

```

#include<Wire.h> //include Arduino wire library (required for I2C devices)
#include<Adafruit_GFX.h> //include Adafruit graphics library
#include<Adafruit_SSD1306.h> //include Adafruit SSD1306 OLED display driver
#include<Adafruit_BME280.h> //include Adafruit BME280 sensor library
#include<Adafruit_Sensor.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 32 // OLED display height, in pixels
#define OLED_RESET 4 //define display reset pin
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
//initialize Adafruit display library

#define BME_280_I2C_ADDRESS 0x76 //define device I2C address: 0x76 or 0x77(0x77 is
library default address)

Adafruit_BME280 bme280; //initialize Adafruit BME280 library

// pin assignments
int UVOUT = A2; //Output from the sensor
int REF_3V3 = A1; //3.3V power on the Arduino board
int LDR_BRIGHTNESS = A3;
int HUMIDITY_SOIL=A0;
int RAIN_ANALOG_IN = A4;
int RAIN_DIGITAL_IN = 22;

boolean isRaining = false;
String strRaining;
int x,minX;

//Takes an average of readings on a given pin
//Returns the average
int averageAnalogRead(int pinToRead)
{
  byte numberOfReadings = 8;
  unsigned int runningValue = 0;

  for(int x = 0 ; x < numberOfReadings ; x++)
    runningValue += analogRead(pinToRead);
  runningValue /= numberOfReadings;

  return(runningValue);
}

//The Arduino Map function but for floats
//From: http://forum.arduino.cc/index.php?topic=3922.0
float mapfloat(float x, float in_min, float in_max, float out_min, float out_max)
{
  return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
}

void setup() {

```

```

// put your setup code here, to run once:

delay(1000); // wait a second

pinMode(HUMIDITY_SOIL,INPUT);
pinMode(UVOUT, INPUT);
pinMode(REF_3V3, INPUT);
pinMode(LDR_BRIGHTNESS, INPUT);
pinMode(RAIN_DIGITAL_IN,INPUT);

// initialize the SSD1306 OLED display with I2C address = 0x3C for 128x32
display.begin(SSD1306_SWITCHCAPVCC, 0x3C);

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

display.setTextSize(0); // text size = 0.5
display.setTextColor(WHITE,BLACK); // set text color to white and black background

display.setTextWrap(false); // disable text wrap

x=display.width();

display.setCursor(8,0);
display.print(" - WEATHER STATION - \n");

display.display();

// initialize the BME280 sensor
if( bme280.begin(BME_280_I2C_ADDRESS) == 0 )
{
    // connection error or device address wrong!
    display.setCursor(34, 0);
    display.print("Connection");
    display.setCursor(49, 23);
    display.print("Error");
    display.display(); // update the display
    while(1); // stay here
}
}

char _bufferTemp[12],_bufferHumid[12],_bufferPress[12];

void loop() {

// read temperature, humidity and pressure from the BME280 sensor
float temp = bme280.readTemperature(); // get temperature in degree Celsius
float humi = bme280.readHumidity(); // get humidity in rH%
float pres = bme280.readPressure(); // get pressure in Pa

// 1: save temperature
delay(5);

```

```

if(temp < 0)
    sprintf(_bufferTemp, "-%02u.%02u C", (int)abs(temp), (int)(abs(temp) * 100) % 100 );
else
    sprintf(_bufferTemp, " %02u.%02u C", (int)temp, (int)(temp * 100) % 100 );

// 2: save humidity
sprintf(_bufferHumid, "%02u.%02u %%", (int)humi, (int)(humi * 100) % 100 );

// 3: save pressure
sprintf(_bufferPress, "%03u.%02u hPa", (int)(pres/100), (int)((uint32_t)pres % 100) );

int uvLevel = averageAnalogRead(UVOUT);
int refLevel = averageAnalogRead(REF_3V3);

//Use the 3.3V power pin as a reference to get a very accurate output value from sensor
float outputVoltage = 3.3 / refLevel * uvLevel;

/* read UV intensity */
float uvIntensity = mapfloat(outputVoltage, 0.99, 2.9, 0.0, 15.0);

/* read brightness intensity from pin A3 */
int v = analogRead(LDR_BRIGHTNESS);

/* read the humidity of soil */
int humidity_soil=analogRead(HUMIDITY_SOIL);

humidity_soil=map(humidity_soil,1023,0,0,100);

/* get rain moisture */
int rain_value = analogRead(RAIN_ANALOG_IN);
rain_value=map(rain_value,1023,0,0,100);
isRaining = !(digitalRead(RAIN_DIGITAL_IN));

/* set a string variable to see if it rains or not */
if(isRaining){
    strRaining = "YES";
}
else{
    strRaining = "NO";
}

/* set the message which will be shown on the display */
String message=(String)"TEMPERATURE: "+_bufferTemp+", "+(String)"HUMIDITY:
"+_bufferHumid+", "+(String)"PRESSURE: "+_bufferPress+", "+" UV Intensity (mW/cm^2):
"+uvIntensity+", "+"Brightness intensity: "+v+", "+(String)"Soil humidity:
"+humidity_soil+"%", "+ "Rain moisture: "+rain_value+"%", "+", Rains? => "+strRaining+". ";

/* algorithm for the text to be scrolled from right to left on the display */
int length_message=message.length();

/* // 10 = 5 pixels/character * text size 2 */
minX=-10*length_message;

```

```
display.setCursor(x,18);
```

```
/* print the message on display*/  
display.print(message);
```

```
display.display();
```

```
if(--x < minX) x = display.width();
```

```
delay(20); // wait a bit
```