

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KỸ THUẬT MÁY TÍNH**



**THIẾT KẾ HỆ THỐNG NHÚNG KHÔNG DÂY
CE232.021**

**BÁO CÁO TIẾN ĐỘ GIỮA KỲ
SMART HOME**

NHÓM 9 - LỚP 2

Hoàng Phan Thành Bách - 21520599

Dương Mai Quý - 21521362

Nguyễn Đức Lưu - 21522314

Kiều Thị Hiền Trinh - 21522715

Giảng viên hướng dẫn

Ths. Trần Hoàng Lộc

Tp. Hồ Chí Minh, ngày 08 tháng 5 năm 2024

MỤC LỤC

I.	TỔNG QUAN ĐỀ TÀI.....	2
II.	KẾ HOẠCH THỰC HIỆN ĐỒ ÁN.....	3
	1. Xây dựng Block Diagram.....	3
	2. Xây dựng Kịch bản kiểm thử	7
	3. Thiết kế và chạy kiểm thử	11
III.	ĐÁNH GIÁ TIẾN ĐỘ, NHẬN XÉT.....	19

I. TỔNG QUAN ĐỀ TÀI

- SMART HOME - ngôi nhà thông minh không còn là chủ đề xa lạ đối với lĩnh vực nghiên cứu phát triển hệ thống hệ thống nhúng & IOT trong thời gian gần đây. Một ngôi nhà được trang bị các thiết bị hiện đại sẽ cần có một hệ thống quản lý thông minh để giúp người tiêu dùng dễ dàng kiểm soát chúng.
- Đề tài “SMART HOME” được thực hiện bởi nhóm 9 với mục đích nghiên cứu, phát triển một hệ thống ngôi nhà thông minh căn bản với khả năng quản lý các thiết bị điện tử chỉ bằng một chạm trên điện thoại hay máy tính cá nhân.

Mục tiêu:

MỤC TIÊU NGHIÊN CỨU	GIÁ TRỊ THỰC TIỄN, GIỚI HẠN THỰC HIỆN
<p>Đồ án “SMART HOME” không chỉ giúp nhóm tìm hiểu sâu hơn về vi điều khiển ESP32, những giao thức của chúng mà còn là cơ hội thực hành trực tiếp và chi tiết dựa trên các chức năng, yêu cầu đề ra cho hệ thống nhà thông minh.</p> <p>Vận dụng các kiến thức đã học trong môn CE232 để hoàn thành đồ án. Thiết kế giao diện người dùng thân thiện, dễ sử dụng. Xây dựng CSDL có thể cập nhật trạng thái thiết bị liên tục.</p>	<p>“SMART HOME” là một đề tài có tính thực tiễn cao áp dụng trong các mô hình ngôi nhà thông minh. Đề tài nhằm tạo ra một hệ thống dễ sử dụng mà người dùng có thể kiểm soát thiết bị trong nhà từ xa hiệu quả, nhanh chóng.</p> <p>Do tính chất dừng lại ở đồ án môn học, đồ án “SMART HOME” tồn tại một số giới hạn như việc hạn chế về thiết bị, công nghệ để quản lý hệ thống chặt chẽ, chính xác. Ngoài ra, giao diện người dùng chỉ hoàn thành ở mức cơ bản nhằm giúp người sử dụng điều khiển tiện lợi mà chưa tối ưu được trải nghiệm người dùng.</p> <p>Vì thế, đồ án thực hiện trên 1 đèn, 1 quạt, 1 cửa và 1 cảm biến. Với đèn có thể hẹn giờ hay cửa có thể mở từ xa tối đa 5m. Trong thực tế, ngôi nhà thông minh sẽ cần số lượng thiết bị nhiều hơn rất nhiều so với con số kể trên đồng thời yêu cầu quy mô kết nối rộng hơn.</p>

Bảng 1.1. Mục tiêu nghiên cứu đề tài SMARTHOME

Tổng quan chức năng: (Dựa trên yêu cầu đề tài trong đồ án)

Thiết bị	Yêu cầu
Cửa	Khóa/mở cửa từ xa trong phạm vi 5 - 10m, cửa khóa/ mở delay không quá 2s. Cho phép người dùng khóa/mở cửa từ xa thông qua ứng dụng di động hoặc web. Ghi nhận trạng thái của cửa vào Cơ sở dữ liệu (CSDL). Hiển thị trạng thái này trên thiết bị người dùng.
Đèn	Bật/tắt đèn từ xa (<5m) thông qua ứng dụng di động hoặc web. Hẹn giờ tự động bật/tắt đèn (Nhận lệnh và thực thi trong vòng 5 giây) Ghi nhận trạng thái của đèn vào CSDL. Hiển thị trạng thái này trên thiết bị người dùng.
Quạt	Bật/tắt quạt từ xa (<10m) Điều chỉnh tốc độ quạt. (Nhẹ/ Vừa/ Mạnh) Ghi nhận trạng thái của quạt vào CSDL. Hiển thị trạng thái này trên thiết bị người dùng.
Cảm biến nhiệt	Ghi nhận dữ liệu về nhiệt độ từ cảm biến độ nhiệt vào CSDL hệ thống. (Hiển thị chính xác độ trễ <1s, hiển thị đơn vị tới số thứ hai sau dấu phẩy) Hiển thị biểu đồ độ nhiệt trên giao diện người dùng.
Giao diện người dùng:	Hiện thực giao diện trên web app hoặc mobile app để người dùng có thể quản lý và kiểm soát các thiết bị. Cung cấp chức năng đăng nhập, hiển thị trạng thái của các thiết bị, thêm/xóa thiết bị và xem biểu đồ độ nhiệt.

