**TỔNG QUAN CODE PHÒNG SERVER**

* Vì báo cáo có cả code nên khó nhìn, vì vậy, khi vào word chọn View -> Nhìn khung cửa sổ Show -> Tích vào Navigation Pane để hiện thị Mục Lục.

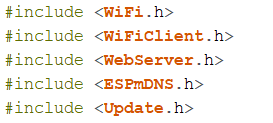
1. **Tổng quan về những module được sử dụng:**

* Email – Arduino.
* DHT11.
* Time – Arduino.
* OTA
* MQTT.
* Firebase.
* Một số linh kiện hỗ trợ khác.

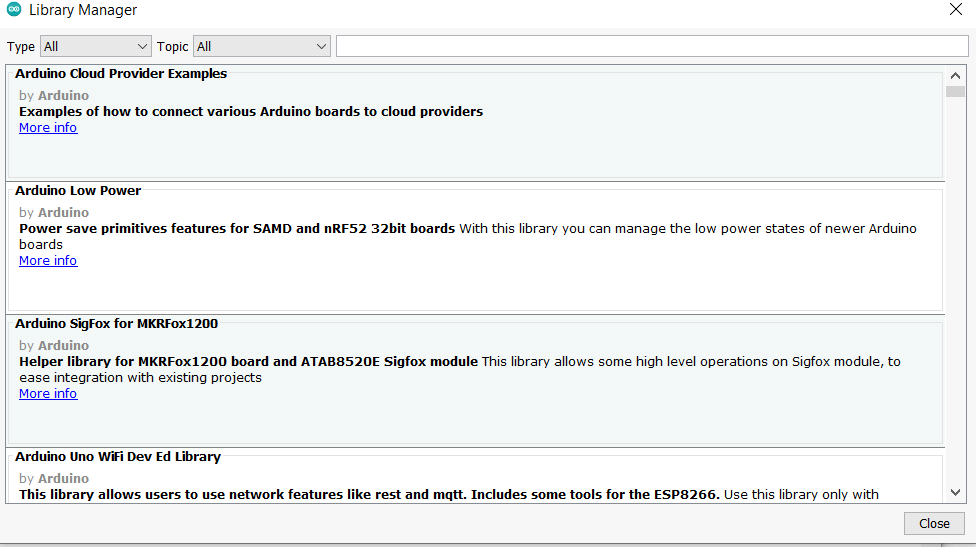
1. **OTA ( Bản cập nhật của Chính):**

## 1. Code nạp OTA

* Khai báo các thư viện cần thiết



Để sử dụng thư viện trước tiên cần phải Install thư viện, ấn **Ctrl + Shift + I,** ở cửa sổ **Library Manager** ta ấn tìm kiếm tên thư viện rồi chọn Install để cài đặt.



* Thêm code phần tạo Web Server:

Code này ta thêm vào phía trước hàm setup(), sau phần khai báo thư viện

WebServer server**(**80**);**

*//Web Server Code*

*/\* Style \*/*

*String style* ***=***

*"<style>#file-input,input{width:100%;height:44px;border-radius:4px;margin:10px auto;font-size:15px}"*

*"input{background:#f1f1f1;border:0;padding:0 15px}body{background:#3498db;font-family:sans-serif;font-size:14px;color:#777}"*

*"#file-input{padding:0;border:1px solid #ddd;line-height:44px;text-align:left;display:block;cursor:pointer}"*

*"#bar,#prgbar{background-color:#f1f1f1;border-radius:10px}#bar{background-color:#3498db;width:0%;height:10px}"*

*"form{background:#fff;max-width:258px;margin:75px auto;padding:30px;border-radius:5px;text-align:center}"*

*".btn{background:#3498db;color:#fff;cursor:pointer}</style>"****;***

*/\* Login page \*/*

*String loginIndex* ***=***

*"<form name=loginForm>"*

*"<h1>ESP32 OTA Update Login </h1>"*

*"<input name=userid placeholder='User ID'> "*

*"<input name=pwd placeholder=Password type=Password> "*

*"<input type=submit onclick=check(this.form) class=btn value=Login></form>"*

*"<script>"*

*"function check(form) {"*

*"if(form.userid.value=='Hyperlogy' && form.pwd.value=='Hyper@123')" //Autheticate user before OTA Update*

*"{window.open('/serverIndex')}"*

*"else"*

*"{alert('Error Password or Username')}"*

*"}"*

*"</script>"* ***+*** *style****;***

*/\* Server Index Page \*/*

*String serverIndex* ***=***

*"<script src='https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js'></script>"*

*"<form method='POST' action='#' enctype='multipart/form-data' id='upload\_form'>"*

*"<input type='file' name='update' id='file' onchange='sub(this)' style=display:none>"*

*"<label id='file-input' for='file'> Choose file...</label>"*

*"<input type='submit' class=btn value='Update'>"*

*"<br><br>"*

*"<div id='prg'></div>"*

*"<br><div id='prgbar'><div id='bar'></div></div><br></form>"*

*"<script>"*

*"function sub(obj){"*

*"var fileName = obj.value.split('\\\\');"*

*"document.getElementById('file-input').innerHTML = ' '+ fileName[fileName.length-1];"*

