//Khai báo thư viện

#include<WiFi.h>

#include<HTTPClient.h>

#include<Arduino\_JSON.h>

#include "DHT.h"

//định nghĩa chân DHT11

#define DHTPIN 15

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

//tạo biến đường link server

const char\* serverNameSendData = "http://dangbxiot.000webhostapp.com/loaptrung/api.php";

const char\* serverNameGetData = "http://dangbxiot.000webhostapp.com/loaptrung/action.php?action=get\_data\_json&board=1";

//khởi tạo biến timer

const long interval = 1000;

const long intervalSendData = 1000;

unsigned long previousMillis = 0;

unsigned long previousMillisTimer = 0;

unsigned long previousMillisSendData = 0;

unsigned long timerInterval;

//khởi tạo chuỗi json

String dataJson;

//khai báo các biến cần thiết

float tempsetValue, humdsetValue, timesetValue, temp, humd, h, t, pretempsetValue, prehumdsetValue, pretimesetValue;

float timer=0;

int den, quatIn, quatOut, dongCo,i, dem=0;

//khai báo task handle

TaskHandle\_t Task1Core0, Task2Core0;

TaskHandle\_t Task1Core1, Task2Core1;

void setup() {

.//cấu hình các chân GPIO

pinMode(19, OUTPUT);

pinMode(18, OUTPUT);

pinMode(21, OUTPUT);

pinMode(22, OUTPUT);

//khởi tạo cổng serial

Serial.begin(115200);

//kết nối wifi smartconfig

//Init WiFi as Station, start SmartConfig

WiFi.mode(WIFI\_AP\_STA);

WiFi.beginSmartConfig();

//Wait for SmartConfig packet from mobile

Serial.println("Waiting for SmartConfig.");

while (!WiFi.smartConfigDone()) {

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("SmartConfig received.");

//Wait for WiFi to connect to AP

Serial.println("Waiting for WiFi");

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

delay(500);

Serial.print(".");

}

Serial.println("WiFi Connected.");

Serial.print("IP Address: ");

Serial.println(WiFi.localIP());

//Khởi tạo biến dht

dht.begin();

//khởi tạo tác vụ trên 2 lõi

//========================CORE 0================================================

xTaskCreatePinnedToCore(Task1codeCore0, "Task1codeCore0", 5000, NULL, 1, &Task1Core0, 0);

//========================CORE 1================================================

xTaskCreatePinnedToCore(Task1codeCore1, "Task1codeCore1", 5000, NULL, 1, &Task1Core1, 1);

}

//chương trình task1 trên lõi 0

void Task1codeCore0( void \* pvParameters ){

for(;;){

unsigned long currentMillisSendData = millis();

if(currentMillisSendData - previousMillisSendData >= intervalSendData){

if(WiFi.status()== WL\_CONNECTED){

sendDatatoServer();

previousMillisSendData = currentMillisSendData;

vTaskDelay(2000);

}

}

}

}

//chương trình task1 trên lõi 1

void Task1codeCore1( void \* pvParameters ){

for(;;){

if(WiFi.status()== WL\_CONNECTED){

deviceset();

if(tempsetValue > 0 && humdsetValue > 0 && timesetValue > 0 ){

setGPIO();

setTimerInterval();

}

}

vTaskDelay(1000);

}

}

//=============chuong trinh con===========================

//phân tích, xử lý dữ liệu Json

void deviceset(){

dataJson = httpGETRequestDevice(serverNameGetData);

Serial.println(dataJson);

JSONVar deviceObj = JSON.parse(dataJson);

if(JSON.typeof(deviceObj)== "undefined"){

Serial.println("Parsing input failed!");

return;

}

Serial.print("JSON Object Device = ");

Serial.println(deviceObj);

JSONVar keys = deviceObj.keys();

for(int i=0;i < keys.length();i++){

JSONVar value = deviceObj[keys[i]];

if(i==0){

den = atoi(value);

if(den==0){digitalWrite(19,LOW);}

}

else if(i==1){

quatIn = atoi(value);

if(quatIn==0){digitalWrite(18,LOW);}

}

else if(i==2){

quatOut = atoi(value);

if(quatOut==0){digitalWrite(21,LOW);}

}

else if(i==3){

dongCo = atoi(value);

if(dongCo==0){digitalWrite(22,LOW);}

}

else if(i==4){

tempsetValue = atof(value);

}

else if(i==5){

humdsetValue = atof(value);

}

else if(i==6){

timesetValue = atof(value);

}

}

Serial.print("den = ");

Serial.println(den);

Serial.print("quatIn = ");

Serial.println(quatIn);

Serial.print("quatOut = ");

Serial.println(quatOut);

Serial.print("dongCo = ");

Serial.println(dongCo);