Bảng 1.2. Xác định các yêu cầu của đồ án SMARTHOME

II. KẾ HOẠCH THỰC HIỆN ĐỒ ÁN

1. Xây dựng Block Diagram

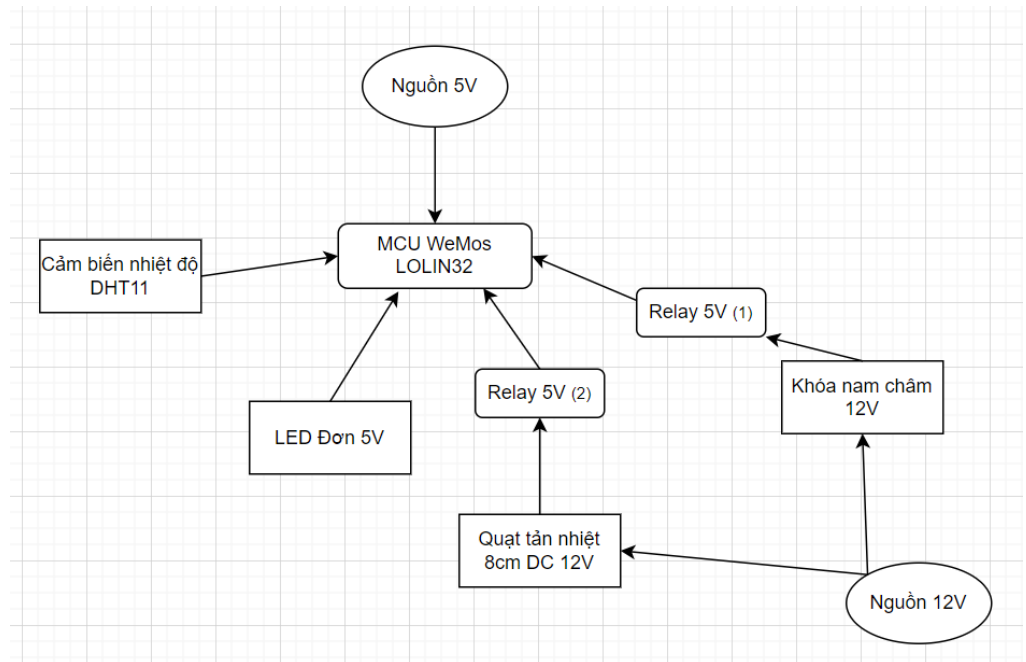
+ Lý thuyết:

- Block Diagram(sơ đồ khối) là một loại biểu đồ được sử dụng để mô tả cấu trúc và cấu hình của một hệ thống hoặc quy trình bằng các khối và liên kết giữa chúng.
- Trong một block diagram, các thành phần của hệ thống được biểu diễn dưới dạng các hình khối, và các liên kết giữa chúng được thể hiện bằng các đường

nổi.

