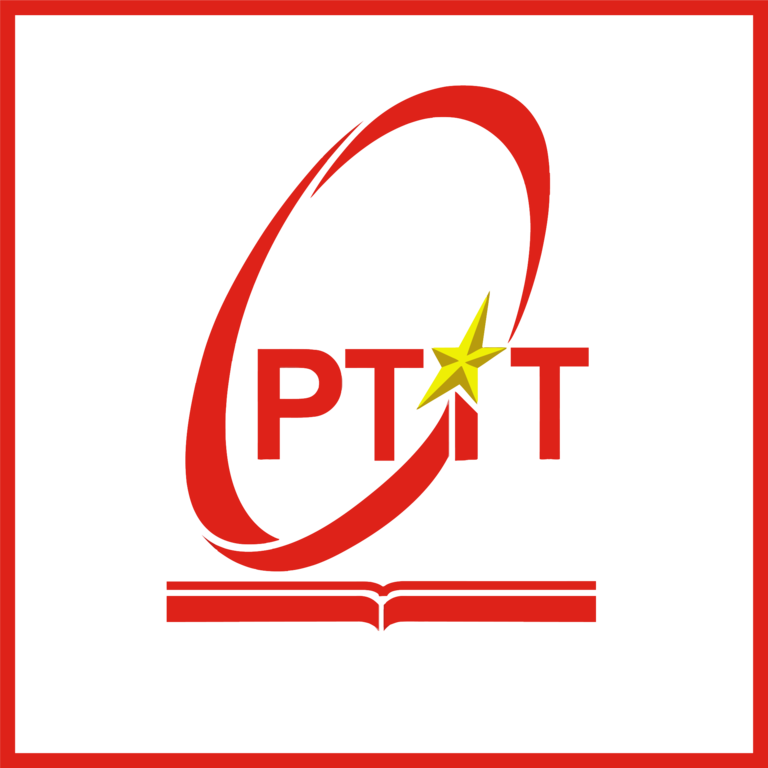
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA KỸ THUẬT ĐIỆN TỬ I



**BÀI TẬP LỚN**

**MÔN HỌC: XỬ LÝ TÍN HIỆU TRONG HỆ THỐNG TRUYỀN THÔNG**

*ĐỀ TÀI:*

*Hệ thống giám sát và điều khiển thiết bị nhà thông minh*

|  |  |  |
| --- | --- | --- |
| Giảng viên | : | Trần Tuấn Anh |
| Sinh viên | : | Vũ Quang Trường |
| Mã sv | : | B19DCDT250 |
| Lớp | : | D19DTMT02 |
|  |  | ‘ |

HÀ NỘI – 201...

**MỤC LỤC**

[LỜI CẢM ƠN 4](#_Toc152932187)

[CHƯƠNG 1: ĐẶT VẤN ĐỀ 5](#_Toc152932188)

[1.1 Giới thiệu về hệ thống nhà thông minh. 5](#_Toc152932189)

[1.2 Đặt bài toán 7](#_Toc152932190)

[1.3 Nội dung nghiên cứu. 7](#_Toc152932191)

[CHƯƠNG 2: TÌM HIỂU CÁC LINK LIỆN VÀ GIAO THỨC 8](#_Toc152932192)

[2.1 Giao thức MQTT. 8](#_Toc152932193)

[**2.1.1 Khái niêm. 8**](#_Toc152932194)

[**2.1.2 Cách MQTT hoạt động 8**](#_Toc152932195)

[**2.1.3 Ưu điểm của MQTT 9**](#_Toc152932196)

[**2.1.4 Nhược điểm của MQTT 9**](#_Toc152932197)

[2.2 WebSocket 10](#_Toc152932198)

[**2.2.1. WebSocket là gì? 10**](#_Toc152932199)

[**2.2.2 Các sự kiện chính trong WebSocket 10**](#_Toc152932200)

[**2.2.3 Ưu và nhược điểm của WebSocket 11**](#_Toc152932201)

[2.3 NodeMCU ESP8266 12](#_Toc152932202)

[**2.3.1 giới thiệu. 12**](#_Toc152932203)

[**2.3.2 Cấu tạo. 12**](#_Toc152932204)

[**2.3.3 Cấu hình sơ đồ bảng phát triển NodeMCU 13**](#_Toc152932205)

[2.4 Động cơ servo sg-90 14](#_Toc152932206)

[2.5 Cảm biến DHT11 15](#_Toc152932207)

[2.6 Cảm biến ánh sáng. 16](#_Toc152932208)

[2.7 Cảm biến khí Gas MQ2 17](#_Toc152932209)

[2.8. *MÀN HÌNH OLED 1.3IN GIAO TIẾP BẰNG I2C* 18](#_Toc152932210)

[CHƯƠNG 3 : LẬP TRÌNH THIẾT KẾ DỰ ÁN 20](#_Toc152932211)

[3.1 Sơ đồ liên kết 20](#_Toc152932212)

[3.2 Sơ đồ kết nối 21](#_Toc152932213)

[3.3 Thiết kế xây dựng web 21](#_Toc152932214)

[**3.3.1 Backend. 21**](#_Toc152932215)

[**3.3.2 Frontend. 22**](#_Toc152932216)

[3.4 Cơ sở dữ liệu Mysql. 26](#_Toc152932217)

[3.5 Lập trình ESP8266. 26](#_Toc152932218)

[KẾT LUẬN 31](#_Toc152932219)

[TÀI LIỆU THAM KHẢO 32](#_Toc152932220)

**DANH MỤC CÁC HÌNH VẼ**

[Hình 1.1: Mô hình nhà thông minh 14](#_Toc135475757)

[Hình 2.1 Sơ đồ hoạt động MQTT 41](#_Toc135475774)

[Hình 2.2 Cấu tạo esp8266 43](#_Toc135475775)

[Hình 2.3 Gpio của esp8266 44](#_Toc135475776)

[Hình 2.4 Hình ảnh thực tế động cơ servo sg-90…………………………………….…45](#_Toc135475777)

[Hình 2.5 Cảm biến DHT11 46](#_Toc135475778)

[Hình 2.6 Cảm biến ánh sáng 47](#_Toc135475779)

[Hình 2.7 Cảm biến khí Gas 49](#_Toc135475780)

[Hình 2.8 Màn hình Oled chuẩn truyền I2C 51](#_Toc135475781)

[Hình 3.1 Sơ đồ kết nối mạch 51](#_Toc135475781)

[Hình 3.2 Trang web 52](#_Toc135475782)

[Hình 3.3 Database Mysql 59](#_Toc135475783)

# LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành nhất đến Thầy Tuấn Anh vì sự hướng dẫn, giảng dạy và động viên trong quá trình thực hiện đồ án "Xây dựng Hệ thống giám sát và điều khiển thiết bị nhà thông minh". Sự quan tâm và sự hỗ trợ của Thầy đã đóng góp rất lớn vào thành công của dự án.

