#include <LiquidCrystal.h>

const int rs = 15, en = 14, d4 = 13, d5 = 12, d6 = 11, d7 = 10;

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

UART Serial2(8, 9, NC, NC);//9rx 8tx

String data1="";

String data2="";

int temp=0;

void setup()

{

lcd.begin(16, 2);

lcd.print("hello, world");

Serial.begin(9600);

Serial2.begin(9600);//9rx 8tx

}

void loop()

{

back:

while(Serial2.available())

{

data1=Serial2.readString(); //PH:37.76, W: 0, L: 54, T: 61,

lcd.clear();

int len=data1.length();

int i=0;lcd.clear();

for(i=1;i<=len;i++)

{

lcd.print(data1[i]);delay(1);

if(i==15)

lcd.setCursor(0,1);delay(1);

}

temp=temp+1;

if(temp==10)

{

int len=data1.length();

int i=0;

for(i=0;i<len;i++)

{

if(data1[i]==' ' || data1[i]==',' || data1[i]==':')

{

data1[i]='\_';

}

}

Serial2.println(data1);delay(1);

lcd.clear();lcd.print("DATA UPLOADED.....");delay(1000);

temp=0;

data1.replace("","")

Serial.println(data1);

}

}

}

#include <OneWire.h>

#include <DallasTemperature.h>

#include <String.h>

// GPIO where the DS18B20 is connected to

const int oneWireBus = 4;

// Setup a oneWire instance to communicate with any OneWire devices

OneWire oneWire(oneWireBus);

// Pass our oneWire reference to Dallas Temperature sensor

DallasTemperature sensors(&oneWire);

//#define BLYNK\_TEMPLATE\_ID "TMPLyCfpOWwP"

//#define BLYNK\_TEMPLATE\_NAME "HOME AUTOMATION"

//#define BLYNK\_AUTH\_TOKEN "sWkLrTAlhVD0wHicyaWrkFZk6Sty4DZo"

#define BLYNK\_TEMPLATE\_ID "TMPL390hXLvCC"

#define BLYNK\_TEMPLATE\_NAME "HOME AUTOMATION"

#define BLYNK\_AUTH\_TOKEN "9b2VKbkFYoafDZgxTmTEcpoBANUs3mjo"

int l1=25;//25

int l2=26;//26

int l3=32;//32

int pp=33;//33

#define BLYNK\_PRINT Serial

//#include <ESP8266WiFi.h>

//#include <BlynkSimpleEsp8266.h>

#include <WiFi.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>

char auth[] = BLYNK\_AUTH\_TOKEN;

// Your WiFi credentials.

// Set password to "" for open networks.

char ssid[] = "ZTE-sUQdqa";

char pass[] = "5hjgxyh9";

//char ssid[] = "project";

//char pass[] = "project.123";

uint8\_t connection\_state = 0;

uint16\_t reconnect\_interval = 10000;

WiFiClient client;

String data1="";

String data2="cmd";

String data=" MESSAGE";

BlynkTimer timer;

// This function sends Arduino's up time every second to Virtual Pin (5).

// In the app, Widget's reading frequency should be set to PUSH. This means

// that you define how often to send data to Blynk App.

void upload()

{

const char\* server4 = "api.thingspeak.com";

const char\* \_getLink4 = "https://api.thingspeak.com/update?api\_key=PWX6Z0ZH2CQXFG3X&field1="; // Thingspeak.com

//const char\* \_getLink4 = "https://api.thingspeak.com/update?api\_key=8QEMT0G004GKO1L8&field1="; // Thingspeak.com

// Serial.println("data uploading");delay(1000);

client.connect(server4,80);

if (client.connect(server4,80)) // "184.106.153.149" or api.thingspeak.com https://api.thingspeak.com/apps/thinghttp/send\_request?api\_key=CT9B331KB5PLM1G5

{

String getStr4 = \_getLink4;

client.print("GET "+getStr4+data1+"\n");

client.print("HTTP/1.1\n");

client.print("Host: api.thingspeak.com\n");

client.print("Connection: close\n\n\n");

}

client.stop();

}

void myTimerEvent()

{

while(Serial.available())

{

data1=Serial.readString();

Serial.println(data1);

data1.replace("\r\n","\_");

data1.replace("\_PH","PH");

data1.replace("\_\_","\_");

//upload();

Blynk.virtualWrite(V4,data1);

sensors.requestTemperatures();

float temperatureC = sensors.getTempCByIndex(0);

float temperatureF = sensors.getTempFByIndex(0);

//Serial.println(temperatureC);

