

Rapport

Ayant un mac en ma possession je n'ai pas pu tester les programmes finalisés pendant les vacances. Ainsi lors de cette séance, en empruntant l'ordinateur de mon binôme j'ai pu continuer le Bluetooth et j'ai ainsi remarqué qu'il n'y avait pas de transmission de donnée du maître (HC05) à l'esclave (HC06). L'objectif de ces deux modules Bluetooth est d'obtenir la température, la température ressentie et l'humidité via le capteur DHT22 (rover) sur l'écran LCD (télécommande). De plus une difficulté corsait le tout : le module HC06 avait du mal à se connecter car il y avait trop de module Bluetooth dans la salle de TP.

Programme du module Bluetooth HC05 et du capteur de température : (il faut encore le modifier)

```
#include<SoftwareSerial.h>
#include <DHT.h>
#define RX 3
#define TX 2
#define DHTPIN 6 // branchée sur la pin D6

// Le type de DHT utilisé
#define DHTTYPE DHT22 // ici on utilise DHT22

//Instanciation de la librairie DHT
DHT dht(DHTPIN, DHTTYPE);

//Instanciation de la librairie SoftwareSerial
SoftwareSerial BlueT(RX,TX);

String reponse;
String message;

void setup() {
  Serial.begin(9600);
  Serial.println("5");
  Serial.println("Bonjour -Pret pour les commandes AT");
  Serial.println();
  dht.begin();
  BlueT.begin(9600);
}

void loop() {
  //lire les réponses de l'esclave
  while (BlueT.available()){
    delay(10);
    if (BlueT.available()>0) {
      char c = BlueT.read();
      answer += c;
    }
  }
  // Lecture des données
  float tauxHumidite = dht.readHumidity(); // Lecture du taux d'humidité en pourcentage
  float tempCelsius = dht.readTemperature(); // Lecture de la température en degrés Celsius

  // Vérification
  if (isnan(tauxHumidite) || isnan(tempCelsius)) {
    Serial.println("Aucune valeur retournée. vérifier le branchement");
    delay(2000);
    return; // Si aucune valeur reçue, on a une attente de deux secondes, puis redémarrage de cette fonction
  }
  // Calcul de la température ressentie
  float tempResentieCelsius = dht.computeHeatIndex(tempCelsius, tauxHumidite, false); // le false permet de montrer qu'on est
  en celsius et non en fahrenheit
```

```

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

//envoyer data vers esclave
Serial.print("Le maître envoie : ");
Serial.println(tempCelsius);
BlueT.print(tempCelsius);

//Serial.print("Le maître envoie : ");
//Serial.println(tauxHumidite);
//BlueT.print(tauxHumidite);

//Serial.print("Le maître envoie : ");
//Serial.println(tempResentieCelsius);
//BlueT.print(tempResentieCelsius);

//Send answer to monitor
if(answer!=""){
  Serial.print("L'esclave reçoit : ");
  Serial.println(answer);
  answer="";
}
delay(2000);
Serial.flush();
}

```

Programme du module Bluetooth HC06 et de l'écran LCD : (il faut encore le modifier)

```

#include<SoftwareSerial.h>
#include <LiquidCrystal.h>
#define RX 11
#define TX 10

String message;
float t=0;
float h=0;

SoftwareSerial BlueT(RX,TX);
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);

void setup() {
  Serial.begin(9600);
  Serial.println("6");
  lcd.begin(20, 4);
  BlueT.begin(9600);
}

void loop() {
  if (BlueT.available()){
    Data=BlueT.read();
    if (Data=='A') {
      PWM=BlueT.parseFloat();
      Serial.println(PWM);
      analogWrite(LED, PWM);
    }
  }
  //Humi=BleuT.read();
  //res=BleuT.read();
}

```

```
//Affichages :  
// Ligne 1:  
lcd.setCursor(0, 0);  
// positionne le curseur à la colonne 1 et à la ligne 1  
lcd.print("Humidite: ");  
lcd.print(t);  
//lcd.setCursor(10, 0);  
//lcd.print(h);  
//lcd.setCursor(15, 0);  
//lcd.print("%");  
//lcd.setCursor(0,2);  
//lcd.print("mode :");  
//lcd.setCursor(10,2);  
//lcd.print("sport");  
  
// Ligne 2:  
//lcd.setCursor(0, 1);  
// positionne le curseur à la colonne 1 et à la ligne 2  
//lcd.print("Temp: ");  
//lcd.setCursor(10, 1);  
//lcd.print(t);  
  
delay(3000);  
}
```