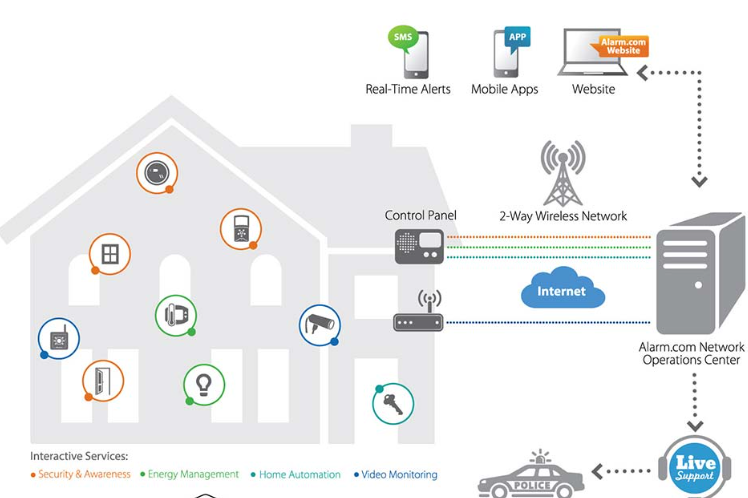
Chúng em cảm kích sự tận tâm và kiến thức sâu sắc mà Thầy đã chia sẻ với chúng em. Thầy đã không ngừng truyền đạt kiến thức chuyên môn và giúp chúng em hiểu rõ hơn về hệ thống iot và giao thức MQTT. Thầy đã luôn sẵn lòng trả lời những câu hỏi và hỗ trợ chúng em vượt qua những khó khăn trong quá trình nghiên cứu và thiết kế.

Cuối cùng, chúng em xin gửi lời cảm ơn sâu sắc đến Thầy. Dự án này đã trở thành một trải nghiệm học tập và phát triển rất đáng nhớ đối với chúng em. Chúng em sẽ tiếp tục ứng dụng những kiến thức đã học được và không ngừng nỗ lực để phát triển và trở thành những kỹ sư chuyên nghiệp.

Em xin chân thành cảm ơn!

# CHƯƠNG 1: ĐẶT VẤN ĐỀ

## Giới thiệu về hệ thống nhà thông minh.



##### Hình 1.1: Mô hình nhà thông minh

Nhà thông minh (tiếng Anh là "Smart Home") hoặc hệ thống nhà thông minh là một ngôi nhà căn hộ được trang bị hệ thống tự động tiên tiến dành cho điều khiển đèn chiếu sáng, nhiệt độ, truyền thông đa phương tiện, an ninh, cửa tự động và nhiều tính năng khác nhằm mục đích làm cho cuộc sống ngày càng tiện nghi, an toàn và góp phần sử dụng hợp lý các nguồn tài nguyên.

Theo wiseGeek, một ngôi nhà (hoặc căn hộ) được coi là "thông minh" bởi vì hệ thống máy tính điện thoại của nó có thể giám sát được nhiều khía cạnh diễn ra xung quanh ngôi nhà của chúng ta thường ngày.Một trong những ví dụ cơ bản nhất của nhà thông minh là một hệ thống kiểm soát mức độ chiếu sáng của hệ thống đèn giúp tiết kiệm điện và phù hợp với khung cảnh, chẳng hạn như cài đặt đèn ánh sáng nhẹ cho các bữa tiệc tối. Hệ thống cũng có thể điều chỉnh rèm cửa theo yêu cầu, kiểm soát. nhiệt độ, hệ thống báo sự cố khí gas, hệ thống đóng mở cửa tự động, hệ thống phòng ngừa trộm…Nhà thông minh ngoài ra còn có một số ứng dụng sáng tạo hơn, gồm hệ thống điều khiển giải trí loa đài với công suất khác nhau, hệ thống tưới nước...

Các chức năng này có thể được thực hiện nhờ việc ghép nối các thiết bị điệntrong nhà với bộ điều khiển trung tâm thông qua hệ thống máy tính điện thoại để có thể theo dõi các trạng thái và ra các quyết định điều khiển phù hợp.

Nhà thông minh đã được hình dung trong các tác phẩm khoa học viễn tưởng từ nhiều năm nhưng nó chỉ trở thành hiện thực kể từ thế kỷ 20 sau sự phát triển rộng rãi của ngành điện ­ điện tử và những tiến bộ nhanh chóng của công nghệ thông tin, theo Wikipedia. Công nghệ nhà thông minh ngày càng trở nên tinh vi,các tín hiệu điều khiển và giám sát được mã hóa truyền đi thông qua hệ thống dây dẫn hoặc mạng không dây đến các bộ chuyển mạch, ổ điện được lập trình sẵn để vận hành các đồ gia dụng và thiết bị điện tử ở trong ngôi nhà. Sự tự động hóa của ngôi nhà đặc biệt có ích cho người lớn tuổi và người tàn tật, những người muốn sống tự lập.

Việc lắp đặt các sản phẩm thông minh đem lại cho ngôi nhà và chủ nhân của nó rất nhiều lợi ích – tương tự như những lợi ích mà công nghệ và máy tính cá nhân đã đem lại cho chúng ta 30 năm qua, bao gồm: sự tiện nghi, tiết kiệm thời gian, tiền bạc và năng lượng. Các thành phần của hệ thống nhà thông minh bao gồm các cảm biến (như cảm biến nhiệt độ, cảm biến chuyển động, cảm biến ánh sáng, cảm biến âm thanh, cảm biến khí gas…), các bộ điều khiển (PLC, vi điều khiển…), máy tính, khối thu phát tín

hiệu (wifi, Bluetooth…)và các thiết bị chấp hành khác (Rơ le trung gian…). Nhờ hệ thống cảm biến, các bộ điều khiển và máy tính điện thoại mà con người có thể theo dõi các trạng thái bên trong ngôi nhà dù chúng ta đang ở bất cứ đâu, để đưa ra các quyết định điều khiển các thiết bị chấp hành một cách phù hợp nhằm đảm bảo sự an toàn cũng như đem lại cho người sử dụng một môi trường sống tốt nhất.

## Đặt bài toán

Xuất phát từ nhu cầu mở rộng quy mô ứng dụng hệ thống nhà thông minh không chỉ tập trung cho các nhà biệt thự, khách sạn… phục vụ cho các tầng lớp giàu có thượng lưu, mà còn đem lại cho người có mức thu nhập trung bình có được cơ hội sở hữu ngôi nhà thông minh với một số tính năng tự động nhất định của riêng mình trong khả năng kinh tế có hạn. Vì vậy trong phạm vi của đề tài đề xuất xây dựng mô hình nhà nhà thông minh, đảm bảo cho ngôi nhà có được một số tính năng tự động như sau:

* Thu thập dữ liệu nhiệt độ độ ẩm từ cảm biến DHT11, cường độ sáng từ cảm biến ánh sáng, đo khí gas từ cảm biến khí gas với vi xử lý trung tâm esp8266.
* Hiển thị dữ liệu thu được lên màn hình Oled 1.3 inch
* Gửi dữ liệu thu được lên MQTT broker
* Hiển thị các dữ liệu thu được từ cảm biến lên web để dễ theo dõi
* Điều khiển các thiết bị như servo ,led ,còi từ web
* Tự động phát hiện và cảnh báo qua còi và gửi tín hiệu lên web khi phát hiện rò rỉ khí gas
* Tổng hợp dữ liệu thu được lên database mysql

## Nội dung nghiên cứu.

* Nội dung 1: nghiên cứu tài liệu về KIT NodeMCU ESP8266 và giao thức MQTT và websocket.
* Nội dung 2: Nghiên cứu các mô hình nhà thông minh .
* Nội dung 3:Thiết kế web hiển thị dữ liệu và điều khiển thiết bị
* Nội dung 4: Thiết kế và tính toán thiết kế mạch phần cứng cho thiết bị.
* Nội dung 5: Thử nghiệm và điều chỉnh hệ thống cũng như chương trình để hệ thống được tối ưu. Đánh giá các thông số của mô hình so với thực tế.
* Nội dung 6: Viết báo cáo thực hiện
* Nội dung 7: Bảo vệ luận án.