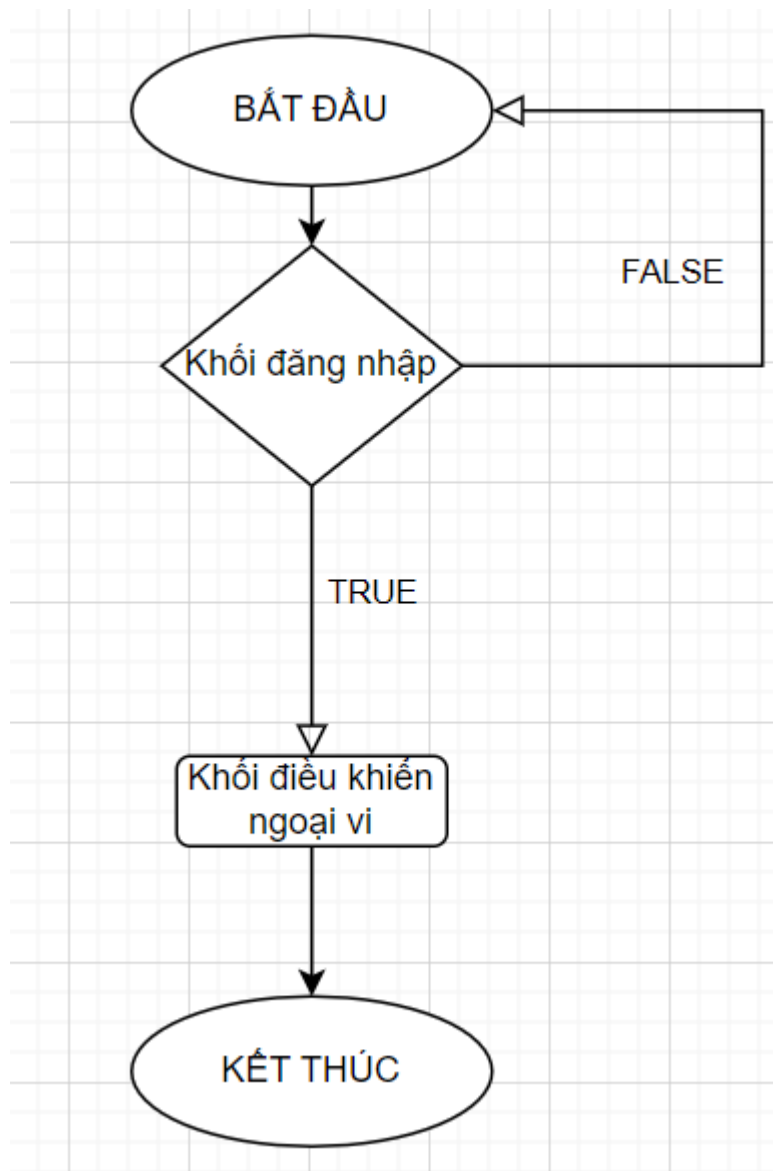
+ **Thực hiện:**

- Sau khi tìm hiểu về các dạng block diagram, nhóm thực hiện vẽ sơ đồ cho hệ thống của mình như sau. Đây là sơ đồ tổng thể cho hệ thống ngôi nhà thông minh của nhóm, tuy nhiên các thiết bị hay chức năng có thể được thay đổi trong quá trình phát triển. Trong phần này, nhóm sẽ trình bày tóm tắt thông qua sơ đồ phần cứng và lưu đồ giải thuật phần mềm như sau:
- Sơ đồ phần cứng:



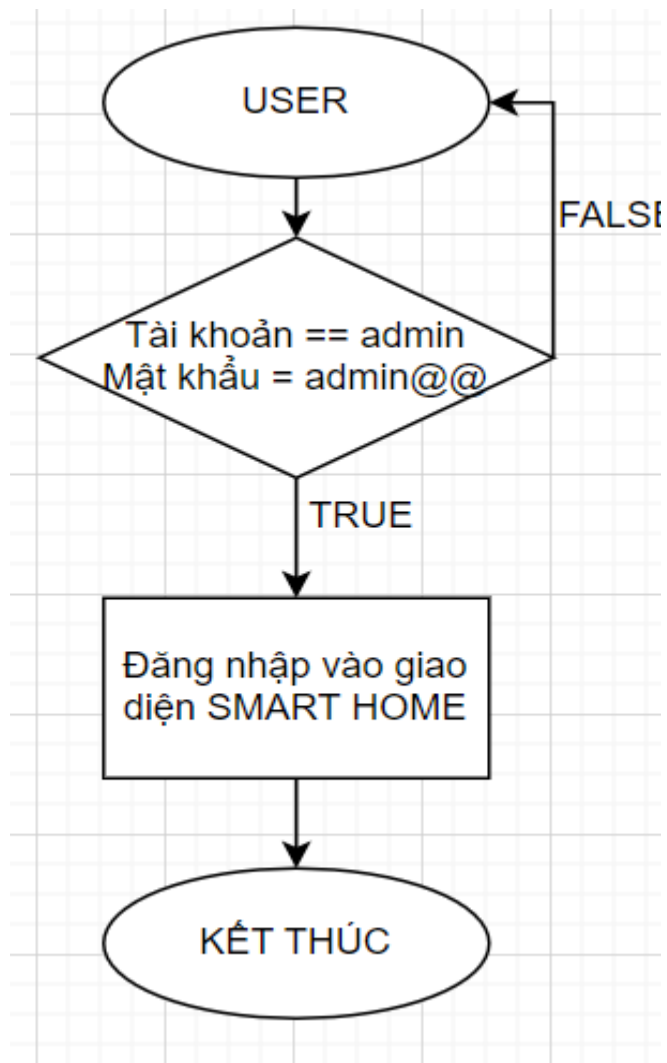
Hình 1. Sơ đồ phần cứng

- Lưu đồ giải thuật phần mềm chương trình chính:



Hình 2. Lưu đồ giải thuật phần mềm chính

- Lưu đồ giải thuật khối đăng nhập:



Hình 3. Lưu đồ giải thuật khối đăng nhập

- Lưu đồ giải thuật khối điều khiển ngoại vi:



Hình 4. Lưu đồ giải thuật khối ngoại vi

2. Xây dựng Kịch bản kiểm thử

+ Lý thuyết:

Kịch bản kiểm thử là một tài liệu mô tả các bước cụ thể mà một nhóm sẽ thực hiện để kiểm tra tính năng, hiệu suất hoặc các khía cạnh khác của một sản phẩm, ứng dụng hoặc hệ thống. Mục tiêu của kịch bản kiểm thử là đảm bảo rằng sản phẩm hoạt động như mong đợi và đáp ứng được các yêu cầu đã đề ra.

+ **Các bước xây dựng bản kiểm thử:**

Bước thực hiện	Nội dung thực hiện
Xác định Mục tiêu Kiểm thử	<p>Mục tiêu của bước này nhằm xác định các yếu tố cần phải kiểm tra kỹ sau khi thiết kế hoàn tất. Nhóm đã đề xuất ra các mục tiêu cần đạt được dựa trên yêu cầu khi kiểm thử bao gồm:</p> <ul style="list-style-type: none"> + Chức năng: Bảo đảm thiết bị hoạt động đúng với chức năng đề ra. (Quạt quay, đèn sáng hay cửa có thể bật tắt) + Hiệu suất: Tốc độ phản hồi của các thiết bị, khả năng kết nối và khả năng mở rộng mô hình (Tốc độ phản hồi nhanh chóng, kết nối trong khoảng cách phù hợp với nhà thông minh). + Độ tin cậy: Kiểm tra khả năng hoạt động lâu dài của thiết bị. Ngoài ra khả năng tích hợp cũng cần được chú ý cho các mô hình lớn hơn, mở rộng sang sân vườn, ao nước...
Phân tích yêu cầu và tài liệu thiết kế	Dựa trên mô tả đồ án được cung cấp và Block Diagram đã xây dựng, nhóm sắp xếp độ ưu tiên của các mục tiêu kiểm thử. Đồng thời hiểu rõ về đề tài nhằm tránh đưa ra các kịch bản kiểm thử quá khả năng với mô hình
Xác định kịch bản kiểm thử	Xác định các trường hợp cần kiểm thử, từ đó chia thành các kịch bản nhỏ phù hợp cho từng khía cạnh cụ thể của sản phẩm

