```
/*
  Arduino with SH1106 OLED display (128x64 Pixel) and BME280 sensor.
  BME280 is barometric pressure, temperature and humidity sensor.
  This is a free software with NO WARRANTY.
*/
#include <Wire.h> // include Arduino wire library (required for I2C devices)
#include <Adafruit_GFX.h> // include Adafruit graphics library
#include <Adafruit_SH1106.h> // include Adafruit SH1106 OLED display driver
#include <Adafruit_BME280.h> // include Adafruit BME280 sensor library
#include <OneButton.h>
#include <math.h>

// Declaration for an SH1106 display connected to I2C (SDA, SCL pins)
#define OLED_RESET 4
Adafruit_SH1106 display(OLED_RESET);

// define device I2C address: 0x76 or 0x77 (0x77 is library default address)
#define BME280_I2C_ADDRESS 0x76
Adafruit_BME280 bme; // initialize Adafruit BME280 library

// Setup a OneButton on pin A0.
OneButton button (A0, true);
byte a = 0;

/* Attachez les cathodes (patte courte de la LED) au Ground,
  via une résistance 220 Ohms.
  Connectez les anodes des LED
  bleu => pin 6, verte => pin 7, jaune => pin 8 & rouge => pin 9
  rouge clignotante => pin 2 & buzzer => pin 3
*/
// N° de la broche où sont branchées les LEDs & buzzer
int led_a = 6;
int led_b = 7;
int led_c = 8;
int led_d = 9;

int alarme = 2;
int buzzer = 3;
int Frequence = 440; // 440 Hz

double debut = millis();
float pression_init = 0;

float x = 0;
float y = 0;
float A = 0;
float Tr = 0;

void setup()
{
 delay(1000);
 button.attachClick(choix);
```

```
button.attachLongPressStart(nobuzzer);

// initialize the SH1106 OLED display with I2C address = 0x3C
display.begin(SH1106_SWITCHCAPVCC, 0x3C);

// clear the display buffer.
display.clearDisplay();

display.setTextSize(1); // text size = 1
display.setTextColor(WHITE, BLACK); // set text color to white and black background

// initialize the BME280 sensor
if (bme.begin(BME280_I2C_ADDRESS) == 0 )

  // Test BME280
  // si sensor absent ou mauvaise connection ou mauvaise adresse
  // le sketch restera bloqué sur l'écran "BMP280 Connection Error"
{
  display.setCursor(46, 10);
  display.print("BME280");
  display.setCursor(34, 23);
  display.print("Connection");
  display.setCursor(49, 37);
  display.print("Error");
  display.display();
  while (1);
}
// Ecran d"accueil
display.clearDisplay();
display.setTextSize(2);
display.setCursor(20, 0);
display.print(F("BIENVENUE"));
display.setCursor(18, 25);
display.print(F("A bord du"));
display.setCursor(30, 50);
display.print(F("BLEIMOR"));
display.display();
delay(1000);

pinMode(led_a, OUTPUT); // Configure la broche 6
pinMode(led_b, OUTPUT); // Configure la broche 7
pinMode(led_c, OUTPUT); // Configure la broche 8
pinMode(led_d, OUTPUT); // Configure la broche 9

pinMode(alarme, OUTPUT); // Configure la broche 2

// test LEDs & Buzzer
digitalWrite(led_a, HIGH); //bleu
delay(500);
digitalWrite(led_a, LOW);
digitalWrite(led_b, HIGH); //vert
delay(500);
```

```
  digitalWrite(led_b, LOW);
  digitalWrite(led_c, HIGH); //jaune
  delay(500);
  digitalWrite(led_c, LOW);
  digitalWrite(led_d, HIGH); //rouge
  delay(500);
  digitalWrite(led_d, LOW);

  digitalWrite(alarme, HIGH); // rouge clignotant
  delay(1000);
  digitalWrite(alarme, LOW);
  tone(3, Frequence); //buzzer
  delay(1000);
  noTone (3);

  pression_init = (bme.readPressure() + 45) / 100; // get pressure in hPa + offset(45)/Port Camargue

}
char _buffer[12];

void loop()
{

  delay(500);

  // read temperature, humidity and pressure from the BME280 sensor
  float tempe = bme.readTemperature(); // get temperature in degree Celsius
  float humid = bme.readHumidity();    // get humidity in rH%
  float pression_now = (bme.readPressure() + 45) / 100; // get pressure in hPa + offset(45)/Port
Camargue
  delay(500);

  /* Calcul du Point de Ros&e (Dew Point)
     Formule de Heinrich Gustav Magnus-Tetens
     Domaine de validité :
        T, température mesurée : 0 °C < T < 60 °C
        RH, humidité relative : 0,01 (1 %) < RH < 1,00 (100 %)
        Tr, point de rosée : 0 °C < Tr < 50 °C

     Tr = b α ( T , RH )/ a − α ( T , RH )
     avec : α ( T ,  H ) = (aT /b + T) + ln RH
     a = 17,27 et b = 237,7 [°C].