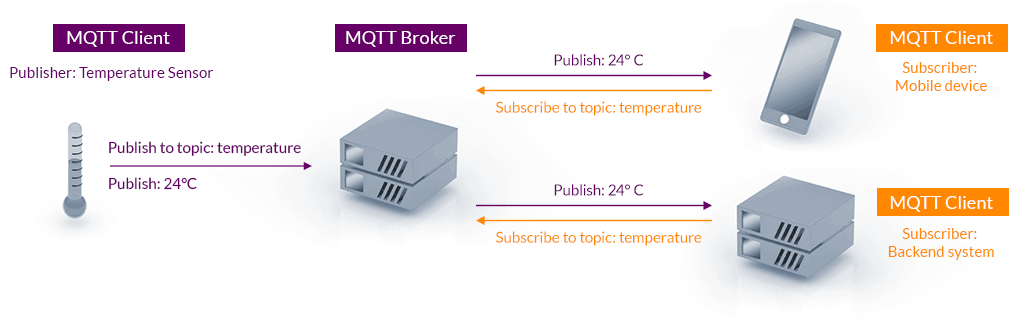
# CHƯƠNG 2: TÌM HIỂU CÁC LINK LIỆN VÀ GIAO THỨC

## 2.1 Giao thức MQTT.

### 2.1.1 Khái niêm.

MQTT (Message Queuing Telemetry Transport) là một giao thức nhắn tin tiêu chuẩn OASIS cho Internet of Things (IoT). Nó được thiết kế như một phương tiện truyền tải tin nhắn publish/subscribe (xuất bản/đăng ký) cực kỳ nhẹ, lý tưởng để kết nối các thiết bị từ xa với băng thông mạng thấp. MQTT ngày nay được sử dụng trong nhiều ngành công nghiệp, chẳng hạn như ô tô, sản xuất, viễn thông, dầu khí, v.v.

### 2.1.2 Cách MQTT hoạt động



##### Hình 2.1 Sơ đồ hoạt động MQTT

Một phiên MQTT được chia thành bốn giai đoạn: kết nối, xác thực, giao tiếp và kết thúc. Client (máy khách) bắt đầu bằng cách tạo kết nối Transmission Control Protocol Internet Protocol (TCP/IP) tới broker bằng cách sử dụng cổng tiêu chuẩn hoặc cổng tùy chỉnh được xác định bởi các nhà phát triển broker. Khi tạo kết nối, điều quan trọng là phải nhận ra rằng máy chủ có thể tiếp tục một phiên cũ nếu nó được cung cấp ID máy khách mà được sử dụng lại.

Các cổng tiêu chuẩn là 1883 cho giao tiếp không mã hóa và 8883 cho giao tiếp được mã hóa – sử dụng Lớp cổng bảo mật (SSL) Bảo mật lớp truyền tải (TLS). Trong quá trình giao tiếp SSL/TLS, máy khách cần kiểm chứng và xác thực máy chủ. Máy khách cũng có thể cung cấp tính xác thực máy khách cho broker trong quá trình giao tiếp. Broker có thể sử dụng điều này để xác thực máy khách. Mặc dù không phải là một phần cụ thể của đặc trưng MQTT, nhưng các broker đã trở thành thông lệ để hỗ trợ xác thực máy khách bằng SSL/TLS phía máy khách.

### 2.1.3 Ưu điểm của MQTT

MQTT cho phép hệ thống [SCADA](https://mesidas.com/scada/) của bạn truy cập dữ liệu IIoT. MQTT mang lại nhiều ưu điểm vượt trội cho hệ thống của bạn:

Phân phối thông tin hiệu quả hơn

Tăng khả năng mở rộng

Giảm đáng kể tiêu thụ băng thông mạng

Giảm tốc độ cập nhật xuống giây

Rất phù hợp cho điều khiển và viễn thám

Tối đa hóa băng thông đang sử dụng

Chi phí đầu tư cực kỳ thấp

Rất an toàn (bảo mật dựa trên sự cấp phép)

Đã được sử dụng bởi các ngành công nghiệp dầu khí, Amazon, Facebook và các doanh nghiệp lớn khác

Giảm thiểu thời gian phát triển

Giao thức publish/subscribe thu thập nhiều dữ liệu hơn với ít băng thông hơn so với giao thức cũ.

### 2.1.4 Nhược điểm của MQTT

MQTT có chu kỳ truyền chậm hơn so với CoAP.

Tài nguyên của MQTT hoạt động dựa trên subscribe động, trong khi CoAP sử dụng hệ thống tài nguyên tĩnh – ổn định.

MQTT không được mã hóa. Thay vào đó, nó sử dụng TLS / SSL để mã hóa bảo mật. Rất khó để tạo ra một mạng MQTT có thể mở rộng toàn cầu.

## 2.2 WebSocket

### 2.2.1. WebSocket là gì?

WebSocket là một giao thức truyền tải dữ liệu hai chiều (full-duplex) trên web, cho phép truyền tải dữ liệu trong thời gian thực giữa trình duyệt web và máy chủ. WebSocket cho phép một kết nối duy trì giữa máy khách và máy chủ, vì vậy các thông tin có thể được gửi đi và nhận lại một cách hiệu quả và nhanh chóng mà không cần phải thiết lập kết nối mới mỗi khi truyền tải thông tin.

WebSocket thường được sử dụng để phát triển các ứng dụng web thời gian thực như trò chơi trực tuyến, ứng dụng trò chuyện và các ứng dụng cập nhật dữ liệu trực tiếp. WebSocket được thiết kế để thay thế các giải pháp truyền tải dữ liệu trực tiếp khác như Ajax hoặc Comet, vì nó có hiệu suất cao hơn và tiết kiệm năng lượng hơn.

### 2.2.2 Các sự kiện chính trong WebSocket

WebSocket cho phép các ứng dụng web tạo kết nối thời gian thực, có khả năng truyền dữ liệu hai chiều liên tục giữa máy chủ và trình duyệt. Các sự kiện chính của WebSocket bao gồm:

Mở kết nối (Open): Khi một kết nối WebSocket được thiết lập giữa trình duyệt và máy chủ, sự kiện "open" được kích hoạt.

Nhận dữ liệu (Message): Khi trình duyệt hoặc máy chủ nhận được dữ liệu mới, sự kiện "message" được kích hoạt.

