ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA VẬT LÝ – VẬT LÝ KỸ THUẬT

CHUYÊN NGÀNH VẬT LÝ TIN HỌC

----------------🙞🙜----------------

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC

*Đề tài:*

MẠCH ĐIỀU KHIỂN QUẠT ĐA NĂNG

SVTH: Nguyễn Cao Kỳ

CBHD: TS. Nguyễn Huỳnh Tuấn Anh[[1]](#footnote-1)

----------------------------------

TP HỒ CHÍ MINH – 2021

ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA VẬT LÝ – VẬT LÝ KỸ THUẬT

CHUYÊN NGÀNH VẬT LÝ TIN HỌC

----------------🙞🙜----------------

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC

*Đề tài:*

MẠCH ĐIỀU KHIỂN QUẠT ĐA NĂNG

SVTH: Nguyễn Cao Kỳ

CBHD: TS. Nguyễn Huỳnh Tuấn Anh[[2]](#footnote-2)

----------------------------------

TP HỒ CHÍ MINH – 2021

Lời Cảm Ơn

Trong suốt quá trình thực hiện khóa luận, mặc dù gặp phải rất nhiều khó khăn nhưng

với sự giúp đỡ và hỗ trợ kịp thời từ quý Thầy Cô và các bạn nên khóa luận đã được

hoàn thành đúng tiến độ.

Đầu tiên chúng em xin chân thành gửi lời cảm ơn đến thầy Nguyễn Huỳnh Tuấn Anh,

người trực tiếp hướng dẫn chúng em trong khóa luận này. Thầy đã tận tình hướng

dẫn, chỉ bảo kinh nghiệm quý báu cũng như hỗ trợ nhiều phương tiện thí nghiệm

trong suốt quá trình tìm hiểu, nghiên cứu và thực hiện đề tài.

Em cũng xin cảm ơn tất cả các thành viên trong bộ môn Vật lý Tin học đã có

những ý kiến đóng góp, bổ sung, cũng như động viên, khích lệ tinh thần giúp chúng

em hoàn thành tốt đề tài.

Mặc dù đã cố gắng hoàn thiện khóa luận, nhưng trong quá trình soạn thảo và

nghiên cứu thực hiện đề tài, với kiến thức còn hạn chế nên vẫn có thể còn nhiều hạn

chế, thiếu sót. Nhóm thực hiện đề tài rất mong nhận được sự góp ý và chỉ dẫn của

quý Thầy Cô cùng các bạn sinh viên.

Chúng em xin chân thành cảm ơn!

Tp. Hồ Chí Minh, tháng … năm 2021

Nguyễn Cao Kỳ

# Mục lục

[Mục lục i](#_Toc93446102)

[Bảng các từ viết tắt iii](#_Toc93446103)

[Danh sách các hình v](#_Toc93446104)

[Lời mở đầu 1](#_Toc93446105)

[Chương 1: TỔNG QUAN 2](#_Toc93446106)

[I- TỔNG QUAN VỀ PHẦN CỨNG SỬ DỤNG TRONG KHÓA LUẬN 2](#_Toc93446107)

[1.1 Động cơ DC không chổi than 2](#_Toc93446108)

[1.1.1 Giới thiệu 2](#_Toc93446109)

[1.1.2.1 Stator 3](#_Toc93446110)

[1.1.2.2 Rotor 3](#_Toc93446111)

[1.1.2.3 Cảm biến vị trí Hall 3](#_Toc93446112)

[1.1.3 Nguyên lý hoạt động 3](#_Toc93446113)

[1.1.4 Ứng dụng 4](#_Toc93446114)

[1.1.5 Ưu điểm và nhược điểm 4](#_Toc93446115)

[1.2 WeMos D1 Mini ESP8266 5](#_Toc93446116)

[1.2.1 Giới thiệu 5](#_Toc93446117)

[1.2.2 Thông số kỹ thuật 5](#_Toc93446118)

[1.2.3 Sơ đồ chân 6](#_Toc93446119)

[1.3 IC ổn áp 7805 8](#_Toc93446120)

[1.3.1 Giới thiệu 8](#_Toc93446121)

[1.3.2 Thông số kỹ thuật 8](#_Toc93446122)

[1.3.3 Sơ đồ chân 9](#_Toc93446123)

[1.4 Màn hình OLED 0.96 inch I2C 9](#_Toc93446124)

[1.4.1 Giới thiệu 9](#_Toc93446125)

[1.4.2 Thông số kỹ thuật 10](#_Toc93446126)

[1.4.3 Sơ đồ chân 10](#_Toc93446127)

[1.5 Module điều khiển hồng ngoại IR1838 10](#_Toc93446128)

[1.5.1 Giới thiệu 11](#_Toc93446129)

[1.5.2 Thông số kỹ thuật 11](#_Toc93446130)

[1.5.3 Sơ đồ chân 11](#_Toc93446131)

[II- TỔNG QUAN VỀ PHẦN MỀM SỬ DỤNG TRONG KHÓA LUẬN 12](#_Toc93446132)

[2.1 Flutter 12](#_Toc93446133)

[2.1.1 Giới thiệu 12](#_Toc93446134)

[2.1.2 Kiến trúc Flutter 13](#_Toc93446135)

[2.1.2.1 Widget 14](#_Toc93446136)

[2.1.2.2 Gestures 15](#_Toc93446137)

[2.1.2.3 State 15](#_Toc93446138)

[2.1.2.4 Layers 16](#_Toc93446139)

[2.1.3 Dart - Ngôn ngữ lập trình Flutter 17](#_Toc93446140)

[2.1.4 Điểm mạnh 17](#_Toc93446141)

[III- CƠ SỞ LÝ THUYẾT 18](#_Toc93446142)

[3.1 Kết nối phần cứng với phần mềm 18](#_Toc93446143)

[3.2 Giao tiếp giữa vi điểu khiển với động cơ 21](#_Toc93446144)

[CHƯƠNG 2: THỰC HÀNH 23](#_Toc93446145)

[II- PHẦN CỨNG 23](#_Toc93446146)

[III- PHẦN MỀM 31](#_Toc93446147)

[Chương 3: TỔNG KẾT 35](#_Toc93446148)

[I- KẾT LUẬN 35](#_Toc93446149)

[II- ƯU ĐIỂM 35](#_Toc93446150)

[III- KHUYẾT ĐIỂM 35](#_Toc93446151)

