BÀI TẬP LỚN HỌC PHẦN

TÊN ĐỀ TÀI:

SỬ DỤNG MODULE IOT ESP 32 TRONG GIÁM SÁT VÀ ĐIỀU KHIỂN HỆ THỐNG CHIẾU SÁNG QUA ỨNG DỤNG

NHÓM THỰC HIỆN: 12

MỤC LỤC

PHẦN MỞ ĐẦU

1. Lý do lựa chọn đề tài

Hiện tại, thế giới đang trong một kỷ nguyên mới về công nghệ, là thời đại của vạn vật kết nối qua Internet. Các thiết bị điện tử kết nối mạng ngày càng trở nên dễ dàng tiếp cận hơn, phổ biến mạnh mẽ trên phạm vi toàn cầu. Việc ứng dụng IoT vào cuộc sống cũng trở nên thực tế, tiện dụng, hữu ích và mang lại nhiều lợi ích, trợ giúp người dùng nhiều hơn. Một trong những mảng phát triển mãnh mẽ và phổ biển hiện nay của IoT là nhà thông minh – "smart home", với các thiết bị gia đình thông minh nâng cao sự tiện lợi, thoải mái, hiệu quả năng lượng và an ninh. Giúp căn nhà trở nên tự động hóa, "thông minh" hơn và có thể kiểm soát từ xa.

Đèn thông minh trong nhà là một phần của "smart home", giúp người dùng có thể kiểm soát, điều khiển hệ thống đèn trong nhà mình nhanh gọn và tiện lợi hơn so với cách thủ công cũ.

Đèn thông minh sẽ càng phát triển và được sử dụng rộng rãi trong một tương lai gần. Vì vậy, với kiến thức được tiếp thu qua môn Xây dựng các hệ thống nhúng trong kỳ vừa qua, chúng em lựa chọn đề tài "Sử dụng Module IoT ESP 32 trong giám sát và điều khiển hệ thống chiếu sáng qua ứng dụng" cho bài tập lớn môn học của mình.

2. Phạm vi bài toán

Xây dựng một ứng dụng website có thể kiểm soát việc bật/tắt, điều chỉnh độ sáng hệ thống đèn cũng như kiểm soát trạng thái của đèn đang hoạt động hay đang ngắt thông qua Wifi với module ESP 32.

3. Phương hướng thực hiện

- Xây dựng mạch kết nối đèn LED với module ESP 32.
- Kết nối với WiFi và đẩy chương trình được lập trình trên Arduino lên ESP 32.
- Viết ứng dụng website giúp người dùng bật/tắt, điều chỉnh độ sáng đèn, kiểm soát trạng thái của hệ thống đèn (của mạch được lắp ráp phía trên) qua đó.

PHẦN NỘI DUNG

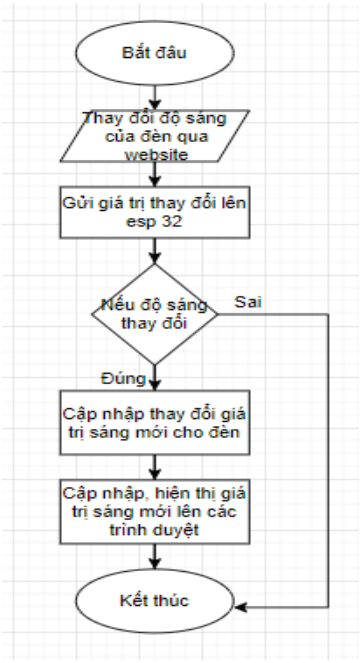
1. Công nghệ sử dụng

Với đề tài "Sử dụng Module IoT ESP 32 trong giám sát và điều khiển hệ thống chiếu sáng qua ứng dụng" thì việc xây dựng một ứng dụng website kết nối với module ESP 32 là điều thiết yếu. Ở đây, em sử dụng giao thức Websocket để có thể điều khiển hệ thống đèn qua ESP 32 với ứng dụng website được xây dựng. Trạng thái đầu ra của các bóng đèn sẽ tự động cập nhật lên giao diện của trang web.

Để thực hiện kết nối, máy khách (client) kết nối đến máy chủ (server) qua một quy trình gọi là "WebSocket handshake". Quá trình này bằng đầu với HTTP request/response, cho phép server xử lý các kết nối HTTP và Websocket trên cùng một cổng. Kết nối được duy trì để ta có thể viết và nhận dữ liệu bằng JavaScript như khi đang sử dụng một TCP socket đơn thuần. [1]

Với giao thức Websocket, server (ESP 32) có thể gửi thông tin đến tất cả các máy khách. Điều này có thể hiểu với đề tài trên rằng: Khi độ sáng của đèn được thay đổi trên một client thì sẽ được gửi thông tin thay đổi đến tất cả trình duyệt của các client khác.

2. Lưu đồ thuật toán



Hình 1 Lưu đồ thuật toán

Lưu đồ thuật toán trên mô tả ngắn gọn cách thức thiết kế đề và các quy trình thực hiện từng bước với ứng dụng mà em xây dựng.

