

EEP LAB PROJECT FILE

BATCH 'B'

GROUP NUMBER - 4

GROUP MEMBERS:

- | | |
|------------------------|-------------|
| 1. SHREYAS BHATLAWANDE | (201060062) |
| 2. UDIT KUMAR DHANKAR | (201060063) |
| 3. ATHARVA BENDRE | (201060064) |
| 4. HIMANSHU KOHALE | (201060018) |

AIM: weather monitoring by Arduino-uno

COMPONENTS:

1. Microcontroller unit (MCU) / Arduino-Uno
2. Sensors
 - (a) DHT11
 - (b) BMP180
3. Wires
4. Breadboard

THEORY:

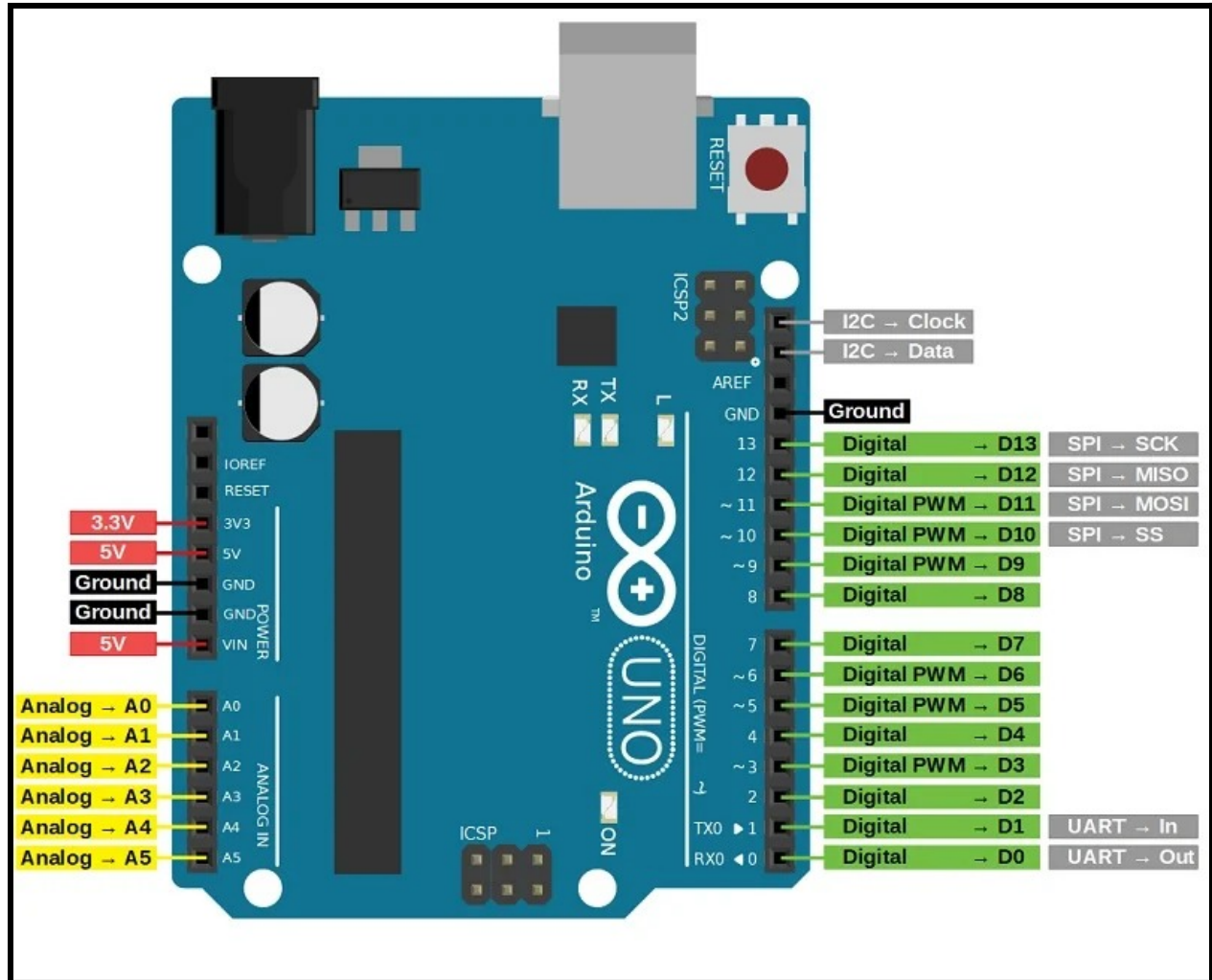
BASIC:

In this project, we've made a device that monitors the parameters of the environment like temperature, pressure, humidity and altitude.