[IV- HƯỚNG PHÁT TRIỂN 36](#_Toc93446152)

[TÀI LIỆU THAM KHẢO 37](#_Toc93446153)

# Bảng các từ viết tắt

DC Direct Current

IR Infrared

TCP/IP Transmission Control Protocol/ Internet Protocol

PWM Pulse Width Modulation

I2C Inter-Intergrated Circuit

USB Universal Serial Bus

WPA Wi-Fi Protected Access

TCP Transmission Control Protocol

UDP User Datagram Protocol

I/O Input/Output

GPIO General Purpose Input Output Pins

IC Integrated Circuit

RX Transmit

TX Receive

GHz Gigahertz

SCL Clock line

SDA Data line

SDK Software Development Kit

UI User Interface

SMTP Simple Mail Transfer Protocol

FTP File Transfer Protocol

HTTP Hypertext Transfer Protocol

MQTT MQ Telemetry Transport

CoAP Constrained Application Protocol

CPU Central Processing Unit

DNS Domain Name System

SYN Synchronize

SSDP Simple Service Discovery Protocol

OLED Organic Light-Emitting Diode

# Danh sách các hình

[Hình 1.1 Cấu tạo của động cơ DC không chổi than 2](file:///C:\Users\Admin\Desktop\BAOCAO.docx#_Toc93393839)

[Hình 1.2 Sơ đồ chân của Wemos D1 Mini 6](#_Toc93393840)

[Hình 1.3 Mô hình IC 7805 8](file:///C:\Users\Admin\Desktop\BAOCAO.docx#_Toc93393841)

[Hình 1.4 Sơ đồ chân IC 7805 9](file:///C:\Users\Admin\Desktop\BAOCAO.docx#_Toc93393842)

[Hình 1.5 OLED 0.96 inch I2C 9](file:///C:\Users\Admin\Desktop\BAOCAO.docx#_Toc93393843)

[Hình 1. 6 Module điều khiển hồng ngoại 11](file:///C:\Users\Admin\Desktop\BAOCAO.docx#_Toc93393844)

[Hình 1.7 Flutter và những ứng dụng 12](file:///C:\Users\Admin\Desktop\BAOCAO.docx#_Toc93393845)

[Hình 1.8 Các widget đặc trưng 14](file:///C:\Users\Admin\Desktop\BAOCAO.docx#_Toc93393846)

[Hình 1.9 Các thao tác tương tác 15](file:///C:\Users\Admin\Desktop\BAOCAO.docx#_Toc93393847)

[Hình 1.10 Thao tác cập nhật lại giao diện 15](file:///C:\Users\Admin\Desktop\BAOCAO.docx#_Toc93393848)

[Hình 1.11 Các lớp của Flutter 16](file:///C:\Users\Admin\Desktop\BAOCAO.docx#_Toc93393849)

[Hình 1.12 Dart- Ngôn ngữ lập trình trong Flutter 17](file:///C:\Users\Admin\Desktop\BAOCAO.docx#_Toc93393850)

[Hình 1. 13 Giao thức IoT được ánh xạ tới mô hình TCP/IP 18](file:///C:\Users\Admin\Desktop\BAOCAO.docx#_Toc93393851)

[Hình 1.14 Tiến trình bắt tay 3 bước 20](#_Toc93393852)

[Hình 1.15 Mô hình truyền dữ liệu 21](file:///C:\Users\Admin\Desktop\BAOCAO.docx#_Toc93393853)

[Hình 1.16 Đồ thị dạng xung điều chế PWM 22](file:///C:\Users\Admin\Desktop\BAOCAO.docx#_Toc93393854)

[Hình 2.1 BH55FT15-07 23](file:///C:\Users\Admin\Desktop\BAOCAO.docx#_Toc93446066)

[Hình 2.2 Hình 2. 2 Sơ đồ chân chiếu theo thứ tự như ảnh 2.1 23](#_Toc93446067)

[Hình 2.3 Sơ đồ mạch ổn áp 5V 24](file:///C:\Users\Admin\Desktop\BAOCAO.docx#_Toc93446068)

[Hình 2.4 WeMos D1 Mini ESP8266 25](file:///C:\Users\Admin\Desktop\BAOCAO.docx#_Toc93446069)

[Hình 2.5 Lưu cấu hình wifi bằng EEPROM 26](file:///C:\Users\Admin\Desktop\BAOCAO.docx#_Toc93446070)

[Hình 2.6 Sơ đồ mạch kết nối OLED với ESP 27](file:///C:\Users\Admin\Desktop\BAOCAO.docx#_Toc93446071)

[Hình 2.7 Sơ đồ mạch kết nối biến trở với ESP 28](#_Toc93446072)

[Hình 2.8 Sơ đồ mạch kết nối mạch thu hồng ngoại với ESP 29](file:///C:\Users\Admin\Desktop\BAOCAO.docx#_Toc93446073)

[Hình 2.9 Remote IR1838 30](file:///C:\Users\Admin\Desktop\BAOCAO.docx#_Toc93446074)

[Hình 2.10 Sơ đồ thuật toán khởi động 30](#_Toc93446075)

[Hình 2.11 Thư viện ping\_discover\_network\_forked 31](#_Toc93446076)

[Hình 2.12 Thư viện wifi\_iot 31](#_Toc93446077)

[Hình 2.13 Giao diện kết nối thiết bị 32](file:///C:\Users\Admin\Desktop\BAOCAO.docx#_Toc93446078)

[Hình 2.14 Nguyên lý dò thiết bị 32](file:///C:\Users\Admin\Desktop\BAOCAO.docx#_Toc93446079)

[Hình 2.15 Giao diện điều khiển và cấu kết nối 33](file:///C:\Users\Admin\Desktop\BAOCAO.docx#_Toc93446080)

[Hình 2.16 Quá trình gửi dữ liệu 34](file:///C:\Users\Admin\Desktop\BAOCAO.docx#_Toc93446081)

# Lời mở đầu

Trong quá trình công nghiệp hóa và hiện đại hóa, máy móc là một vật dụng không thể thiếu với đời sống mỗi người. Cùng với đó không thể không kể tới sự ra đời của các loại động cơ khác nhau. Mỗi động cơ đảm nhận một vai trò nhiệm vụ cùng với ưu điểm, nhược điểm khác nhau. Đặc biệt là động cơ DC không chổi than.