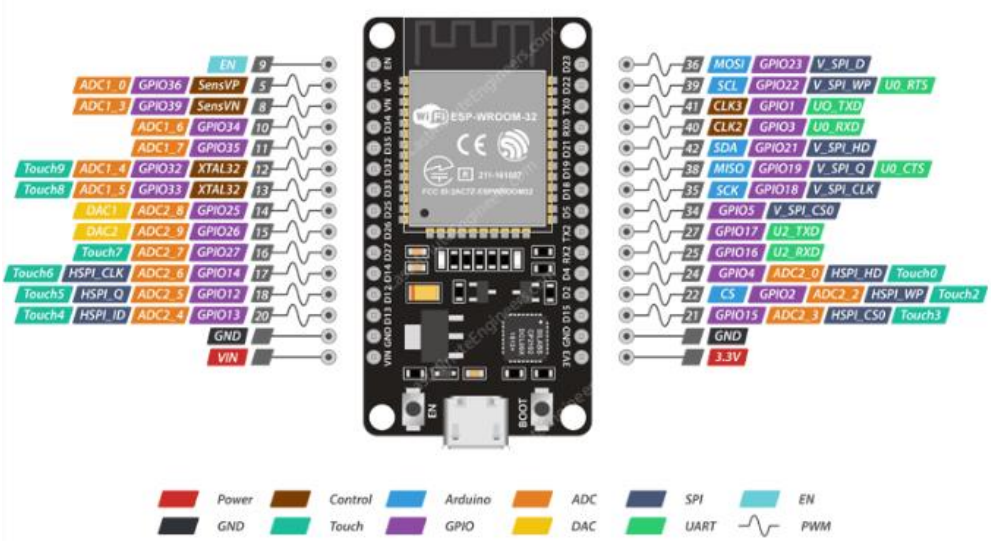
Sau khi tải chương trình lập trình từ Arduino lên ESP 32 và kết nối thành công với Wifi, ta sẽ thu được địa chỉ IP là url của ứng dụng website được xây dựng điều khiển hệ thống đèn.

Theo lưu đồ, từ trình duyệt ta thay đổi độ sáng của 1 đèn bất kỳ. Giá trị sáng của đèn thay đổi sẽ được gửi lên ESP 32. Sau đó, ESP 32 sẽ tăng/giảm độ sáng của đèn bị thay đổi ứng với giá trị được yêu cầu. Cuối cùng, giá trị sáng của đèn thay đổi sẽ được cập nhập hiển thị lên giao diện của tất cả các client đang truy cập.

3. Module phần cứng

3.1 Module ESP32

- Thông số cơ bản: Vi điều khiển: ESP-WROM-32; điện áp logic: 3.3V; GPIO: 30; Flash memory: 4MB; SRAM: 520KB; Clock Speed: 240MHz; Wifi: 802.11b/g/n Wi-Fi transceiver; Bluetooth: Bluetooth 4.2/BLE; Nguồn điện khuyên dùng: 5V. [2]
- Chức năng: Vi điều khiển hỗ trợ các kết nối Wifi, Bluetooth và các chế độ hoạt động sử dụng năng lượng thấp.
- Hình ảnh:



Hình 2 ESP 32

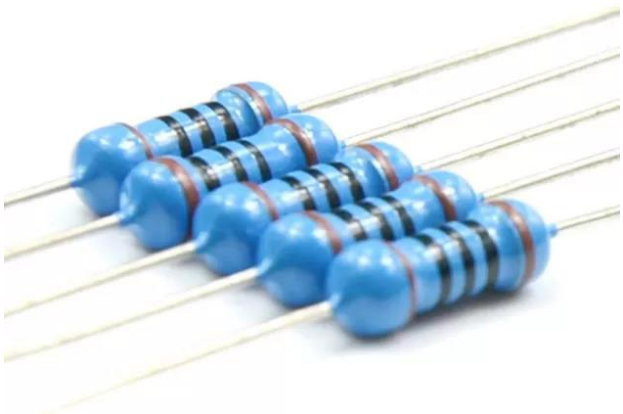
3.2 Đèn LED

- Thông số cơ bản: Ánh sáng trắng; điện áp: 3V-3.2V; kích thước bóng: đường kính 3mm; số chân: 2; chiều dài chân: 28-29mm. [3]
- Chức năng: Thắp sáng, được sử dụng để lắp mạch điện đơn giản, làm mạch led trái tim, …
- Hình ảnh:

Hình 3 Đèn Led 3mm

3.3 Điện trở

- Thông số cơ bản: Điện trở vạch; giá trị trở: 200 ôm; Công suất định mức: 1/4W; sai số: 5%. [4]
- Chức năng: Điều chỉnh mức độ tín hiệu, hạn chế cường độ dòng điện chảy trong mạch.
- Hình ảnh:



Hình 4 Điện trở vạch 200 ôm

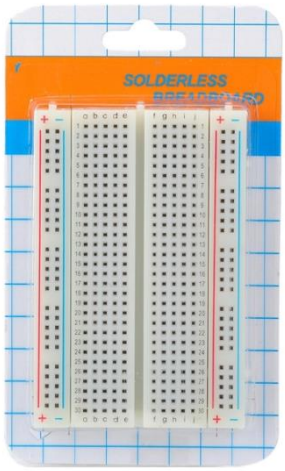
3.4 Dây nối

- Thông số cơ bản: M-F Jumper Wire; dây lõi đồng mã thiếc nhiều sợi có độ dẫn điện cao; độ dài: 20cm. [5]
- Chức năng: dẫn điện, nối các thiết bị điện với nhau.
- Hình ảnh:

Hình 5 Dây nối

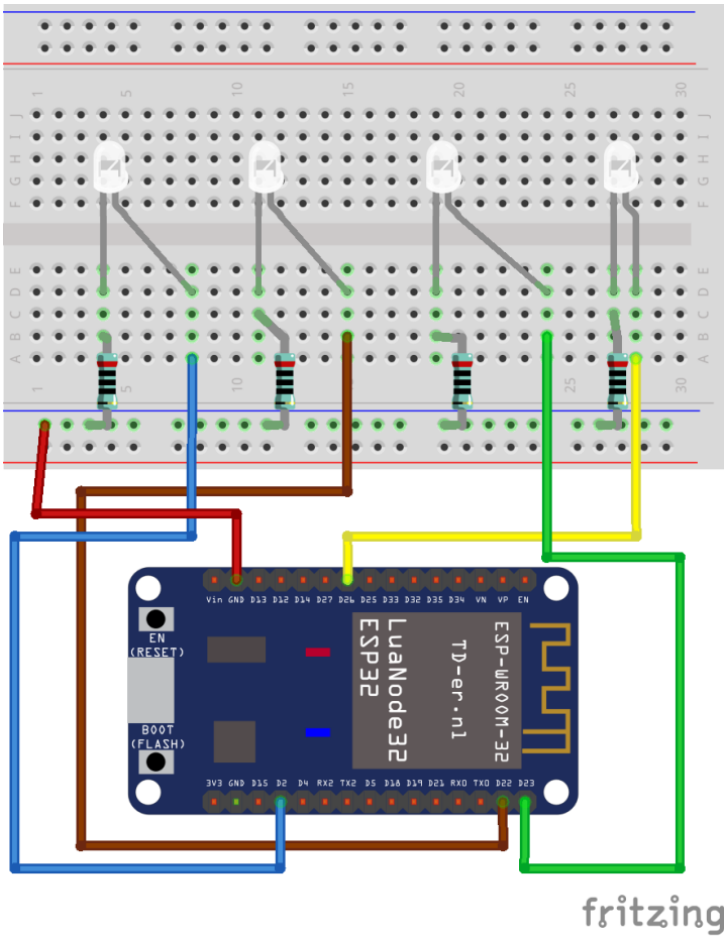
3.5 Bảng mạch Breadbroad

- Thông số cơ bản: 400 lỗ; kích thước: 8.2x5.5x0.85 cm; điện áp/dòng điện: 300V/3-5A. [6]
- Chức năng: bảng mạch không hàn, lắp ráp mạch linh kiện điện tử.
- Hình ảnh:



Hình 6 Bảng mạch breadbroad

4. Sơ đồ mạch



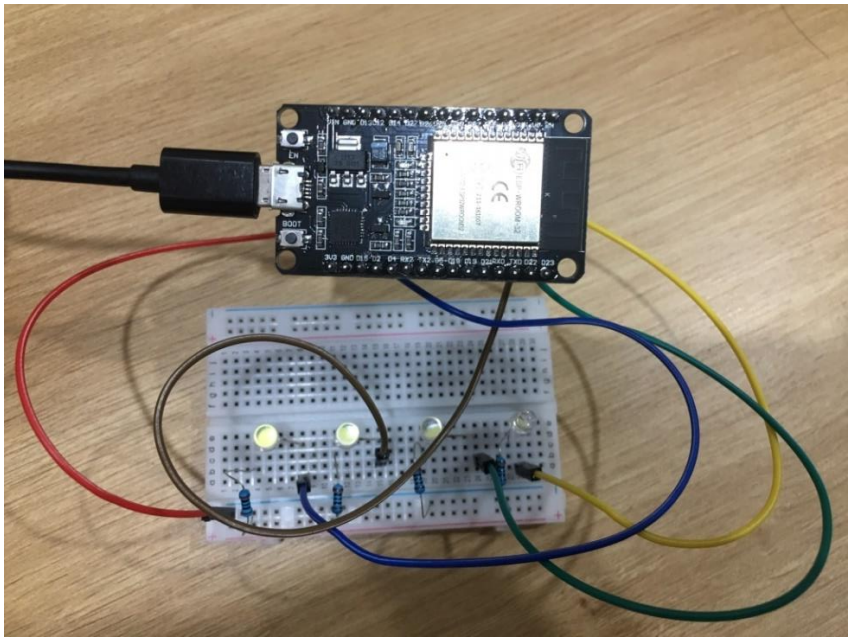
Hình 7 Mạch mô phỏng được vẽ bằng ứng dụng fritzing

Trên đây là sơ đồ mạch được em vẽ mô phỏng qua ứng dụng Fritzing nhằm mục đính giúp xem rõ kết nối các thành phần trong mạch.

Qua hình 7 ta có thể thấy, mạch điện này được xây dựng với 5 thành phần (đã được đề cập chi tiết các thông số thiết bị sử dụng ở mục phía trên): module ESP 32, bóng đèn LED trắng, điện trở 200 ôm, dây nối và bảng mạch breadbroad.

Qua mạch mô phỏng, ta có thể thấy rõ bóng đèn led 1 được nối qua điện trở và với chân D2 của ESP32; bóng đèn led 2 tương tự và được nối với chân D22; bóng led 3 cũng như vậy được nối với D23; cuối cùng, bóng đèn led 4 được nối với chân D26. Và chân GND của module được nối với bảng breadbroad để cung cấp điện cho toàn mạch.

Và dưới đây là mạch lắp rắp được trong thực tế theo với mạch mô phỏng và các thiết bị kể trên.