Viết kịch bản kiểm thử	<p>Vạch ra các dữ liệu đầu vào, kết quả mong muốn.</p> <p>+ Đèn:</p> <p>Test 1: Với nguồn 5V, Led đơn có thể sáng tốt và bật tắt theo yêu cầu của người sử dụng.</p> <p>Test 2: Hẹn giờ tắt mở trong ngày (Có thể mở vào 18h00 pm, tắt vào 6h00 am).</p> <p>Test 3: Hiển thị trạng thái trên giao diện người dùng và cập nhật trên CSDL.</p> <p>+ Cửa:</p> <p>Test 1: Cửa thông minh có thể đóng/mở từ xa (tối đa 5m).</p> <p>Test 2: Hiển thị trạng thái trên giao diện người dùng và cập nhật trên CSDL.</p> <p>+ Cảm biến nhiệt độ/ độ ẩm:</p> <p>Test 1: Cảm biến có thể gửi thông tin đo được về máy đồng thời hiển thị trên giao diện người dùng.</p> <p>Test 2: Cảm biến hiển thị kết quả theo dạng biểu đồ trên giao diện người dùng</p> <p>+ Quạt:</p> <p>Test 1: Bật tắt theo yêu cầu người dùng</p> <p>Test 2: Hiển thị trạng thái trên giao diện người dùng và cập nhật trên CSDL</p>
Tiến hành kiểm thử	Từ các kịch bản đã xây dựng, nhóm tiến hành kiểm thử và so sánh kết quả mong muốn với kết quả thực tế.

Bảng 2.2. Xác định các bước thực hiện bản kiểm thử đề tài SMART HOME

Từ việc xác định các bước thực hiện bản kiểm thử, đầu tiên nhóm đề ra những yêu cầu kiểm thử như sau:

Kịch bản kiểm thử	Chi tiết công việc
-------------------	--------------------

1/ Kiểm tra hoạt động của các linh kiện	<ul style="list-style-type: none"> - Kiểm thử hoạt động của đèn LED đơn với hiệu điện thế 3,3V và 5V - Kiểm thử hoạt động thu thập nhiệt độ của cảm biến - Kiểm thử hoạt động của Khóa nam châm 12V với Relay 5V. - Kiểm thử hoạt động của quạt - Kiểm thử khả năng chạy của cả hệ thống
2/ Kiểm tra đèn với các yêu cầu	<ul style="list-style-type: none"> - Kiểm tra đèn bật/tắt với hiệu điện thế 3,3V. - Kiểm tra khả năng tăng giảm cường độ sáng của đèn - Kiểm tra hoạt động hẹn giờ bật/tắt của đèn (Đèn tự động bật/ tắt khi tới khung giờ thiết lập) (Dự kiến: 6h sáng tắt – 6h tối bật)
3/ Kiểm tra cảm biến với các yêu cầu	<ul style="list-style-type: none"> - Kiểm tra hoạt động thu thập của cảm biến, so sánh giá trị với các thiết bị di động, điện tử có tính năng tương tự. - Kiểm tra khả năng nhận và cập nhật dữ liệu (thấp hơn 2 giây)
4/ Kiểm tra cửa khóa nam châm, Relay	<ul style="list-style-type: none"> - Kiểm tra hoạt động của Relay 5V, cửa bật/ tắt với từng khoảng cách (tối đa 5m)
5/ Kiểm tra quạt	<ul style="list-style-type: none"> - Kiểm tra hoạt động tắt mở/ quạt với Relay 5V

6/ Kiểm tra trạng thái người dùng trên CSDL	- Kiểm tra trạng thái của đèn, cửa, cảm biến và quạt đã được gửi lên hay chưa
7/ Kiểm tra giao diện người dùng trên Web/ App	<ul style="list-style-type: none"> - Kiểm tra giao diện, hoạt động của Web/App, kiểm tra tương tác của người dùng với hệ thống thông qua Web/App. - Kiểm tra chức năng đăng nhập, hiển thị nhiệt độ theo biểu đồ

Từ các yêu cầu kiểm thử, nhóm thực hiện viết kịch bản kiểm thử gồm:

Trường hợp 1: Chạy thử hệ thống với đèn, quạt, cảm biến và cửa được kết nối. Hoạt động của các thiết bị được điều khiển trực tiếp thông qua việc lập trình với những yêu cầu chi tiết sau:

- + Đèn sáng với chế độ bật/tắt mỗi 5 giây.
- + Cảm biến trả về kết quả nhiệt độ trên Serial Monitor.
- + Cửa đóng/mở sau khi nhấn nút.