Bước vào thời đại cách mạng công nghiệp 4.0, các loại quạt có cánh kiểu cũ cũng dần nhường chỗ cho những loại quạt thông minh. Dù chưa phổ biến ở Việt Nam, một số nhà phân phối đã bắt đầu nhập các mẫu mã quạt thông minh để bán cho đối tượng khách hàng là những người có thu nhập cao, sẵn sàng bỏ tiền để trải nghiệm một thiết bị đem đến một tổng thể thông minh cho cả căn nhà.

Quạt thông minh trên thị trường hiện nay chủ yếu tập trung vào khả năng điều khiển từ xa. Đây là một chức năng cơ bản giúp loại bỏ được sự bất tiện của việc di chuyển để bật/tắt hay chỉnh mức gió quạt. Có hai kiểu điều khiển từ xa là thao tác trên điện thoại thông qua kết nối không dây (Wifi, Zigbee, Z-Wave, IR…) hoặc ra lệnh thông qua trợ lý ảo thông minh (Google Assistant, Apple HomeKit hoặc Amazon Alexa).

Để hiểu rõ hơn về động cơ DC không chổi than nói chung và quạt thông minh nói riêng, em quyết định chọn đề tài này.

# Chương 1: TỔNG QUAN

# I- TỔNG QUAN VỀ PHẦN CỨNG SỬ DỤNG TRONG KHÓA LUẬN

## 1.1 Động cơ DC không chổi than

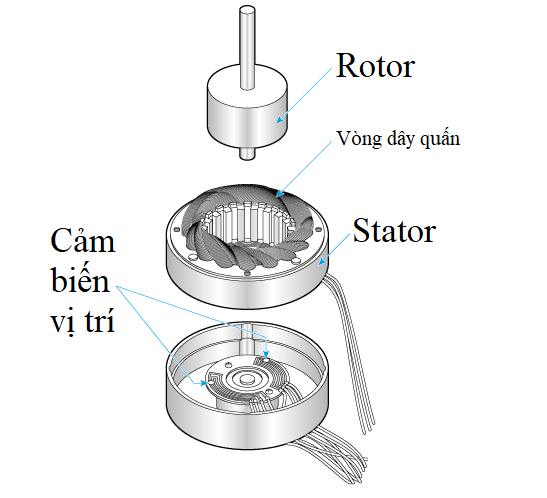
### **1.1.1 Giới thiệu**

Trên thị trường hiện nay có rất nhiều động cơ DC khác nhau. Nhưng chủ yếu bao gồm hai loại chính là động cơ không chổi than và động cơ chổi than.

Động cơ chổi than là loại đông cơ sử dụng cổ góp và chổi than để cung cấp dòng điện cho cuộn dây. Hoạt động dựa trên cơ chế tiếp xúc giữa cổ góp, chổi than để giúp cung cấp điện năng cho cuộn dây.

Động cơ không chổi than là loại động cơ được hoạt động dựa vào từ trường vĩnh cữu và cảm biến xác định vị trí, không sử dụng chổi than giúp triệt tiêu ma sát, giảm tiếng ổn cho động cơ máy vận hành êm ái, sử dụng tiết kiệm điện.

So với động cơ chổi than, động cơ không chổi than có những ưu điểm nổi trội hơn về hiệu suất lẫn tính năng. Nhưng ít phổ biến do giá thành đắt.

**1.1.2 Cấu tạo**

Hình 1.1 Cấu tạo của động cơ DC không chổi than

Cấu tạo của động cơ DC không chổi than bao gồm ba bộ phận chính: dây quấn stator, rotor và cảm biến vị trí.

#### 1.1.2.1 Stator

Stator của động cơ DC không chổi than được cấu tạo từ các lá thép kỹ thuật điện ghép cách điện với nhau với các cuộn dây quấn được đặt phía trong của stator

#### 1.1.2.2 Rotor

Rotor bao gồm trục động cơ và các nam châm vĩnh cửu được bố trí xen kẽ giữa các cực bắc và nam. Để đạt được moment xoắn cực đại yêu cầu phải có mật độ từ trường cao cho nên người ta phải lựa chọn chất liệu nam châm tương ứng

#### 1.1.2.3 Cảm biến vị trí Hall

Do đặc thù của sức phản điện động của động cơ DC không chổi than có dạng hình thang nên cấu hình điều khiển thông thường của nó cũng cần có cảm biến xác định vị trí của từ trường rotor trong tương quan với các pha của cuộn dây stator. Để làm được điều đó, người ta thường sử dụng cảm biến hiệu ứng Hall, có thể gọi tắt là Hall sensor.

### **1.1.3 Nguyên lý hoạt động**

Động cơ không chổi than không lan truyền dòng điện đến các cuộn dây bởi vì các cuộn dây này không hề nằm trên rotor. Thay vào đó, rotor chính là một nam châm vĩnh cửu, nó có các cuộn dây không quay mà được cố định vào vị trí ở trên stator. Vì các cuộn dây này không di chuyển nên người ta không cần chổi than cũng như cổ góp.

Trong động cơ không chổi than, người ta sẽ quay nam châm vĩnh cửu bằng cách thay đổi hướng của từ trường được tạo ra bởi các cuộn dây được sắp xếp đứng yên xung quanh nó. Để điều khiển chuyển động quay đó, bạn cần điều chỉnh độ lớn kết hợp hướng của dòng điện chạy vào các cuộn dây này.

Nguyên lý hoạt động của động cơ không chổi than là xác định vị trí của của rotor để điều khiển quá trình đóng ngắt các khóa bán dẫn cung cấp nguồn cho các cuộn dây stator theo trình tự hợp lý. Nếu không động cơ không thể thay đổi chiều quay và khởi động tự động được.

Hoạt động dựa trên tương tác của từ trường do stator và nam châm vĩnh cửu trên rotor tạo ra, Khi dòng điện chạy qua một trong ba cuộc dây sẽ tạo ra lực hút các nam châm vĩnh cửu trái dấu ở gần.

Để điều khiển tốc độ của động cơ không chổi than người ta sử dụng phương pháp điều chế độ rộng xung. Bộ điều khiển xác định vị trí trục rotor và xuất điện áp điều khiển đóng/mở các khóa bán dẫn cấp điện áp cho động cơ.

### **1.1.4 Ứng dụng**

Động cơ không chổi than từ lâu đã được sử dụng nhiều trong việc chế tạo máy giặt, máy điều hòa không khí cũng như các thiết bị điện tử gia dụng khác. Và gần đây, động cơ này còn xuất hiện trong các loại quạt, chính vì hiệu suất cao của chúng đã góp phần giảm thiểu đáng kể lượng điện năng tiêu thụ cho quạt.