We are using the "DHT11 sensor" which is easily capable of measuring the temperature and humidity, and the "BMP180" sensor is used for measuring the altitude and pressure, also we are using the arduino uno there which helps in displaying the values in the serial monitor.

DHT11 and BMP180 sensors are connected to the Arduino-uno.

ARDUINO-UNO:



Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, and a 16 MHz ceramic resonator.

The board can operate on an external supply from 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the

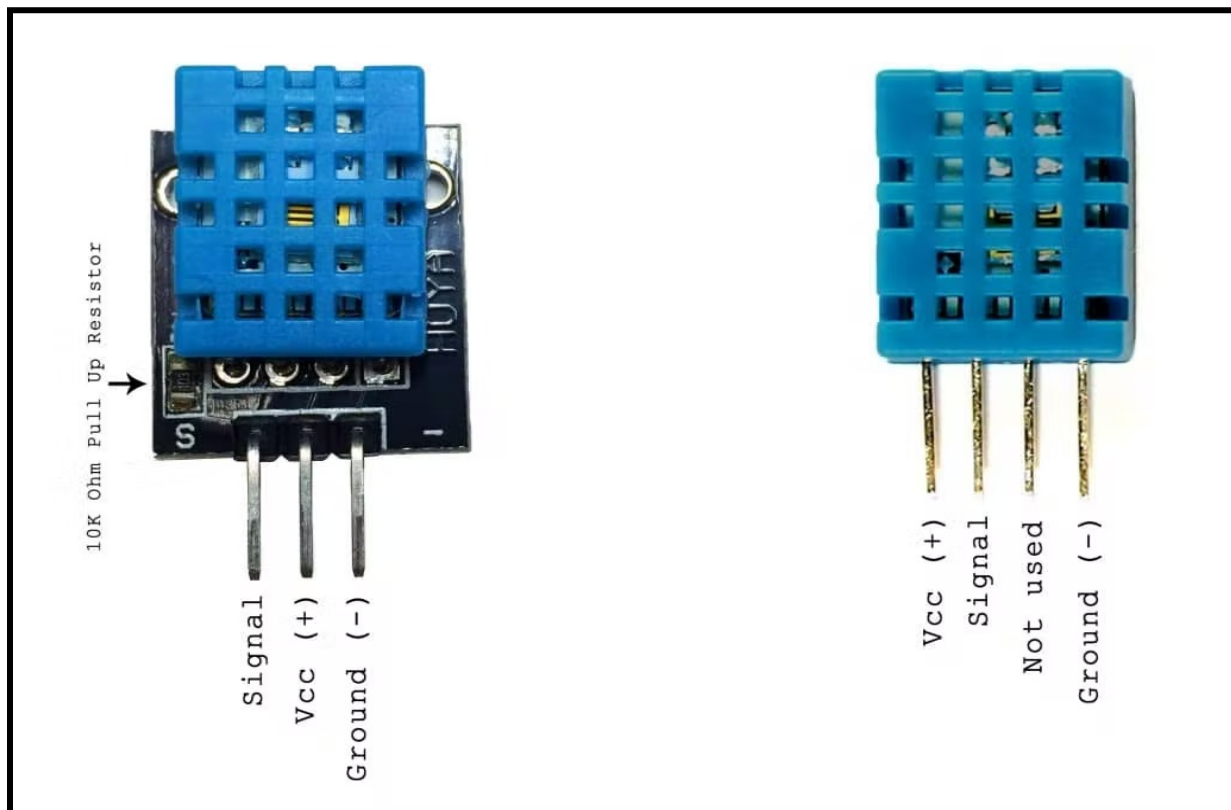
voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The ATmega328 has 32 KB (with 0.5 KB occupied by the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM

The ATmega328 also supports I2C (TWI) and SPI communication.

The ATmega328 provides UART (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX).

THE DHT11:



The DHT11 sensor in the weather **monitoring system** measures the temperature and humidity by the inbuilt sensors.

There are two sensors, one of them measures the temperature and the other one always measures the humidity.

There is an IC too which sends the data in the multiplex form and we always connect this DHT11 sensor with the Arduino and Arduino gets this information.

After getting the information the Arduino works on the data. according to the code and inbuilt sensor library. To make this weather monitoring system Arduino calibrates the sensor by using the coding which we will upload to the Arduino.