Gửi dữ liệu (Send): Khi trình duyệt hoặc máy chủ gửi dữ liệu, sự kiện "send" được kích hoạt.

Đóng kết nối (Close): Khi kết nối WebSocket bị đóng, sự kiện "close" được kích hoạt.

Lỗi (Error): Nếu có lỗi trong quá trình thiết lập hoặc sử dụng kết nối WebSocket, sự kiện "error" được kích hoạt.

Các sự kiện này đều cung cấp các thông tin cần thiết cho ứng dụng để xử lý các tình huống khác nhau khi sử dụng WebSocket.

### 2.2.3 Ưu và nhược điểm của WebSocket

WebSocket cung cấp một cách kết nối liên tục, toàn diện hơn để truyền tải dữ liệu giữa hai bên, so với các phương pháp truyền tải dữ liệu truyền thống như HTTP. Dưới đây là một số ưu và nhược điểm của WebSocket:

Ưu điểm:

Thời gian đáp ứng nhanh hơn: Vì WebSocket cho phép duy trì kết nối mở liên tục giữa máy khách và máy chủ, việc truyền tải dữ liệu được thực hiện nhanh hơn so với HTTP.

Kích thước dữ liệu nhỏ hơn: WebSocket giảm thiểu kích thước gói tin và số lần giao tiếp yêu cầu/phản hồi so với HTTP. Do đó, kích thước dữ liệu được truyền tải sẽ nhỏ hơn, làm giảm lưu lượng mạng.

Công nghệ phù hợp cho ứng dụng web thời gian thực: WebSocket là một công nghệ phù hợp cho các ứng dụng web thời gian thực như trò chuyện trực tuyến, trò chơi trực tuyến, cập nhật dữ liệu thời gian thực và các ứng dụng tương tác người dùng khác.

Không cần phải gửi yêu cầu mới để nhận dữ liệu mới: Khi sử dụng HTTP, trình duyệt phải gửi yêu cầu mới để nhận dữ liệu mới từ máy chủ. Tuy nhiên, với WebSocket, máy chủ có thể gửi dữ liệu mới cho trình duyệt mà không cần phải yêu cầu mới.

Nhược điểm:

Tốn nhiều tài nguyên: WebSocket sử dụng kết nối mở liên tục giữa trình duyệt và máy chủ, vì vậy nó sẽ tốn nhiều tài nguyên hơn so với HTTP.

Không được hỗ trợ bởi một số trình duyệt cũ: Một số phiên bản cũ của các trình duyệt không hỗ trợ WebSocket, điều này có thể gây khó khăn cho các nhà phát triển khi triển khai ứng dụng của họ.

Thiếu tính năng bảo mật: Mặc dù WebSocket cung cấp các tính năng bảo mật cơ bản, nhưng nó vẫn có thể bị tấn công bởi các cuộc

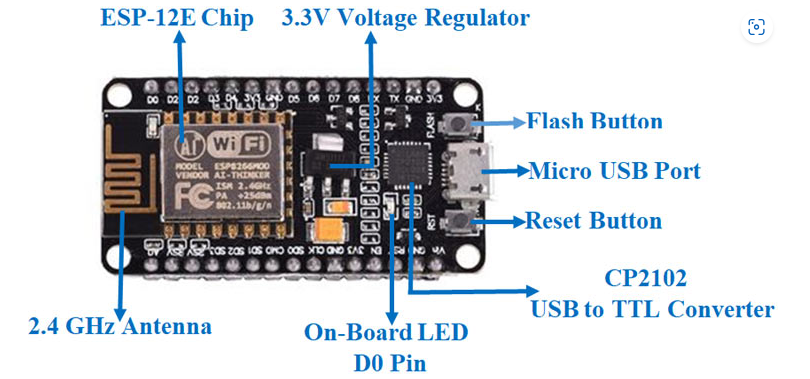
## 2.3 NodeMCU ESP8266

### 2.3.1 giới thiệu.

ESP8266 không chỉ đơn thuần là một vi xử lý thông thường, mà còn là một vi xử lý tích hợp mạnh mẽ đặc biệt được thiết kế với mục tiêu giải quyết các thách thức về kết nối và truyền thông trong lĩnh vực Internet of Things (IoT). Được ra đời như một tương lai cho việc kết nối, ESP8266 mang trong mình sức mạnh để biến các thiết bị thông thường thành những phần tử thông minh, có khả năng tương tác và kết nối với Internet một cách đơn giản và hiệu quả.

Với khả năng tích hợp chức năng Wi-Fi, ESP8266 trở thành cầu nối thông tin quan trọng giữa thế giới vật lý và không gian kỹ thuật số. Nó giúp cho các thiết bị, từ những đèn đơn giản đến các hệ thống phức tạp, có khả năng kết nối và giao tiếp trực tiếp với mạng Internet.

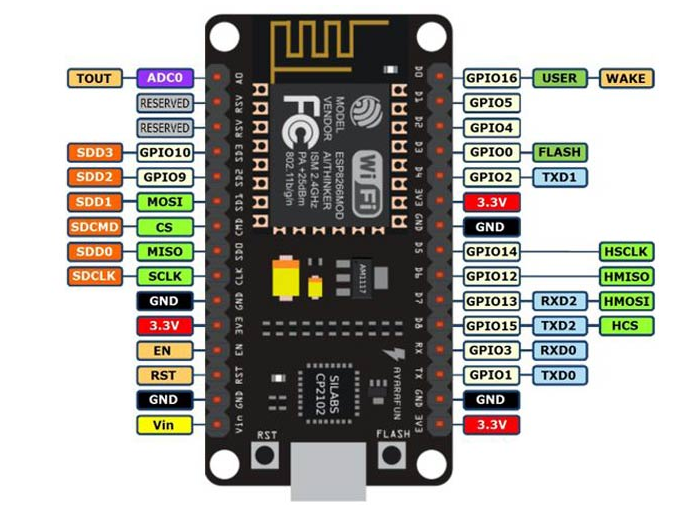
### 2.3.2 Cấu tạo.



##### Hình 2.2 Cấu tạo esp8266

* Bộ vi điều khiển: CPU RISC 32-bit Tensilica Xtensa LX106
* Điện áp hoạt động: 3.3V
* Điện áp đầu vào: 7-12V
* Chân I / O kỹ thuật số (DIO): 16
* Chân đầu vào tương tự (ADC): 1
* UARTs: 1
* SPI: 1
* I2Cs: 1
* Bộ nhớ Flash: 4 MB
* SRAM: 64 KB
* Tốc độ đồng hồ: 80 MHz
* USB-TTL dựa trên CP2102 được bao gồm trên bo mạch, cho phép Plug n Play
* Ăng-ten PCB
* Mô-đun có kích thước nhỏ để phù hợp thông minh bên trong các dự án IoT của bạn

### 2.3.3 Cấu hình sơ đồ bảng phát triển NodeMCU



##### Hình 2.3 Gpio của esp8266

Power:

* Micro-USB: NodeMCU có thể được cấp nguồn thông qua cổng USB.
* 3.3V: Chân này có thể được cấp nguồn 3.3V cho bo mạch.
* GND: Chân nối đất.
* Vin: Chân này dùng để cấp nguồn từ nguồn điện bên ngoài.

