**CLOUD BASED INDUSTRY MONITORING AND CONTROLLING SYSTEM**

#include <DHT.h>

#include <DHT\_U.h>

#include <ETH.h>

#include <WiFi.h>

#include <WiFiAP.h>

#include <WiFiClient.h>

#include <WiFiGeneric.h>

#include <WiFiMulti.h>

#include <WiFiScan.h>

#include <WiFiServer.h>

#include <WiFiSTA.h>

#include <WiFiType.h>

#include <WiFiUdp.h>

#include <Adafruit\_Sensor.h>

#define DHTPIN 2

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

#include <ThingSpeak.h>

float ldr;

float gas;

float temperature;

float humidity;

const char\* ssid = "Sandhya";

const char\* password = "bhanu123";

void funcldr();

void Gas();

void temp();

WiFiClient client;

unsigned long myChannelNumber = 805513;

const char \*myWriteAPIKey ="P9VUNSXXY5VS5DOV";

WiFiServer server(80);

void setup() {

dht.begin();

pinMode(33,INPUT);

pinMode(15, OUTPUT);

pinMode(23,OUTPUT);

// lcd.begin(16, 2);

pinMode(4,OUTPUT);

Serial.begin(115200);

pinMode(34, INPUT);

Serial.println();

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi is connected");

server.begin();

Serial.println("Server started");

Serial.println(WiFi.localIP());

ThingSpeak.begin(client);

}

void loop() {

{

ldr = analogRead(33);

Serial.print("ldr value:");

Serial.println(ldr);

delay(200);

if(ldr<=3000)

{

digitalWrite(23,HIGH);

delay(500);

}

else

{

digitalWrite(23,LOW);

delay(500);

}

}

{

gas = analogRead(34);

Serial.print("gas value:");

Serial.println(gas);

delay(200);

if(gas>=500)

{

digitalWrite(15,HIGH);

delay(500);

Serial.println("BAD Air");

}

else

Serial.println("GOOD air");

}

{

temperature = dht.readTemperature();

humidity = dht.readHumidity();

Serial.print("Temperature ");

Serial.println(temperature);

delay(500);

Serial.println("Humidity");

Serial.println(humidity);

delay(500);

}

WiFiClient client = server.available();

client.println("HTTP/1.1 200 OK");

client.println("Content-Type: text/html");

client.println("Connection: close");

client.println("Refresh: 10");

client.println();

client.print("<p style='text-align: center;'><span style='color: #0000ff;'><strong style='font-size: large;'>Humidity = ");

client.println(humidity);

client.print("</p>");

client.println("</html>");

delay(5000);

static boolean data\_state = false;

ThingSpeak.writeField(myChannelNumber, 1, temperature, myWriteAPIKey);

delay(15000);

client.println("<!DOCTYPE HTML>");

client.println("<html>");

client.print("<p style='text-align: center;'><span style='font-size: x-large;'><strong>Digital Temperature meter</strong></span></p>");

client.print("<p style='text-align: center;'><span style='color: #0000ff;'><strong style='font-size: large;'>Temperature (\*C)= ");

client.println(temperature);

//ThingSpeak.writeField(myChannelNumber, 1, humidity, myWriteAPIKey);

ThingSpeak.writeField(myChannelNumber, 2, gas, myWriteAPIKey);

delay(30000);

}