*"};"*

*"$('form').submit(function(e){"*

*"e.preventDefault();"*

*"var form = $('#upload\_form')[0];"*

*"var data = new FormData(form);"*

*"$.ajax({"*

*"url: '/update',"*

*"type: 'POST',"*

*"data: data,"*

*"contentType: false,"*

*"processData:false,"*

*"xhr: function() {"*

*"var xhr = new window.XMLHttpRequest();"*

*"xhr.upload.addEventListener('progress', function(evt) {"*

*"if (evt.lengthComputable) {"*

*"var per = evt.loaded / evt.total;"*

*"$('#prg').html('progress: ' + Math.round(per\*100) + '%');"*

*"$('#bar').css('width',Math.round(per\*100) + '%');"*

*"}"*

*"}, false);"*

*"return xhr;"*

*"},"*

*"success:function(d, s) {"*

*"console.log('success!') "*

*"},"*

*"error: function (a, b, c) {"*

*"}"*

*"});"*

*"});"*

*"</script>"* ***+*** *style****;***

*//@Web\_Server\_OTA*

* *Thêm phần code để kết nối đến mạng WiFi và để xem địa chỉ IP được cấp*

*Code này đặt cho chạy đầu tiên ở trong hàm setup*

*//@Dang nhap Wifi va nap firmware qua OTA*

*Serial****.****println****(****"Booting"****);***

*WiFi****.****mode****(****WIFI\_STA****);***

*WiFi****.****begin****(****ssid****,*** *password****);***

***while******(****WiFi****.****waitForConnectResult****()******!=*** *WL\_CONNECTED****)***

***{***

*Serial****.****println****(****"Connection Failed! Rebooting..."****);***

*delay****(****2000****);***

*ESP****.****restart****();***

***}***

*Serial****.****println****(****""****);***

*Serial****.****print****(****"Connected to "****);***

*Serial****.****println****(****ssid****);***

*Serial****.****print****(****"IP address: "****);***

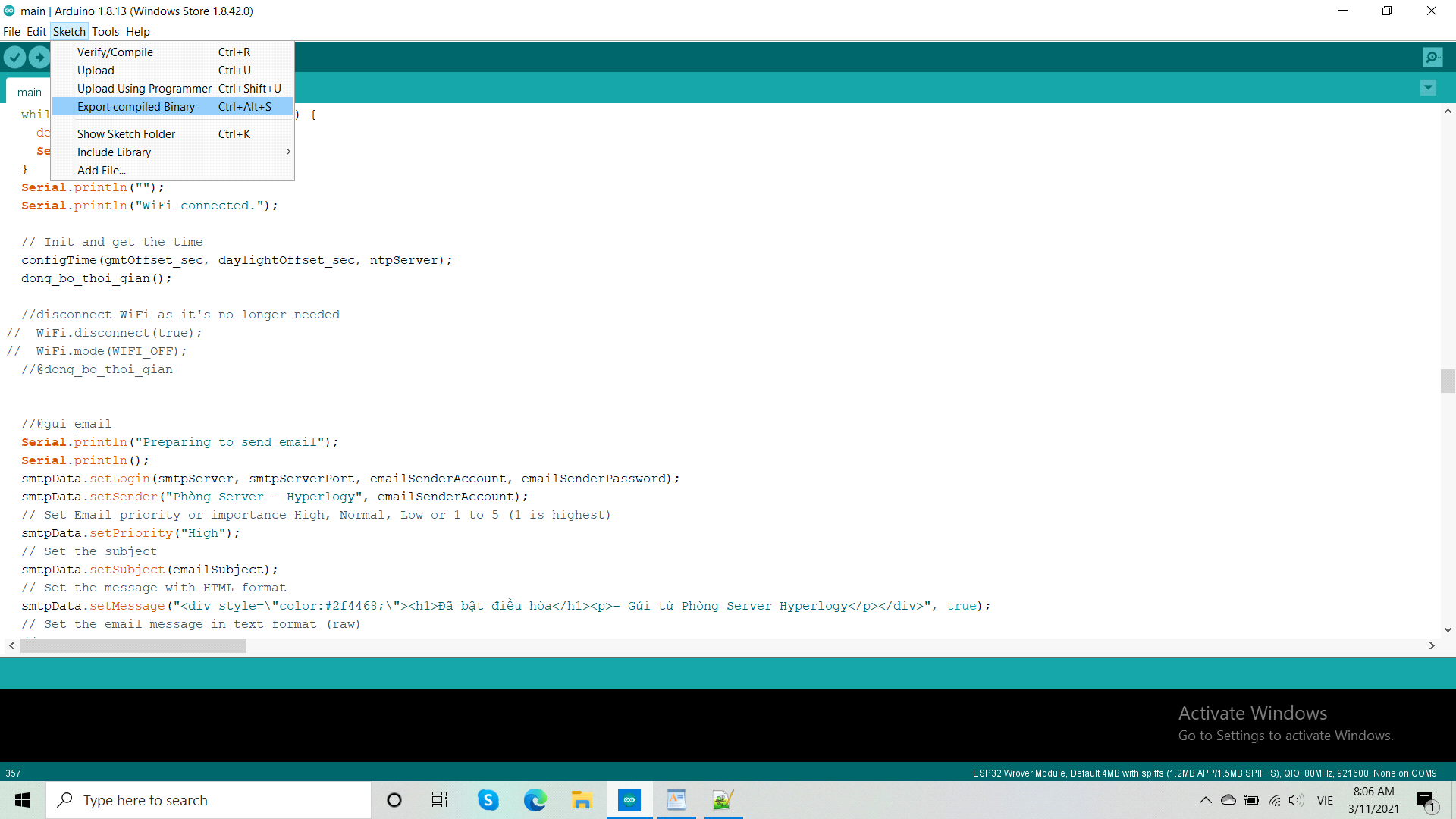
*Serial****.****println****(****WiFi****.****localIP****());***

Cuối cùng đặt dòng code này vào hàm Loop để liên tục kiểm tra server xem có chương trình mới cần nạp vào hay không

