

PROJECT

Mini weather station using ThingSpeak

PROJECT REPORT

Submitted by

KURUBA BULLEY BHARADWAJ

Mail ID:

kbulleybharadwaj_eee180223@mgit.ac.in

**ELECTRICAL AND ELECTRONICS
ENGINEERING**

In

**MAHATMA GANDHI INSTITUTE OF
TECHNOLOGY-HYDERABAD**

TITLE ; Mini weather station using ThingSpeak

ELECTRONIC COMPONENTS

- Arduino UNO
- ESP8266
- Arduino proto shield
- DHT22 Temperature Sensor
- Seeed Grove – Light Sensor
- Seeed Grove – UV Sensor
- Seeed Grove – Barometer Sensor
- Seeed Grove – Dust Sensor
- Resistors(1k,10k,4.75k)ohms
- Jumper Wires

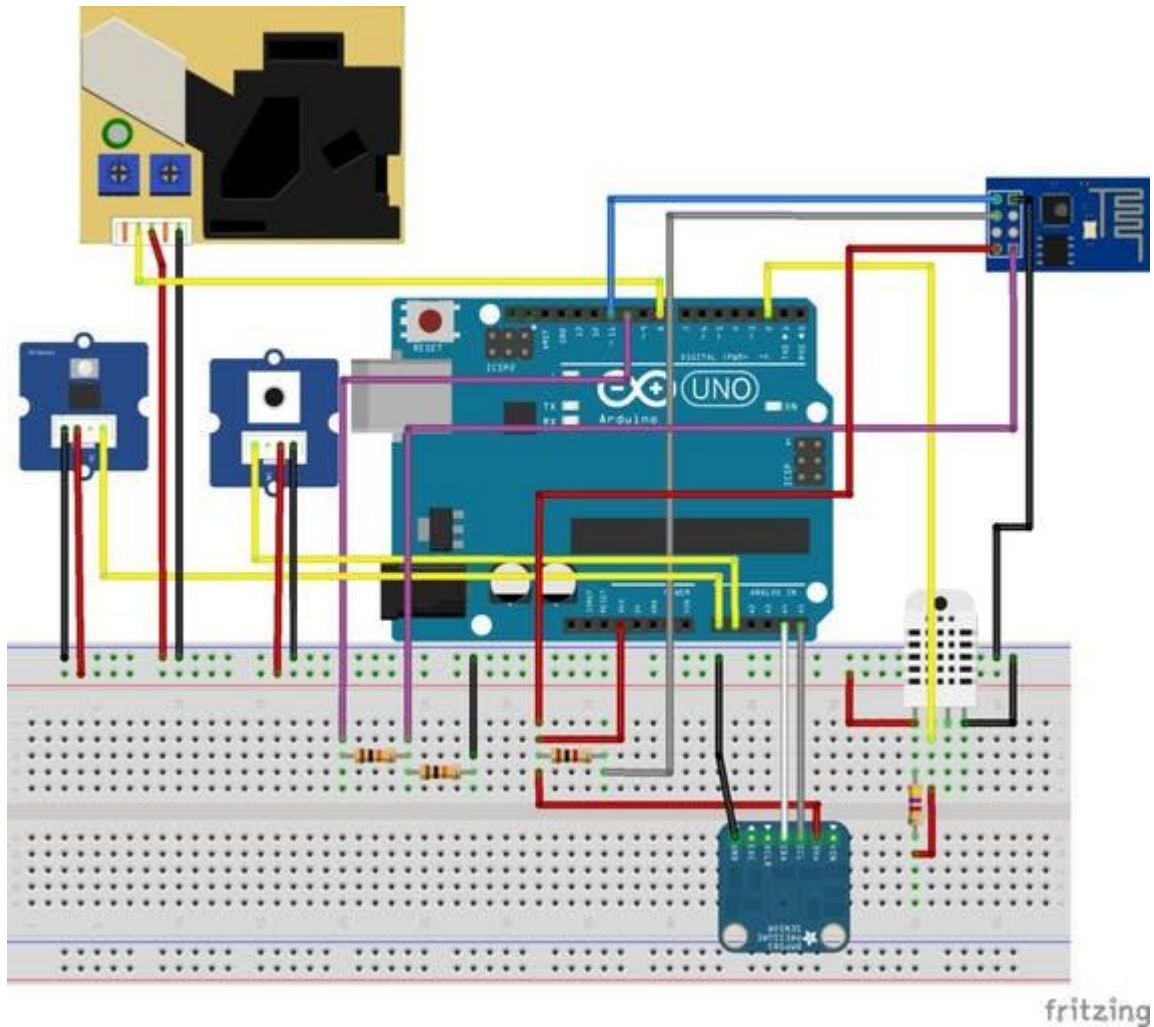
Services used

- Arduino IDE
- ThingSpeak

The weather station collects the following data related to the weather and environment using different sensors:

- Temperature
- Humidity
- Atmospheric pressure
- Light intensity
- UV index
- Dust concentration

CIRCUIT DIAGRAM



CODE

```
#include <stdlib.h>
#include <SoftwareSerial.h>
#include <DHT.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP085_U.h>
#define SSID "Bharadwaj"
#define PASS "18261A0223"
```

```
#define IP "184.106.153.149" // thingspeak.com IP
```

```
#define DHT22_PIN 2
```

```
String GET = "GET /update?key= WS96FCCUUM5RD2U8&field1=";
```

```
SoftwareSerial monitor(10, 11); //Serial communication to ESP8266  
module (RX, TX)
```

```
dht DHT;
```

```
Adafruit_BMP085_Unified bmp = Adafruit_BMP085_Unified(10085);
```

```
//Variables
```

```
int luminancePin = A0;
```

```
int uvPin = A1;
```

```
int dustPin = 8;
```

```
unsigned long duration;
```

```
unsigned long starttime;
```

```
unsigned long sampletime_ms = 30000;
```

```
unsigned long delay_time = 60000;
```

```
unsigned long lowpulseoccupancy = 0;
```

```
float ratio = 0;
```

```
float concentration = 0;
```

```
//setup
```

```
void setup()
```

```
{
```

```
  //start serial communications
```

```
  Serial.begin(9600);
```

```
  monitor.begin(9600);
```

```
  Serial.println("Initializing...");
```

```
  //configure Arduino pins
```

```
  pinMode(dustPin, INPUT);
```

```
  //initialize pressure sensor
```

```
  Serial.println("Detecting BMP085 pressure sensor...");
```

```
  if(!bmp.begin())
```

```
  {
```

```
    Serial.println("BMP085 sensor wasn't detected. Verify your  
connections or I2C ADDR!");
```

```
    while(1);
```

```
  }
```

```
  Serial.println("BMP085 detected!");
```

```
  //communication with wifi module
```

```
  monitor.flush();
```

```
  monitor.println("AT");
```

```

    delay(2000);

    if(monitor.find("OK")){
        Serial.println("Communication with ESP8266 module: OK");
    }
    else {
        Serial.println("ESP8266 module ERROR");
    }

    //connect wifi router
    connectWiFi();

    Serial.print("Sampling (");
    Serial.print(sampletime_ms/1000);
    Serial.println("s)...");

    //initialize timer
    starttime = millis();
}

void loop(){

    //measuring dust particles
    duration = pulseIn(dustPin, LOW);
    lowpulseoccupancy = lowpulseoccupancy + duration;

    //30 seconds cicle
    if ((millis() - starttime) >= sampletime_ms)
    {
        ratio = lowpulseoccupancy/(sampletime_ms*10.0); // percentage
        (de 0 a 100%)
        concentration = 1.1*pow(ratio,3)-
3.8*pow(ratio,2)+520*ratio+0.62; // from datsheet
        lowpulseoccupancy = 0;

        //read other sensors
        char buffer[10];
        //light sensor
        float luminance = analogRead(luminancePin);
        //UV sensor
        float uv = analogRead(uvPin);
        uv = uv * 0.0049; //convert values to volts
        uv = uv * 307; //convert to mW/m²
        uv = uv/200; //calculate UV index
        //temperature and humidity

```

```

int chk = DHT.read22(DHT22_PIN);
float humidity = DHT.humidity;
float temperature = DHT.temperature;
//pressure and temperature1
sensors_event_t event;
bmp.getEvent(&event);
float pressure = 0;
float temperature1 = 0;
if (event.pressure)
{
    pressure = event.pressure;
    bmp.getTemperature(&temperature1);
}

//convert sensor values to strings
String luminanceStr = dtostrf(luminance, 4, 1, buffer);
luminanceStr.replace(" ", "");
String uvStr = dtostrf(uv, 4, 1, buffer);
uvStr.replace(" ", "");
String humidityStr = dtostrf(humidity, 4, 1, buffer);
humidityStr.replace(" ", "");
String temperatureStr = dtostrf(temperature, 4, 1, buffer);
temperatureStr.replace(" ", "");
String dustStr = dtostrf(concentration, 4, 1, buffer);
dustStr.replace(" ", "");
String pressureStr = dtostrf(pressure, 4, 1, buffer);
pressureStr.replace(" ", "");
String temperature1Str = dtostrf(temperature1, 4, 1, buffer);
temperature1Str.replace(" ", "");

//send data to ThingSpeak
updateSensors(luminanceStr, humidityStr, temperatureStr, uvStr,
dustStr, pressureStr, temperature1Str);

//wait next sampling cycle
Serial.print("Wait ");
Serial.print(delay_time/1000);
Serial.println("s for next sampling");
Serial.println();
delay(delay_time);

//initialize new cycle
Serial.println();
Serial.print("Sampling (");
Serial.print(sampletime_ms/1000);
Serial.println("s) ...");