Trường hợp 2: Chạy thử hệ thống với web server cơ bản, bao gồm chức năng bật/tắt mở đèn, quạt và đóng/ mở cửa trên web, yêu cầu tốc độ phản hồi của các lệnh đưa ra thấp hơn 3 giây.

Trường hợp 3: Chạy thử hệ thống và theo dõi trạng thái thay đổi trên database. Nhóm sử dụng Firebase – dịch vụ cơ sở dữ liệu trên nền tảng đám mây. Theo dõi tốc độ trả về kết quả, trạng thái mới của từng thiết bị sau các lệnh đưa ra.

Trường hợp 4: Hoàn thành các yêu cầu đặt ra, kiểm tra lần cuối hoạt động cập nhật trạng thái trên CSDL cũng như tốc độ phản hồi trên giao diện người dùng (thấp hơn 3 giây). Kiểm tra hoạt động hẹn giờ của đèn (đúng giờ bật/tắt), hiển thị biểu đồ giá trị nhiệt độ từ DHT11.

Sau khi hoàn thành xây dựng kịch bản kiểm thử, nhóm sẽ chuyển qua giai đoạn thiết kế để hoàn tất mô hình. Thiết kế xong sẽ chuyển về lại quá trình kiểm thử này, ghi nhận những cái được và chưa được nhằm khắc phục trong thời gian sớm nhất.

3. Thiết kế, chạy và kiểm thử

+ Xác định phần cứng sử dụng:

- Để bắt đầu thiết kế hệ thống SMART HOME, nhóm cần tìm hiểu trước về các phần cứng cần sử dụng để hiện thực mô hình. Trong quá trình nghiên cứu, nhóm liệt kê danh sách các phần cứng cần chuẩn bị ứng với từng thiết

bị:

Thiết bị	Thành phần phần cứng
Vi điều khiển	LOLIN32 ESP32 là một bảng mạch phát triển được xây dựng xung quanh bộ vi điều khiển Espressif ESP-WROOM-32. Nó được trang bị bộ nhớ 4MB và cung cấp kết nối WiFi và Bluetooth.
Đèn	LED đơn 5V
Cửa	Khóa cửa nam châm điện 12V - 0.4A Relay 5V Nguồn pin 12V
Cảm biến	Cảm biến nhiệt độ & độ ẩm DHT11
Quạt	Quạt tản nhiệt 12V 9x9CM Relay 5V Nguồn pin 12V

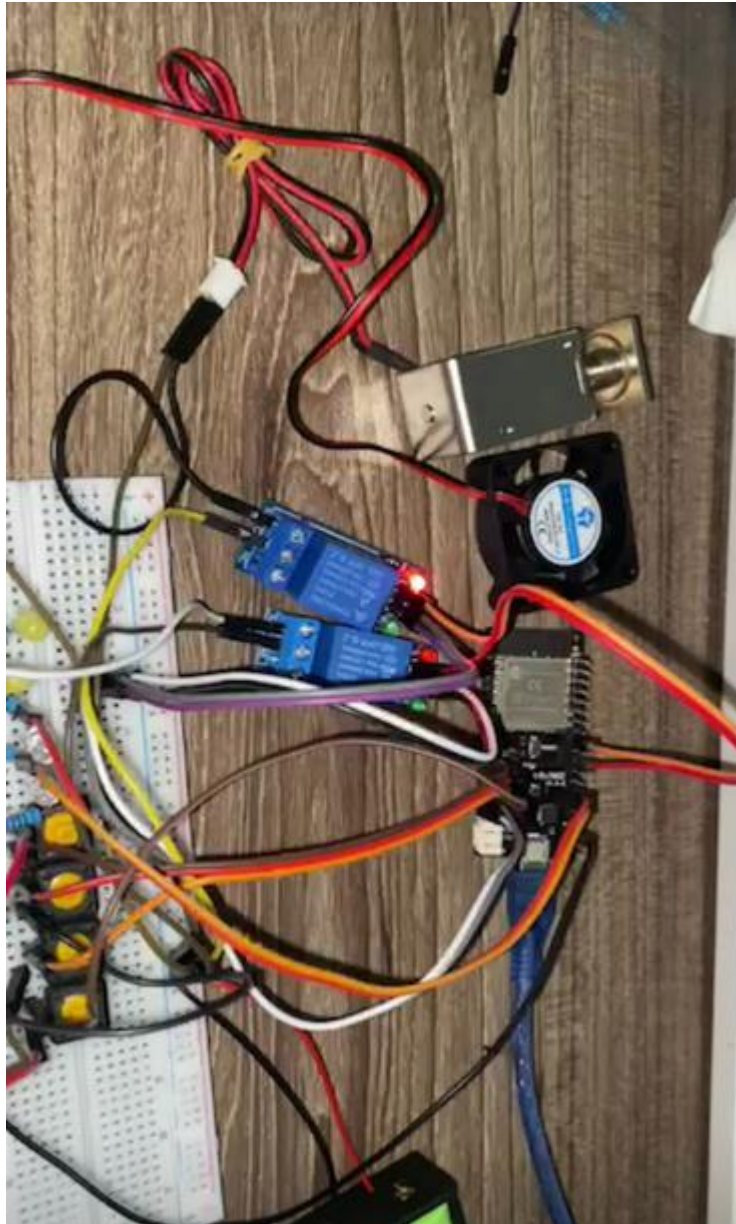
Bảng 2.4. Danh sách các thiết bị phần cứng cần chuẩn bị để thiết kế mô hình SMARTHOME

+ Thực hiện thiết kế:

Đầu tiên, nhóm tiến hành ghép nối từng linh kiện với nhau thông qua MCU chính là WeMos LOLIN 32.