Đồng thời cũng là một bộ phận không thể thiếu của máy hút chân không. Trong một số trường hợp, sự thay đổi trong việc thiết lập chương trình điều khiển của chúng sẽ dẫn đến một bước đột phá lớn về tốc độ quay. Đây chính là một ví dụ về khả năng điều khiển bậc nhất của các động cơ này.

Với khả năng điều khiển chính xác khiến cho chúng đặc biệt phù hợp với những chiếc máy bay không người lái đa hành trình, bằng cách điều khiển chính xác vị trí, tốc độ quay của mỗi cánh quạt.

### **1.1.5 Ưu điểm và nhược điểm**

a) Ưu điểm:

Động cơ DC không chổi than có các ưu điểm của động cơ đồng bộ nam châm vĩnh cửu như: tỷ lệ moment quán tính lớn, tỷ lệ công suất/khối lượng cao. Do máy được kích từ bằng nam châm vĩnh cửu nên trên rotor hiệu suất động cơ cao hơn.

Động cơ kích từ nam châm vĩnh cửu không cần chổi than và vành trượt nên không tốn chi phí bảo trì chổi than. Ta cũng có thể thay đổi đặc tính động cơ bằng cách thay đổi đặc tính của nam châm kích từ và cách bố trí nam châm trên rotor.

b) Nhược điểm:

Với những bước đột phá đã được cải tiến của động cơ không chổi, giúp khắc phục hoàn toàn những nhược điểm của động cơ có chổi than, tăng tuổi thọ và hiệu quả cao, đồng thời, động cơ DC không chổi than được chế tạo từ nam châm vĩnh cửu và cảm biến Hall để điều khiển tốc độ và momen động cơ nên có giá thành tương đối cao.

## 1.2 WeMos D1 Mini ESP8266

### **1.2.1 Giới thiệu**

WeMos D1 Mini tiền thân của nó là một vi mạch Wifi giá rẻ có hỗ trợ bộ giao thức TCP/IP và có thể tích hợp vào vi điều khiển được sàn xuất ở Trung Quốc. Chip ESP8266 lần đầu tiên được các nhà sản xuất phương Tây chú ý tới là vào năm 2014 với module ESP-01 do bên thứ 3 Ai Thinker sản xuất. Module này cho phép các vi điều khiển kết nối mạng Wifi và thực hiện các kết nối TCP/IP đơn giản bằng cách sử dụng các kiểu Hayes (tập lệnh AT).

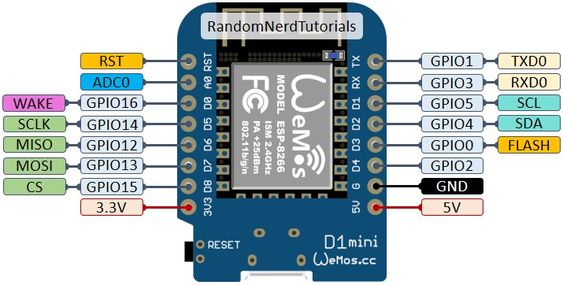
WeMos D1 Mini là board mach được phát triển dựa trên Module Esp8266-12S, là thiết bị nhỏ gọn được tích hợp Wifi để dễ dàng thực hiện các ứng dụng thu thập dữ liệu và điều khiển thiết bị thông qua Wifi.

WeMos D1 Mini được hỗ trợ để nạp trực tiếp thông qua USB mà không cần thực hiện các thao tác thủ công (sử dụng nút reset và flash) như NodeMCU nha72m đơn giản hóa quá trình làm việc với board mạch

### **1.2.2 Thông số kỹ thuật**

* Vi điều khiển: ESP8266EX
* Điện áp hoạt động: 3.3V
* Số chân I/O: 11 (tất cả các chân I/O đều có Interrupt/PWM/I2C/One-wire, trừ chân D0)
* Số chân Analog Input: 1 (điện áp vào tối đa 3.3V)
* Bộ nhớ Flash: 4MB
* Giao tiếp: Cable Micro USB
* Nguồn áp: 5V được cung cấp qua cổng Micro USB
* Wifi: 2.4 GHz
* Hỗ trợ bảo mật: WPA/WPA2
* Tích hợp giao thức TCP/IP
* Kích thước: 34.2mmx25.6mm
* Lập trình trên các ngôn ngữ: C/C++, Micropython, NodeMCU – Lua

### **1.2.3 Sơ đồ chân**



Hình 1.2 Sơ đồ chân của Wemos D1 Mini

Một điều đáng lưu ý là hầu hết các loại ESP8266 hiện nay số chân GPIO thì không trùng khớp với nhãn tên trên bo mạch. Ví dụ D0 tương ứng với GPIO16 và D1 tương ứng với GPIO5.

Mỗi nhãn tên và số chân GPIO có một nhiệm vụ và chức năng đặc trưng riêng, cần chú ý khi tới nó khi sử dụng vào các dự án khác nhau.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nhãn tên | GPIO | Đầu vào | Đầu ra | Lưu ý |
| D0 | GPIO16 | Không hỗ ngắt | Không hỗ trợ PWM và I2C | Mức 1 khi khởi động |
| D1 | GPIO5 | Phù hợp | Phù hợp | Sử dụng như SCL (I2C) |
| D2 | GPIO4 | Phù hợp | Phù hợp | Sử dụng như SDA (I2C) |
| D3 | GPIO0 |  | Phù hợp | Kết nối tới nút FLASH, khởi động thất bại nếu xuống mức 0 |
| D4 | GPIO2 |  | Phù hợp | Mức 1 khi khởi động, kết nối tới LED trên mạch, khởi động thất bại nếu xuống mức 0 |
| D5 | GPIO14 | Phù hợp | Phù hợp |  |
| D6 | GPIO12 | Phù hợp | Phù hợp |  |
| D7 | GPIO13 | Phù hợp | Phù hợp |  |
| D8 | GPIO15 |  | Cần lưu ý | Khởi động thất bại khi lên mức 1 |
| RX | GPIO3 | Cần lưu ý | Không hỗ trợ chân RX | Mức 1 khi khởi động |
| TX | GPIO1 | Không hỗ trợ chân TX | Cần lưu ý | Mức 1 khi khởi động. |
| A0 | ADC0 | Analog | Không hỗ trợ |  |

Chân GPIO6 tới GPIO11 thường được kết nối kết chip FLASH trên mạch nên các chân đó thường không khuyên dùng. Một số chân khác có thể làm ESP8266 không thể khởi động nếu nó được kéo lên mức cao hoặc mức thấp. Ngoài ra ESP8266 chỉ hỗ trợ một chân analog duy nhất, nó được gọi là ADC0 với nhãn tên là A0. Đầu vào tối đa của chân ADC0 này là từ 0 tới 3.3V. Đồng thời ESP8266 cho phép tạo xung PWM và hỗ trợ ngắt trên hầu hết tất cả các chân.

