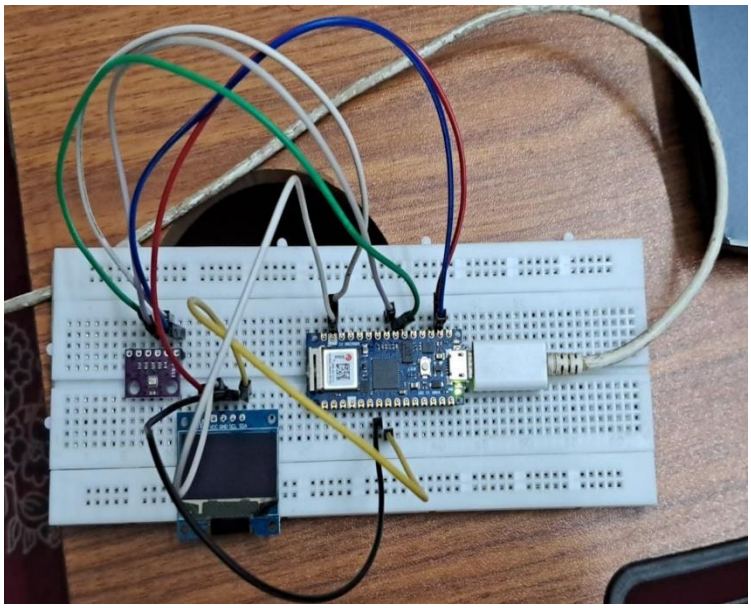
Tiny Weather Monitoring Station with IoT and Android interface (Smart City)

By A.S.Poornash (2101CS01)

Anuj Sharma (2101CS11)

Deepanker Jauhari (2101CS23)



OVERVIEW-

We have made a project that can measure real-time humidity, pressure, and temperature and have built an android app with a user-friendly interface that outputs the values.

We are also using a cloud service called Blynk in order to store and retrieve data into our android interface.

Now let's see the components involved in this project-

COMPONENTS USED -

1. Arduino Nano 33 IoT
2. BME 280 Weather Sensor
3. 0.96 in OLED display – 128 x 64
4. Micro USB cable
5. Male-to Female breadboard jumper wires
6. USB Wall chargers
7. Arduino IDE
8. Blynk Software
9. Breadboard
10. Power Bank

IMPLEMENTATION and REQUIREMENTS–

We use the Arduino Nano 33 IoT which handles WiFi, a BME280 sensor board with temperature, humidity, and barometric pressure all in one unit, and a 1-inch OLED display. All three are run at 3.3 volts.

Both the sensor and the display interface with I2C. One I2C can support multiple devices and hence by importing simply importing the project we can use this project across different devices.

Power is supplied through the USB cable. That way, it can be programmed or powered by a single cable in a simple manner. That cable can go to the computer when programming or to a plug-in USB power supply when being used as a weather station.

The software running on the Arduino is a complicated mixture of use of the library examples for the sensor, the display, and Blynk. The libraries involved are the Adafruit_BME280 Library for the sensor, the ss_oled library for the display, and the Blynk library for Blynk.

All three are to be downloaded directly from Arduino's library manager. We also have added the library: #include <avr/dtostrf.h>as this was necessary to convert the numbers coming out of the sensor into strings suitable to display on the OLED display.

After uploading the code, we see that the Pressure, Temperature, and Humidity values are seen in both the Serial Monitor and, in the Android Interface, powered by Blynk software. The Wi-Fi-enabled IoT service helps the Arduino Nano 33 board to connect to the internet and send the values to the app in a Realtime manner

PIN NUMBERS USED AND CONNECTIONS–

- VIN pin of BME and the Vcc pin of OLED are connected to the REF 3.3V pin of Arduino nano
- GND pin of the BME and OLED display is connected to the GND pin of the Arduino nano
- SCL pin of BME is connected to the A5 pin of Arduino nano
- SDA pin of BME is connected to the A4 pin of Arduino nano
- SCL pin of the OLED display is connected to the D4 pin of the Arduino nano