Control Pins:

* EN (Enable): Chân này kích hoạt hoặc vô hiệu hóa bo mạch.
* RST (Reset): Chân reset, sử dụng để làm mới bo mạch.

Analog Pin:

* A0: Chân đo điện áp tương tự trong khoảng 0-3.3V.

GPIO Pins (General-Purpose Input/Output):

* GPIO1 đến GPIO16: NodeMCU có 16 chân GPIO đầu vào/đầu ra.

SPI Pins (Serial Peripheral Interface):

* SD1, CMD, SD0, CLK: Chân kết nối với giao diện SPI.

UART Pins (Universal Asynchronous Receiver-Transmitter):

* TXD0, RXD0: Giao diện UART0.
* TXD2, RXD2: Giao diện UART1, thường được sử dụng để tải lên phần mềm/chương trình.

I2C Pins (Inter-Integrated Circuit):

* NodeMCU hỗ trợ chức năng I2C, nhưng bạn cần tìm hiểu xem chân nào được sử dụng cho I2C, vì chúng có thể khác nhau tùy theo phiên bản cụ thể của bo mạch NodeMCU bạn đang sử dụng.

## **2.4 Động cơ servo sg-90**

- Động cơ RC Servo 9G là động phổ biến dùng trong các mô hình điều khiển nhỏ và đơn giản như cánh tay robot. Động cơ có tốc độ phản ứng nhanh, được tích hợp sẵn Driver điều khiển động cơ, dễ dàng điều khiển góc quay bằng phương pháp điều độ rộng xung PWM.

**Đặc điểm:**

Kích thước: 23mmX12.2mmX29mm

Trọng lượng: 9gram

Điện áp hoạt động: 4.2-6V

Nhiệt độ: 0 ℃ --55 ℃

Tốc độ: 0.3 giây / 60 độ



##### Hình 2.4 Hình ảnh thực tế động cơ servo sg-90

## 2.5 Cảm biến DHT11

DHT11 là một cảm biến nhiệt độ và độ ẩm giá rẻ và phổ biến, thường được sử dụng trong các dự án IoT và các ứng dụng đo lường môi trường. Dưới đây là mô tả về cấu tạo và chức năng cơ bản của cảm biến DHT11:

Cấu Trúc Fisical:

* Cảm Biến: Bên trong, DHT11 có một cảm biến nhiệt độ và một cảm biến độ ẩm, được đặt trong một vỏ chống ẩm.

Nguyên Lý Hoạt Động:

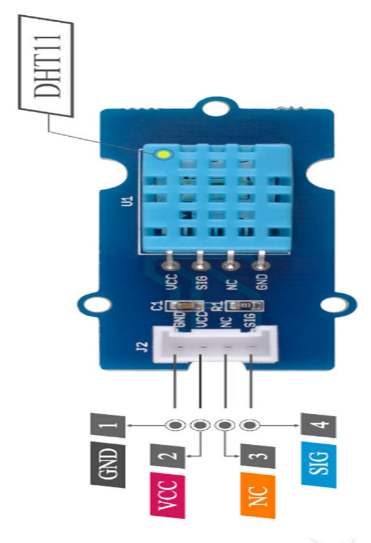
* Cảm Biến Nhiệt Độ và Độ Ẩm: Sử dụng nguyên tắc làm thay đổi điện trở của một chất nhậy cảm với nhiệt độ và độ ẩm. Điện trở của chất nhậy thay đổi theo nhiệt độ và độ ẩm xung quanh, và thông qua quá trình đọc điện trở, DHT11 có thể đo lường nhiệt độ và độ ẩm.

Chức Năng Giao Tiếp:

* Digital Signal Output: Dữ liệu đo được từ cảm biến được truyền dưới dạng tín hiệu kỹ thuật số.
* Giao Tiếp: DHT11 sử dụng giao tiếp dạng 1-wire, có nghĩa là nó chỉ yêu cầu một dây để truyền dữ liệu.

Thông Số Kỹ Thuật:

* Dải Đo Nhiệt Độ: Thường là từ 0 đến 50 độ C với độ chính xác khoảng ±2 độ C.
* Dải Đo Độ Ẩm: Thường là từ 20% đến 90% với độ chính xác khoảng ±5%.



##### Hình 2.5 Cảm biến DHT11

## 2.6 Cảm biến ánh sáng.

****

##### Hình 2.6: Cảm biến ánh sáng

Cảm biến ánh sáng quang điện trở (LDR) là 1 loại cảm biến ánh sáng đơn giản, nguyên tắc hoạt động dựa vào hiện tượng quang điện trong.

Nguyên lý làm việc của quang điện trở là khi ánh sáng chiếu vào chất bán dẫn (có thể là Cadmium sulfide – CdS, Cadmium selenide – CdSe) làm phát sinh các điện tử tự do, tức sự dẫn điện tăng lên và làm giảm điện trở của chất bán dẫn. Các đặc tính điện và độ nhạy của quang điện trở tùy thuộc vào vật liệu dùng trong chế tạo.

Cảm biến ánh sáng quang điện trở (LDR) được làm từ vật liệu bán dẫn tiếp xúc như cadmium sulphide, làm thay đổi điện trở từ vài nghìn Ohms trong bóng tối thành chỉ vài trăm Ohms khi ánh sáng chiếu vào nó.

Vật liệu được sử dụng làm chất bán dẫn cho cảm biến ánh sáng phổ biến nhất là Cadmium Sulphide ( Cds ).

Cadmium sulphide được sử dụng trong sản xuất các tế bào quang điện vì đường phản ứng quang phổ của nó gần giống với mắt người và có độ nhạy sáng cao. Thông thường, nó có bước sóng nhạy cảm đỉnh ( λp ) khoảng 560nm đến 600nm trong phạm vi phổ khả kiến.

## 2.7 Cảm biến khí Gas MQ2

Cảm biến khí Gas MQ2 là một trong những cảm biến được sử dụng rộng rãi nhất trong các dòng cảm biến MQ. Nó là một cảm biến MOS (Metal Oxide Semiconductor). Cảm biến oxit kim loại hay còn được gọi là (**Điện trở hóa trị**) vì cảm biến dựa trên sự thay đổi điện trở của cảm biến khi tiếp xúc với khí.

Module cảm biến khí Gas MQ2 khá đơn giản có hai đầu ra AO và DO, có nhiệm vụ dùng để phát hiện khí ở trong môi trường.