*server****.****handleClient****();*** *//Nap FW qua OTA*

*delay****(****1****);***

## 2. Cách tạo file .bin để nạp code

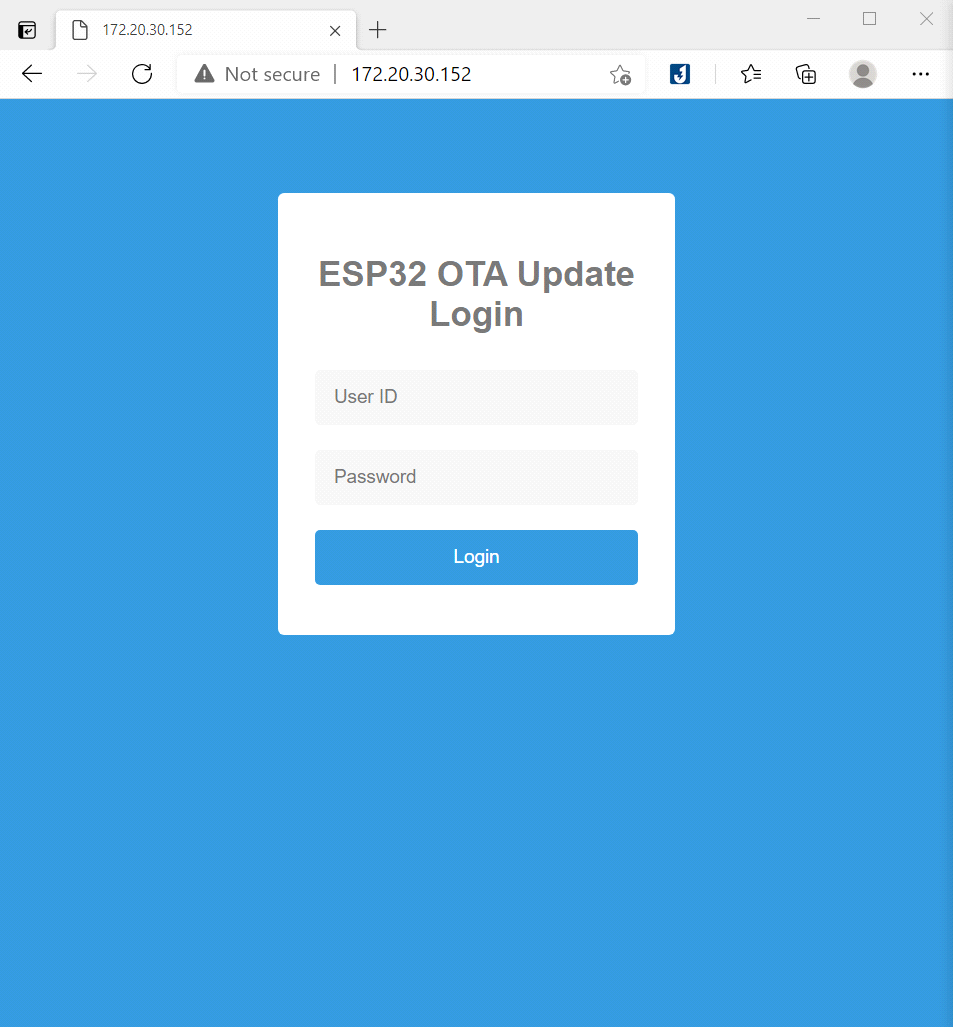


Chọn **Sketch -> Export compiled Binary** hoặc ấn tổ hợp phím **Ctrl+Alt+S**

File .bin sẽ được tạo và lưu ở cùng vị trí với file project arduino

## 3. Cách đăng nhập và nạp code trên Web

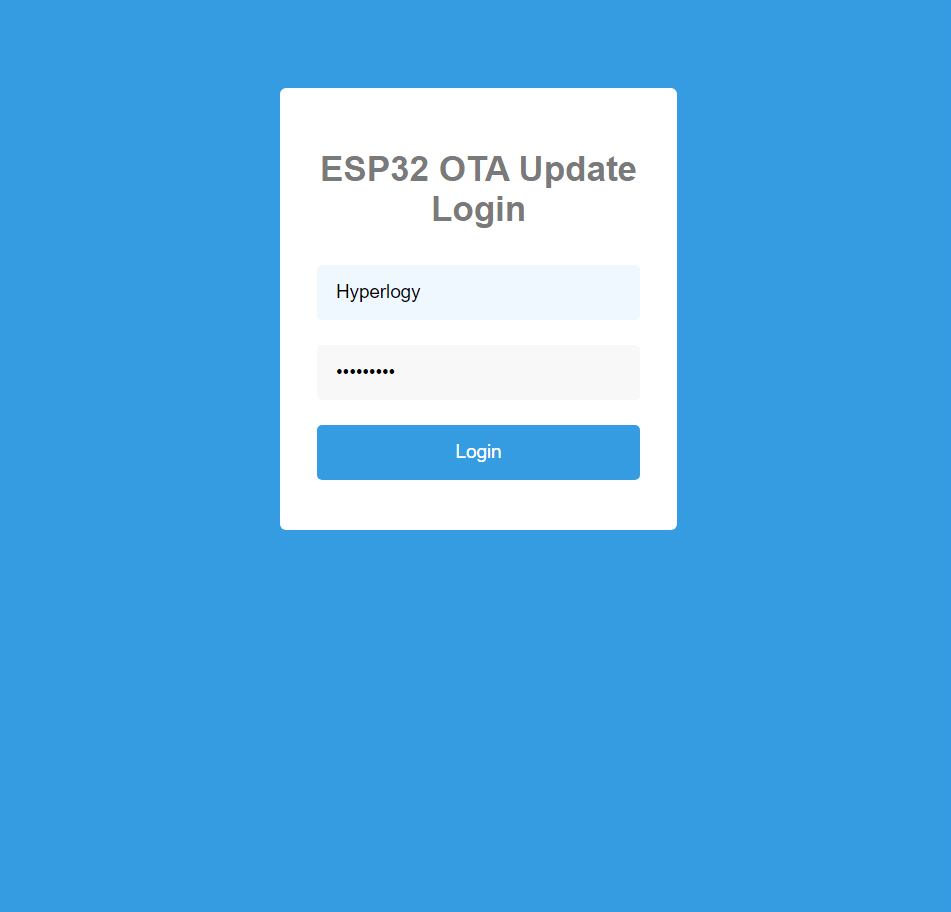
Đầu tiên vào địa chỉ IP: <http://172.20.30.152/>



