F

  // Affichage des valeurs
  Serial.print("Humidité = "); Serial.print(tauxHumidite); Serial.println(" %");
  Serial.print("Température = "); Serial.print(temperatureEnCelsius); Serial.println(" °C");
  Serial.print("Température ressentie = "); Serial.print(temperatureResentieEnCelsius); Serial.println(" °C");
  Serial.println();

  // Temporisation de 2 secondes (pour rappel : il ne faut pas essayer de faire plus d'1 lecture toutes les 2 secondes, avec le DHT22, selon le fabricant)
  delay(2000);
}
```



The screenshot shows the Arduino IDE interface with the serial monitor open. The serial monitor displays the following output:

```
Programme de test du DHT22
=====

Température = 23.40 °C
Température ressentie = 22.69 °C

Humidité = 34.50 %
Température = 23.40 °C
Température ressentie = 22.70 °C

Humidité = 34.70 %
Température = 23.40 °C
Température ressentie = 22.70 °C

Humidité = 34.90 %
Température = 23.40 °C
Température ressentie = 22.71 °C
```

The serial monitor settings are configured to COM6, 9600 baud, and the 'Défilement automatique' (Automatic scroll) option is checked. The code in the background shows the DHT22 library being used to read humidity and temperature data.

J'ai modifié ce code afin d'obtenir uniquement l'humidité et la température « réelle ».



### 3. Tableau LCD

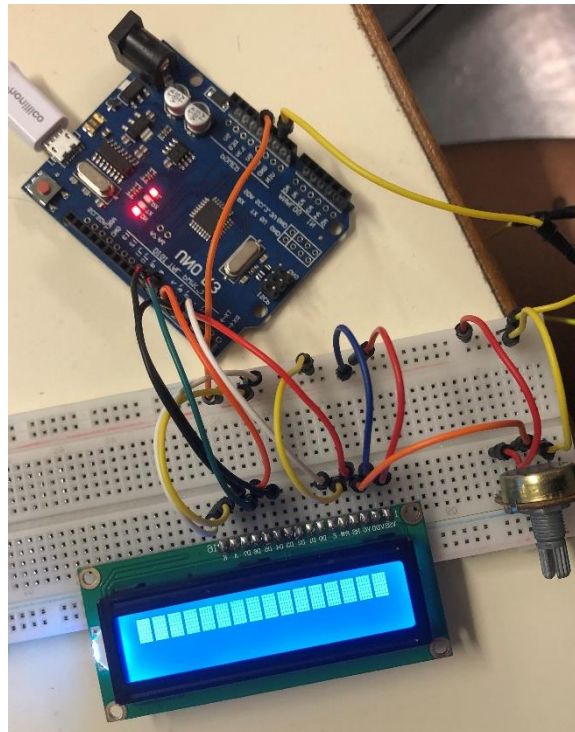
J'ai fini ma séance sur l'apprentissage de l'affichage sur un tableau LCD, sur lequel je voulais retranscrire les résultats trouvés avec le module DHT22. Sur le cours d'initiation de l'arduino de Monsieur Masson j'ai trouvé un montage et un code simple que j'ai reproduis afin de me familiariser avec ce nouveau composant. Voici le code, le montage et le résultat obtenu.

```
//affichage d'un texte sur un LCD
#include <LiquidCrystal.h>

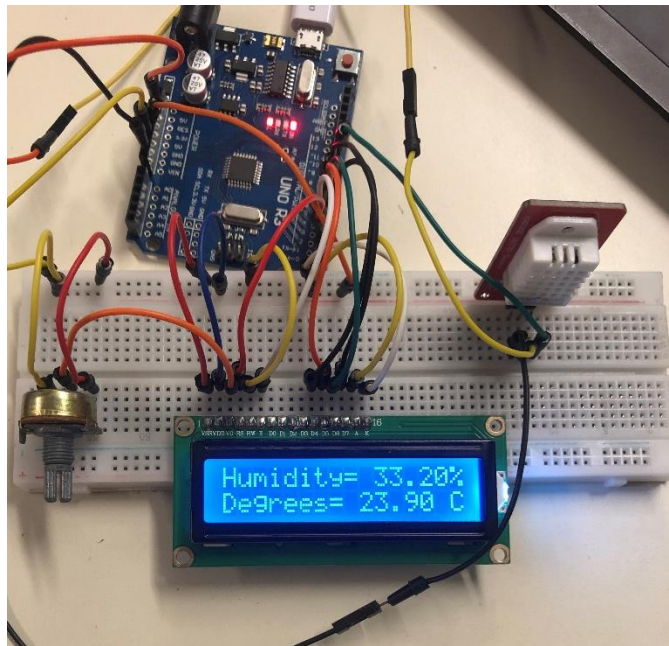
LiquidCrystal lcd(2,3,4,5,6,7);

void setup() {
  // put your setup code here, to run once:
  lcd.begin(16,2);
  lcd.print("Température ");
  lcd.setCursor(0,1);
  lcd.print("je suis en PEIP2");
}

void loop() {
  // put your main code here, to run repeatedly:
}
```



Enfin j'ai fusionné le précédent montage avec celui du module DHT22 et complété le code afin d'afficher la température et l'humidité. Cependant le tableau ne permet pas d'afficher les accents donc j'ai choisi un affichage en anglais.



```

#include <DHT.h>

#define brocheBranchementDHT 10 // Si la ligne de données du DHT22 est branchée
                                //sur la pin D6 de votre Arduino, par exemple

#define typeDeDHT DHT22 // Si votre DHT utilisé est un DHT22 (mais cela
                        //pourrait aussi bien être un DHT11 ou DHT21, par exemple)

DHT dht(brocheBranchementDHT, typeDeDHT);

#include <LiquidCrystal.h>

LiquidCrystal lcd(2,3,4,5,6,7);

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Serial.println("Programme de test du DHT22");
  Serial.println("-----");
  Serial.println();

  // Initialisation du DHT22;
  dht.begin();

  //initialise le lcd
  lcd.begin(16,2);
}

void loop() {
  // put your main code here, to run repeatedly:

  // Lecture des données
  float tauxHumidite = dht.readHumidity(); // Lecture du taux d'humidité (en %)
  float temperatureEnCelsius = dht.readTemperature(); // Lecture de la température, exprimée en degrés Celsius

  // Vérification si données bien reçues
  if (isnan(tauxHumidite) || isnan(temperatureEnCelsius)) {
    Serial.println("Aucune valeur retournée par le DHT22. Est-il bien branché ?");
    delay(2000);
    return; // Si aucune valeur n'a été reçue par l'Arduino, on attend 2 secondes, puis on redémarre la fonction loop()
  }

  // Affichage des valeurs
  Serial.print("Humidité = "); Serial.print(tauxHumidite); Serial.println(" %");
  Serial.print("Température = "); Serial.print(temperatureEnCelsius); Serial.println(" °C");
  Serial.println();

  //affiche sur lcd
  lcd.setCursor(0,0);
  lcd.print("Humidity= ");lcd.print(tauxHumidite); lcd.print("%");
  lcd.setCursor(0,1);
  lcd.print("Degrees= ");lcd.print(temperatureEnCelsius); lcd.print(" °C");

  // Temporisation de 2 secondes (pour rappel : il ne faut pas essayer de faire plus d'1 lecture toutes les 2 secondes, avec le DHT22, selon le fabricant)
  delay(2000);
}

```