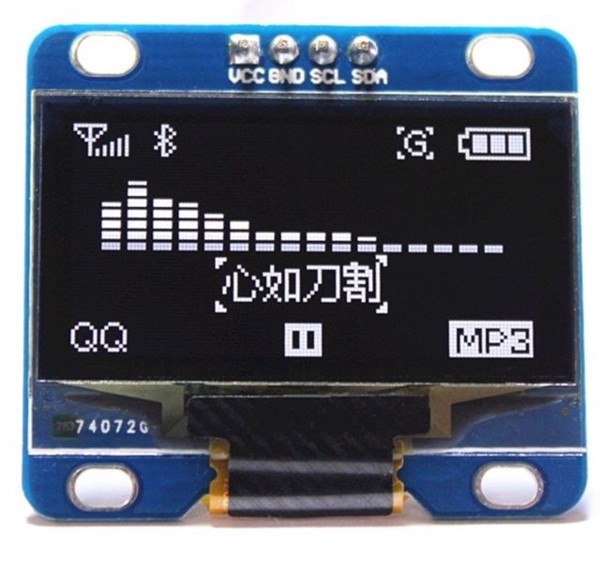


##### Hình 2.7 Cảm biến khí Gas

Sự thay đổi điện áp của chân đầu ra của cảm biến (tại chân AO) thay đổi tỷ lệ thuận với nồng độ khói/khí. Nồng độ khí càng cao, điện áp đầu ra càng cao; nồng độ khí càng thấp, điện áp đầu ra càng thấp.

## 2.8. MÀN HÌNH OLED 1.3IN GIAO TIẾP BẰNG I2C

Màn hình Oled 1.3 inch giao tiếp I2C cho khả năng hiển thị đẹp, sang trọng, rõ nét vào ban ngày và khả năng tiết kiệm năng lượng tối đa với mức chi phí phù hợp, màn hình sử dụng giao tiếp I2C cho chất lượng đường truyền ổn định và rất dễ giao tiếp chỉ với 2 chân GPIO. Màn hình OLED SH1106 với kích thước 1.3 inch, cho khả năng hiển thị hình ảnh tốt với khung hình 128×64 pixel. Ngoài ra, màn hình còn tương thích với hầu hết các vi điều khiển hiện nay thông qua giao tiếp SPI. Màn hình sử dụng driver SH1106 cùng thiết kế nhỏ gọn sẽ giúp bạn phát triển các sản phẩm DIY hoặc các ứng dụng khác một cách nhanh chóng.



##### ****Hình 2.8 Màn hình Oled chuẩn truyền I2C****

THÔNG SỐ KỸ THUẬT

* Điện áp sử dụng: 2.2~5.5VDC
* Công suất tiêu thụ: 0.04w
* Góc hiển thị: lớn hơn 160 độ

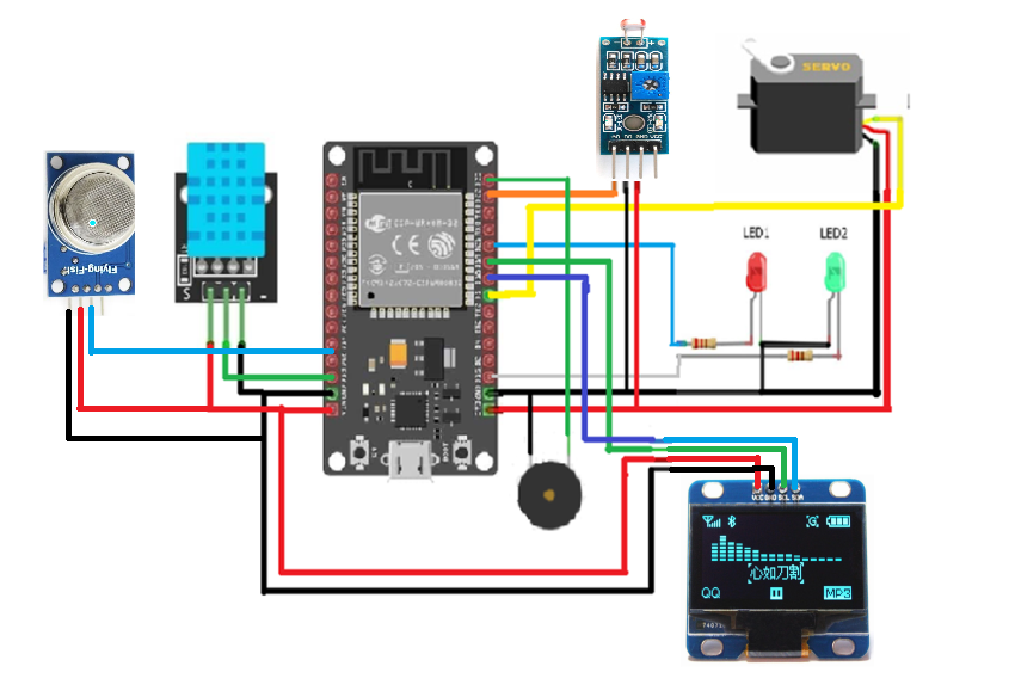
Số điểm hiển thị: 128×64 điểm.

* Độ rộng màn hình: 1.3 inch.
* Màu hiển thị: Trắng / Xanh Dương.
* Giao tiếp: I2C hoặc SPI tùy loại
* Driver: SH1106
* Kích thước 1.3 inch (128x64px)
* Góc nhìn tối đa: 160°
* Nhiệt độ làm việc: -30°V đến 80°C
* Tương thích với hầu hết các board như: Arduino, ESP8266, ESP32, STM32,

