float val, voltage, temp;

String ssid = "Simulator Wifi"; // SSID to connect to

String password = ""; //virtual wifi has no password

String host = "api.thingspeak.com"; // Open Weather Map API

const int httpPort = 80;

String url = "/update?api\_key=M3SQL09NDRKCJ6JA&field1=";

//Replace XXXXXXXXXXXXXXXX by your ThingSpeak Channel API Key

const int trigger = 11;

const int echo = 12;

int pushButton = 2;

int led = 13;

#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 10, 9, 8, 7);

int bpm[100] = {72,70,65,67,73,80,100,85,80,75,95,76,65,70,77,99,85,88,72,75};

int bpm1 = bpm[0];

int temp1 = 0;

float duration, distance;

void setupESP8266(void) {

// Start our ESP8266 Serial Communication

Serial.begin(115200); // Serial connection over USB to computer

Serial.println("AT"); // Serial connection on Tx / Rx port to ESP8266

delay(10); // Wait a little for the ESP to respond

if (Serial.find("OK"))

Serial.println("ESP8266 OK!!!");

// Connect to Simulator Wifi

Serial.println("AT+CWJAP=\"" + ssid + "\",\"" + password + "\"");

delay(10); // Wait a little for the ESP to respond

if (Serial.find("OK"))

Serial.println("Connected to WiFi!!!");

// Open TCP connection to the host:

//ESP8266 connects to the server as a TCP client.

Serial.println("AT+CIPSTART=\"TCP\",\"" + host + "\"," + httpPort);

delay(50); // Wait a little for the ESP to respond

if (Serial.find("OK"))

Serial.println("ESP8266 Connected to server!!!") ;

}

void anydata(void) {

digitalWrite(trigger, LOW);

delayMicroseconds(2);

digitalWrite(trigger, HIGH);

delayMicroseconds(10);

digitalWrite(trigger, LOW);

//duration = pulseIn(echo,HIGH);

//distance = (duration \*0.000001 \* 3400) / 2; // dist in 'm'

//temp = distance;

int buttonState = digitalRead(pushButton);

Serial.println(buttonState);

if (buttonState == 1) {

digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)

String httpPacket = "GET " + url + String(temp) + "&field2=" + String(bpm[temp1]) + "&field3=" + String(buttonState) + " HTTP/1.1\r\nHost: " + host + "\r\n\r\n";

} else {

digitalWrite(led, LOW); // turn the LED off by making the voltage LOW

}

val=analogRead(A0);

voltage=val\*0.0048828125;

temp = (voltage - 0.5) \* 100.0;

// Construct our HTTP call

if (temp1 == 15){

temp1 = 0;

}

String httpPacket = "GET " + url + String(temp) + "&field2=" + String(bpm[temp1]) + "&field3=" + String(buttonState) + " HTTP/1.1\r\nHost: " + host + "\r\n\r\n";

temp1 = temp1 + 1;

int length = httpPacket.length();

//int length1 = httpPacket1.length();

// Send our message length

Serial.print(String(temp) +String(bpm[0]));

Serial.print("AT+CIPSEND=");

Serial.println(length);

//Serial.print("AT+CIPSEND=");

//Serial.println(length1);

delay(10); // Wait a little for the ESP to respond if (!Serial.find(">")) return -1;

// Send our http request

Serial.print(httpPacket);

delay(10); // Wait a little for the ESP to respond

if (Serial.find("SEND OK\r\n"))

Serial.println("ESP8266 sends data to the server");

}

void setup() {

pinMode(trigger, OUTPUT);

pinMode(echo, INPUT);

Serial.begin(9600);

setupESP8266();

pinMode(pushButton, INPUT);

pinMode(led, OUTPUT);

// set up the LCD's number of columns and rows:

lcd.begin(16, 2);

// Print a message to the LCD.

lcd.print("Temp:");

}

void loop() {

anydata();

lcd.setCursor(0, 1);

lcd.print(temp);

delay(10000);

}