## Buy L7805CV IC Linear Voltage Regulator Online at the Best Price in India1.3 IC ổn áp 7805

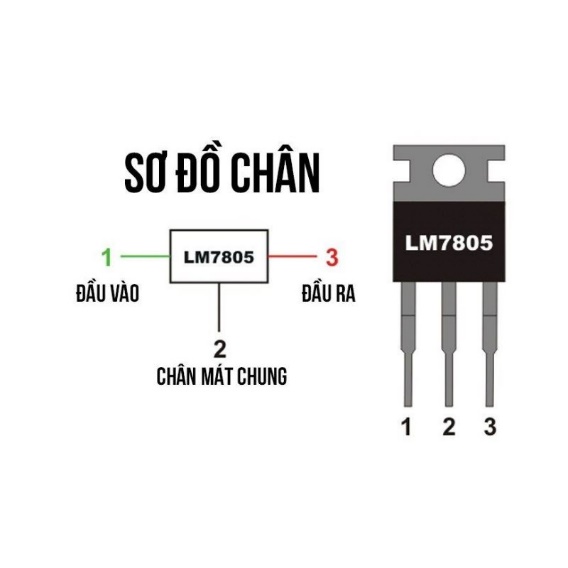
Hình 1.3 Mô hình IC 7805

### **1.3.1 Giới thiệu**

IC ổn áp 7805 là mạch tích hợp sẵn trong gói TO-220 với một điện áp đầu ra cố định là 5V, yêu cầu điện áp đầu vào tối thiếu là 7V. IC 7805 có thế cung cấp điện áp đầu ra với dòng điện lên đến 1A.

### **1.3.2 Thông số kỹ thuật**

* Điện áp đầu vào tối thiểu: 2V
* Dòng cực đại có thể duy trì: 1A.
* Dòng đỉnh: 2.2A.
* Công suất tiêu tán cực đại nếu không dùng tản nhiệt: 2W
* Công suất tiêu tán nếu dùng tản nhiệt đủ lớn: 15W

**1.3.3 Sơ đồ chân**

Hình 1.4 Sơ đồ chân IC 7805

Dựa vào số thứ tự chân trong ảnh ta có thông tin của các chân như bảng dưới đây:

|  |  |
| --- | --- |
| 1 | 12 – 24V (Ngõ vào) |
| 2 | GND |
| 3 | 5V (Ngõ ra) |

## Màn hình LCD OLED 1.3inch, 0.96inch, 0.91inch 128x64 Giao Tiếp I2C - Thiết bị văn phòng | ĐiệnThoạiDiĐộng.vn1.4 Màn hình OLED 0.96 inch I2C

Hình 1.5 OLED 0.96 inch I2C

### **1.4.1 Giới thiệu**

Màn hình OLED là một đối thủ cạnh tranh đáng gờm có thể thay thế cho màn hình LCD kiểu cũ. Có cấu tạo gồm những lớp bao gồm tấm nền, Anode, lớp hữu cơ và cathode. Phát ra ánh sáng theo cách tương tự như đèn LED.

Ưu điểm nổi bật của màn hình OLED này là những lớp nhựa hữu cơ mỏng, nhẹ mềm dẻo hơn những lớp tinh thể trên LED hay LCD nhờ vậy có thể ứng dụng OLED vào nhiều mục đích khác nhau. Độ sáng của OLED cũng tốt hơn LED, không cần đèn nền như trên LCD nên tiết kiện năng lượng hơn.

Màn hình OLED 0.96 Inch này 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 I2C. Màn hình sử dụng driver SSD1306 cùng thiết kế nhỏ gọn sẽ phù hợp để phát triển các sản phẩm DIY hoặc các ứng dụng khác một cách nhanh chóng. Đồng thời 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.

### **1.4.2 Thông số kỹ thuật**

* Điện áp sử dụng: 3.3V đến 5V (DC)
* Công suất tiêu thụ: 0.04W
* Góc hiển thị: Lớn hơn 160 độ
* Độ phân giải: 128X64 pixel
* Độ rộng màn hình: 0.96inch
* Giao tiếp: I2C
* Màu: Trắng và Đen
* Driver: SSD1306

**1.4.3 Sơ đồ chân**

|  |  |
| --- | --- |
| VCC | 3.3 - 5V |
| GND | GND |
| SCL | Xung clock |
| SDA | Truyền dữ liệu |

**1.5 Module điều khiển hồng ngoại IR1838**

**1.5.1 Giới thiệu**

Hình 1. 6 Module điều khiển hồng ngoại

Module thu hồng ngoại IR1838 nhận hồng ngoại thông qua đầu thu hồng ngoại 1838. Ánh sáng điện trở, nhiễu điện từ mạnh,được xây dựng trong IC hồng ngoại chuyên dụng, có thể làm việc dưới cường độ ánh sáng 500lux. Nó được sử dụng rộng rãi trong: âm thanh nổi, TV, máy video, đĩa máy, hộp set-top,khung ảnh kỹ thuật số, xe hơi stereo, đồ chơi điều khiển từ xa, thu vệ tinh,máy nghe nhạc đĩa cứng, máy lạnh, máy sưởi, quạt điện, ánh sáng và các thiết bị gia dụng khác.

**1.5.2 Thông số kỹ thuật**

* Kích thước: 6.4 x 7.4 x 5.1mm
* Góc thu nhận: 90 °
* Điện áp làm việc: 2,7 ~ 5,5V
* Tần số: 37.9KHz
* Phạm vi nhận: 18m
* Mạch thu: IR1838
* Dạng ngõ ra: Digital.

