#include <WiFi.h>

#include <WiFiClient.h>

#include <WiFiServer.h>

#include <WiFiUdp.h>

#include <LiquidCrystal.h>

LiquidCrystal lcd(4,5,18,19,22,23);

#include <ThingSpeak.h>

const char\* ssid = "SmartBridge";

const char\* password = "smartbridge@sb";

WiFiClient client;

float temp\_celsius = 0;

float temp\_fahrenheit = 0;

WiFiServer server(80);

unsigned long myChannelNumber = 804337;

const char \* myWriteAPIKey = "7HVZVZ6OQO9J14OZ";

String header;

String output27State = "off";

const int output27 = 27;

String output12State = "off";

const int output12 = 12;

String output13State = "off";

const int output13 = 13;

String output15State = "off";

const int output15 = 15;

String output4State = "off";

const int output4 = 4;

String output5State = "off";

const int output5 = 5;

String output26State = "off";

const int output26 = 26;

void setup()

{

lcd.begin(16, 2);

Serial.begin(115200);

pinMode(34, INPUT);

pinMode(output27, OUTPUT);

pinMode(output12, OUTPUT);

pinMode(output13, OUTPUT);

pinMode(output15, OUTPUT);

pinMode(output26, OUTPUT);

pinMode(output5, OUTPUT);

digitalWrite(output26, LOW);

digitalWrite(output5, LOW);

pinMode(output4, OUTPUT);

digitalWrite(output4, LOW);

digitalWrite(output27, LOW);

digitalWrite(output12, LOW);

digitalWrite(output13, LOW);

digitalWrite(output15, LOW);

Serial.println();

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

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

{

delay(500);

Serial.println(".");

}

Serial.println("");

Serial.println("WiFi connected.");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

server.begin();

ThingSpeak.begin(client);

}

void loop()

{

static boolean data\_state = false;

float a = analogRead(34);

temp\_celsius = (a/9.31) ;

temp\_fahrenheit = temp\_celsius \* 1.8 + 32.0;

ThingSpeak.writeField(myChannelNumber, 1, temp\_celsius, myWriteAPIKey);

ThingSpeak.writeField(myChannelNumber, 2, temp\_fahrenheit, myWriteAPIKey);

Serial.print(" Temperature = ");

Serial.print(temp\_celsius);

Serial.print(" Celsius, ");

Serial.print(temp\_fahrenheit);

Serial.println(" Fahrenheit");

WiFiClient client = server.available();

if (client) { // If a new client connects,

Serial.println("New Client."); // print a message out in the serial port

String currentLine = ""; // make a String to hold incoming data from the client

while (client.connected())

{ // loop while the client's connected

if (client.available()) { // if there's bytes to read from the client,

char c = client.read(); // read a byte, then

Serial.write(c); // print it out the serial monitor

header += c;

if (c == '\n')

{

if (currentLine.length() == 0)

{

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

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

client.println("Connection: close");

client.println("Refresh: 10");

client.println();

if (header.indexOf("GET /27/on") >= 0) {

Serial.println("GPIO 27 on");

output27State = "on";

digitalWrite(output27, HIGH);

lcd.println("R-1");

} else if (header.indexOf("GET /27/off") >= 0) {

Serial.println("GPIO 27 off");

output27State = "off";

digitalWrite(output27, LOW);

}

if (header.indexOf("GET /12/on") >= 0) {

Serial.println("GPIO 12 on");

output12State = "on";

digitalWrite(output12, HIGH);

lcd.println("R-2");

} else if (header.indexOf("GET /12/off") >= 0) {

Serial.println("GPIO 12 off");

output12State = "off";

digitalWrite(output12, LOW);

}

if (header.indexOf("GET /13/on") >= 0) {

Serial.println("GPIO 13 on");

output13State = "on";

digitalWrite(output13, HIGH);

lcd.println("R-3");

} else if (header.indexOf("GET /13/off") >= 0) {

Serial.println("GPIO 13 off");

output13State = "off";

digitalWrite(output13, LOW);

}

if (header.indexOf("GET /15/on") >= 0) {

Serial.println("GPIO 15 on");

output15State = "on";

digitalWrite(output15, HIGH);

lcd.println("R-4");

} else if (header.indexOf("GET /15/off") >= 0) {

Serial.println("GPIO 15 off");

output15State = "off";

digitalWrite(output15, LOW);

}

if(temp\_celsius>30)

{

output26State = "on";

digitalWrite(output26, HIGH);

}

else

{

output26State = "off";

digitalWrite(output26, LOW);

}

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

client.println("<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">");

client.println("<link rel=\"icon\" href=\"data:,\">");

client.println("<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center;}");

client.println(".button { background-color: #4CAF50; border: none; color: white; padding: 16px 40px;");

client.println("text-decoration: none; font-size: 30px; margin: 2px; cursor: pointer;}");

client.println(".button2 {background-color: #555555;}</style></head>");

client.println("<body><h1>ESP32 Web Server</h1>");

client.println("<p>GPIO 27 - State " + output27State + "</p>");

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(temp\_celsius);

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

client.println(temp\_fahrenheit);

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

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

if (output27State=="off") {

client.println("<p><a href=\"/27/on\"><button class=\"button\">ON</button></a></p>");

} else {

client.println("<p><a href=\"/27/off\"><button class=\"button button2\">OFF</button></a></p>");

}

if (output12State=="off") {

client.println("<p><a href=\"/12/on\"><button class=\"button\">ON</button></a></p>");

} else {

client.println("<p><a href=\"/12/off\"><button class=\"button button2\">OFF</button></a></p>");

}

if (output13State=="off") {

client.println("<p><a href=\"/13/on\"><button class=\"button\">ON</button></a></p>");

} else {

client.println("<p><a href=\"/13/off\"><button class=\"button button2\">OFF</button></a></p>");

}

if (output15State=="off") {

client.println("<p><a href=\"/15/on\"><button class=\"button\">ON</button></a></p>");

} else {

client.println("<p><a href=\"/15/off\"><button class=\"button button2\">OFF</button></a></p>");

}

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

client.println();

break;

} else { // if you got a newline, then clear currentLine

currentLine = "";

}

} else if (c != '\r') { // if you got anything else but a carriage return character,

currentLine += c; // add it to the end of the currentLine

}

}

}

// Clear the header variable

header = "";

// Close the connection

client.stop();

Serial.println("Client disconnected.");

Serial.println("");

}

}