```

```

        starttime = millis();
    }
}

//Send data to ThingSpeak
void updateSensors(String luminanceStr, String humidityStr, String
temperatureStr, String uvStr, String dustStr, String pressureStr,
String temperature1Str) {

    String cmd = "AT+CIPSTART=\"TCP\", \"";
    cmd += IP;
    cmd += "\",80";
    monitor.println(cmd);
    delay(2000);

    cmd = GET;
    cmd += luminanceStr;
    cmd += "&field2=";
    cmd += humidityStr;
    cmd += "&field3=";
    cmd += temperatureStr;
    cmd += "&field4=";
    cmd += uvStr;
    cmd += "&field5=";
    cmd += dustStr;
    cmd += "&field6=";
    cmd += pressureStr;
    cmd += "&field7=";
    cmd += temperature1Str;
    cmd += "\r\n";
    delay(1000);
    int strsize = cmd.length();
    monitor.println("AT+CIPSEND=" + String(strsize));
    delay(2000);

    monitor.print(cmd);
    if(monitor.find("OK")){
        Serial.println("Transmission completed with success");
    }else{
        Serial.println("Transmission failed!");
    }
}

void sendDebug(String cmd) {
    Serial.print("SEND: ");
    Serial.println(cmd);
}

```

```
    monitor.println(cmd);
}

boolean connectWiFi() {
    Serial.println("Connecting wi-fi...");
    String cmd = "AT+CWMODE=1";
    monitor.println(cmd);
    delay(2000);
    monitor.flush(); //clear buffer
    cmd="AT+CWJAP=\"";
    cmd+=SSID;
    cmd+="\", \"";
    cmd+=PASS;
    cmd+="\"";
    monitor.println(cmd);
    delay(5000);

    if(monitor.find("OK")){
        Serial.println("Connection succeeded!");
        return true;
    }else{
        Serial.println("Connection failed!");
        return false;
    }
    Serial.println();
}
```


ThingSpeak Configuration

Channel ID: 1360334

Name: XYZ

Description: Weather Station

Field 1: Temperature ☒

Field 2: Humidity ☒

Field 3: Atmospheric Pressure ☒

Field 4: Light Intensity ☒

Field 5: UV Index ☒

Field 6: Dust Concentration ☒

Field 7: ☐

Field 8: ☐

Metadata:

Tags:
(Tags are comma separated)

Link to External Site: http://

Channel Settings

- **Percentage complete:** Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, video, and tags to complete your channel.
- **Channel Name:** Enter a unique name for the ThingSpeak channel.
- **Description:** Enter a description of the ThingSpeak channel.
- **Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- **Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- **Tags:** Enter keywords that identify the channel. Separate tags with commas.
- **Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- **Show Channel Location:**
 - **Latitude:** Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072.
 - **Longitude:** Specify the longitude position in decimal degrees. For example, the longitude of the city of London is -0.1275.
 - **Elevation:** Specify the elevation position meters. For example, the elevation of the city of London is 35.052.
- **Video URL:** If you have a YouTube[®] or Vimeo[®] video that displays your channel information, specify the full path of the video URL.
- **Link to GitHub:** If you store your ThingSpeak code on GitHub[®], specify the GitHub repository URL.