**1.5.3 Sơ đồ chân**

|  |  |
| --- | --- |
| VCC | 3.3 – 5V |
| GND | GND |
| IN | Truyền dữ liệu |

# II- TỔNG QUAN VỀ PHẦN MỀM SỬ DỤNG TRONG KHÓA LUẬN

2.1 Flutter

### Tìm hiểu Flutter là gì, ưu và khuyết điểm của Flutter - Fullstack Station**2.1.1 Giới thiệu**

Hình 1.7 Flutter và những ứng dụng

Phát triển ứng dụng di động là một công việc phức tạp và nhiều khó khăn. Có rất nhiều framework hỗ trợ phát triển một ứng dụng trên di động. Android cung cấp một framework cơ bản dựa trên ngôn ngữ lập trình Java còn iOS thì cung cấp framework dựa trên Objective-C/ Swift.

Tuy nhiên hầu hết các ứng dụng hiện nay đều hỗ trợ cả 2 nền tảng Android và iOS, do đó cùng lúc phát triển 2 dự án khác nhau với 2 framework khác nhau là một công việc phức tạp và lãng phí thời gian. Do đó người ta đã phát triển các framework lập trình đa nền tảng để giải quyết vấn đề này. Một framework rất phổ biến hiện nay là React Native được phát triển bới Facebook đang được sử dụng rất rộng rãi. Tuy nhiên React Native vẫn thông qua các api của các framework gốc như Android hay iOS do đó bị hạn chế và tốc độ kém.

Như một sự phát triển của tương lai, Flutter là một nền tảng hỗ trợ phát triển cho các ứng dụng đa nền cho hệ iOS và Android do tập đoàn Google phát triển. Nó được sử dụng vô cùng phổ biến cho nhiệm vụ tạo ra các ứng dụng gốc dành cho Google.

Flutter gồm 2 thành phần quan trọng:

Một SDK (Software Development Kit): SDK được biết đến là bộ sưu tập những công cụ giúp cho người dùng phát triển được ứng dụng nền của mình. Người dùng sử dụng công cụ này để hỗ trợ phần lớn các thao tác phát triển ứng dụng. Trong đó bao gồm các trình biên dịch mã thành các mã gốc dành riêng cho hai hệ điều hành IOS và Android.

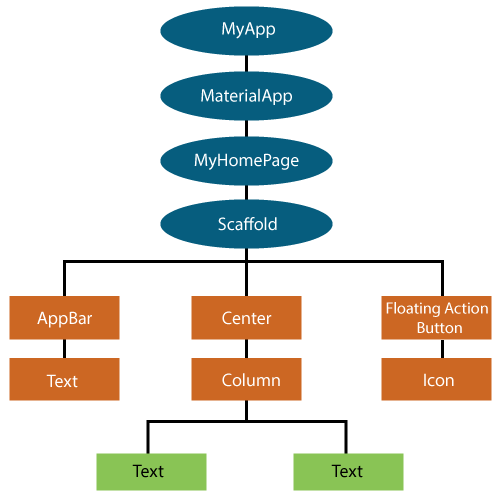
Một Framework (UI Library based on widgets): Framework cho phép người dùng tập hợp những thành phần của giao diện. Chính vì điều đó, việc tái sử dụng những mã code trên Framework trở nên dễ dàng hơn. Tiết kiệm thời gian và tính hữu dụng của Framework các bạn có thể tự tìm hiểu để biết thêm chi tiết. Tuy nhiên, điều này thật sự có ích với việc cá nhân hóa tùy theo nhu cầu của người dùng.

### **2.1.2 Kiến trúc Flutter**

Trong Flutter, tất cả đều quy về các widget, một widget phức hợp sẽ bao gồm các widget khác bên trong. Nhờ sự kết hợp linh hoạt này chúng ta có thể tạo ra bất kì ứng dụng phức tạp nào.Và các tính năng tương tác sẽ được tích hợp bất cứ khi nào nhờ GestureDetector widget.

Trạng thái của các widget được quản lý cập nhật bởi StatefulWidget widget. Đồng thời Flutter cung cấp thiết kế class để bất kỳ lớp nào cũng có thể được lập trình tùy thuộc vào độ phức tạp của tác vụ.

#### 2.1.2.1 Widget



Hình 1.8 Các widget đặc trưng

Widget là khái niệm cốt lõi nhất của Flutter, mọi thứ đều là widget. Widget là thành phần giao diện cơ bản nhất tạo nên toàn bộ giao diện người dùng của ứng dụng.

Trong Flutter, bản thân chính ứng dụng đã là một widget. Mỗi ứng dụng chính là một top-level widget và nó bao gồm một hoặc nhiều các widget con, mỗi widget này lại có thể bao gồm một hoặc nhiều widget con khác. Nhờ sự kết hợp linh hoạt này chúng ta có thể tạo ra bất kì ứng dụng phức tạp nào.

Trong Flutter tất các widget được phân loại dựa trên chức năng thành 4 nhóm sau:

* Các widget giao diện đặc thù theo từng nền tảng
* Các widget hỗ trợ bố trí giao diện
* Các widget quản lý trạng thái
* Các widget cơ bản độc lập với nền tảng

#### 2.1.2.2 Gestures

Hình 1.9 Các thao tác tương tác

Flutter widget hỗ trợ tương tác thông qua một widget đặc biệt gọi là GestureDetector. GestureDetector là một tiện ích không hiển thị trên giao diện nhưng có khả năng nắm bắt các thao tác của người dùng như nhấp, kéo, vuốt, chạm,… Phần lớn widget gốc của Flutter hỗ trợ tương tác giao diện thông qua GestureDetector.

#### State Management With Provider | raywenderlich.com2.1.2.3 State

Hình 1.10 Thao tác cập nhật lại giao diện

Flutter widgets quản lý các State thông qua một widget đặc biệt gọi là StatefulWidget. Tất cả các Widget cần StatefulWidget để quản lý các trạng thái và kết nối với các widget khác. Flutter widgets là một dạng reactive gốc. Nó tương tự như reactjs và StatefulWidget sẽ tự động thay đổi giao diện hiển thị khi thay đổi trang thái. Sự thay đổi giao diện này được tối ưu bằng cách tìm kiếm sự khác biệt giữa UI widget cũ và mới sau đó chỉ vẽ lại những thay đổi cần thiết.

