

# WEATHER STATION USING ARDUINO

## Code:

```
#include <LiquidCrystal_I2C.h>

#include <SoftwareSerial.h>

#include <dht.h>

#include <Wire.h>

#include <BMP180.h>

dht DHT;

LiquidCrystal_I2C lcd(0x27, 16, 2);

SoftwareSerial mySerial(10, 11);

BMP180 myBMP(BMP180_ULTRAHIGHRES);

#define DHT11_PIN A0

#define mq135_pin A2

#define LDR A1

void ReadDHT(void);

void ReadBMP(void);

void ReadAir(void);

void send_data(void);

bool BMP_flag = 0;

bool DHT_flag = 0;


void setup()

{
```

```
mySerial.begin(115200);  
pinMode(mq135_pin, INPUT);  
pinMode(LDR, INPUT);  
lcd.init();  
lcd.backlight();  
lcd.setCursor(0, 0);  
lcd.print(" IoT Weather ");  
lcd.setCursor(0, 1);  
lcd.print("Monitor System");  
delay(1500);  
}
```

```
void loop()  
{  
  ReadDHT();  
  ReadBMP();  
  ReadAir();  
  Readlight();  
  send_data();  
}
```

```
void ReadDHT(void)  
{  
  lcd.clear();
```

```
int chk = DHT.read11(DHT11_PIN);

switch (chk)
{
    case DHTLIB_OK:
        DHT_flag = 1;
        lcd.setCursor(0, 0);
        lcd.print("Temp: ");
        lcd.print(DHT.temperature, 1);
        lcd.print(" *C");
        lcd.setCursor(0, 1);
        lcd.print("Humi: ");
        lcd.print(DHT.humidity, 1);
        lcd.print(" %");
        break;

    case DHTLIB_ERROR_CONNECT:
        lcd.setCursor(0, 0);
        lcd.print("NO DHT11 SENSOR");
        lcd.setCursor(0, 1);
        lcd.print("  FOUND!  ");
        break;

    case DHTLIB_ERROR_CHECKSUM:
    case DHTLIB_ERROR_TIMEOUT:
    case DHTLIB_ERROR_ACK_L:
    case DHTLIB_ERROR_ACK_H:
```

default:

```
DHT_flag = 0;

lcd.setCursor(0, 0);

lcd.print(" DHT11 SENSOR ");

lcd.setCursor(0, 1);

lcd.print("  ERROR  ");

break;

}

delay(2000);

}
```

void ReadBMP(void)

```
{

  lcd.clear();

  if (myBMP.begin() != true)

  {

    lcd.setCursor(0, 0);

    lcd.print(" BMP180 SENSOR ");

    lcd.setCursor(0, 1);

    lcd.print("  NOT FOUND  ");

    BMP_flag = 0;

    delay(2000);

  }

  else
```

```

{
  BMP_flag = 1;
  lcd.setCursor(0, 0);
  lcd.print("Pa(Grnd):");
  lcd.print(myBMP.getPressure());
  lcd.setCursor(0, 1);
  lcd.print("Pa(sea) :");
  lcd.print(myBMP.getSeaLevelPressure(115));
}
delay(2000);
}

```

```

void ReadAir(void)
{
  int airqlty = 0;
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("AIR QUALITY:");
  airqlty = analogRead(mq135_pin);
  lcd.print(map(analogRead(mq135_pin), 0, 1024, 99, 0));
  lcd.print("%");
  lcd.setCursor(0, 1);
  if (airqlty <= 180)
    lcd.print("GOOD!");
}

```

```
else if (airqlty > 180 && airqlty <= 225)
    lcd.print("POOR");
else if (airqlty > 225 && airqlty <= 300)
    lcd.print("VERY BAD");
else
    lcd.print("TOXIC");
delay(2000);
}
```

```
void Readlight(void)
{
    lcd.clear();
    lcd.setCursor(3, 0);
    lcd.print("LIGHT :");
    lcd.print(map(analogRead(LDR), 0, 1024, 0, 99));
    lcd.print("%");
    lcd.setCursor(0, 1);
    lcd.print("*****");
    delay(2000);
}
```

```
void send_data()
{
    mySerial.print('*'); // Starting char
```

```
if (DHT_flag == 1)
{
    mySerial.print(DHT.temperature, 0); //2 digit data
    mySerial.print(DHT.humidity, 0); //2 digit data
}
else
{
    mySerial.print("0000"); // Send dummy data
}
if (BMP_flag == 1)
{
    mySerial.print(myBMP.getPressure()); //5 digit data
}
else
{
    mySerial.print("00000");// Send dummy data
}
mySerial.print(map(analogRead(LDR), 0, 1024, 0, 99)); //2 digit data
mySerial.print(map(analogRead(mq135_pin), 0, 1024, 99, 0)); //2 digit data
mySerial.println('#'); // Ending char
}
```

**Image:**