Humidity calculation formula by DHT11 :

$$RH = \left(\frac{\rho_w}{\rho_s} \right) \times 100\%$$

RH : Relative Humidity

ρ_w : Density of water vapor

ρ_s : Density of water vapor at saturation

DHT11 Specifications

1. Operating Voltage: 3.5V to 5.5V
2. Operating current: 0.3mA (measuring) 60uA (standby)
3. Output: Serial data

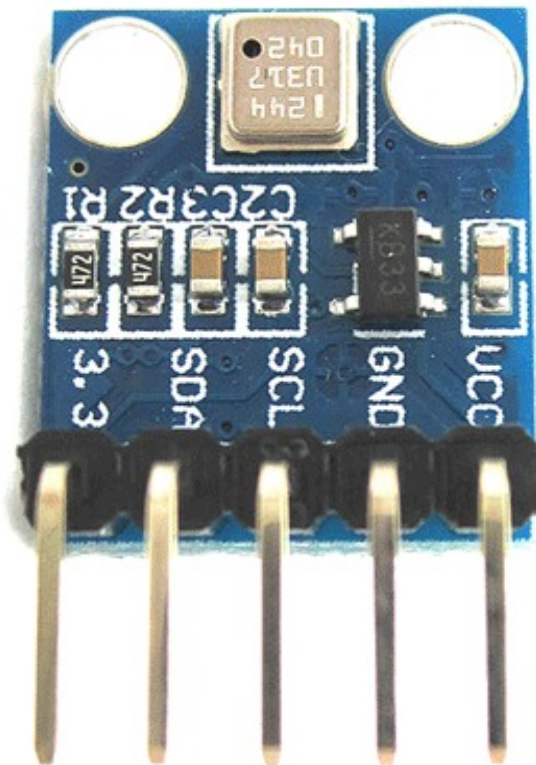
4. Temperature Range: 0°C to 50°C

5. Humidity Range: 20% to 90%

6. Resolution: Temperature and Humidity both are 16-bit

7. Accuracy: $\pm 1^\circ\text{C}$ and $\pm 1\%$

BMP180:



The BMP180 is the function compatible successor of the BMP085, a new generation of high precision digital pressure sensors for consumer applications.

The BMP180 is designed to be connected directly to a microcontroller of a mobile device via the I2C bus.

I2C stands for the inter-integrated controller. This is a serial communication protocol that can connect low-speed devices. It is a master-slave communication in which we can connect and control multiple slaves from a single master. In this, each slave device has a specific address.

The pressure and temperature data has to be compensated by the calibration data of the E2PROM of the BMP180.

With the measured pressure p and the pressure at sea level p_0 e.g. 1013.25hPa, **the altitude** in meters can be calculated with the international barometric formula:

$$\text{altitude} = 44330 * \left(1 - \left(\frac{p}{p_0} \right)^{\frac{1}{5.255}} \right)$$

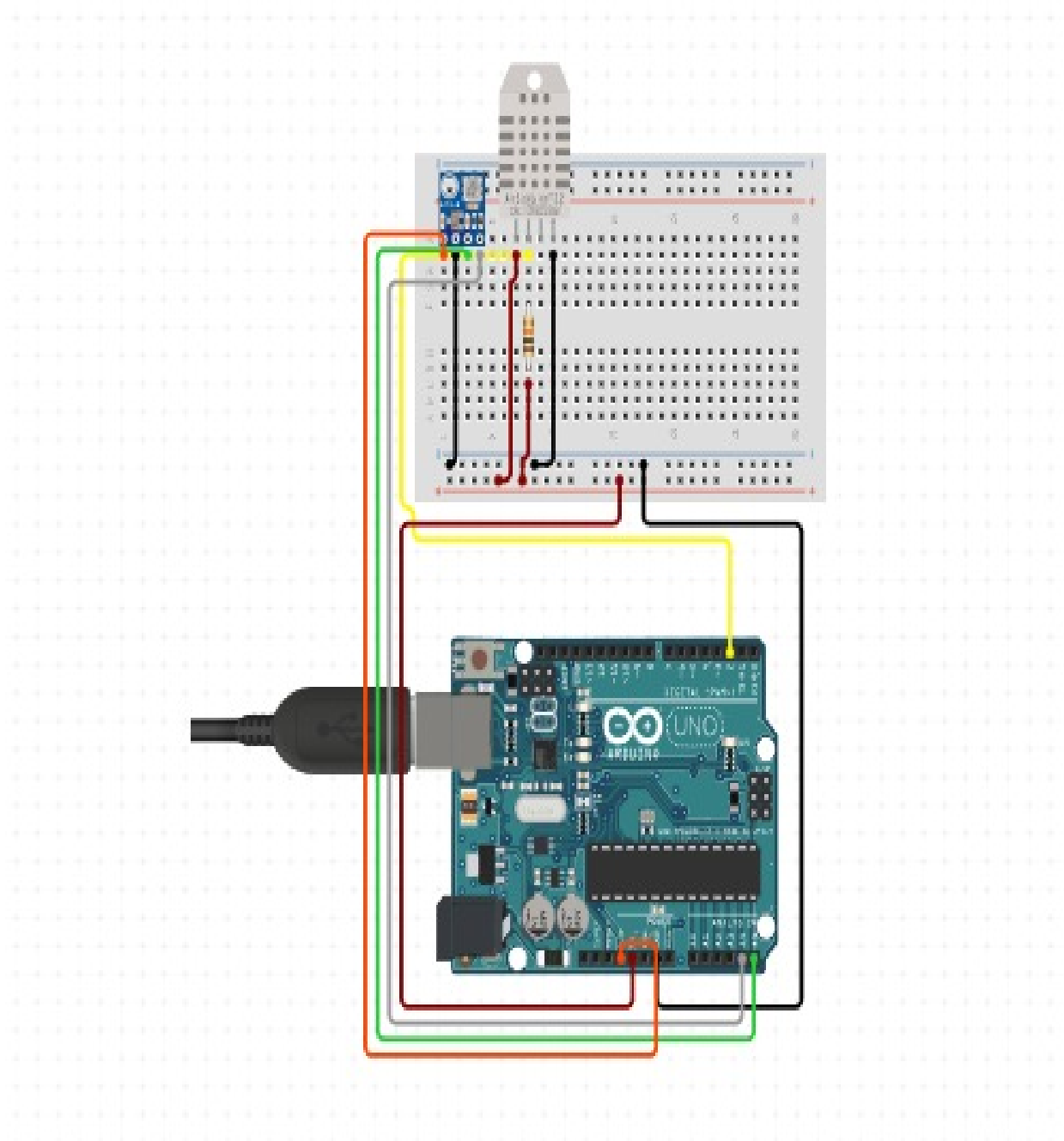
With the measured pressure p and the absolute altitude, **the pressure at sea level can be calculated:**