  */

  tempe = tempe - 2.3; // Offset
  humid = humid + 2.7; // Offset

  x = 17.27;
  y = 237.7;
  A = (x * tempe) / (y + tempe) + log(humid / 100);
  Tr = (y * A) / (x - A);
```

```
/* Relation de base : 1 min. = 60000 ms
   1 heure = 60 min. = 3600000 ms
*/
if (millis() - debut >= 3600000)
{
  if ((pression_init - pression_now) >= 2) {
    digitalWrite(alarme, HIGH);
    tone(3, Frequence);
  }
  if ((pression_init - pression_now) <= 0.5) {
    digitalWrite(alarme, LOW);
    noTone (3);
  }
  debut = millis();
  pression_init = pression_now;
}

/*Méthode Moreux
  http://meteonet.info/html/moreux.html
  Période: Eté
  Vent de sud/sud-est
  > 1020 hPa Beau, mais orages possibles.Très chaud le jour, chaud la nuit.(led_a)
  De 1013 à 1020 hPa Beau, avec possibilité d'averses orageuses. Assez chaud. (led_b)
  De 1006 à 1013 hPa Temps lourd et orages avec averses. Chaud. (led_c)
  < 1006 hPa Averses orageuses et vents violents. Chaud et humide.(led_d)
*/

if (pression_now >= 1020); {
  digitalWrite(led_b, LOW);
  digitalWrite(led_c, LOW);
  digitalWrite(led_d, LOW);
  digitalWrite(led_a, HIGH); //bleu
}
if ((pression_now) >= 1013 & (pression_now) < 1020) {
  digitalWrite(led_a, LOW);
  digitalWrite(led_c, LOW);
  digitalWrite(led_d, LOW);
  digitalWrite(led_b, HIGH); //vert
}
if ((pression_now) > 1006 & (pression_now) < 1013) {
  digitalWrite(led_a, LOW);
  digitalWrite(led_b, LOW);
  digitalWrite(led_d, LOW);
  digitalWrite(led_c, HIGH); //jaune
}
if ((pression_now) <= 1006) {
  digitalWrite(led_a, LOW);
  digitalWrite(led_b, LOW);
  digitalWrite(led_c, LOW);
  digitalWrite(led_d, HIGH); //rouge
}
button.tick();
```

```
switch (a) {
  //-------------------PRESSURE----------------------------------
  case 0:

    display.clearDisplay();
    display.setTextSize(1);
    display.setCursor(0, 0);
    display.print(F("Pressure:"));
    display.setTextSize(2);
    display.setCursor(0, 25);
    display.print(pression_now);
    display.setTextSize(1);
    display.setCursor(87, 28);
    display.print(F("hPa"));
    display.display();
    break;


  //---------------------TEMPERATURE-----------------------------
  case 1:

    display.clearDisplay();
    display.setTextSize(1);
    display.setCursor(0, 0);
    display.print(F("Temperature:"));
    if (tempe < 0) {
      sprintf(_buffer, "-%02u.%02u", (int)abs(tempe), (int)(abs(tempe) * 100) % 100 );
    }
    else
      sprintf(_buffer, " %02u.%02u", (int)tempe, (int)(tempe * 100) % 100 );
    display.setTextSize(2);
    display.setCursor(0, 25);
    display.print(_buffer);
    display.setTextSize(1);
    display.setCursor(87, 28);
    display.print(F("C"));
    // print degree symbols ( ° )
    display.drawRect(80, 25, 3, 3, WHITE);
    display.display();
    break;

  //---------------------HUMIDITY------------------------------
  case 2:

    display.clearDisplay();
    display.setTextSize(1);
    display.setCursor(0, 0);
    display.print(F("Humidity:"));
    sprintf(_buffer, "%02u.%02u", (int)humid, (int)(humid * 100) % 100 );
    display.setTextSize(2);
    display.setCursor(15, 25);
```

```
      display.print(_buffer);
      display.setTextSize(1);
      display.setCursor(80, 28);
      display.print(F("%"));
      display.display();
      break;

    //-------------------DEW POINT--------------------------
    case 3:

      display.clearDisplay();
      display.setTextSize(1);
      display.setCursor(0, 0);
      display.print(F("Dew Point:"));
      display.setTextSize(2);
      display.setCursor(15, 25);
      display.print(Tr);
      display.setTextSize(1);
      display.setCursor(87, 28);
      display.print(F("C"));
      // print degree symbols ( ° )
      display.drawRect(80, 25, 3, 3, WHITE);
      display.display();
      break;
  }
}
//-------------------Fonction CHOIX--------------------------
void choix() {
  a++;
  if (a > 3) {
    a = 0;
  }
  return ;
}
//-------------------Fonction NOBUZZER-----------------------
void nobuzzer() {
  noTone(3);
}

// end of code.
```