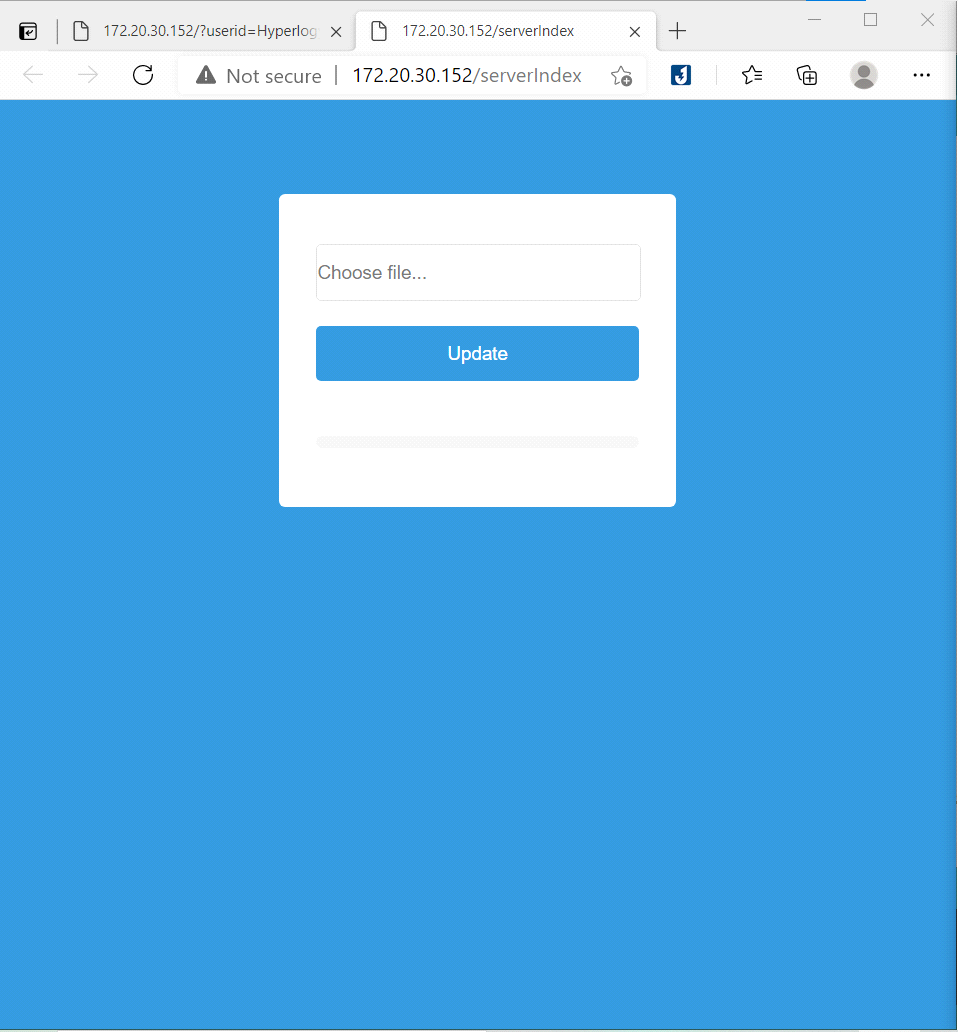
*Giao diện web đăng nhập*

Điền USER ID: Hyperlogy

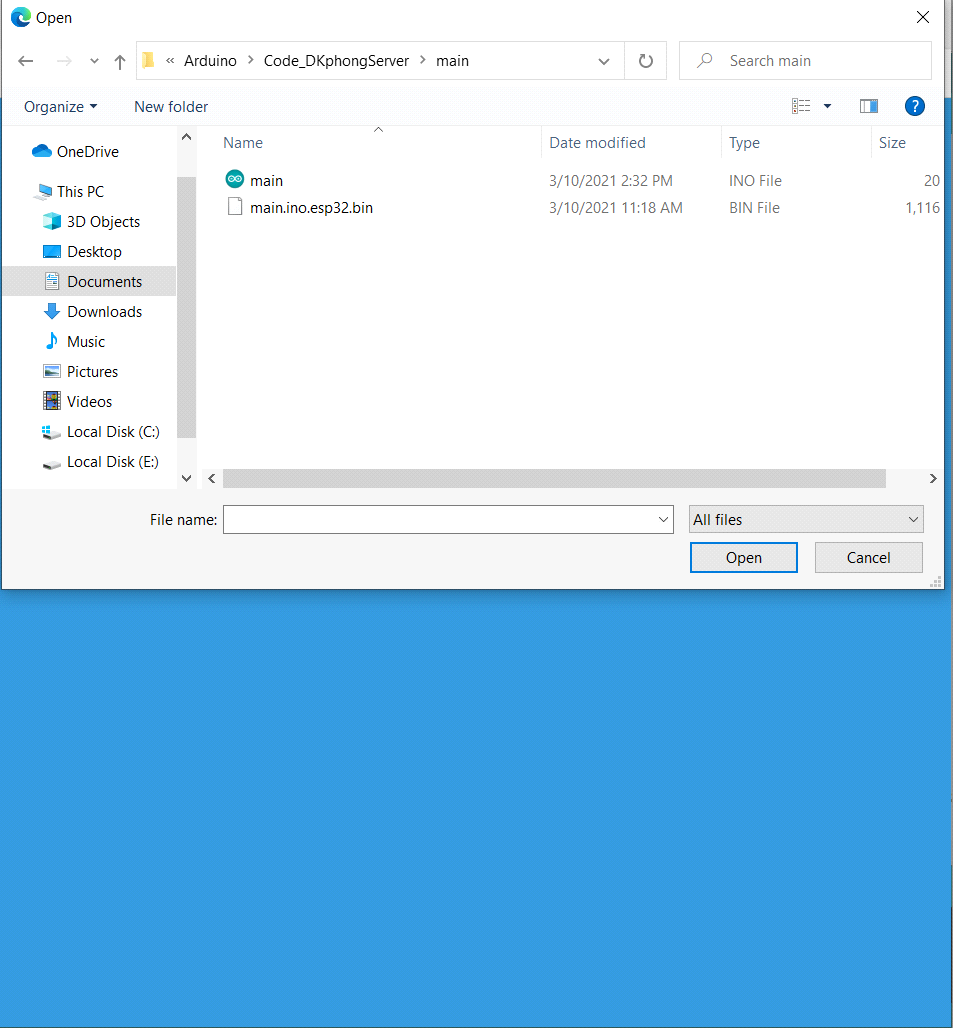
Điền Password: Hyper@123



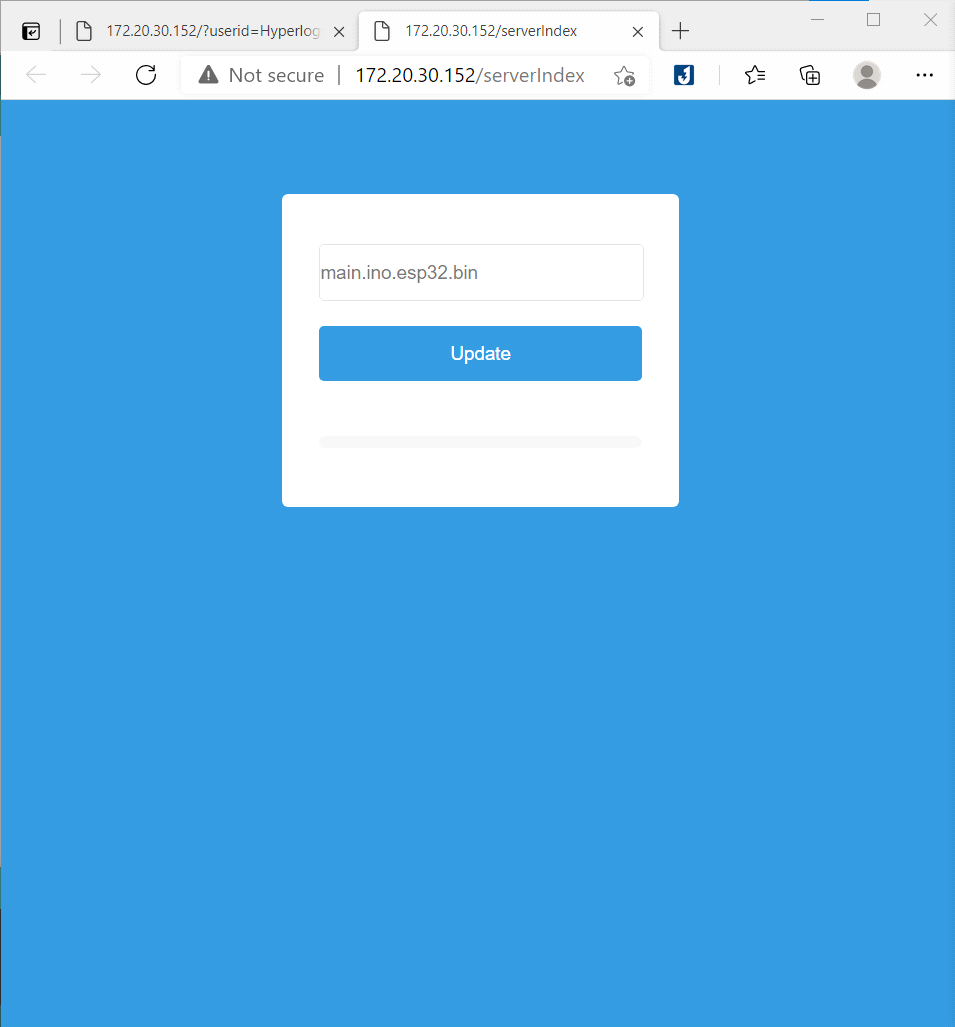
Tại màn hình chọn file ấn **Choose file...**



Chọn file .bin được lưu cùng vị trí với file project arduino



Ấn **Update**



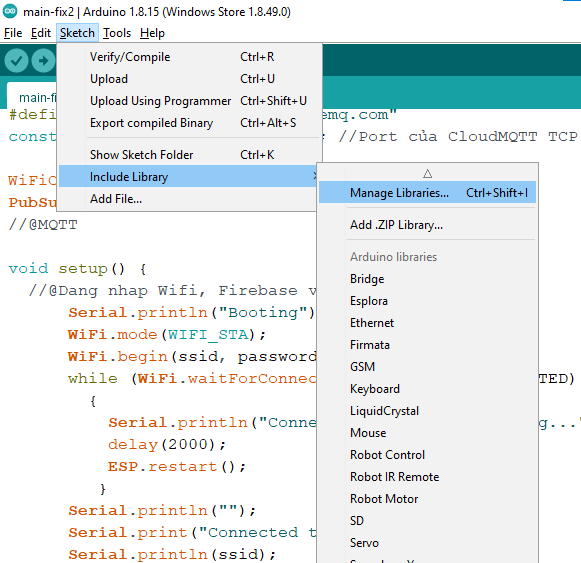
Sau khi nạp xong ấn nút reset trên mạch

1. **DHT11:**

* Mục tiêu: Đọc giá trị nhiệt độ để hiện thị và đưa ra các quyết đinh.

1. **Cài đặt thư viện DHT11:**

* Vào Sketch sau đó chọn Include Library sau đó chọn Manage Libraries … sau đó điền DHT11 và download.



* Khai báo thư viện:

*#include "DHT.h"*

* Các chân sử dụng:

*//@nhiet\_do\_va\_do\_am*

*#define DHTPIN\_in 0 // Digital pin connected to the DHT sensor*

*#define DHTPIN\_out 4 // Digital pin connected to the DHT22*

*#define DHTTYPE DHT11 // DHT 11*

*#define DHTTYPE\_2 DHT11 // DHT 11*

*//@nhiet\_do\_va\_do\_am*

* Config DHT:

*//@nhiet\_do\_va\_do\_am*

*DHT dht\_in(DHTPIN\_in, DHTTYPE);*

*DHT dht\_out(DHTPIN\_out, DHTTYPE\_2);*

*//@nhiet\_do\_va\_do\_am*

*//@nhiet\_do\_va\_do\_am*

*dht\_in.begin(); // DHT11 (1) enable*

*dht\_out.begin(); // DHT11 (2)*

*//@nhiet\_do\_va\_do\_am*

* Đọc nhiệt độ với độ ẩm:

*float dht\_in\_h = dht\_in.readHumidity();*

*float dht\_in\_t = dht\_in.readTemperature();*

*//DHT (out)*

*float dht\_out\_h = dht\_out.readHumidity();*

*float dht\_out\_t = dht\_out.readTemperature();*

1. **Đồng bộ thời gian:**

* Mục tiêu: Đọc thời gian thực.