Hình 8 Mạch được lắp ráp trong thực tế

5. IDE sử dụng

5.1 Arduino

Arduino IDE là một nền tảng với mã nguồn mở, linh hoạt, dễ dàng sử dụng. Đây là một IDE hộ trợ đa nền tảng, chạy trên nhiều hệ điều hành như Mac, Window và Linux. Arduino IDE là môi trường để lập trình, kết nối với các bo mạch để tải chương trình và giao tiếp với chúng. Từ đó, ta có thể đọc đầu vào và kiểm soát đàu ra: bật đèn LED, kích hoạt động cơ. Ta có thể ra lệnh cho bo mạch bằng cách gửi chương trình lập trình sẵn đến bộ vi điều khiển của bo mạch. Các file chương trình trên Arduino sẽ được lưu với phần mở rộng là .ino. [7]

Chương trình chính của ứng dụng cho đề tài này được xây dựng trên IDE này. Trên Arduino sẽ là phần chương trình khai báo kết nối với hệ thống đèn kèm ESP 32 được lắp ráp trên bo mạch, với wifi cũng như kết nối với phần frontend (giao diện) của website điều khiển, kiểm soát đèn.

5.2 Visual Studio Code

Visual Studio Code là một IDE dùng để lập trình đa ngôn ngữ và hỗ trợ đa tảng với nhiều tích hợp giúp người dùng thuận tiện, tối ưu hơn trong quá trình sử dụng. [8]

IDE được em sử dụng để xây dựng phần giao diện (frontend) cho ứng dụng website cho đề tài với các file .html, .css và .js.

6. Chương trình thực thi

Chương trình phần mềm cho đề tài "Sử dụng Module IoT ESP 32 trong giám sát và điều khiển hệ thống chiếu sáng qua ứng dụng" gồm 2 phần chính:

- Arduino code: là mã chạy trên vi điều khiển, triển khai máy chủ web.
- Website: giao diện người dùng cho hệ thống điều khiển đèn LED, gồm các thành phần của trang web(html, js,css) kết nối với server thông qua giao thức WebSocket, cung cấp giao diện thân thiện cho người dùng để điều khiển đèn LED.

- Arduino code:

Đây là mã Arduino triển khai máy chủ web và máy chủ WebSocket để điều khiển bốn đèn LED sử dụng tín hiệu PWM và giao diện web. Mã này sử dụng thư viện AsyncWebServer và AsyncWebSocket, cho phép thực hiện các thao tác không chặn. Máy chủ được khởi tạo để chạy trên cổng 80 và kết nối WiFi được thiết lập bằng cách kết nối với SSID và mật khẩu được chỉ định.

Dưới đây là khởi tạo và 1 số hàm chính trong đoạn mã:

- Mã khai báo bốn chân LED và thiết lập các tham số PWM của chúng, bao gồm tần số, kênh và độ phân giải. Giao diện web bao gồm bốn thanh trượt, một thanh cho mỗi đèn LED và vị trí hiện tại của chúng được lưu trữ trong chuỗi sliderValue1-4.
- Hàm getSliderValues: truy xuất các giá trị thanh trượt hiện tại và trả về chúng dưới dạng chuỗi JSON. Khi máy chủ nhận được tin nhắn từ máy khách WebSocket, hàm handleWebSocketMessage sẽ trích xuất vị trí thanh trượt từ tin nhắn và cập nhật giá trị thanh trượt tương ứng và chu kỳ nhiệm vụ PWM. Sau đó, các giá trị thanh trượt được cập nhật sẽ được gửi tới tất cả các máy khách được kết nối bằng cách sử dụng hàm notifyClients.
- Hàm onEvent: được gọi khi một sự kiện WebSocket xảy ra, chẳng hạn như một máy khách mới kết nối hoặc ngắt kết nối. Mã này cũng bao gồm một thư viện SPIFFS cho phép máy chủ phục vụ các tệp từ hệ thống tệp của ESP32, có thể hữu ích để phục vụ các trang web hoặc các tệp tĩnh khác.

Nhìn chung, mã này cho phép điều khiển từ xa bốn đèn LED bằng giao diện web, giao diện này có thể được truy cập từ bất kỳ thiết bị nào được kết nối với cùng một mạng với ESP32.

- Website:

Gồm 3 phần chính html, css, js cấu hình nên giao diện người dùng. Trong đó, có một số hàm chính bằng Javascript sau để kết nối đến máy chủ, cập nhập giá trị sách của đèn và update lên biểu đồ. Đoạn mã js này cung cấp một trang web tương tác, cho phép người dùng điều khiển các thiết bị thông qua các thanh trượt và biểu đồ, và cập nhật trạng thái của các thiết bị. Nó sử dụng WebSocket để gửi và nhận các thông điệp từ máy chủ, và sử dụng thư viện Chart.js để vẽ các biểu đồ. Các điểm dữ liệu trong biểu đồ được cập nhật trong thời gian thực và có thể được tùy chỉnh thông qua thanh trượt. Các giá trị mới được cập nhật lên trang web bằng cách sử dụng JavaScript để thay đổi nội dung của các phần tử HTML với các hàm chính như sau:

- Hàm onload(event): được gọi khi trang web được tải hoàn tất, gọi hàm initWebSocket() để khởi tạo kết nối với máy chủ WebSocket.
- Hàm initWebSocket(): tạo kết nối WebSocket tới máy chủ ở địa chỉ gateway thông qua hàm new WebSocket(), và gán các hàm callback cho các sự kiện onopen, onclose, onmessage của WebSocket.
- Hàm getValues(): gửi chuỗi`"getValues" tới máy chủ WebSocket để yêu cầu nhận giá trị từ máy chủ.
- Hàm onOpen(event): được gọi khi kết nối WebSocket được mở thành công. Gọi hàm getValues() để yêu cầu giá trị từ máy chủ.
- Hàm onClose(event): được gọi khi kết nối WebSocket bị đóng, sau đó đợi 2 giây và gọi lại hàm initWebSocket() để khởi tạo kết nối mới.
- Hàm updateSliderPWM(element): được gọi khi thanh trượt của slider thay đổi giá trị, cập nhật giá trị của thanh trượt và gửi giá trị mới tới máy chủ WebSocket thông qua websocket.send().

