

### Arduino Code:

```
#include "DHT.h"
#define DHTPIN 12
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

int trig = 9;
int echo = 10;
long duration, inches, cm;

const int DOUTpin1 = 7; //MQ6
int limit1;
int value1;

const int DOUTpin2 = 8; //MQ135
int limit2;
int value2;

const int DOUTpin3 = 6; //MQ2
int limit3;
int value3;

#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_ADXL345_U.h>
Adafruit_ADXL345_Unified accel = Adafruit_ADXL345_Unified();

float acceleration_x, acceleration_y, acceleration_z;

String readstringdata = "";

void setup()
{
  Serial.begin(9600);

  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);

  dht.begin();
```

```

    accel.begin();
}

void loop()
{
    readstringdata = "";

    delay(1000);

    /*****DHT11 Environment Temperature and Humidity Sensor*****/
    // Reading temperature or humidity takes about 250 milliseconds!
    // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
    float h = dht.readHumidity();
    // Read temperature as Celsius (the default)
    float t = dht.readTemperature();
    // Read temperature as Fahrenheit (isFahrenheit = true)
    float f = dht.readTemperature(true);

    // Check if any reads failed and exit early (to try again).
    if (isnan(h) || isnan(t) || isnan(f)) {
        Serial.println("Failed to read from DHT sensor!");
        return;
    }

    // Compute heat index in Fahrenheit (the default)
    float hif = dht.computeHeatIndex(f, h);
    // Compute heat index in Celsius (isFahreheit = false)
    float hic = dht.computeHeatIndex(t, h, false);

    #ifdef serial
    Serial.print("Humidity: ");
    Serial.print(h);
    Serial.print(" %\t");
    Serial.print("Temperature: ");
    Serial.print(t);
    Serial.print(" *C ");
    Serial.print(f);
    Serial.print(" *F\t");
    Serial.print("Heat index: ");
    Serial.print(hic);
    #endif
}

```

```
Serial.print(" *C ");
Serial.print(hif);
Serial.println(" *F");
#endif
delay(500);

/*.....GAS (MQ6).....*/
value1 = analogRead(A1);
limit1 = digitalRead(DOUTpin1);

#ifdef serial
Serial.print("MQ6 value: ");
Serial.print(value1);//prints the alcohol value
Serial.print(" Limit: ");
Serial.println(limit1);/*prints the limit reached as either LOW or HIGH (above or
underneath) */
#endif
delay(500);

/*.....GAS (MQ135).....*/
value2 = analogRead(A2);
limit2 = digitalRead(DOUTpin2);

#ifdef serial
Serial.print("MQ135 value: ");
Serial.print(value2);//prints the alcohol value
Serial.print(" Limit: ");
Serial.println(limit2);/*prints the limit reached as either LOW or HIGH (above or
underneath) */
#endif
delay(500);

/*.....GAS (MQ2).....*/
value3 = analogRead(A0);
limit3 = digitalRead(DOUTpin3);

#ifdef serial
Serial.print("MQ2 value: ");
Serial.print(value3);//prints the alcohol value
Serial.print(" Limit: ");
Serial.println(limit3);/*prints the limit reached as either LOW or HIGH (above or
```

```

underneath) */
#endif
delay(500);

/*****Ultrasonic Sensor*****/
digitalWrite(trig, LOW);
delayMicroseconds(2);
digitalWrite(trig, HIGH);
delayMicroseconds(5);
digitalWrite(trig, LOW);

duration = pulseIn(echo, HIGH);
//this returns the time duration taken
//for the ultrasonics to hit an obstacle and return

inches = duration / 74 / 2;//converts the time duration into inches
cm = duration / 29 / 2;//converts the time duration to cm

//Serial.print("Distance: ");
//Serial.print(cm);
//Serial.println(" cm");
delay(500);

/*****ADXL345 Acceleration Gait Sensor*****/
sensors_event_t event;
accel.getEvent(&event);

acceleration_x = event.acceleration.x;
acceleration_y = event.acceleration.y;
acceleration_z = event.acceleration.z;
#ifdef serial
Serial.print("X: "); Serial.print(event.acceleration.x); Serial.print(" ");
Serial.print("Y: "); Serial.print(event.acceleration.y); Serial.print(" ");
Serial.print("Z: "); Serial.print(event.acceleration.z); Serial.print(" ");
#endif
delay(500);

readstringdata += String(h);
readstringdata += String(",");
readstringdata += String(t);
readstringdata += String(",");

```

```

readstringdata += String(cm);
readstringdata += String(",");
readstringdata += String(value3);
readstringdata += String(",");
readstringdata += String(value1);
readstringdata += String(",");
readstringdata += String(value2);
readstringdata += String(",");
readstringdata += String(acceleration_x);
readstringdata += String(",");
readstringdata += String(acceleration_y);
readstringdata += String(",");
readstringdata += String(acceleration_z);
readstringdata += String('#');
Serial.println(readstringdata);
delay(500);

#ifdef serial
Serial.println("*****");
Serial.println(" ");
#endif

readstringdata = "";

delay(5000);
}