- SDA pin of the OLED display is connected to D5 pin of Arduino nano

CODE-

```
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_BME280.h>
#include <avr/dtostrf.h>
#include <ss_oled.h>
#include <WiFiNINA.h>
#include <BlynkSimpleWiFiNINA.h>


char auth[] = "wQpvBA1EyKYuHJ4FPLinJQOVy9K-xURA";
char ssid[] = "DEEJAY-PREDATOR";
char pass[] = "26122003";

Adafruit_BME280 bme; // use I2C interface
Adafruit_Sensor *bme_temp = bme.getTemperatureSensor();
Adafruit_Sensor *bme_pressure = bme.getPressureSensor();
Adafruit_Sensor *bme_humidity = bme.getHumiditySensor();

#define SDA_PIN 5
#define SCL_PIN 4
#define RESET_PIN -1
#define OLED_ADDR -1
#define FLIP180 0
#define INVERT 0
#define USE_HW_I2C 0
#define MY_OLED OLED_128x64
#define OLED_WIDTH 128
#define OLED_HEIGHT 64

SSOLED ssoled;
BlynkTimer timer;


void setup() {
  Serial.println("anything0");
```

```
  if (!bme.begin()) {
    while (1) delay(10);
    Serial.println(F("Could not find a valid BME280 sensor, check wiring or "
                     "try a different address!"));
  }

  bme_temp->printSensorDetails();
  bme_pressure->printSensorDetails();
  bme_humidity->printSensorDetails();

  oledInit(&ssoled, MY_OLED, OLED_ADDR, FLIP180, INVERT,
USE_HW_I2C, SDA_PIN, SCL_PIN, RESET_PIN, 400000L);
  oledFill(&ssoled, 0x0, 1);

  Blynk.begin(auth, ssid, pass);
  timer.setInterval(5000L, myupdate);  // runs every 5 seconds
  myupdate();
}

void loop() {
   Blynk.run();
   timer.run();
}

void myupdate(){
  sensors_event_t temp_event, pressure_event, humidity_event;
  bme_temp->getEvent(&temp_event);
  bme_pressure->getEvent(&pressure_event);
  bme_humidity->getEvent(&humidity_event);

  float ctemp = temp_event.temperature;  // get temp in centigrade
  float ftemp = 32 + (9*ctemp)/5;  // convert temp to fahrenheit
  float hum = humidity_event.relative_humidity;  // get relative humidity
  float mpress = pressure_event.pressure; // get pressure in mm

  char mytemp[8];
  dtostrf(ftemp, 6, 2, mytemp); // convert temp to a string
  char myhum[8];
  dtostrf(hum, 6, 2, myhum); // convert humidity to a string
  char mypress[8];
  dtostrf(mpress, 6, 2, mypress); // convert pressure to a string

  // update OLED display
```

```
  oledWriteString(&ssoled, 0,2,1,(char *)"Weather Station", FONT_NORMAL,
0, 1);
  oledWriteString(&ssoled, 0,2,3,(char *)"Temp = ", FONT_NORMAL, 0, 1);
  oledWriteString(&ssoled, 0,50,3,(char *)mytemp, FONT_NORMAL, 0, 1);
  oledWriteString(&ssoled, 0,105,3,(char *)"F", FONT_NORMAL, 0, 1);
  oledWriteString(&ssoled, 0,2,5,(char *)"Humid=", FONT_NORMAL, 0, 1);
  oledWriteString(&ssoled, 0,50,5,(char *)myhum, FONT_NORMAL, 0, 1);
  oledWriteString(&ssoled, 0,105,5,(char *)"%", FONT_NORMAL, 0, 1);
  oledWriteString(&ssoled, 0,2,7,(char *)"Press=", FONT_NORMAL, 0, 1);
  oledWriteString(&ssoled, 0,50,7,(char *)mypress, FONT_NORMAL, 0, 1);
  oledWriteString(&ssoled, 0,105,7,(char *)"in", FONT_NORMAL, 0, 1);

  // push new weather data to Blynk
  Blynk.virtualWrite(V3, ftemp);
  Blynk.virtualWrite(V4, hum);
  Blynk.virtualWrite(V5, mpress);
}
```

CONTRIBUTIONS-

- **Selecting the Project and buying components –** A.S.Poornash, Deepanker Jauhari
- **Hardware configuring –** Anuj Sharma, A.S.Poornash
- **Writing the code and integrating it with blynk and Arduino –** Deepanker Jauhari, Anuj Sharma
- **Report, PPT and Video –** A.S.Poornash, Anuj Sharma , Deepanker Jauhari