1. **Khai báo thư viện:**

*#include "time.h"*

1. **Hàm đọc thời gian thực:**

*void dong\_bo\_thoi\_gian()*

*{*

*struct tm timeinfo;*

*if(!getLocalTime(&timeinfo)){*

*Serial.println("Failed to obtain time");*

*return;*

*}*

*Serial.println(&timeinfo, "%A, %B %d %Y %H:%M:%S");*

*Serial.print("Day of week: ");*

*Serial.println(&timeinfo, "%A");*

*Serial.print("Month: ");*

*Serial.println(&timeinfo, "%B");*

*Serial.print("Day of Month: ");*

*Serial.println(&timeinfo, "%d");*

*Serial.print("Year: ");*

*Serial.println(&timeinfo, "%Y");*

*Serial.print("Hour: ");*

*Serial.println(&timeinfo, "%H");*

*Serial.print("Hour (12 hour format): ");*

*Serial.println(&timeinfo, "%I");*

*Serial.print("Minute: ");*

*Serial.println(&timeinfo, "%M");*

*Serial.print("Second: ");*

*Serial.println(&timeinfo, "%S");*

*Serial.println("Time variables");*

*char timeHour[3];*

*strftime(timeHour,3, "%H", &timeinfo);*

*Serial.println(timeHour);*

*char timeWeekDay[10];*

*strftime(timeWeekDay,10, "%A", &timeinfo);*

*Serial.println(timeWeekDay);*

*Serial.println();*

*}*

1. **Email:**

* Mục tiêu: Có thể gửi thông báo dựa trên điều kiện mình mong muốn xảy ra. Ví dụ trước khi bật hoặc tắt điều hòa sẽ có thông báo.
* Dựa trên giao thức SMTP ( Simple Mail Transfer Protocol) là giao thức truyền tải thư tín đơn giản hóa. Giao thức này thực hiện nhiệm vụ chính là gửi gmail. Còn phía nhận Gmail sẽ được các giao thức khác đảm bảo.

1. **Khai báo thư viện:**

*#include "ESP32\_MailClient.h"*

* Khai báo tên cấu hình Email:

*SMTPData smtpData;*

1. **Define tài khoản Email gửi và một số tài khoản nhận email đó.**

*#define emailSenderAccount "cle75736@gmail.com"*

*#define emailSenderPassword "t0lachua"*

*#define emailRecipient "dinhchinh42@gmail.com"*

*#define emailRecipient2 "tung.pham@hyperlogy.com"*

*#define emailRecipient3* [*thang.le@hyperlogy.com*](mailto:thang.le@hyperlogy.com)

* Define server SMTP:

*#define smtpServer "smtp.gmail.com"*

*#define smtpServerPort 465*

*#define emailSubject "[Alert]Phòng Server"*

1. **Cấu hình tên tiêu đề, nội dung tin nhắn:**

* Đăng nhập tài khoản:

*smtpData.setLogin(smtpServer, smtpServerPort, emailSenderAccount, emailSenderPassword);*

*smtpData.setSender("Phòng Server - Hyperlogy", emailSenderAccount);*

*smtpData.setPriority("High");*

* Tên tiêu đề cần gửi:

*smtpData.setSubject(emailSubject);*

* Nội dung tin nhắn: Được cấu hình sử dụng HTML vì vậy nội dung sẽ trong thẻ h của HTML, phần Style là định dạng chữ gửi.

*smtpData.setMessage("<div style=\"color:#2f4468;\"><h1>Đã bật điều hòa, temp > 35</h1><p>- Gửi từ Phòng Server Hyperlogy</p></div>", true);*

*smtpData.addRecipient(emailRecipient);*

*smtpData.setSendCallback(sendCallback);*

1. **Cách sử dụng:**

*MailClient.sendMail(smtpData);*

* Do đã cấu hình hết trong smtpData vì vậy chỉ cần send cấu hình.
* Nếu như muốn gửi 2 Gmail khác nhau:

+ Cách 1: Hãy thêm phần tên Gmail khác và cấu hình như trên nhưng với nội dung mình mong muốn.

+ Cách 2: Sử dụng 2 mảng String với 2 nội dung khác nhau. Sau đó khi nào cần tin nhắn như thế nào thì cấu hình lại hàm *smtpData.setMessage* là xong.

1. **Firebase:**

* Firebase là một dịch vụ cơ sở dữ liệu thời gian thực hoạt động trên nền tảng cloud được cung cấp bởi Google nhằm giúp phát triển nhanh các ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu.
* Link tham khảo: https://hocarm.org/esp8266-va-firebase/

1. **Khai báo thư viện:**

*#include <FirebaseESP32.h>*

1. **Link tài khoản host nhận dữ liệu:**

*#define FIREBASE\_HOST "https://esp32-device-default-rtdb.firebaseio.com/"*

*#define FIREBASE\_AUTH "vfRpihYU4RzT8Ge6wLAh3MPEFFmr6KgP5uNOm7Wf"*

FirebaseData firebaseData;