#### 2.1.2.4 Layers

Hình 1.11 Các lớp của Flutter

Đây là một khái niệm quan trọng của Flutter. Nó là các thành phần sẽ được nhóm lại theo độ phức tạp và được sắp xếp rõ ràng trong các tầng có độ phức tạp giảm dần. Trên cùng của tất cả các layer là các widget đặc biệt cho Android và iOS. Layer tiếp theo là widget gốc của flutter. Tiếp nữa là Rendering layer, đây là cấp bậc thấp nhất trong việc tạo ra các thành phần của ứng dụng Flutter. Layer tiếp theo là nền tảng gốc hệ điều hành.

### Ngôn ngữ lập trình Flutter**2.1.3 Dart - Ngôn ngữ lập trình Flutter**

Hình 1.12 Dart- Ngôn ngữ lập trình trong Flutter

Dart là ngôn ngữ lập trình mã nguồn mở được phát triển bởi Google, hiện đã được chấp thuận bởi tổ chức Ecma. Dart được sử dụng để xây dựng các loại ứng dụng: web, server, di động thậm trí là lập trình nhúng với Flutter.

Dart là ngôn ngữ hỗ trợ lập trình hướng đối tượng, cú pháp kiểu C, Dart có thể biên dịch thành JavaSript để chạy trên trình duyệt.

Mục đích của Dart là tạo giao diện người dùng frontend cho web và ứng dụng dành cho thiết bị di động. Nó được biên dịch sang mã máy để xây dựng ứng dụng di động, lấy cảm hứng từ các ngôn ngữ lập trình khác như Java, JavaScript, C #. Vì Dart là một ngôn ngữ biên dịch nên bạn không thể thực thi code của mình trực tiếp, thay vào đó, trình biên dịch phân tích cú pháp của nó và chuyển nó thành mã máy.

### **2.1.4 Điểm mạnh**

Flutter đi kèm với nhiều widget đẹp và có độ tuỳ biến cao giúp phát triển ứng dụng hiệu năng cao vượt trội đáp ứng mọi nhu cầu và tuỳ biến. Bên cạnh đó Flutter còn có những điểm mạnh sau:

Dart có một kho lớn các gói phần mềm cho phép bạn mở rộng khả năng cho ứng dụng của mình

Các lập trình viên chỉ cần viết một chương trình duy nhất cho tất cả các ứng dụng (Android và iOS) . Flutter có thể mở rộng ra các nền tảng khác trong thời gian tới.

Flutter dễ dàng kiểm thử hơn do tiết kiệm thời gian kiểm thử trên từng nền tảng.

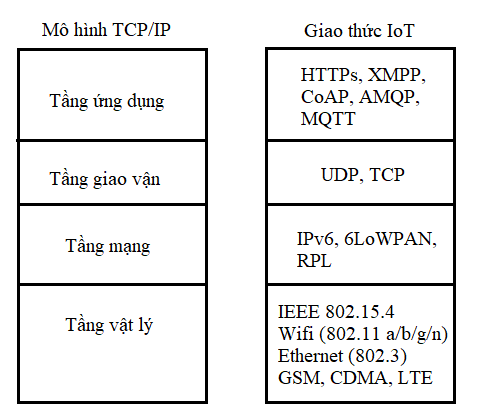
Nhờ sự đơn giản của mình, Flutter là lựa chọn hàng đầu cho các ứng dụng mới. Nó còn dễ dàng tuỳ biến và mở rộng lên càng mạnh mẽ hơn

Với Flutter, lập trình viên có toàn quyền để sắp xếp bổ trí điều khiển các widget

Flutter có bộ công cụ phát triển rất hoàn thiện và đầy đủ, đặc biệt với tính năng hot reload đẩy nhanh tốc độ build ứng dụng đáng kinh ngạc.

# III- CƠ SỞ LÝ THUYẾT

## 3.1 Kết nối phần cứng với phần mềm

Internet of Things là thuật ngữ rất phổ biến hiện nay, dùng để chỉ một tập hợp các thiết bị có khả năng kết nối internet và kết nối được với nhau để thực hiện một công việc nào đó. Để khai thác hết được tiềm năng của mô hình này, các thiết bị kết nối cần phải giao tiếp bằng các giao thức nhẹ mà không làm tiêu tốn quá nhiều tài nguyên của CPU.

Hình 1. 13 Giao thức IoT được ánh xạ tới mô hình TCP/IP

Chúng ta có các giao thức truyền tải dữ liệu phổ biến có thể được sử dụng trong các mô hình Internet of Things đó là MQTT, CoAP, HTTP,… Đa phần các giao thức đó đều là giao thức trên lớp ứng dụng.

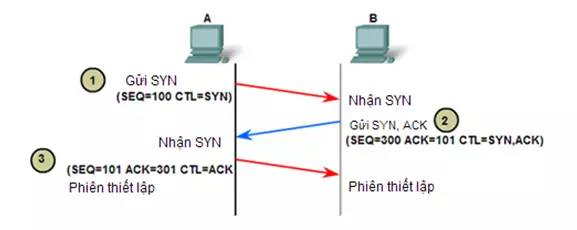
Tầng ứng dụng là tầng gần với người sử dụng nhất. Nó cung cấp phương tiện cho người dùng truy nhập các thông tin và dữ liệu trên mạng thông qua chương trình ứng dụng. Tầng này là giao diện chính để người dùng tương tác với chương trình ứng dụng, và qua đó với mạng. Một số ví dụ về các ứng dụng trong tầng này bao gồm Telnet, Giao thức truyền tập tin FTP và Giao thức truyền thư điện tử SMTP, HTTP, X.400 Mail remote.

Khi sử dụng các giao thức thuộc tầng này, dữ liệu người dùng đưa vào sẽ được đi từ tầng trên xuống tầng dưới bắt đầu từ tầng ứng dụng, qua mỗi tầng dữ liệu được thêm vào thông tin điều khiển gọi là Header. Khi nhận dữ liệu thì quá trình xảy ra ngược lại, dữ liệu được truyền từ tầng dưới lên và qua mỗi tầng thì phần header tương ứng sẽ được lấy đi và khi đến tầng trên cùng thì dữ liệu không còn phần header nữa.

Tuỳ vào nhu cầu sử dụng có thể bỏ qua việc sử dụng bộ giao thức IoT để gửi gói tin trực tiếp bằng giao thức TCP hoặc UDP của tầng giao vận. Do nhu cầu của khoá luận là một mạch điều khiển quạt thông minh trên đồng thời có kết nối internet và không kết nối internet, không yêu cầu mức độ bảo mật cao nhưng phải đảm bảo tốc độ nên một trong hai giao thức TCP và UDP là phù hợp nhất.