- Hàm onMessage(event): được gọi khi nhận được dữ liệu từ máy chủ WebSocket, sau đó cập nhật các giá trị nhận được vào các phần tử HTML tương ứng, đồng thời cập nhật giá trị của các slider và vẽ biểu đồ tương ứng.

Các hàm này kết hợp với nhau để tạo ra một ứng dụng điều khiển đơn giản, cho phép điều khiển các thiết bị đang kết nối với máy chủ WebSocket và hiển thị các giá trị trên giao diện người dùng.

7. Phương pháp thực hiện

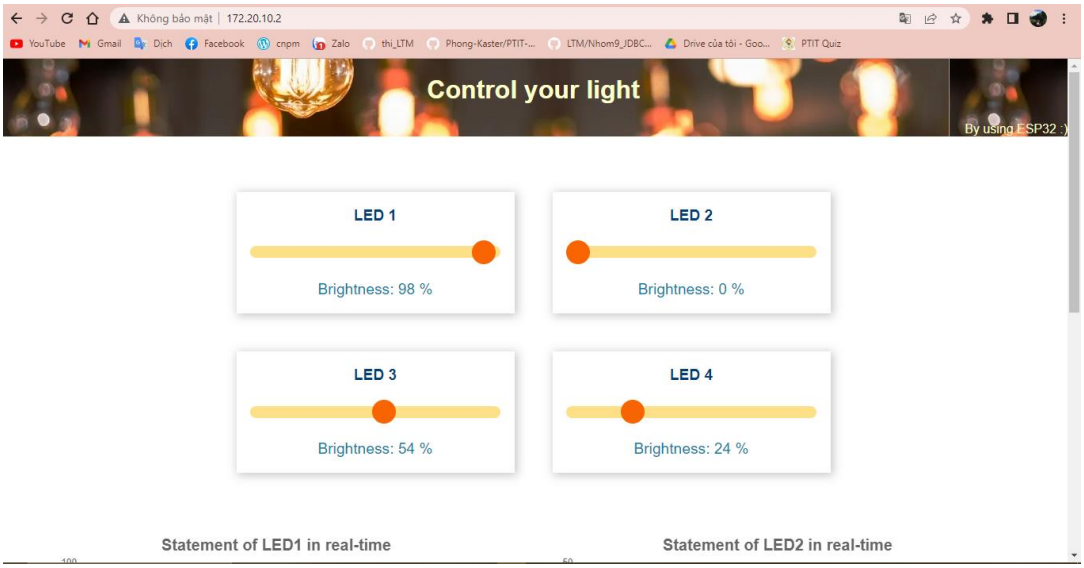
Để cài đặt hệ thống cho chương trình điều khiển đèn esp32 thông qua website sử dụng websocket, chúng ta cần thực hiện các bước sau:

- Chuẩn bị thiết bị: chuẩn bị các module phần cứng và lắp ráp mạch theo như sơ đồ mạch được mô tả ở các phần phía trên.
- Cài đặt môi trường phát triển: cài đặt trình biên dịch Arduino IDE để chạy chương trình và phát triển cũng như Visual Studio Code IDE để lập trình giao diện.
- Cài đặt một số thư viện và tool cần thiết: ESP32FS upload file và hình ảnh lên ESP32, thư viện ESPAsyncWebServer và AsyncTCP kết nối mạng cho ESP32,…
- Đảm bảo rằng broads đang sử dụng là của Esp32 và đúng cổng COM kết nối với bảng mạch.
- Tải Arduino code lên Esp32 broad.
- Tải file(website: html, css, js, img) lên Esp32 thông qua Tool > ESP32 Data Sketch Upload trên Arduino.
- Sau khi tải lên thành công thì mở Serial Monitor ở 115200 baud và nhấn “EN” trên ESP32.
- Lúc đó, 1 địa chỉ IP sẽ được in ra trên Serial Monitor. Với các thiết bị trong cùng mạng wifi với ESP32 thì sau khi truy cập vào địa chỉ IP đã cho phía trên sẽ có thể điều khiển hệ thống đèn thông qua giao diện web được thiết lập.

8. Mô tả ứng dụng xây dựng

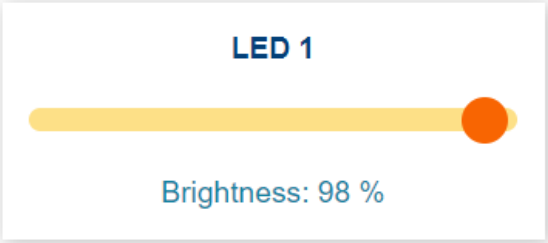
8.1 Điều chỉnh độ sáng đèn

Sau khi kết nối thành công, ta sẽ được cung cấp cho một địa chỉ IP là url của website ứng dụng điều khiển hệ thống đèn. Để có thể truy cập đường dẫn trên, client (máy khách) phải sử dụng chung một mạng dữ liệu với server (ESP 32). Hình 9 phía dưới là giao diện hiện thị khi kết nối thành công giữa client và server. Thông qua website, người dùng có hệ điều chỉnh độ sáng, trạng thái bật/tắt với từng đèn, cụ thể ở đây là với 4 đèn tương ứng với mạch được lắp ráp như đã nêu phía trên. Độ sáng của đèn sau khi được thay đổi đều được cập nhập trên các client.