$$p_0 = \frac{p}{\left(1 - \frac{\text{altitude}}{44330} \right)^{5.255}}$$

BMP180 SPECIFICATIONS:

- 1.operating voltage of BMP180: 1.3V – 3.6V
- 2.Input voltage of BMP180MODULE: 3.3V to 5.5V
- 3.Peak current : 1000uA
- 4.Consumes 0.1uA standby
- 5.Maximum voltage at SDA , SCL : $VCC + 0.3V$
- 6.Operating temperature: $-40^{\circ}C$ to $+80^{\circ}C$

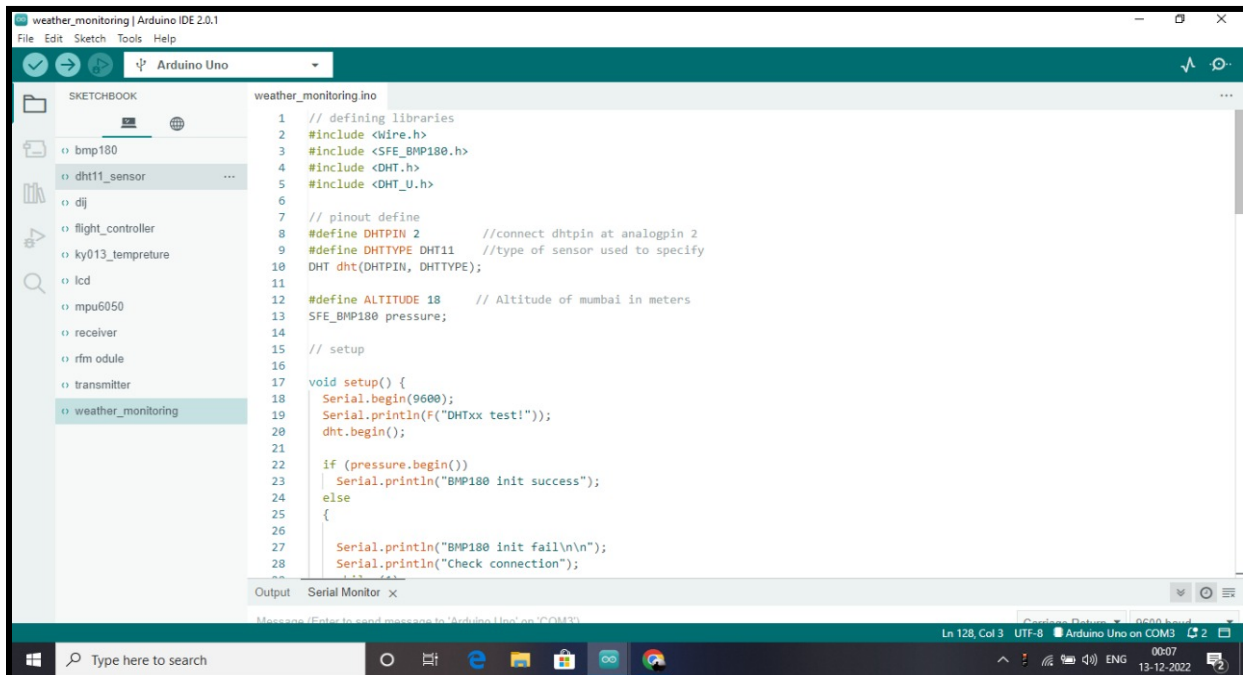
CIRCUIT DIAGRAM OF THE PROJECT:



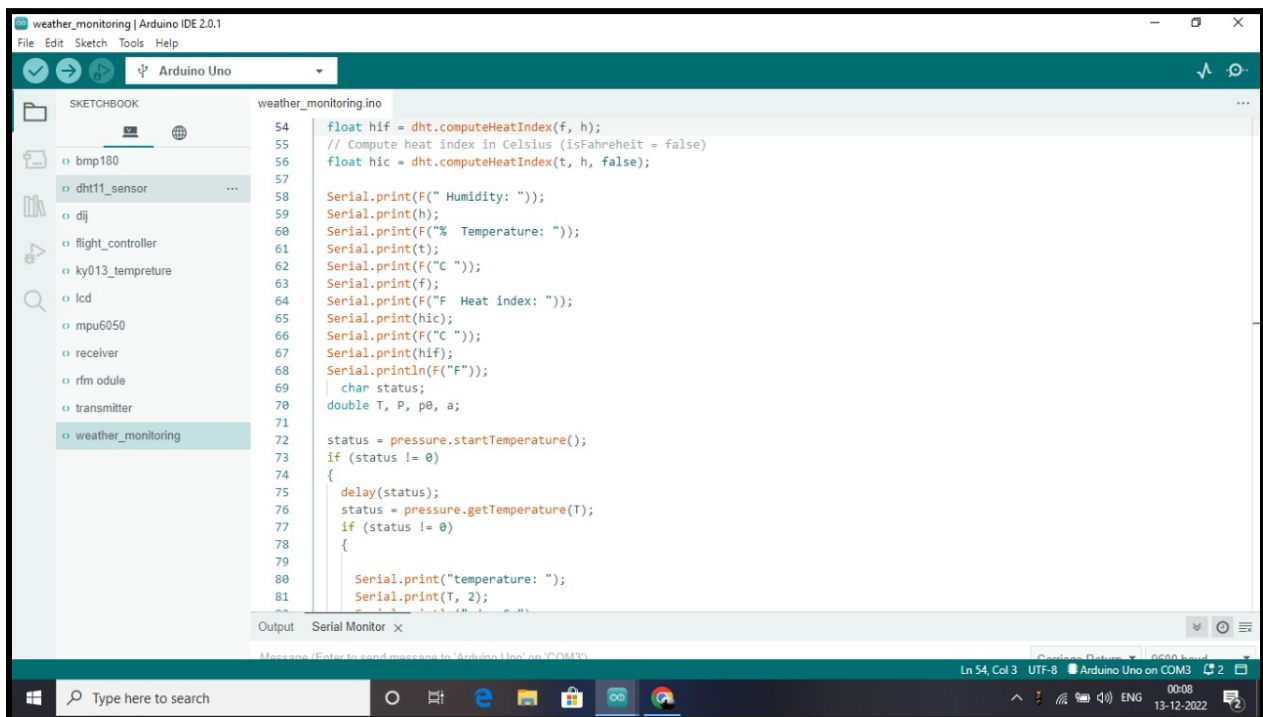
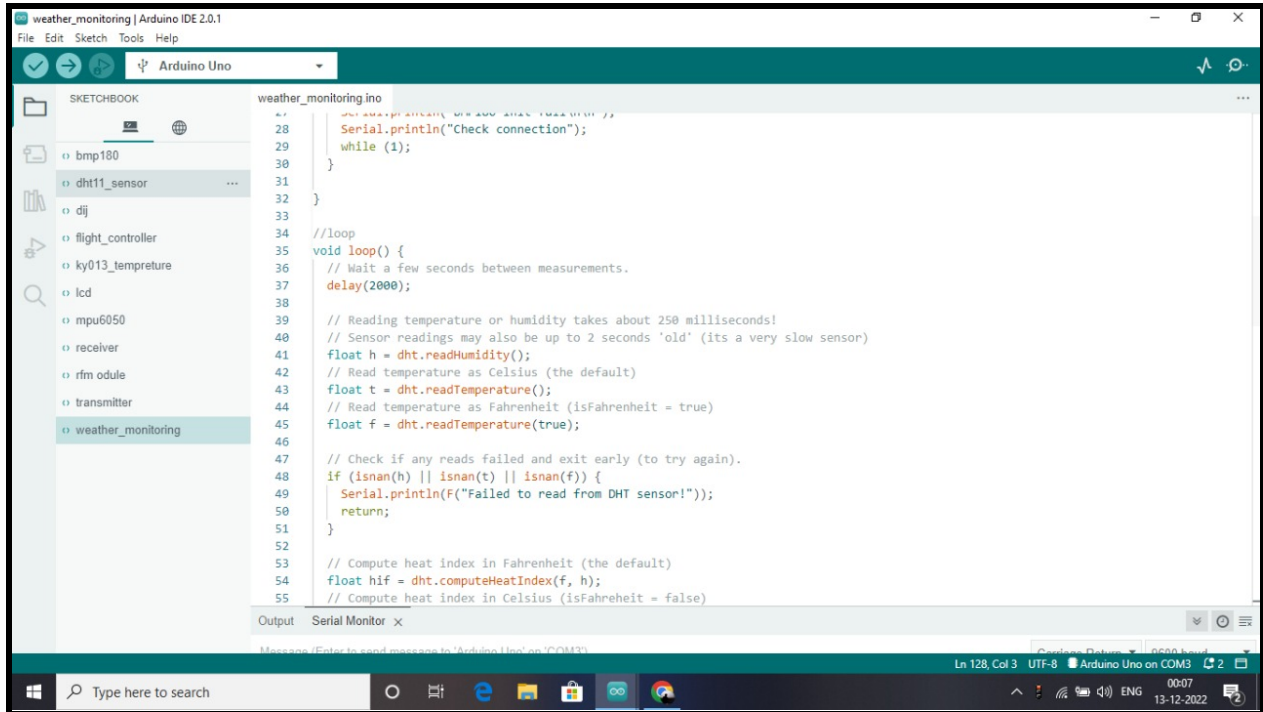
WORKING:

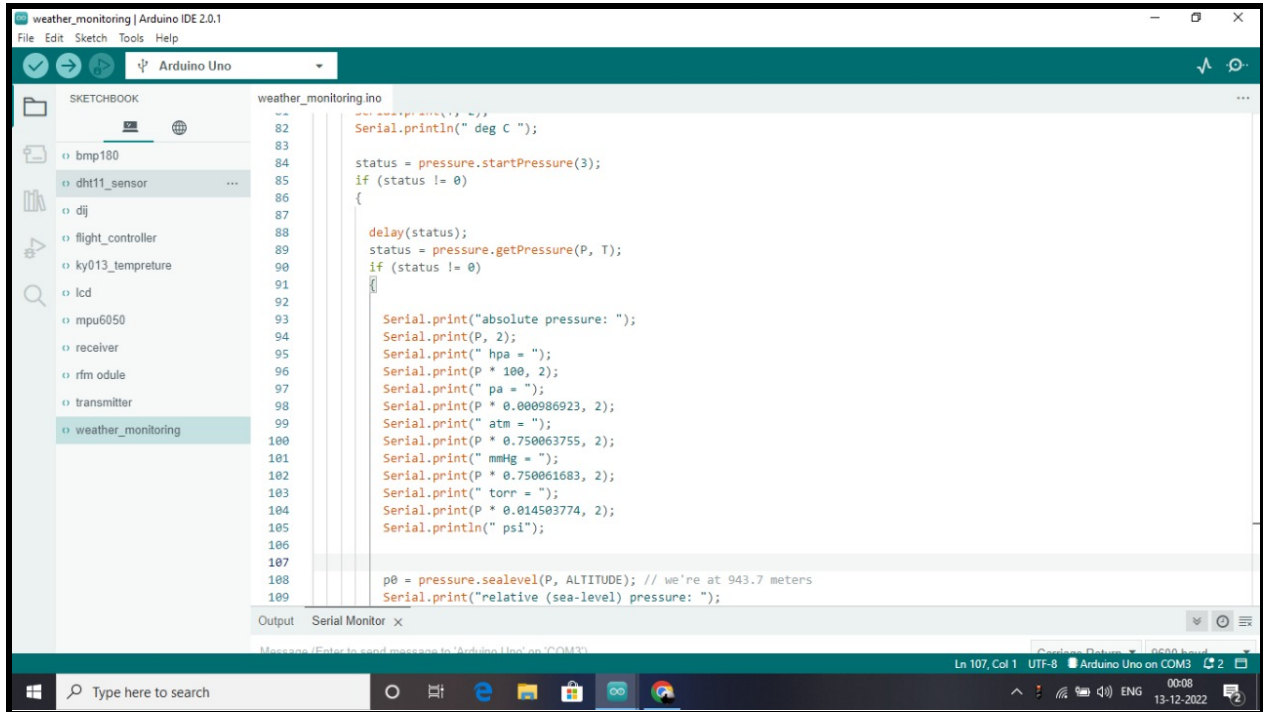
1. In the above diagram, DHT11 and BMP180 sensors are connected to arduino-uno
2. **DHT11** has 3 pins [power,ground and data pin] . Data pin is connected to digital pin number 2 of arduino-uno. Power pin is connected to 5v , ground is connected to the ground pin of arduino-uno
3. **BMP180** runs on I2C protocols. I2C is a serial communication protocol that has 2 lines i.e SDA and SCL. SCL works as a clock, SDA line is connected to BMP180 sensor. Power pin and ground pins of BMP180 are connected to 5v and ground pin of arduino-uno
4. After hardware connections, we implement the code in the arduino IDE .
5. After flashing the code in the arduino IDE, we get the required values of all weather parameters in the **serial monitor** of the arduino IDE.