# CHƯƠNG 3 : LẬP TRÌNH THIẾT KẾ DỰ ÁN

## 3.1 Sơ đồ liên kết

## 3.2 Sơ đồ kết nối



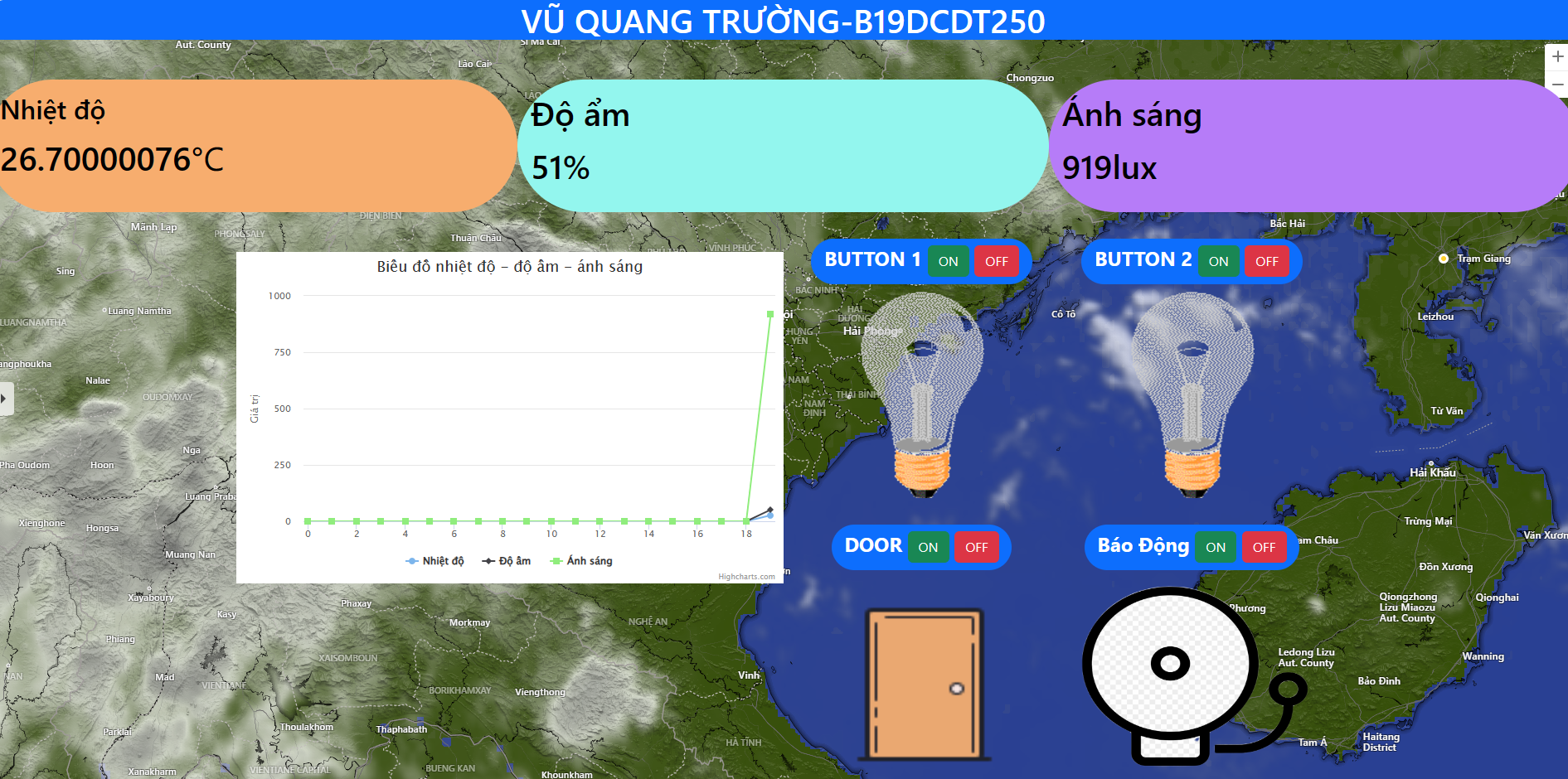
##### Hình 3.1 Sơ đồ kết nối mạch

## 3.3 Thiết kế xây dựng web

### 3.3.1 Backend.

Xây dựng trang web giúp chúng ta có thể dễ dàng quan sát và điều khiển các thiết bị. Mục tiêu :

* Hiển thị các giữ liệu nhiệt độ độ ẩm ánh sáng từ cảm biến DHT11,cường độ ánh sáng.
* Hiển thị các trạng thái của led ,servo,còi.
* Dùng biểu đồ để dễ dàng quan sát sự thay đổi của các thông số .
* Có thể điều khiển các thiết bị trên web.



##### Hinh 3.2 Trang web

### 3.3.2 Frontend.

Phần frontend được chia thành 2 phần chính là truyền nhận dữ liệu từ MQTT và WebSocket ,truyền nhận dữ liệu từ WebSocket và web.Ngôn ngữ javascrit là ngôn ngữ phù hợp để lập trình 2 phần trên.

* Truyền nhận dữ liệu từ MQTT và Websocket:

const mqtt = require('mqtt');

const WebSocket = require('ws');

const mqttBroker = 'mqtt://192.168.43.63';

const mqttTopic = 'DHT';

const mqttClient = mqtt.connect(mqttBroker);

const wss = new WebSocket.Server({ port: 3000 });

wss.on('connection', (ws) => {

console.log('WebSocket client connected');

mqttClient.subscribe(mqttTopic);

mqttClient.on('message', (topic, message) => {

if (topic === mqttTopic) {

const data = message.toString();

console.log(`MQTT sub: ${data}`);

ws.send(data);

}

});

ws.on('message', (message) => {

console.log(`Nhận từ client WebSocket: ${message}`);

try {

const jsonData = JSON.parse(message);

if (jsonData.action && jsonData.deviceId) {

mqttClient.publish(jsonData.action, jsonData.deviceId);

} else {

console.log('Định dạng JSON không hợp lệ. Cần có các thuộc tính: action, deviceId');

}

} catch (error) {

console.error('Lỗi khi phân tích JSON:', error);

}

});

* Ta sẽ khai báo host MQTT và topic để lấy dữ liệu từ MQTT
* Ta tạo một server websocket port 3000 để đẩy dữ liệu vừa nhận được lên
* Đồng thời nhận dữ liệu json từ client websocket đẩy lại MQTT với topic = action và mess = deviceId để điều khiển các thiết bị.
* Hiển thị các thiết bị nhận được lên terminal.
* Truyền nhận dữ liệu liệu từ websocket và web.

const socket = new WebSocket('ws://localhost:3000');

socket.addEventListener('open', () => {

console.log('WebSocket connected');

});

socket.addEventListener('close', () => {

console.log('WebSocket disconnected');

});

socket.addEventListener('message', (event) => {

try {

const data = JSON.parse(event.data);

if (data && data.temperature\_C !== undefined && data.humidity\_percent !== undefined && data.light\_lux !== undefined && data.gas\_value !== undefined) {

let nhietDo = data.temperature\_C;

let doAm = data.humidity\_percent;

let anhSang = data.light\_lux;

let gas = data.gas\_value;

document.getElementById("ND2").innerHTML = nhietDo + "℃";

document.getElementById("DA2").innerHTML = doAm + "%";

document.getElementById("AS2").innerHTML = anhSang + "lux";

function bieuDo() {

dataAS.push(anhSang); dataAS.shift();

dataDA.push(doAm); dataDA.shift();

dataND.push(nhietDo); dataND.shift();

}

bieuDo();

* Phần này ta sẽ kết nối với websocket với port 3000
* Ta phải xử lý dữ liệu json nhận về gán biến cho các gía trị
* Sau đó đẩy sang file HTML để hiển thị lên web và biểu đồ

function aon() {

if (confirm("Bạn có muốn bật") == true) {

document.getElementById("myImage").src = "bongdensang1.gif";

document.getElementById('nutnhan').style.backgroundColor = "yellow";

sendWebSocketData('led1', 'ON');

}

}

function aof() {

if (confirm("bạn có muốn tắt") == true) {

document.getElementById("myImage").src = "bongden1.gif"

document.getElementById('nutnhan').style.backgroundColor = "yellow"

}

sendWebSocketData('led1', 'OFF');

}

function bon() {

if (confirm("bạn có muốn bật ") == true) {

document.getElementById("myImage1").src = "bongdensang1.gif"

document.getElementById('nutnhan2').style.backgroundColor = "yellow"

}

sendWebSocketData('led2', 'ON');

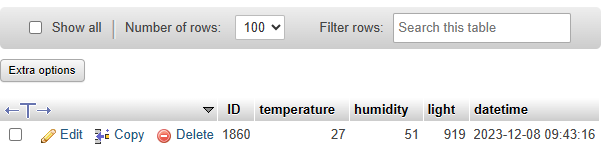
}

* Trong file này còn các chương trình câu điều kiện if else để xử lý các giá trị để thay đổi màu sắc phụ thuộc vào giá trị nhận về.
* Ta tạo các chương trình nhận dữ liệu từ các biến id từ HTML để gửi lại dữ liệu lên websocket.

## 3.4 Cơ sở dữ liệu Mysql.

MySQL là **hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới** và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh.