Hình 9 Giao diện website phần điều chỉnh độ sáng đèn

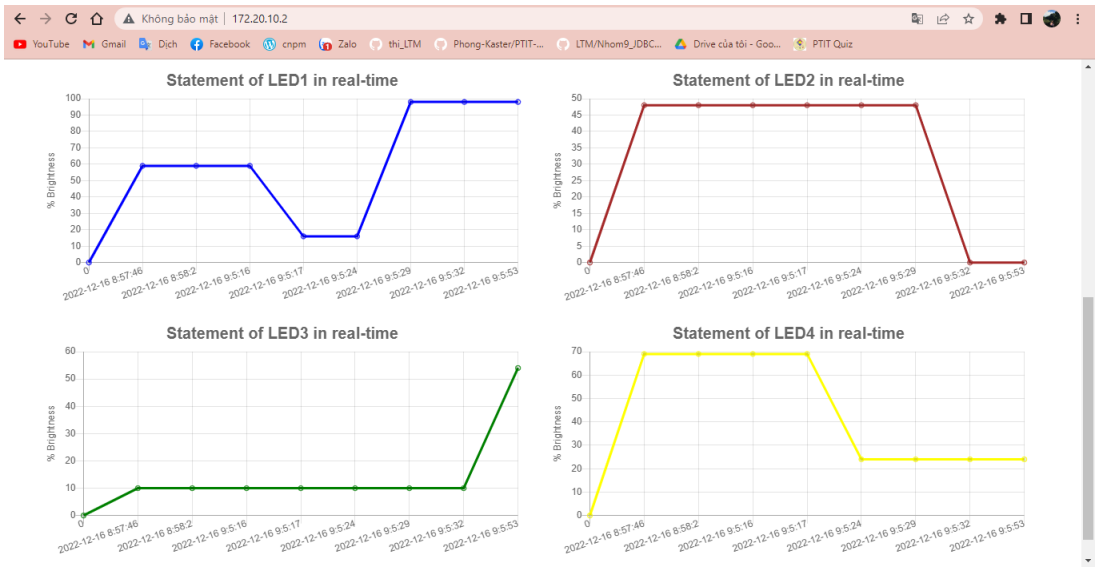
Cụ thể với từng đèn, ta điều chỉnh độ sáng bằng cách kéo thả thanh trượt đến độ sáng mong muốn theo hình 10. Giá trị thanh trượt hiển thị là độ sáng đèn từ 0% đến 100% tương ứng từ đèn tắt đến đèn sáng với cường độ cao nhất.



Hình 10 Giao diện điều chỉnh độ sáng của đèn LED 1.

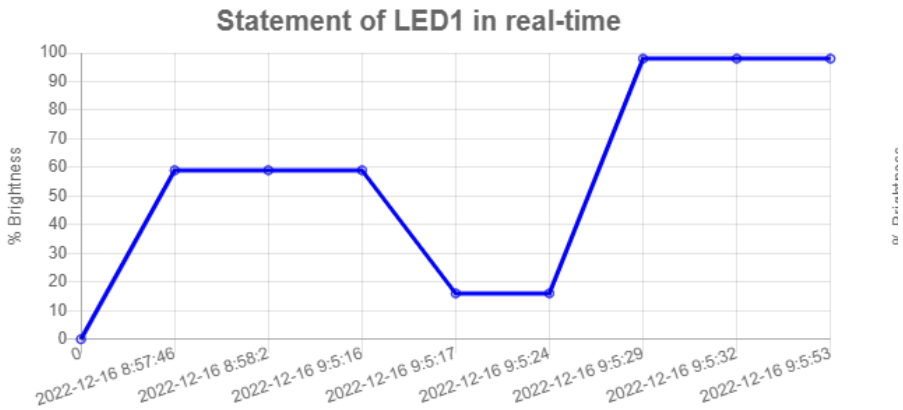
8.2 Theo dõi độ sáng của hệ thống đèn trong thời gian thực qua biểu đồ

Bên dưới giao diện phần điều chỉnh độ sáng đèn, ta có thể quan sát trạng thái (độ sáng) từng đèn trên thời gian thực qua đồ thị đường (hình 11). Mỗi khi độ sáng của đèn được điều chỉnh, sự thay đổi đó đều được cập nhập lên biểu đồ. Đối với client, biểu đồ chỉ có thể cập nhập trạng thái đèn kể từ thời điểm kết nối.



Hình 11 Giao diện đồ thị trạng thái đèn trên thời gian thực

Mỗi lần điều chỉnh độ sáng của một đèn trong hệ thống đèn, biểu đồ của đèn được thay đổi nói riêng sẽ cập nhập lại giá trị sáng mới và các đèn còn lại cũng sẽ được cập nhập giá trị sáng của chúng tại thời điểm đấy lên biểu đồ. Như hình 11, ta thấy tại thời điểm cập nhập độ sáng mới nhất, đèn led 3 vừa được thay đổi độ sáng đèn, các đèn khác vẫn giữ độ sáng cũ tại thời điểm đó.



