// Libraries

#include <SPI.h>

#include <WiFi.h>

#include <WiFiClient.h>

//#include <dht.h>

// WiFi Client

WiFiClient client;

// Your network name also called SSID

char ssid[] = "curiousfly\_devs";

// your network password

char password[] = "talk2e2016";

// your network key Index number (needed only for WEP)

int keyIndex = 0;

// Dweet parameters

char \* server\_name = "www.dweet.io";

#define thing\_name "xtrematix"

const int LED3\_WIFICONNECT =21;

const int LED4\_ONLINE\_STATUS =3;

const int LED5\_DEVICE\_RUNNING\_STATUS =4;

void setup() {

 Serial.begin(115200); // Initialize serial communication

 pinMode(LED3\_WIFICONNECT, OUTPUT); //WIFI CONNECTIVITY STATUS

 pinMode(LED4\_ONLINE\_STATUS, OUTPUT); //ONLINE CONNECTED OR NOT

 pinMode(LED5\_DEVICE\_RUNNING\_STATUS, OUTPUT);

 digitalWrite(LED3\_WIFICONNECT,LOW);

 digitalWrite(LED5\_DEVICE\_RUNNING\_STATUS,LOW);

 digitalWrite(LED4\_ONLINE\_STATUS, LOW);

 // attempt to connect to Wifi network:

 Serial.print("Attempting to connect to Network named: ");

 // print the network name (SSID);

 Serial.println(ssid);

 // Connect to WPA/WPA2 network. Change this line if using open or WEP network:

 WiFi.begin(ssid, password);

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

 // print dots while we wait to connect

 Serial.print(".");

 delay(300);

 }

 digitalWrite(LED3\_WIFICONNECT,HIGH);

 // Pin for the PIR sensor

 //pinMode(8,INPUT);

 Serial.println("\nYou're connected to the network");

 Serial.println("Waiting for an ip address");

 while (WiFi.localIP() == INADDR\_NONE) {

 // print dots while we wait for an ip addresss

 Serial.print(".");

 delay(300);

 }

printWifiStatus();

}

void loop() {

 // Measure motion

 int motion = digitalRead(8);

 //Serial.println(motion);

 // Measure temperature & humidity

 float temperature=53.0, humidity;

 digitalWrite(LED5\_DEVICE\_RUNNING\_STATUS,!digitalRead(LED5\_DEVICE\_RUNNING\_STATUS));

 // Send data to server

 if (client.sslConnect(server\_name,443))

 {

 //Serial.println("Connected");

 //Serial.print(F("Sending request... "));

 client.print(F("GET /dweet/for/"));

 client.print(thing\_name);

 client.print(F("?temperature="));

 client.print(temperature);

 client.print(F("&humidity="));

 client.print(humidity);

 client.print(F("&motion="));

 client.print(motion);

 client.println(F(" HTTP/1.1"));

 client.println(F("Host: dweet.io"));

 client.println(F("Connection: close"));

 client.println(F(""));

 //Serial.println(F("done."));

 }

 // Read answer

 //Serial.println(F("Reading answer..."));

 while (client.connected())

 {

 while (client.available())

 {

 char c = client.read();

 Serial.print(c);

 }

 }

 //Serial.println(F(""));

 // Close connection

 client.stop();

 // Serial.println(F("Closing connection"));

 // Serial.println(F(""));

 if( WiFi.status() != WL\_CONNECTED)

 {

 digitalWrite(LED3\_WIFICONNECT,LOW);

 }

}

void printWifiStatus() {

 Serial.print("WiFi.status():");

 Serial.println(WiFi.status());

 // print the SSID of the network you're attached to:

 Serial.print("SSID: ");

 Serial.println(WiFi.SSID());

 // print your WiFi shield's IP address:

 IPAddress ip = WiFi.localIP();

 Serial.print("IP Address: ");

 Serial.println(ip);

 IPAddress Gip = WiFi.gatewayIP();

 Serial.print("GATEWAY IP Address: ");

 Serial.println(Gip);

 IPAddress Sip = WiFi.subnetMask();

 Serial.print("SUBNET MASK IP Address: ");

 Serial.println(Sip);

 // print the received signal strength:

 long rssi = WiFi.RSSI();

 Serial.print("signal strength (RSSI):");

 Serial.print(rssi);

 Serial.println(" dBm");

}