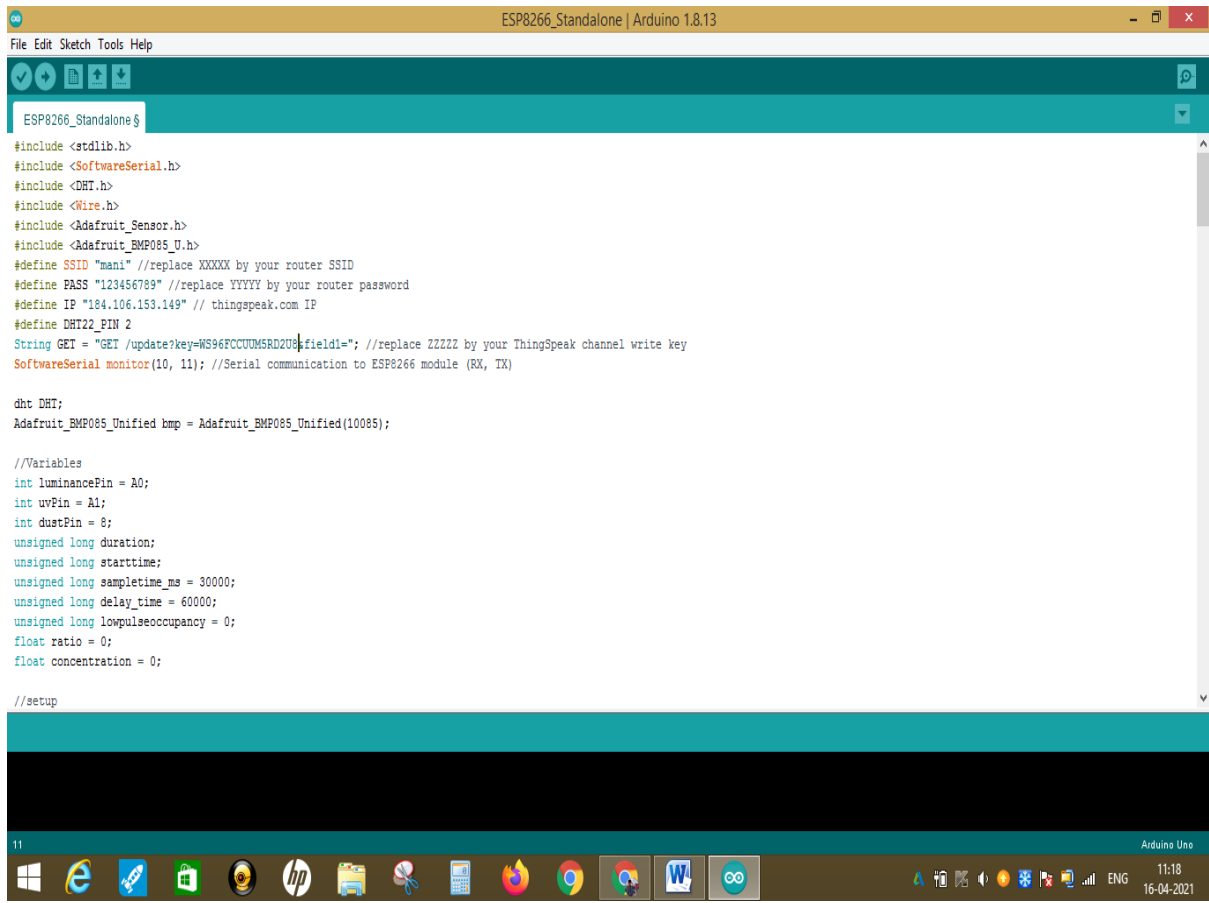
Using the Channel

You can get data into a channel from a device, website, or another ThingSpeak channel. You can then visualize data and transform it using ThingSpeak Apps.

Set the fields like :

- channel 1 = temperature
- channel 2 = humidity
- channel 3 = atmospheric pressure
- channel 4 = light intensity
- channel 5 = UV index
- channel 6 = dust concentration

Uploading the Code



```
ESP8266_Standalone $
#include <stdlib.h>
#include <SoftwareSerial.h>
#include <DHT.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP085_U.h>
#define SSID "mani" //replace XXXXX by your router SSID
#define PASS "123456789" //replace YYYY by your router password
#define IP "184.106.153.149" // thingspeak.com IP
#define DHT22_PIN 2
String GET = "GET /update?key=NS96FCCUUMSRD2U8&field1="; //replace ZZZZ by your ThingSpeak channel write key
SoftwareSerial monitor(10, 11); //Serial communication to ESP8266 module (RX, TX)

dht DHT;
Adafruit_BMP085_Unified bmp = Adafruit_BMP085_Unified(10085);

//Variables
int luminancePin = A0;
int uvPin = A1;
int dustPin = 8;
unsigned long duration;
unsigned long starttime;
unsigned long sampletime_ms = 30000;
unsigned long delay_time = 60000;
unsigned long lowpulseoccupancy = 0;
float ratio = 0;
float concentration = 0;

//setup
```