/\*

\*/

#include <SoftwareSerial.h> //Software Serial library

#include <LiquidCrystal.h>

#include "wires.h"

SoftwareSerial espSerial(2, 3); //Pin 2 and 3 act as RX and TX. Connect them to TX and RX of ESP8266

#define DEBUG true

String mySSID = "glucopro"; // WiFi SSID

String myPWD = "gluco#02"; // WiFi Password

String myAPI = "OU8BR0BQW16FU1JU"; // API Key

String myHOST = "api.thingspeak.com";

String myPORT = "80";

String myFIELD = "field1";

int sendVal;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

int sensorValue = 0; // value read from the pot

int outputValue = 0; // value output to the PWM (analog out)

void setup() {

// initialize serial communications at 9600 bps:

Serial.begin(9600);

pinMode(senin, INPUT);

lcd.begin(16, 2);

lcd.print("Gluco Level");

espSerial.begin(115200);

espData("AT+RST", 1000, DEBUG); //Reset the ESP8266 module

espData("AT+CWMODE=1", 1000, DEBUG); //Set the ESP mode as station mode

espData("AT+CWJAP=\""+ mySSID +"\",\""+ myPWD +"\"", 1000, DEBUG); //Connect to WiFi network

/\*while(!esp.find("OK"))

{

//Wait for connection

}\*/

delay(1000);

}

void loop() {

// read the analog in value:

sensorValue = analogRead(s1);

// map it to the range of the analog out:

outputValue = map(sensorValue, 0, 1023, 0, 255);

// change the analog out value:

analogWrite(out1, outputValue);

// print the results to the Serial Monitor:

Serial.print("sensor = ");

Serial.print(sensorValue);

Serial.print("\t output = ");

Serial.println(outputValue);

if(digitalRead(senin))

{

lcd.setCursor(0,1);

lcd.print(" ");

lcd.setCursor(0,1);

lcd.print(0);

sendVal =(0);

}

else

{

lcd.setCursor(0,1);

lcd.print(" ");

lcd.setCursor(0,1);

lcd.print(sensorValue \* 2);

sendVal =(sensorValue \* 2);

}

// wait 2 milliseconds before the next loop for the analog-to-digital

// converter to settle after the last reading:

//delay(50);

//sendVal = random(1000); // Send a random number between 1 and 1000

// sendVal =(sensorValue \* 2);

String sendData = "GET /update?api\_key="+ myAPI +"&"+ myFIELD +"="+String(sendVal);

espData("AT+CIPMUX=1", 1000, DEBUG); //Allow multiple connections

espData("AT+CIPSTART=0,\"TCP\",\""+ myHOST +"\","+ myPORT, 1000, DEBUG);

espData("AT+CIPSEND=0," +String(sendData.length()+4),1000,DEBUG);

espSerial.find(">");

espSerial.println(sendData);

Serial.print("Value to be sent: ");

Serial.println(sendVal);

espData("AT+CIPCLOSE=0",1000,DEBUG);

//delay(10000);

delay(1000);

}

String espData(String command, const int timeout, boolean debug)

{

Serial.print("AT Command ==> ");

Serial.print(command);

Serial.println(" ");

String response = "";

espSerial.println(command);

long int time = millis();

while ( (time + timeout) > millis())

{

while (espSerial.available())

{

char c = espSerial.read();

response += c;

}

}

if (debug)

{

//Serial.print(response);

}

return response;

}