Ta sẽ tạo một bảng với các giá trị là ID, temperature, humidity, light, datetime



##### Hình 3.3 Database Mysql

Ta sẽ tạo 1 file javascirt lấy dữ liệu từ websocket và xử lý dữ liệu json nhận về gắn vào các giá trị mới giống như phần đẩy dữ liệu hiển thị lên web nhưng lên này ta sẽ gán giá trị đó vào các giá trị phù hợp trên bảng đã tạo

## 3.5 Lập trình ESP8266.

ESP8266, được xem như bộ não chính trong hệ thống, đảm nhận nhiều nhiệm vụ quan trọng nhằm xử lý thông tin từ cảm biến và điều khiển các thiết bị. Dưới đây là mô tả chi tiết về các nhiệm vụ cụ thể của ESP8266:

* Kết nối với MQTT Broker.
* Đọc dữ liệu cảm biến DHT11,cường độ sáng,khí gas.
* Chuyển đổi dữ liệu tành mã json sub lên MQTT.
* Nhận dữ liệu từ MQTT để điều khiển đèn và servo.
* Đọc dữ liệu từ cảm biến khí gas để cảnh báo khi phát hiện khí Gas
* Hiển thị dữ liệu lên màn hình Oled

#include <ESP8266WiFi.h>

#include <Wire.h>

#include <PubSubClient.h>

#include <ArduinoJson.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SH110X.h>

#include <DHT.h>

#include <Servo.h>

#define LIGHT\_SENSOR\_PIN A0

#define GAS\_SENSOR\_PIN 2

#define DHTPIN 14

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

#define wifi\_ssid "Truong"

#define wifi\_password "12345678a"

#define mqtt\_server "192.168.43.63"

#define dht\_topic "DHT"

#define led1\_topic "led1"

#define led2\_topic "led2"

#define door\_topic "door"

#define coi\_topic "coi"

Servo doorServo;

const int ledPin1 = 16;

const int ledPin2 = 0;

const int coi = 13;

const int servo = 12;

* Ta khai báo các thư viện cần dùng, khai báo thông tin của wiffi và MQTT để esp có thể kết nói, khai báo các topic và các chân gpio.

void loop() {

if (!client.connected()) {

reconnect();

}

client.loop();

delay(2000);

float h = dht.readHumidity();

float t = dht.readTemperature();

if (isnan(h) || isnan(t)) {

Serial.println("Failed to read from DHT sensor!");

return;

}

DynamicJsonDocument jsonDocument(256);

jsonDocument["temperature\_C"] = t;

jsonDocument["humidity\_percent"] = h;

int lightIntensity = analogRead(LIGHT\_SENSOR\_PIN);

jsonDocument["light\_lux"] = lightIntensity;

int gasValue = digitalRead(GAS\_SENSOR\_PIN);

jsonDocument["gas\_value"] = gasValue;

String jsonString;

serializeJson(jsonDocument, jsonString);

* Đọc dữ liệu từ các cảm biến độ và độ ẩm DHT11,cảm biến ánh sáng khí gas sau đó chuyển thành mã json và sub giữ liệu lên MQTT.

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

Serial.print("Message arrived [");

Serial.print(topic);

Serial.print("] ");

String message = "";

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

message += (char)payload[i];

}

Serial.println(message);

if (String(topic) == led1\_topic) {

if (message == "ON") {

digitalWrite(ledPin1, HIGH);

} else if (message == "OFF") {

digitalWrite(ledPin1, LOW);

}

} else if (String(topic) == led2\_topic) {

if (message == "ON") {

digitalWrite(ledPin2, HIGH);

}else if (message == "OFF") {

digitalWrite(ledPin2, LOW);

}

}else if (String(topic) == door\_topic) {

if (message == "OPEN") {

doorServo.write(180);

}else if (message == "CLOSE") {

doorServo.write(0);

}

}else if (String(topic) == coi\_topic) {

if (message == "ON") {

digitalWrite(coi, HIGH);

}else if (message == "OFF") {

digitalWrite(coi, LOW);

}

}

}

* Tạo hàm call back để nhận dữ liệu từ MQTT sau đó tạo các chương trình để điều khiển các thiết bị phù hợp với các topic nhận được.

// display temperature

display.setTextSize(1);

display.setCursor(0,0);

display.print("Temperature: ");

display.print(t);

display.print(" ");

display.cp437(true);

display.write(167);

display.print("C");

// display Humidity

display.setTextSize(1);

display.setCursor(0,10);

display.print("Humidity: ");

display.print(h);

display.print(" %");

// display light

display.setTextSize(1);

display.setCursor(0,20);

display.print("Light: ");

display.print(lightIntensity);

display.print(" Lux");

display.setTextSize(2);

display.setCursor(0,30);

display.print("Gas ");

display.print(gasValue);

display.print(" !!");

display.display();

* Hiển thị dữ liệu lên LCD:

ESP8266 gửi dữ liệu đã xử lý và được đọc từ các cảm biến lên màn hình LCD để hiển thị thông tin một cách trực quan và thuận tiện.

# KẾT LUẬN

Dự án Thiết kế Nhà Thông minh bằng ESP8266 đã đạt được một số kết quả tích cực và mang lại những lợi ích đáng kể trong việc tối ưu hóa quản lý và tiện ích cho người sử dụng. Việc tích hợp ESP8266, một module IoT mạnh mẽ, đã mang lại khả năng kết nối đa dạng và linh hoạt, giúp tối ưu hóa việc điều khiển các thiết bị trong nhà thông minh.

Các thành công chính của dự án bao gồm khả năng kiểm soát ánh sáng, nhiệt độ, và các thiết bị điện tử khác từ web. Sự tích hợp linh hoạt và sự dễ dàng trong việc mở rộng chức năng đã tạo ra một hệ thống nhà thông minh đáng tin cậy và hiệu quả.

Tuy nhiên, vẫn còn một số điểm cần cải thiện để nâng cao hiệu suất và trải nghiệm người dùng:

* Tăng cường lớp bảo mật để ngăn chặn các mối đe dọa từ bên ngoài.
* Sử dụng giao thức an toàn để đảm bảo an toàn trong việc truyền dữ liệu.
* Tối ưu hóa giao diện người dùng để làm cho ứng dụng trở nên dễ sử dụng hơn và thân thiện với người dùng.
* Thêm các tính năng mô phỏng để người dùng có thể xem trước trạng thái hiện tại của các thiết bị.
* Tích hợp chế độ tiết kiệm năng lượng để tối ưu hóa việc sử dụng nguồn điện.
* Sử dụng cảm biến thông minh để tự động điều chỉnh các thiết bị dựa trên điều kiện

# TÀI LIỆU THAM KHẢO

1. [Eclipse Mosquitto](https://mosquitto.org/)
2. [Công nghệ nhận diện khuôn mặt - QuanTriMang.com](https://quantrimang.com/cong-nghe/cong-nghe-nhan-dien-khuon-mat-51590)
3. [What is ESP8266, how it works and what you can do with ESP32? – Circuit Schools](https://www.circuitschools.com/what-is-esp32-how-it-works-and-what-you-can-do-with-esp32/)