1. **Cài đặt phần Setup:**

*Firebase.begin(FIREBASE\_HOST, FIREBASE\_AUTH);*

*Firebase.reconnectWiFi(true);*

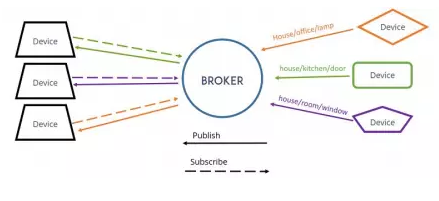
1. **Cách gửi dữ liệu lên:**

*Firebase.setDouble(firebaseData, path + "nhietdo",dht\_in\_t);*

*Firebase.setDouble(firebaseData, path + "doam",dht\_in\_h);*

1. **MQTT:**
2. **MQTT ( Message Queue Telemetry Transport):**

* là giao thức truyền thông điệp (message) theo mô hình publish/subscribe được sử dụng cho các thiết bị IoT với băng thông thấp, độ tin cậy cao và khả năng được sử dụng trong mạng lưới không ổn định. Nó dựa trên một Broker (tạm dịch là “Máy chủ môi giới”) “nhẹ” (khá ít xử lý) và được thiết kế có tính mở (tức là không đặc trưng cho ứng dụng cụ thể nào), đơn giản và dễ cài đặt.
* Mô hình giao thức MQTT publish và subcribe bản tin:



1. **Khai báo thư viện kết nối ESP8266 và Client:**

*#include <ESP8266WiFi.h>*

*#include <PubSubClient.h>*

1. **Khai báo MQTT\_Server và Port thu nhập dữ liệu:**

* Chú ý: Khi sử dụng các server free thì phải để ý port cài đặt của nó.

*#define mqtt\_server "broker.hivemq.com"*

*const uint16\_t mqtt\_port = 1883; //Port của CloudMQTT TCP*

1. **Khai báo Wifi và Client:**

*WiFiClient espClient;*

*PubSubClient client(espClient);*

1. **Setup wifi:**

*void setup\_wifi()*

*{*

*delay(10);*

*Serial.println();*

*Serial.print("Connecting to ");*

*Serial.println(ssid);*

*WiFi.begin(ssid, password);*

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

*delay(500);*

*Serial.print(".");*

*}*

*Serial.println("");*

*Serial.println("WiFi connected");*

*Serial.println("IP address: ");*

*Serial.println(WiFi.localIP());*

*}*

1. **Tạo hàm callback để hiện thị dữ liệu sau khi subcribe về:**

*void callback(char\* topic, byte\* payload, unsigned int length)*

*{*

*Serial.print("Co tin nhan moi tu topic:");*

*Serial.println(topic);*

*for (int i = 0; i < length; i++)*

*Serial.print((char)payload[i]);*

*Serial.println();*

*}*

1. **Viết hàm reconnect để đăng ký kênh:**

*void reconnect()*

*{*

*while (!client.connected()) // Chờ tới khi kết nối*

*{*

*// Thực hiện kết nối với mqtt user và pass*

*if (client.connect("ESP8266\_id1","ESP\_offline",0,0,"ESP8266\_id1\_offline")) //kết nối vào broker*

*{*

*Serial.println("Đã kết nối:");*

*client.subscribe("ESP8266\_read ");*

*}*

*else*

*{*

*Serial.print("Lỗi:, rc=");*

*Serial.print(client.state());*

*Serial.println(" try again in 5 seconds");*

*// Đợi 5s*

*delay(5000);*

*}*

*}*

*}*

1. **Cài đặt setup:**

*Void setup(){*

*setup\_wifi();*

*client.setServer(mqtt\_server, mqtt\_port);*

*client.setCallback(callback);*

*}*

1. **Cách gửi bản tin tròng vòng loop:**

* Kiểm tra kết nối:

*if (!client.connected())// Kiểm tra kết nối*

*reconnect();*

*client.loop();*

* Cách gửi dữ liệu publish:

+ Toan123456 là kênh gửi lên.

+ inputString là topic chứa dữ liệu gửi lên kênh.

*char inputString[6]="Hello";*

*client.publish("Toan123456", inputString);*

1. **Thêm: Gửi bản tin JSON.**

* Vì sẽ có 2 dữ liệu sẽ được gửi lên vì vậy nên thiết kế bản tin JSON cho tiện vẽ biểu đồ để theo dõi.
* Code mẫu:

+ Cấu trúc bản tin JSON buffer:

*StaticJsonDocument<200> doc;*

*doc["Temp\_in"] = tmp\_in;*

*doc["Temp\_out"] = tmp\_out;*

*char buffer[256];*

*size\_t n = serializeJson(doc, buffer);*

+ Gửi bản tin JSON lên với topic

*client.publish("Toan123456", buffer, n);*

*serializeJsonPretty(doc, Serial)*

1. **Phần hiện thị Web hỗ trợ cho MQTT:**
2. **Script Subcribe dữ liệu:**

*<script type = "text/javascript" language = "javascript">*

*var max,at\_OK;*

*function makeid()*

*{*

*var text = "";*

*var possible = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";*

*for (var i = 0; i < 5; i++)*

*text += possible.charAt(Math.floor(Math.random() \* possible.length));*

*return text;*

*}*

*// Create a client instance*

*var client = new Paho.MQTT.Client("broker.hivemq.com", 8000, makeid());*

*// set callback handlers*

*client.onConnectionLost = onConnectionLost;*

*client.onMessageArrived = onMessageArrived;*

*var options = {*

*useSSL: false,*

*userName: "",*

*password: "",*

*onSuccess:onConnect,*

*onFailure:doFail*

*}*

*console.log("Connect to broker.hivemq.com:8000");*

*// connect the client*

*client.connect(options);*

*function doFail(e){*

*console.log(e);*

*}*

*function onConnect() //sự kiên kết nối thành công*

*{*

*console.log("Connect OK");*

*client.subscribe("Toan123456"); //đăng kí kênh*

*}*

*// called when the client loses its connection*

*function onConnectionLost(responseObject)*

*{*

*if (responseObject.errorCode !== 0)*

*{*

*console.log(responseObject.errorMessage);*

*}*

*}*

*// called when a message arrives*

*function onMessageArrived(message)*

*{*

*console.log(message.destinationName + ":" +message.payloadString);*

*document.getElementById("tinnhan").innerHTML = "Tin nhắn từ esp8266: " + message.payloadString;*

*}*

*function public (topic,data)*

*{*

*message = new Paho.MQTT.Message(data);*

*message.destinationName = topic;*

*client.send(message);*

*}*

*</script>*

1. **Function Chart:**

* Hàm sử dụng với dữ liệu dạng JSON:

*function drawGraph(data\_cus) {*

*var chart = new CanvasJS.Chart("chartContainer", {*

*animationEnabled: false,*

*theme: "light2",*

*title:{*

*text: "Simple Line Chart"*

*},*

*data: [{*

*type: "line",*

*indexLabelFontSize: 16,*

*dataPoints: data\_cus,*

*}]*

*});*

*chart.render();*

*}*

*window.onload = function () {*

*drawGraph([{y:1}]);*

*}*

* Cách sử dụng hàm với dữ liệu:

*data = []*

*data1 = []*

*function onMessageArrived(message)*

*{*

*console.log(message.destinationName + ":" +message.payloadString)*

*data.push({y: JSON.parse(message.payloadString)['Temp\_in']})*

*drawGraph(data)*

*data1.push({y: JSON.parse(message.payloadString)['Temp\_out']})*

*drawGraph1(data1)*

*}*

* Thêm phần chart vào phần body:

*<div id="chartContainer" style="height: 300px; width: 60%;"></div>*

*<div id="chartContainer1" style="height: 300px; width: 60%;"></div>*

1. **Phần các linh kiện được sử dụng thêm:**
2. **IR:**

* Mục tiêu: Điều khiển các thiết bị sử dụng hồng ngoại: điều hòa.
* Khai báo thư viện:

*#include <IRremoteESP8266.h>*

*#include <IRsend.h>*

*irsend.begin(); //@ir\_dieu\_khien*

* Điều khiển IR: 2 trường hợp bật ( state = 1) hoặc tắt ( state = 0 )

*void bat\_tat\_dieu\_hoa(bool state)*

*{*

*if(state == 1 )*

*{*

*irsend.sendRaw(rawData, 199, 38);*

*digitalWrite(LED\_state\_air, HIGH);*

*Serial.println("Air on");*

*}*

*if(state == 0)*

*{*

*irsend.sendRaw(off\_air, 199, 38);*

*Serial.println("Air off");*

*digitalWrite(LED\_state\_air, LOW);*

*}*

*}*

1. **Cửa ra vào:**

* Kiểm tra trạng thái của cửa: Nếu là LOW thì là đóng, HIGH là mở.

*void trang\_thai\_cua\_ra\_vao()*

*{*

*if(digitalRead(Door\_sensor) == LOW)*

*{*

*Serial.println(F("Cua CLOSED"));*

*}*

*else*

*{*

*Serial.println(F("Cua OPEN"));*

*}*

*}*

* Điều khiển cửa:

*void Cua\_trang\_thai(bool state)*

*{*

*if(state == 1) // Dong Cua*

*{*

*digitalWrite(Cua\_dieu\_khien, HIGH);*

*digitalWrite(LED\_state\_door, LOW);*

*}*

*else // Mo Cua*

*{*

*digitalWrite(Cua\_dieu\_khien, LOW);*

*digitalWrite(LED\_state\_door, HIGH); }*

*}*

1. **Điều khiển quạt:**

*void Quat\_dieu\_khien(bool state)*

*{*

*if(state == 1)*

*{*

*digitalWrite(Fan\_control, HIGH);*

*digitalWrite(LED\_state\_fan, LOW);*

*Serial.println(F("Quat ON"));*

*}*

*else*

*{*

*digitalWrite(Fan\_control, LOW);*

*digitalWrite(LED\_state\_fan, HIGH);*

*Serial.println(F("Quat OFF"));*

*}*

*}*

1. **MAIN CODE:**
2. **Đọc nhiệt độ trong và ngoài:**

*float dht\_in\_t = dht\_in.readTemperature(); // DHT in*

*float dht\_out\_t = dht\_out.readTemperature();//DHT (out)*

1. **Từ dữ liệu nhiệt độ trong và ngoài đưa ra các quyết định:**

\_ Trạng thái 1: State = 0 ( Trạng thái nóng – cần bật điều hòa ):

* Nếu nhiệt độ trong > 37 hoặc nhiệt độ ngoài >34 thì sẽ gửi Gmail.
* Nếu nhiệt độ trong > 45 hoặc nhiệt độ ngoài > 45 thì bật điều hòa, tắt quạt, đóng cửa và chuyển sang trạng thái state = 1.
* Code mẫu:

*if (state == 0)*

*{*

*if (dht\_in\_t > th\_up || dht\_out\_t > th\_dwn)*

*{*

*MailClient.sendMail(smtpData);*

*}*

*if (dht\_in\_t > 45 || dht\_out\_t > 45)*

*{*

*cnt++;*

*if(dieu\_hoa == false)*

*{*

*bat\_tat\_dieu\_hoa(1);*

*state = 1;*

*//delay(10000);*

*dieu\_hoa = true;*

*Serial.println("Dieu hoa bat: ");*

*Serial.println(dieu\_hoa);*

*}*

*if(cnt > 40)*

*{*

*cnt = 0;*

*Quat\_dieu\_khien(0);*

*}*

*Cua\_trang\_thai(1); // đóng*

*}*

*}*

\_ Trạng thái 2: State = 1 ( Trạng thái mát – tắt điều hòa ):

* Nếu nhiệt độ trong < 34 và nhiệt độ ngoài < 31 thì tắt điều hòa, bật quạt, mở cửa và chuyển sang trạng thái state = 0.
* Code mẫu:

*if (state == 1)*

*{*

*if (dht\_in\_t < th\_up && dht\_out\_t < th\_dwn)*

*{*

*Serial.println("Check: ");*

*Serial.println(dieu\_hoa);*

*if(dieu\_hoa == true)*

*{*

*for(int i = 0; i< 2; i++)*

*{*

*bat\_tat\_dieu\_hoa(0);*

*state = 0;*

*}*

*dieu\_hoa = false;*

*Serial.println("Dieu hoa tat: ");*

*Serial.println(dieu\_hoa);*

*}*

*Quat\_dieu\_khien(1);*

*Cua\_trang\_thai(0);*

*}*

*}*