CODE:

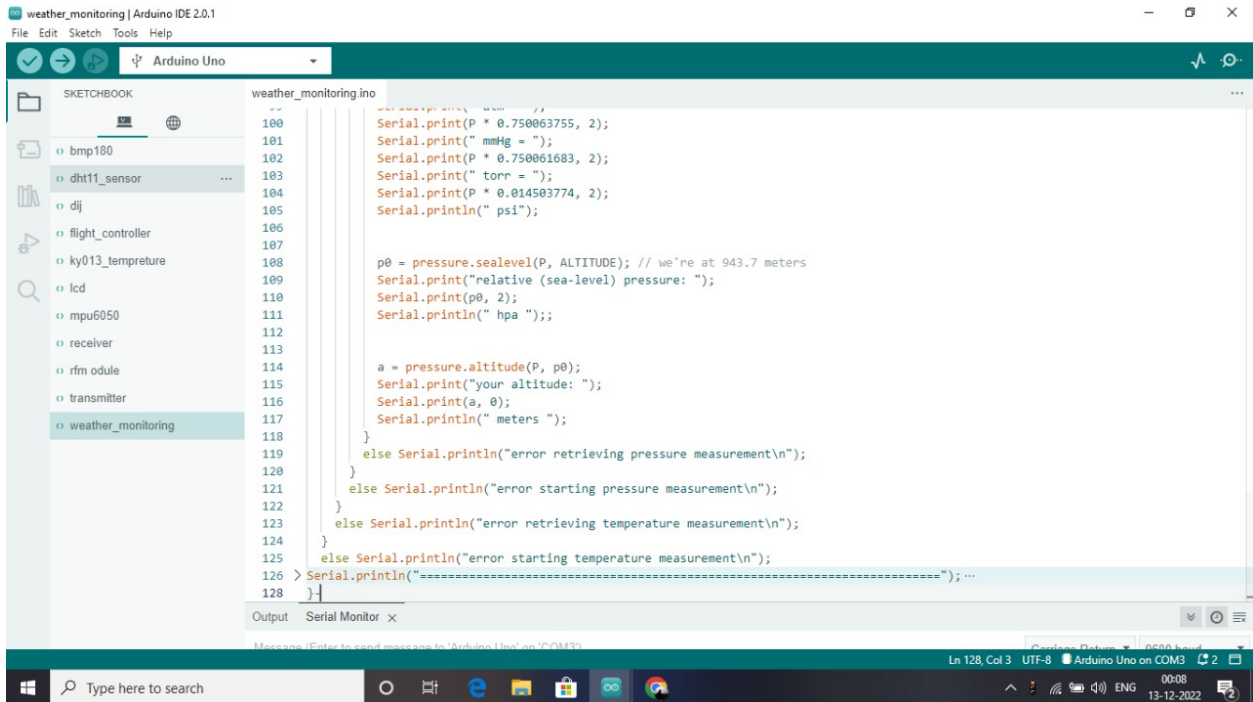


```
1 // defining libraries
2 #include <Wire.h>
3 #include <SFE_BMP180.h>
4 #include <DHT.h>
5 #include <DHT_U.h>
6
7 // pinout define
8 #define DHTPIN 2 //connect dhtpin at analogpin 2
9 #define DHTTYPE DHT11 //type of sensor used to specify
10 DHT dht(DHTPIN, DHTTYPE);
11
12 #define ALTITUDE 18 // Altitude of mumbai in meters
13 SFE_BMP180 pressure;
14
15 // setup
16
17 void setup() {
18   Serial.begin(9600);
19   Serial.println(F("DHTxx test!"));
20   dht.begin();
21
22   if (pressure.begin())
23     Serial.println("BMP180 init success");
24   else
25   {
26     Serial.println("BMP180 init fail\n\n");
27     Serial.println("Check connection");
28   }
29 }
```