Serial.print("tempset = ");

Serial.println(tempsetValue);

Serial.print("humdset = ");

Serial.println(humdsetValue);

Serial.print("timeset = ");

Serial.println(timesetValue);

}

//gửi dữ liệu lên server

void sendDatatoServer(){

read\_dht();

send\_data();

}

//đọc dữ liệu cảm biến dht11

void read\_dht() {

h = dht.readHumidity();

t = dht.readTemperature();

while (isnan(h) || isnan(t)) {

dht.begin();

h = dht.readHumidity();

t = dht.readTemperature();

}

temp = t;

humd = h;

Serial.print(F("Humidity: "));

Serial.print(h);

Serial.print(F("% Temperature: "));

Serial.print(t);

Serial.println(F("°C "));

}

//gửi dữ liệu lên server

void send\_data(){

String postData = (String)"temp=" + temp + "&humd=" + humd;

HTTPClient http;

http.begin(serverNameSendData);

http.addHeader("Content-Type", "application/x-www-form-urlencoded");

auto httpCode = http.POST(postData);

String payload = http.getString();

http.end();

}

//điều khiển GPIO, kiểm soát nhiệt độ

void setGPIO(){

if(tempsetValue > temp){

if(humdsetValue > humd){

//bat den, tat quat vao, tat quat ra

if(den==1){

digitalWrite(19,HIGH);

digitalWrite(18,LOW);

digitalWrite(21,LOW);

}

else{

digitalWrite(18,LOW);

digitalWrite(21,LOW);

}

}

else if(humdsetValue < humd){

//bat den, tat quat vao, bat quat ra

if(den==1 && quatOut==1){

digitalWrite(19,HIGH);

digitalWrite(18,LOW);

digitalWrite(21,HIGH);

}

else if(den==1 && quatOut==0){

digitalWrite(19,HIGH);

digitalWrite(18,LOW);

}

else if(den==0 && quatOut==1){

digitalWrite(21,HIGH);

digitalWrite(18,LOW);

}

else{digitalWrite(18,LOW);}

}

else{

//bat den, tat quat vao, tat quat ra

if(den==1){

digitalWrite(19,HIGH);

digitalWrite(18,LOW);

digitalWrite(21,LOW);

}

else{

digitalWrite(18,LOW);

digitalWrite(21,LOW);

}

}

}

else if(tempsetValue <= temp){

if(humdsetValue > humd){

//bat den, bat quat vao, tat quat ra

if(den==1 && quatIn==1){

digitalWrite(19,HIGH);

digitalWrite(18,HIGH);

digitalWrite(21,LOW);

}

else if(den==1 && quatIn==0){

digitalWrite(19,HIGH);

digitalWrite(21,LOW);

}

else if(den==0 && quatIn==1){

digitalWrite(18,HIGH);

digitalWrite(21,LOW);

}

else{digitalWrite(21,LOW);}

}

else if(humdsetValue < humd){

//tat den, bat quat vao, bat quat ra

if(quatIn==1 && quatOut==1){

digitalWrite(19,LOW);

digitalWrite(18,HIGH);

digitalWrite(21,HIGH);

}

else if(quatIn==1 && quatOut==0){

digitalWrite(18,HIGH);

digitalWrite(19,LOW);

}

else if(quatIn==0 && quatOut==1){

digitalWrite(21,HIGH);

digitalWrite(19,LOW);

}

else{digitalWrite(19,LOW);}

}

else{

//tat den, bat quat vao, tat quat ra

if(quatIn==1){

digitalWrite(18,HIGH);

digitalWrite(19,LOW);

digitalWrite(21,LOW);

}

else{

digitalWrite(19,LOW);

digitalWrite(21,LOW);

}

}

}

}

//điều khiển động cơ đảo trứng

void setTimerInterval(){

unsigned long currentMillisTimer = millis();

if(timer==0 || timer != timesetValue ){

timer=timesetValue;

timerInterval= timesetValue\*60000;

}

if(currentMillisTimer - previousMillisTimer >= timerInterval && timer > 0 && dongCo==1){

digitalWrite(22,HIGH);

delay(5000);

digitalWrite(22,LOW);

previousMillisTimer = currentMillisTimer;

}

}

//khởi tạo chuỗi request to server và lấy dữ liệu từ server

String httpGETRequestDevice(const char\* serverNameGetData){

HTTPClient http;

http.begin(serverNameGetData);

int httpResponseCode = http.GET();

String payload = "{}";

while(httpResponseCode <= 0){

http.begin(serverNameGetData);

httpResponseCode = http.GET();

}

if(httpResponseCode > 0){

payload = http.getString();

}

else{

Serial.print("Erorr code: ");

Serial.println(httpResponseCode);

}

http.end();

return payload;

}

void loop() {

}