TP : Arduino, Capteur de Température, Ultrason et Bluetooth

Objectif du TP

Le but de ce TP est de créer un système IoT simple en utilisant une carte Arduino, un capteur de température, un capteur ultrasonique et un module Bluetooth. Le système doit être capable de mesurer la température, la distance à un obstacle avec le capteur ultrason, et transmettre ces données via Bluetooth à un appareil distant

Matériel nécessaire

1. Arduino Uno
2. Capteur de température
3. Capteur ultrasonique
4. Module Bluetooth (par exemple, HC-05)
5. Breadboard et fils de connexion
6. Deux LED (une verte et une rouge)
7. Résistances (si nécessaire)
8. Smartphone ou ordinateur avec Bluetooth

Montage :

1. Connectez le capteur DHT11 à l'Arduino :
    i. Broche **VCC** du DHT11 à la broche **5V** de l'Arduino.
    ii. Broche **DATA** du DHT11 à la broche **2** de l'Arduino.
    iii. Broche **GND** du DHT11 à la broche **GND** de l'Arduino.
2. Connectez le capteur ultrasonique HC-SR04 à l'Arduino :
    i. Broche **VCC** du HC-SR04 à la broche **5V** de l'Arduino.
    ii. Broche **TRIG** du HC-SR04 à la broche **7** de l'Arduino.
    iii. Broche **ECHO** du HC-SR04 à la broche **6** de l'Arduino.
    iv. Broche **GND** du HC-SR04 à la broche **GND** de l'Arduino.
3. Connectez le module Bluetooth HC-05 à l'Arduino :
    i. Broche **VCC** du HC-05 à la broche **5V** de l'Arduino.
    ii. Broche **GND** du HC-05 à la broche **GND** de l'Arduino.
    iii. Broche **TXD** du HC-05 à la broche **2** de l'Arduino.
    iv. Broche **RXD** du HC-05 à la broche **3** de l'Arduino.

4. LED verte :
    i. Connectez l'anode (le côté plus long) à la broche numérique 10 de l'Arduino via une résistance de 220 ohms.
    ii. Connectez la cathode (le côté plus court) à la masse de l'Arduino.
5. LED rouge :
    i. Connectez l'anode (le côté plus long) à la broche numérique 9 de l'Arduino via une résistance de 220 ohms.
    ii. Connectez la cathode (le côté plus court) à la masse de l'Arduino.

Programme :

1. Configuration des bibliothèques et des broches :

Inclure les bibliothèques nécessaires, dans ce cas, DHT.h pour le capteur DHT11. Définir les broches utilisées pour le capteur DHT11, le capteur ultrasonique et le module Bluetooth.

```
#include <DHT.h>
#include <DHT_U.h>
#include <SoftwareSerial.h>
#define DHTPIN 4
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
#define trigPin 7
#define echoPin 6
#define ledGreen 10
#define ledRed 9
SoftwareSerial bluetooth(2, 3);  // TX, RX
```

2. Configuration du setup() :

Initialiser la communication série pour la surveillance via le moniteur série. Initialiser le capteur DHT11. Configurer les broches du capteur ultrasonique en tant que sortie ou entrée.

```
void setup() {
  Serial.begin(9600);
  bluetooth.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(ledGreen, OUTPUT);
  pinMode(ledRed, OUTPUT);     }
```

3. Boucle principale loop() :

Mesurer la température et l'humidité avec le capteur DHT11.Mesurer la distance avec le capteur ultrasonique. Envoyer les données mesurées via Bluetooth.

```
    void loop() {
      digitalWrite(ledRed, HIGH);   // La LED rouge s'allume pour indiquer l'attente
      digitalWrite(ledGreen, LOW);  // La LED verte est éteinte pour indiquer l'impossibilité d'envoyer des données
      if (bluetooth.available() > 0) {
        char command = bluetooth.read();
        digitalWrite(ledGreen, HIGH);  // La LED verte s'allume pour indiquer la disponibilité d'envoyer des données
        digitalWrite(ledRed, LOW);     // La LED rouge s'éteint
        switch (command) {
          case 'T':
            float tempC;
            tempC = dht.readTemperature();
            bluetooth.println(tempC);
            break;
          case 'H':
            float humidity;
            humidity = dht.readHumidity();
            bluetooth.println(humidity);
            break;
          case 'D':
            float duration, distance;
            digitalWrite(trigPin, LOW);
            delayMicroseconds(2);
            digitalWrite(trigPin, HIGH);
            delayMicroseconds(10);
            digitalWrite(trigPin, LOW);
            duration = pulseIn(echoPin, HIGH);
            distance = (duration / 2) / 29.1;
            bluetooth.println(distance);
            break;
        }
      }
    }
```