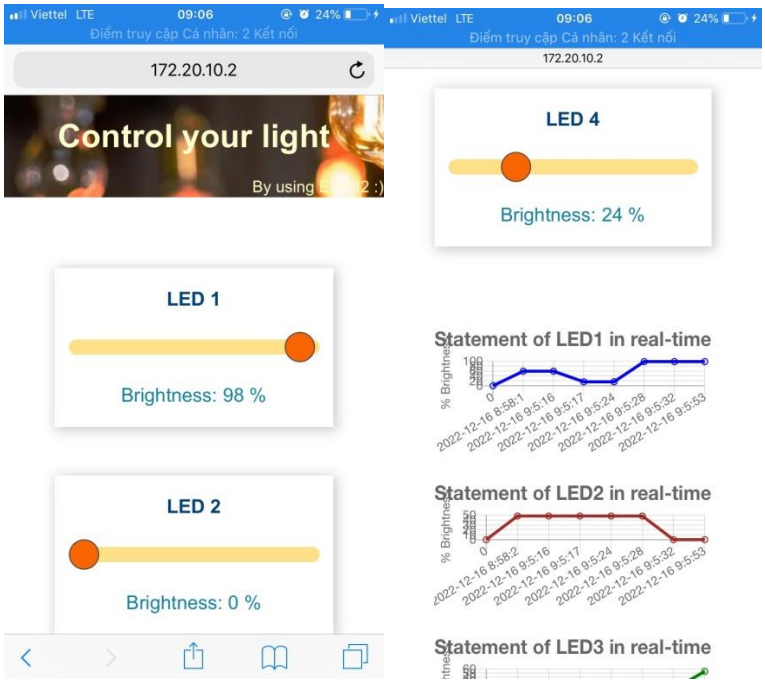
Hình 12 Đồ thị trạng thái của đèn LED 1 trong thời gian thực.

Qua đồ thị đường ở hình 12, ta thấy độ sáng đèn LED 1 được thay đổi thế nào trong thời gian thực tới lần cập nhập gần nhất.

8.3 Website ứng dụng từ điện thoại

Chỉ cần sử dụng trong cùng mạng với server, các client đều có thể truy cập url vào website để điều khiển hệ thống đèn. Khi độ sáng của đèn được thay đổi, giá trị sáng mới, biểu đồ đều sẽ được cập nhập với tất cả các client. Đối với client kết nối sau, client chỉ có thể theo dõi trạng thái đèn kể từ thời điểm kết nối.

Dưới đây là website ứng dụng được truy cập từ trình duyệt của điện thoại.

