#include <DHTesp.h>

#include <Adafruit\_MQTT\_Client.h>

#include <Adafruit\_MQTT.h>

#include <ESP8266WiFi.h>

#include <Wire.h>

#include <PubSubClient.h>

#include "DHT.h"

#define DHTTYPE DHT22

#define DHTPIN 4

DHT dht(DHTPIN, DHTTYPE);

#define wifi\_ssid "\*\*\*\*\*\*\*\*apke wifi ka naam\*\*\*\*\*\*\*\*"

#define wifi\_password "\*\*\*aapke wifi ka password\*\*\*\*\*\*l"

#define mqtt\_server "\*\*\*\*\*\*ip address mqtt ka...\*\*\*\*\*\*"

//#define mqtt\_user "user"

//#define mqtt\_password "password"

#define humidity\_topic "sensor/humidity"

#define temperature\_celsius\_topic "sensor/temp\_c"

#define temperature\_fahrenheit\_topic "sensor/temp\_f"

WiFiClient espClient;

PubSubClient client(espClient);

void setup() {

Serial.begin(115200);

dht.begin();

setup\_wifi();

client.setServer(mqtt\_server, 1883);

}

String macToStr(const uint8\_t\* mac)

{

String result;

for (int i = 0; i < 6; ++i) {

result += String(mac[i], 16);

if (i < 5)

result += ':';

}

return result;

}

void setup\_wifi() {

delay(10);

// We start by connecting to a WiFi network

Serial.println();

Serial.print("Connecting to ");

Serial.println(wifi\_ssid);

WiFi.begin(wifi\_ssid, wifi\_password);

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

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

}

void reconnect() {

// Loop until we're reconnected

while (!client.connected()) {

Serial.print("Attempting MQTT connection...");

// Generate client name based on MAC address and last 8 bits of microsecond counter

String clientName;

clientName += "esp8266-";

uint8\_t mac[6];

WiFi.macAddress(mac);

clientName += macToStr(mac);

clientName += "-";

clientName += String(micros() & 0xff, 16);

Serial.print("Connecting to ");

Serial.print(mqtt\_server);

Serial.print(" as ");

Serial.println(clientName);

// Attempt to connect

// If you do not want to use a username and password, change next line to

if (client.connect((char\*) clientName.c\_str())) {

//if (client.connect((char\*) clientName.c\_str()), mqtt\_user, mqtt\_password)) {

Serial.println("connected");

} else {

Serial.print("failed, rc=");

Serial.print(client.state());

Serial.println(" try again in 5 seconds");

// Wait 5 seconds before retrying

delay(5000);

}

}

}

void loop() {

if (!client.connected()) {

reconnect();

}

client.loop();

// Wait a few seconds between measurements.

delay(2000);

// 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);

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);

Serial.print(" \*C ");

Serial.print(hif);

Serial.println(" \*F");

Serial.print("Temperature in Celsius:");

Serial.println(String(t).c\_str());

client.publish(temperature\_celsius\_topic, String(t).c\_str(), true);

Serial.print("Temperature in Fahrenheit:");

Serial.println(String(f).c\_str());

client.publish(temperature\_fahrenheit\_topic, String(f).c\_str(), true);

Serial.print("Humidity:");

Serial.println(String(h).c\_str());

client.publish(humidity\_topic, String(h).c\_str(), true);

}