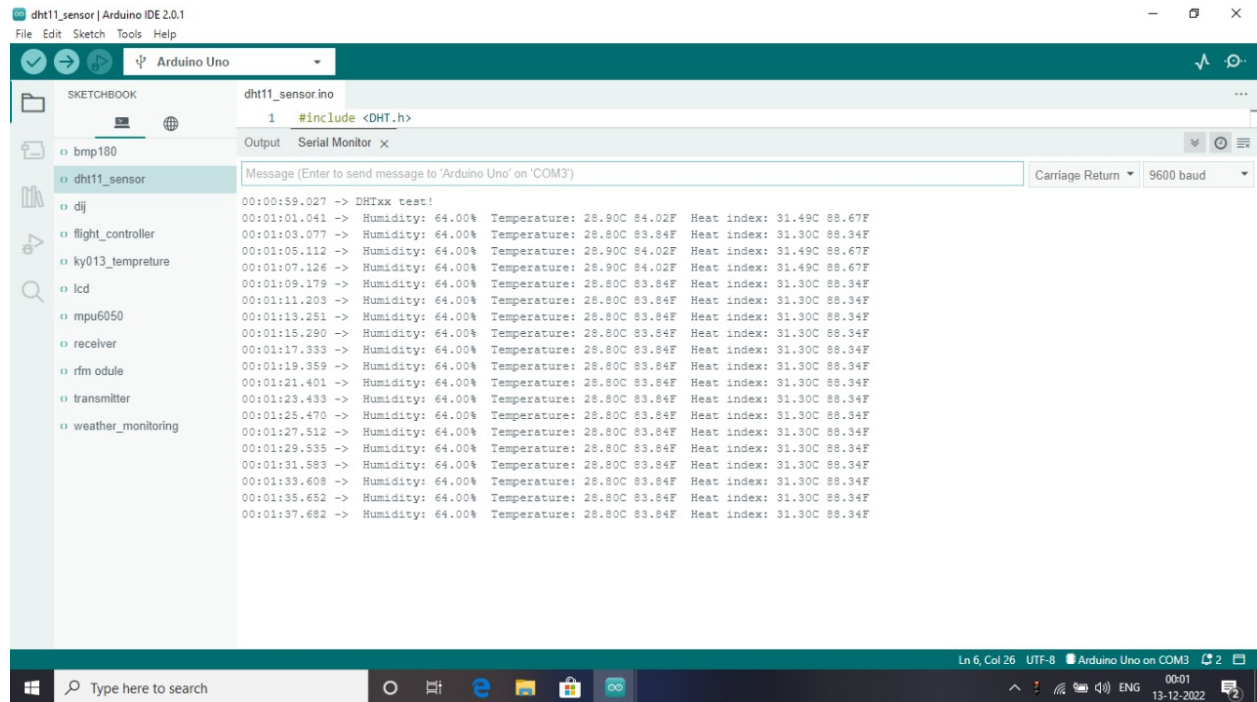


```
weather_monitoring.ino
82 // Serial.print("\n deg C ");
83
84
85 status = pressure.startPressure(3);
86 if (status != 0)
87 {
88     delay(status);
89     status = pressure.getPressure(P, T);
90     if (status != 0)
91     {
92
93         Serial.print("absolute pressure: ");
94         Serial.print(P, 2);
95         Serial.print(" hpa = ");
96         Serial.print(P * 100, 2);
97         Serial.print(" pa = ");
98         Serial.print(P * 0.000986923, 2);
99         Serial.print(" atm = ");
100        Serial.print(P * 0.750063755, 2);
101        Serial.print(" mmHg = ");
102        Serial.print(P * 0.750061683, 2);
103        Serial.print(" torr = ");
104        Serial.print(P * 0.014503774, 2);
105        Serial.println(" psi");
106
107
108        p0 = pressure.sealevel(P, ALTITUDE); // we're at 943.7 meters
109        Serial.print("relative (sea-level) pressure: ");
```



```
100 Serial.print(P * 0.750063755, 2);
101 Serial.print(" mmHg = ");
102 Serial.print(P * 0.750061683, 2);
103 Serial.print(" torr = ");
104 Serial.print(P * 0.014503774, 2);
105 Serial.println(" psi");
106
107
108 p0 = pressure.sealevel(P, ALTITUDE); // we're at 943.7 meters
109 Serial.print("relative (sea-level) pressure: ");
110 Serial.print(p0, 2);
111 Serial.println(" hpa ");
112
113
114 a = pressure.altitude(P, p0);
115 Serial.print("your altitude: ");
116 Serial.print(a, 0);
117 Serial.println(" meters ");
118 }
119 else Serial.println("error retrieving pressure measurement\n");
120 }
121 else Serial.println("error starting pressure measurement\n");
122 }
123 else Serial.println("error retrieving temperature measurement\n");
124 }
125 else Serial.println("error starting temperature measurement\n");
126 > Serial.println("=====");
127 }
128 }
```

READINGS:



CONCLUSION:

We made a weather monitoring system using arduino-uno and with the help of sensors like BMP180 and DHT11 and produced the readings of the required weather parameters.