Hình 5. Ghép nối khóa nam châm 12V với nguồn và MCU chính.



Hình 6. Ghép nối đèn, quạt, cửa khóa với nguồn và MCU chính

Sau khi hoàn thành ghép nối, nhóm tiến hành kiểm tra hoạt động của các thiết bị thông qua lập trình trên Arduino IDE, ESP-IDF. Sau khi kiểm tra và đánh dấu hoàn thành kịch bản kiểm thử 1 – kiểm tra phần cứng, nhóm chuyển sang phát triển giao diện webserver cho hệ thống.

Ở phần thiết kế giao diện webserver, nhóm đã sử dụng giao thức HTTPS (HyperText Transfer Protocol Service) để điều khiển trạng thái của thiết bị. Lựa chọn giao thức HTTPS nhằm mục đích nâng cao tính bảo mật, tính chất xác thực để ngăn chặn các trang web giả mạo. Đồng thời, đối với đồ án ngôi nhà thông minh, việc bảo đảm quyền riêng tư cũng quan trọng – ưu điểm đặc biệt của HTTPS thông qua Secure Socket Layer/ Transport Layer Security (bắt tay, thông báo, xác thực, mã hóa).

Tạo giao diện đăng nhập:

Trước tiên, ta tạo giao diện đăng nhập gồm có Username, Password cần nhập, gọi khối vừa tạo là login – container.

```
</head>
<body>
  <div id="login-container">
    <form id="login-form">
      <h2>Login</h2>
      <input type="text" id="username" placeholder="Username" required>
      <input type="password" id="password" placeholder="Password" required>
      <button type="submit">Login</button>
    </form>
  </div>
```

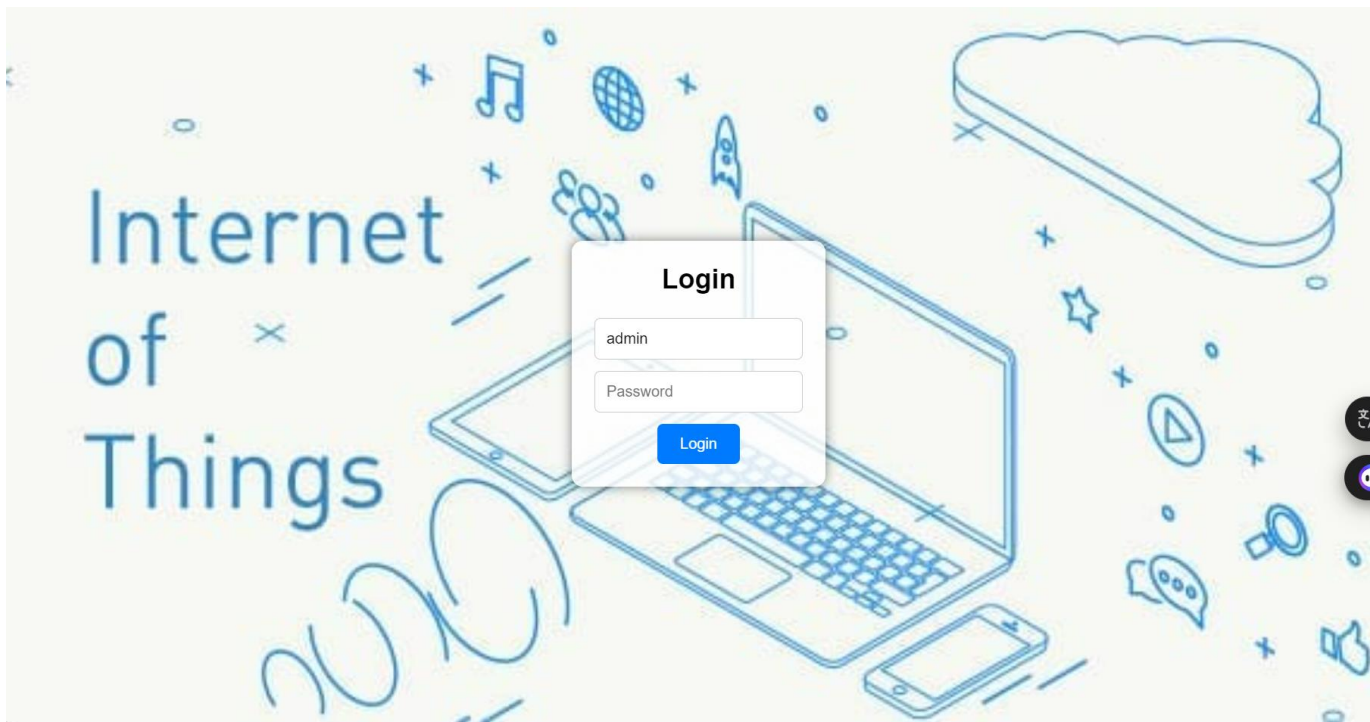
Hình 7. Tạo bố cục, ô chữ đăng nhập và nơi điền tài khoản, mật khẩu

Sau khi hoàn thành việc tạo bố cục đăng nhập, ta đến với phần kiểm tra thông tin, cho phép đăng nhập bằng cách so sánh username, password với tài khoản có sẵn. Nếu trùng khớp, giao diện sẽ được chuyển sang device – container để điều khiển thiết bị. Ngược lại thì sẽ thông báo đã nhập sai tài khoản, mật khẩu.

```
document.getElementById('login-form').addEventListener('submit', function(e) {
  e.preventDefault();
  var username = document.getElementById('username').value;
  var password = document.getElementById('password').value;

  if(username === 'admin' && password === 'admin@@') {
    document.getElementById('login-container').style.display = 'none';
    document.getElementById('device-container').style.display = 'block';
  } else {
    alert('Invalid username or password');
  }
});
```

Hình 8. Kiểm tra tài khoản, mật khẩu, báo Invalid khi đăng nhập không thành công.