String iotdata="WATER\_TEMP\_"+String(temperatureC);

data1=data1+iotdata;

Serial.println(data1);

upload();

Blynk.virtualWrite(V4,iotdata);delay(10);

if(temperatureC>35)

{

digitalWrite(l1,HIGH);

}

else

{

digitalWrite(l1,LOW);

}

}

}

BLYNK\_WRITE(V0)

{

int pinValue = param.asInt(); // assigning incoming value from pin V1 to a variable

//Serial.print("VALUE V0:");//Serial.println(String(pinValue));

if(pinValue==1)

{

Serial.print("1");delay(100);

Blynk.virtualWrite(V4, "FROM IOT FEEDER ON\r\n");

digitalWrite(l1,LOW);delay(6000);digitalWrite(l1,HIGH);delay(100);

Blynk.virtualWrite(V4, "FROM IOT FEEDER OFF\r\n");

}

if(pinValue==0)

{

Serial.print("2");delay(100);

//Blynk.virtualWrite(V4, "FROM IOT FEEDER OFF\r\n");

digitalWrite(l1,LOW);delay(6000);digitalWrite(l1,HIGH);delay(100);

}

}

BLYNK\_WRITE(V1)

{

int pinValue = param.asInt(); // assigning incoming value from pin V1 to a variable

//Serial.print("VALUE V1:");Serial.println(String(pinValue));

//lcd.clear();lcd.print("LIGHT1:");lcd.print(String(pinValue));delay(1000);

if(pinValue==1)

{

Serial.print("3");delay(100);

Blynk.virtualWrite(V4, "Load:2 On\r\n");

digitalWrite(l2,HIGH);delay(1000);

}

if(pinValue==0)

{

Serial.print("4");delay(100);

Blynk.virtualWrite(V4, "Load:2 Off\r\n");

digitalWrite(l2,LOW);delay(1000);

}

}

BLYNK\_WRITE(V2)

{

int pinValue = param.asInt(); // assigning incoming value from pin V1 to a variable

//Serial.print("VALUE V2:");Serial.println(String(pinValue));

if(pinValue==1)

{

Serial.print("5");delay(100);

Blynk.virtualWrite(V4, "Load:3 On\r\n");

digitalWrite(l3,HIGH);delay(1000);

}

if(pinValue==0)

{

Serial.print("6");delay(100);

Blynk.virtualWrite(V4, "Load:3 Off\r\n");

digitalWrite(l3,LOW);delay(1000);

}

}

BLYNK\_WRITE(V3)

{

int pinValue = param.asInt(); // assigning incoming value from pin V1 to a variable

//Serial.print("VALUE V3:");Serial.println(String(pinValue));

if(pinValue==1)

{

Serial.print("7");delay(100);

Blynk.virtualWrite(V4, "Load:4 On\r\n");

digitalWrite(pp,HIGH);delay(1000);

}

if(pinValue==0)

{

Serial.print("8");delay(100);

Blynk.virtualWrite(V4, "Load:4 Off \r\n");

digitalWrite(pp,LOW);delay(1000);

}

}

BLYNK\_WRITE(V4)

{

int pinValue = int(param.asInt()); // assigning incoming value from pin V1 to a variable

Serial.print("DATA FROM TERMINAL:");Serial.println(pinValue);

//Blynk.virtualWrite(V4,"DATA FROM TERMINAL:");Blynk.virtualWrite(V4,pinValue);

if(pinValue==12345)

{

Serial.println("CORRCET PASSWORD LOCKER OPEN");delay(1000);

Blynk.virtualWrite(V4,"LOCKER OPEN");

}

else if(pinValue==54321)

{

Serial.println("CORRCET PASSWORD LOCKER CLOSE");delay(1000);

Blynk.virtualWrite(V4,"LOCKER CLOSE");

}

}

void setup()

{

sensors.begin();

pinMode(l1,OUTPUT);pinMode(l2,OUTPUT);pinMode(l3,OUTPUT);pinMode(pp,OUTPUT);

digitalWrite(l1,LOW);digitalWrite(l2,HIGH);digitalWrite(l3,HIGH);digitalWrite(pp,HIGH);

// Debug console

Serial.begin(9600);

sensors.begin();

Blynk.begin(auth, ssid, pass);delay(1000);

// You can also specify server:

//Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);

//Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);

timer.setInterval(1000L, myTimerEvent);

}

void loop()

{

Blynk.run();

timer.run();

// You can inject your own code or combine it with other sketches.

// Check other examples on how to communicate with Blynk. Remember

// to avoid delay() function!

}