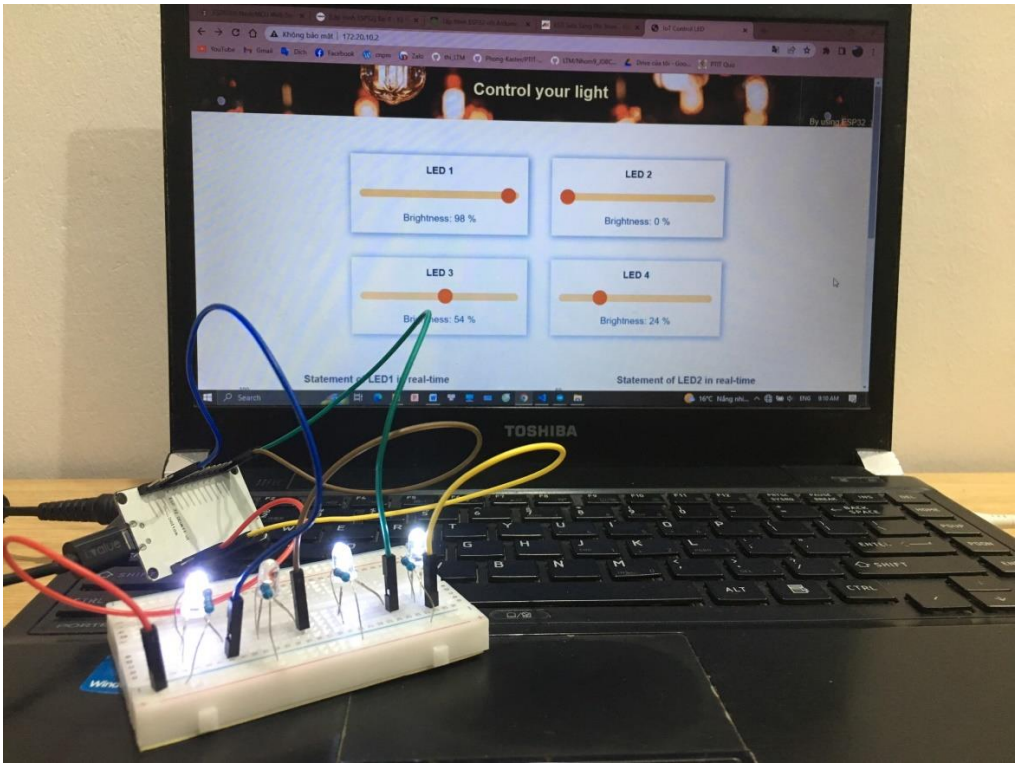


Hình 13 Website hoạt động trên điện thoại

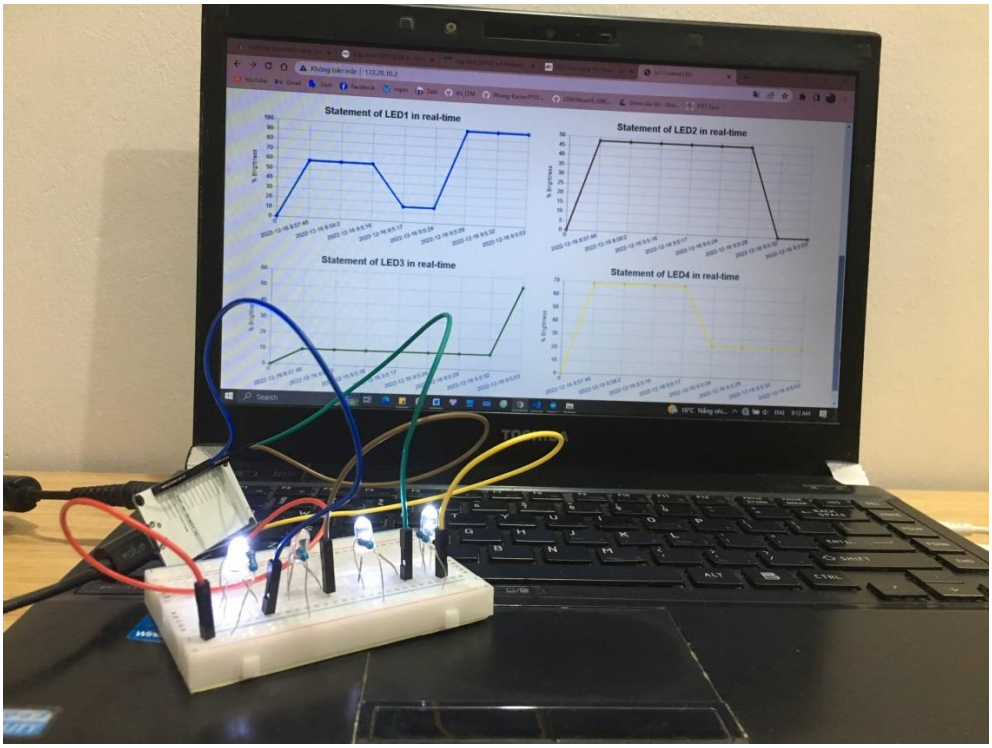
9. Kết quả đạt được

Dưới đây là kết quả thu được thực tế đối với hệ thống đèn và ứng dụng mà chúng em xây dựng:

Ứng dụng website có thể điều khiển hệ thống đèn bật/tắt, thay đổi độ sáng đèn theo mong muốn cũng như theo dõi theo thời gian thực những thay đổi, độ sáng của đèn.



Hình 14 Hệ thống hoạt động trong thực tế 1



Hình 15 Hệ thống hoạt động trong thực tế 2

PHẦN KẾT LUẬN

Với đề tài "Sử dụng Module IoT ESP 32 trong giám sát và điều khiển hệ thống chiếu sáng qua ứng dụng", đã giúp chúng em hiểu hơn về các giải pháp IoT ứng dụng trong cuộc sống và ứng dụng được kiến thức học được qua môn học "Xây dựng các hệ thống nhúng" vào xây dựng một ứng dụng website giám sát và điều khiển đèn sử dụng ESP 32.

Ứng dụng xây dựng được đã đáp ứng được những tiêu chí đặt ra như: Từ website, ta có thể điều khiển bật/tắt của đèn cũng như điều chỉnh độ sáng đèn theo mong muốn và đặc biệt có thể theo dõi trạng thái, độ sáng đèn qua biểu đồ mỗi lần cập nhập thay đổi. Không chỉ vậy, nếu sử dụng cùng mạng cục bộ với ESP 32, ta có thể truy cập được vào website thông qua đường dẫn, tất cả các thông số đều được cập nhập trên tất cả các thiết bị kết nối.

Bên cạnh đấy, do điều kiện và kiến thức chưa đầy đủ, ứng dụng mà em xây dựng còn nhiều hạn chế như: dữ liệu độ sáng, các thay đổi của hệ thống đèn đều chưa được sao lưu mỗi lần ngắt truy cập; website mới chỉ được kiểm tra có thể hoạt động tốt với 3 client (máy khách) kết nối và chưa được kiểm tra được với trường hợp nhiều hơn;…

Trong tương lai, ứng dụng cần hoàn thiện, bổ sung nhiều hơn các chức năng, nâng cấp thêm giao diện để có thể đưa vào ứng dụng.

TÀI LIỆU THAM KHẢO

[1]	[Online]. Available: https://randomnerdtutorials.com/esp32-websocket-server-arduino/.
[2]	[Online]. Available: https://www.makerstore.vn/product/04172456/.
[3]	[Online]. Available: https://mualinhkien.vn/led-sieu-sang-phi-3mm-goi-10-con.
[4]	[Online]. Available: https://dientutuonglai.com/dien-tro-200-ohm-1-4w-1-5-vong-mau.html.
[5]	[Online]. Available: https://hshop.vn/products/day-cam-breadboard-duc-cai-20cm-bo-25-soi-don-loai-tot-m-f-jumper-wire.
[6]	[Online]. Available: https://vietnamese.electronics-breadboard.com/sale-7669708-5a-400-point-8-2-5-5-0-85cm-breadboard-small.html.
[7]	[Online]. Available: https://docs.arduino.cc/learn/starting-guide/the-arduino-software-ide.
[8]	[Online]. Available: https://code.visualstudio.com/.

PHỤ LỤC