Hình 9. Giao diện đăng nhập vào hệ thống.

Trong giao diện điều khiển thiết bị ngoại vi, nhóm sử dụng 3 biến button để thay đổi hoạt động của đèn, quạt và mở/ đóng cửa. Với biểu đồ nhiệt, độ ẩm, nhóm dùng thư viện chart.js để vẽ biểu đồ giá trị.

```
var temperatureChart = new Chart(ctxTemp, {
  type: 'line',
  data: {
    labels: [],
    datasets: [{
      label: 'Temperature (°C)',
      data: [],
      backgroundColor: 'rgba(255, 255, 255, 0.5)',
      borderColor: 'rgba(255, 99, 132, 1)',
      borderWidth: 1
    }]
  },
  options: {
    scales: {
      yAxes: [{
        ticks: {
          beginAtZero: true
        }
      }]
    }
  }
});
```

Hình 9. Vẽ biểu đồ thể hiện giá trị của nhiệt độ đo được từ DHT11

Với các nút, đầu tiên nhóm truy xuất các phần tử HTML dựa trên lớp CSS và ID. Đầu tiên, với lệnh `querySelector('.btnON1')`, chương trình sẽ chọn phần tử đầu tiên trong file HTML có lớp CSS tương ứng. Tương tự với các biến còn lại.

Tương tự ta sẽ truy xuất với lệnh `getElementById` để nhận giá trị từ phần tử có ID tương ứng trong file HTML. Bằng cách truy xuất này, ta có thể thay đổi kiểu dáng cho nút, thêm các sự kiện cho nút.


```

14  const btnON1 = document.querySelector('.btnON1');
15  const btnOFF1 = document.querySelector('.btnOFF1');
16  const btnON2 = document.querySelector('.btnON2');
17  const btnOFF2 = document.querySelector('.btnOFF2');
18  const btnON3 = document.querySelector('.btnON3');
19  const btnOFF3 = document.querySelector('.btnOFF3');
20
21  var btnOn1 = document.getElementById("btnON_01");
22  var btnOff1 = document.getElementById("btnOFF_01");
23  var btnOn2 = document.getElementById("btnON_02");
24  var btnOff2 = document.getElementById("btnOFF_02");
25  var btnOn3 = document.getElementById("btnON_03");
26  var btnOff3 = document.getElementById("btnOFF_03");

```

Hình 10. Truy xuất phần tử HTML thông qua lớp CSS và ID.

Tiếp theo, các nút sẽ thay đổi giá trị tín hiệu dựa trên việc theo dõi trạng thái trong cơ sở dữ liệu trên Firebase. Ví dụ với nút mở/tắt đèn sau, bằng việc tham chiếu đến đường dẫn `"/light"` trong cơ sở dữ liệu Firebase, ta gọi thêm lệnh `".on("value", function(snapshot))"` để tạo một listener nghe sự thay đổi của giá trị value, nếu thay đổi, hàm callback sẽ gọi với tham số snapshot. Từ đó sẽ cập nhật giá trị mới cho biến `light_state`.

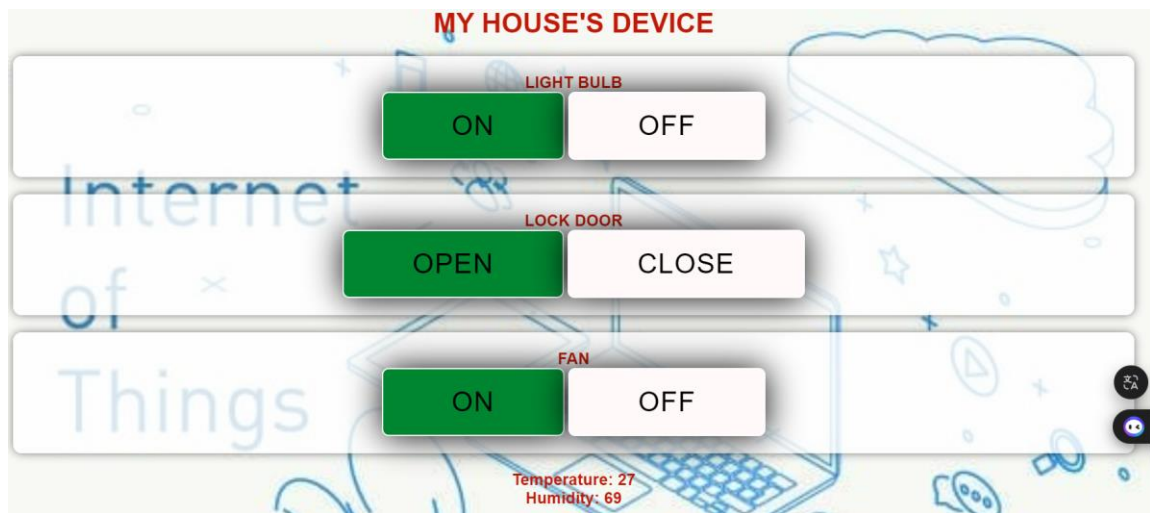
```

database.ref("/light").on("value", function(snapshot) {
  var light_state = snapshot.val();
  if(light_state == "on") {
    btnON1.style.backgroundColor = '#008631';
    btnOFF1.style.backgroundColor = '#ffffaf';
  } else {
    btnON1.style.backgroundColor = '#ffffaf';
    btnOFF1.style.backgroundColor = '#c21807';
  }
});

