

```
#include <SPI.h>

// appel des librairies nécessaires :

#include <Wire.h>

#include <Adafruit_Sensor.h>

#include <Adafruit_BME280.h>

#include "rgb_lcd.h"

#include "FastLED.h"

// définition des variables des capteurs du barometre

#define BME_SCK 13

#define BME_MISO 12

#define BME_MOSI 11

#define BME_CS 10

#define NUM_LEDS 12 // on définit le nombre de leds de notre ring

#define DATA_PIN 6 // on définit la pin du ledring

#define potentiometer A0 // on définit la pin du potentiometre

int potValue = 0; // on initialise la variable du potentiometre

int outputValue = 0; // on initialise la variable de sortie

CRGB leds[NUM_LEDS]; // on définit un tableau pour les leds

#define SEALEVELPRESSURE_HPA (1024.6)

Adafruit_BME280 bme(BME_CS, BME_MOSI, BME_MISO, BME_SCK); // software SPI
```

```
rgb_lcd lcd; // on déclare notre LCD.
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
    lcd.begin(16, 2); // On démarre de lcd
```

```
    lcd.setRGB(0, 127, 255); // On choisit une couleur pour le fond d'écran
```

```
    lcd.setCursor(0, 2); // On place le curseur sur la colonne puis la ligne désirée
```

```
    lcd.print("hello world:"); // Ici, on écrit message de bienvenue
```

```
    delay(4000); // on attend un peu
```

```
    lcd.clear(); // vider le texte de l'écran
```

```
    FastLED.addLeds<NEOPIXEL, DATA_PIN>(leds, NUM_LEDS);
```

```
    FastLED.setBrightness(100);
```

```
    bme.begin(); // On démarre le barometre
```

```
    for(int i = 0; i <=11; i++)
```

```
    {
```

```
        leds[i] = CRGB::Blue;
```

```
        FastLED.show();
```

```
    }
```

```
    delay(1000);
```

```
        for(int i = 0; i <=11; i++)
```

```
        {
```

```
            leds[i] = CRGB::White;
```

```
            FastLED.show();
```

```
        }
```

```
    delay(1000);
    for(int i = 0; i <=11; i++)
    {
        leds[i] = CRGB::Blue;
        FastLED.show();

    }
    delay(1000);

    for(int i = 0; i <=11; i++)
    {
        leds[i] = CRGB::White;
        FastLED.show();

    }
    delay(1000);

}

void loop() {

    potValue = analogRead(A0);
    outputValue = map(potValue, 0, 1023, 0, 255);
    FastLED.setBrightness(outputValue); // affiche le niveau de luminosité selon valeur captée au
    potentiometre qui a été traduite en valeur numérique pour le ledsring

    if(bme.readTemperature())<25){
        for(int i = 0; i <=11; i++)
        {
```

```

    leds[i] = CRGB::Blue;
    FastLED.show();

}
lcd.setRGB(30, 144, 255);// On choisit une couleur pour ce seuil de température
}
else{
    for(int i = 0; i <=11; i++)
    {
        leds[i] = CRGB::Red;
        FastLED.show();
    }
    lcd.setRGB(180, 8, 8);// On choisit une couleur pour ce seuil de température
}
    lcd.setCursor(0, 2); // On place le curseur sur la colonne puis la ligne désirée

// imprime sur l'écran LCD de l'ordinateur la température:
lcd.print("T ");
lcd.print(bme.readTemperature());
lcd.print(" Celsius");
delay(4000);// on attend un peu
lcd.clear();//vider le texte de l'écran

//imprime sur l'écran LCD de l'ordinateur l'humidité
lcd.setCursor(0, 2); // On place le curseur sur la colonne puis la ligne désirée
lcd.print("Humid ");
lcd.print(bme.readHumidity());
lcd.print(" %");
delay(4000); // on attend un peu
lcd.clear(); //vider le texte de l'écran

```

```
Serial.print("Temperature = ");
Serial.print(bme.readTemperature());
Serial.println(" *C");
Serial.print("Pression = ");
Serial.print(bme.readPressure() / 100.0F);
Serial.println(" hPa");
Serial.print("Altitude = ");
Serial.print(bme.readAltitude(SEALEVELPRESSURE_HPA));
Serial.println(" m");
Serial.print("Humidite = ");
Serial.print(bme.readHumidity());
Serial.println(" %");
Serial.println();
delay (1000);

}
```