```

### ESP8266 NodeMCU Code:

```

#include <Arduino.h>
#ifdef ESP32
#include <WiFi.h>
#elif defined(ESP8266)
#include <ESP8266WiFi.h>
#endif

#include <WiFiClient.h> //Client wifi connection library

#include <ThingSpeak.h> //ThingSpeak Cloud library

```

```
#define WIFI_SSID "TP-Link_8E98"  
#define WIFI_PASSWORD "86427920"
```

```
WiFiClient client; //client configuration
```

```
unsigned long myChannelNumber1 = 2835825; //Thingspeak channel number  
const char * myWriteAPIKey1 = "TOI17O33AV7U97NE"; //Thingspeak Write API key
```

```
unsigned long myChannelNumber2 = 2835826; //Thingspeak channel number  
const char * myWriteAPIKey2 = "M3CQ4KBRKFRM8LQ6"; //Thingspeak Write API key
```

```
String readstring = "";
```

```
String hum;  
String temp;  
String dist;  
String mq2;  
String mq6;  
String mq135;  
String acceleration_x;  
String acceleration_y;  
String acceleration_z;
```

```
int ind1; // , locations  
int ind2;  
int ind3;  
int ind4;  
int ind5;  
int ind6;  
int ind7;  
int ind8;  
int ind9;
```

```
void setup()  
{  
  Serial.begin(9600);  
  Serial.println();  
  
  Serial.print("Connecting to AP");
```

```
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
while (WiFi.status() != WL_CONNECTED)
{
    Serial.print(".");
    delay(200);
}
```

```
Serial.println("");
Serial.println("WiFi connected.");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
Serial.println();
```

```
ThingSpeak.begin(client);
}
```

```
void loop()
```

```
{
    readstring = ""; //Reset the variable
```

```
    while (Serial.available())
    { //Check if there is an available byte to read
        delay(10); //Delay added to make thing stable
        char c = Serial.read(); //Conduct a serial read
        if (c == '#') {break;} //Exit the loop when the # is detected after the word
        readstring += c; //build the string
    }
```

```
    if (readstring.length() > 0)
    {
        Serial.println(readstring);
```

```
        ind1 = readstring.indexOf(',');
        hum = readstring.substring(0, ind1);
        ind2 = readstring.indexOf(',', ind1+1);
        temp = readstring.substring(ind1+1, ind2);
        ind3 = readstring.indexOf(',', ind2+1); //finds location of second ,
        dist = readstring.substring(ind2+1, ind3);
        ind4 = readstring.indexOf(',', ind3+1); //finds location of second ,
        mq2 = readstring.substring(ind3+1, ind4);
        ind5 = readstring.indexOf(',', ind4+1); //finds location of second ,
```

```
mq6 = readstring.substring(ind4+1, ind5);
ind6 = readstring.indexOf(',', ind5+1);//finds location of second ,
mq135 = readstring.substring(ind5+1, ind6);
ind7 = readstring.indexOf(',', ind6+1);//finds location of second ,
acceleration_x = readstring.substring(ind6+1, ind7);
ind8 = readstring.indexOf(',', ind7+1);//finds location of second ,
acceleration_y = readstring.substring(ind7+1, ind8);
ind9 = readstring.indexOf(',', ind8+1);//finds location of second ,
acceleration_z = readstring.substring(ind8+1);//captures remain part of data after last ,
```

```
Serial.print("Humidity: ");
Serial.println(hum);
Serial.print("Temperature: ");
Serial.println(temp);
Serial.print("Distance: ");
Serial.println(dist);
Serial.print("MQ2: ");
Serial.println(mq2);
Serial.print("MQ6: ");
Serial.println(mq6);
Serial.print("MQ135: ");
Serial.println(mq135);
Serial.print("Acceleration X: ");
Serial.println(acceleration_x);
Serial.print("Acceleration Y: ");
Serial.println(acceleration_y);
Serial.print("Acceleration Z");
Serial.println(acceleration_z);
```

```
ThingSpeak.setField(1, hum);
ThingSpeak.setField(2, temp);
ThingSpeak.setField(3, dist);
ThingSpeak.setField(4, mq2);
ThingSpeak.setField(5, mq6);
ThingSpeak.setField(6, mq135);
```

```
ThingSpeak.writeFields(myChannelNumber1, myWriteAPIKey1);
delay(2000);
```

```
ThingSpeak.setField(1, acceleration_x);
ThingSpeak.setField(2, acceleration_y);
```



```
ThingSpeak.setField(3, acceleration_z);
```

```
ThingSpeak.writeFields(myChannelNumber2, myWriteAPIKey2);  
delay(2000);
```

```
}
```

```
}
```