```

Hình 11. Theo dõi thay đổi trên CSDL cho trạng thái đèn

Cuối cùng, ta có giao diện điều khiển thiết bị như sau:



Hình 12. Giao diện điều khiển thiết bị ngoại vi

Sau đó, để cập nhật trạng thái thiết bị với cơ sở dữ liệu, nhóm tiếp tục sử dụng HTTPS để giao tiếp với Firebase. Dưới đây là đoạn mã thiết lập kết nối với Firebase bằng cách tạo một tham chiếu (firebaseConfig) vào hệ thống.

```
const firebaseConfig = {
  apiKey: "AIzaSyCmW9JD05hkNFvqXyviT0pYf4USenkuu5A",
  authDomain: "smarthome-68ff3.firebaseio.com",
  databaseURL: "https://smarthome-68ff3-default-rtdb.firebaseio.com",
  projectId: "smarthome-68ff3",
  storageBucket: "smarthome-68ff3.appspot.com",
  messagingSenderId: "775526693080",
  appId: "1:775526693080:web:99b327069ac8fec1a83fad"
};
firebase.initializeApp(firebaseConfig);
var database = firebase.database();
```

Hình 13. Thiết lập kết nối với Firebase

Sau đó, ta có thể dễ dàng thiết lập trạng thái của các thiết bị vào Firebase như sau:

```

btnOn1.onclick = function() {
    firebase.database().ref("/").update({ "light": "on" });
};
btnOff1.onclick = function() {
    firebase.database().ref("/").update({ "light": "off" });
};
btnOn2.onclick = function() {
    firebase.database().ref("/").update({ "lock": "open" });
};
btnOff2.onclick = function() {
    firebase.database().ref("/").update({ "lock": "close" });
};
btnOn3.onclick = function() {
    firebase.database().ref("/").update({ "fan": "on" });
};
btnOff3.onclick = function() {
    firebase.database().ref("/").update({ "fan": "off" });
};

```

Hình 14. Thiết lập trạng thái của các thiết bị trên Firebase

Sau khi thiết lập thành công trạng thái thiết bị trên Firebase, trạng thái của các thiết bị sẽ liên tục được cập nhật, đổi mới mỗi khi có thay đổi. (Nhóm sẽ bổ sung hình database khi báo cáo trực tiếp)

III. ĐÁNH GIÁ TIẾN ĐỘ, NHẬN XÉT VÀ TƯƠNG LAI

HỌ TÊN	NỘI DUNG CÔNG VIỆC	PHẦN TRĂM CÔNG VIỆC
Hoàng Phan Thành Bách (Nhóm trưởng)	Lên danh sách các linh kiện phần cứng cần sử dụng. Thực hiện thiết kế, chạy thử phần cứng cảm biến, cửa khóa. Đưa hệ thống lên Webserver, thực hiện chức năng đăng nhập. Biểu diễn nhiệt độ thông qua biểu đồ. Trang trí giao diện người dùng. Viết báo cáo tổng kết	40%
Nguyễn Đức Lưu	Chịu trách nhiệm bảo trì, kiểm tra và mua mới các linh kiện phần cứng sử dụng cho đồ án. Thực hiện thiết kế, chạy thử phần cứng quạt, đèn. Đưa hệ thống lên Webserver Đưa hệ thống lên CSDL, cập nhật trạng thái người dùng. Thực hiện chức năng hẹn giờ cho đèn.	40%
Kiều Thị Hiền Trinh		10%
Dương Mai Quý		10%

Bảng 3.1. Đánh giá hoàn thành công việc giai đoạn tháng 3 - 5/2024.

***Nhóm nhận xét:**

- Hoàn thành: Về mặt thiết kế, nhóm đã hoàn thành mặt thiết kế cơ bản và kết nối các thiết bị. Các kịch bản kiểm thử cũng đã được hoàn thiện với đúng yêu cầu mà nhóm đề ra khi thiết kế hệ thống. Theo dõi trên tiêu chí chấm điểm đề tài, nhóm tự chấm đã hoàn thành được 10/12 điểm của đề tài.
- Chưa hoàn thành: Bên cạnh những điểm tốt, nhóm cũng thừa nhận hệ thống vẫn còn nhiều thiếu sót. Cụ thể thì nhóm chưa làm được hai chức năng hẹn giờ cho đèn cũng như chỉnh sửa cường độ quạt. Ngoài ra, ở giao diện người dùng vẫn còn chưa bắt mắt, hấp dẫn và cũng chưa có tính năng

thêm/ xóa thiết bị.

Giảng viên nhận xét:

.....

.....

.....

.....