Giao thức TCP và UDP là hai giao thức cốt lõi trong tầng giao vận có khả năng gửi tin nhắn tới máy chủ trong mạng giao thức internet.

TCP là giao thức truyền tải hướng kết nối, nghĩa là phải thực hiện thiết lập kết nối với đầu xa trước khi thực hiện truyền dữ liệu. Tiến trình thiết lập kết nối ở TCP được gọi là tiến trình bắt tay 3 bước.



Hình 1.14 Tiến trình bắt tay 3 bước

Ngược lại với giao thức TCP là giao thức UDP. UDP là giao thức truyền tải hướng không kết nối. Nó sẽ không thực hiện thao tác xây dựng kết nối trước khi truyền dữ liệu mà thực hiện truyền ngay lập tức khi có dữ liệu cần truyền vì vậy truyền tải rất nhanh cho dữ liệu của lớp ứng dụng.

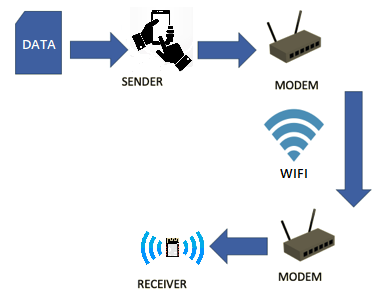
Từ cái khái niệm trên ta rút ra được điểm khác nhau của hai giao thức này:

|  |  |
| --- | --- |
| TCP | UDP |
| Hướng kết nối | Không hướng kết nối |
| Độ tin cậy cao | Độ tin cậy thấp |
| Không cho phép mất gói tin | Cho phép mất gói tin |
| Đảm bảo việc truyền dữ liệu | Không đảm bảo việc truyền dữ liệu |
| Có sắp xếp thứ tự các gói tin | Không sắp xếp thứ tự các gói tin |
| Tốc độ truyền thấp | Tốc độ truyền cao |

Từ đó cho thấy việc sử dụng giao thức UDP là hoàn toàn phù hợp với yêu cầu khoá luận.

ESP8266 đi kèm với thư viện kết nối WiFi hỗ trợ TCP, UDP và các ứng dụng HTTP, mDNS, SSDP, DNS Servers. Đồng thời hỗ trợ cả việc tạo điểm truy cập hoặc máy khách kết nối Wifi. Khá phù hợp cho việc tạo một server UDP để xử lý các gói tin mà các máy khách truyền vào.

Trong mô hình này ESP8266 sẽ có nhiệm vụ phát hoặc kết nối vào mạng Wifi chỉ định sẵn như một server nhận dữ liệu và các thiết bị di động sẽ đóng vai trò như máy khách gửi các đoạn dữ liệu vào server. Khi server UDP của ESP8266 nhận được dữ liệu truyền về từ ứng dụng di động trên máy khách sẽ tiến hành thực hiện các thao tác chỉ định.



Hình 1.15 Mô hình truyền dữ liệu

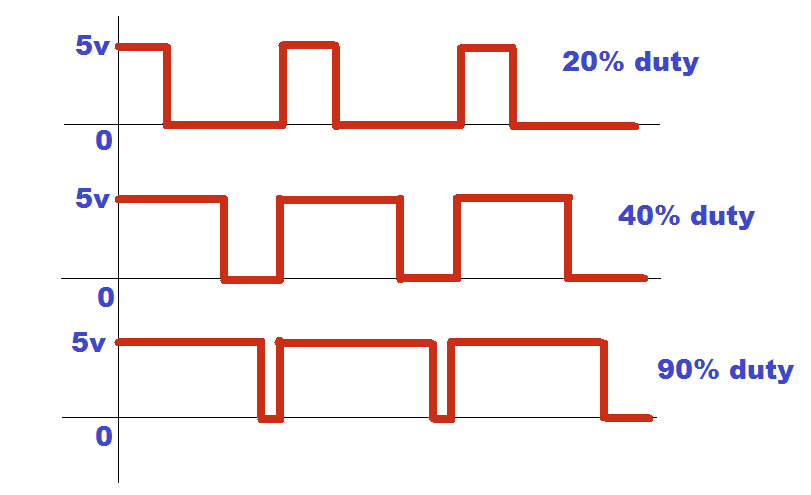
## 3.2 Giao tiếp giữa vi điểu khiển với động cơ

Động cơ BLDC được sử dụng trong khoá luận là loại động cơ điều khiển bằng xung PWM. Vì vậy việc điều chế xung PWM bằng vi điều khiển là một phần quan trọng không thể thiếu.

PWM là một loại tín hiệu có thể được tạo ra từ một vi mạch kỹ thuật số như vi điều khiển hoặc bộ định thời 555. Do đó, tín hiệu được tạo ra sẽ có một nhóm các xung và các xung này sẽ ở dạng một sóng vuông. Có nghĩa là, tại bất kỳ thời điểm cụ thể nào, sóng sẽ cao hoặc sẽ thấp. Đại lượng đặc trưng cho xung là tần số và chu kỳ xung.

Phương pháp điều xung PWM phương pháp điều chỉnh điện áp ra tải, hay nói cách khác, là phương pháp điều chế dựa trên sự thay đổi độ rộng của chuỗi xung vuông, dẫn đến sự thay đổi điện áp ra.

Các PWM khi biến đổi thì có cùng 1 tần số và khác nhau về độ rộng của sườn dương hay sườn âm.



Hình 1.16 Đồ thị dạng xung điều chế PWM

Với các xung có độ rộng khác nhau tốc độ quay của động cơ BLDC cũng khác nhau

# CHƯƠNG 2: THỰC HÀNH

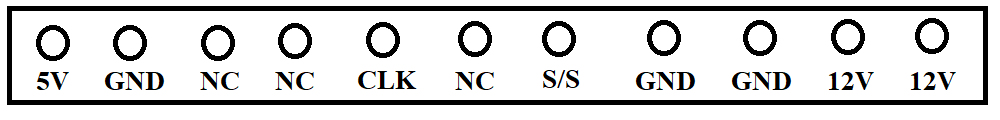
I- PHẦN CỨNG

A close-up of a typewriter

Description automatically generated with low confidenceĐa phần các động cơ BLDC trên thị trường khá đa dạng về các loại sản phẩm nhưng nhìn chung chúng đều giống nhau về cách thức hoạt động và thông số các chân đầu vào.

Hình 2.1 BH55FT15-07

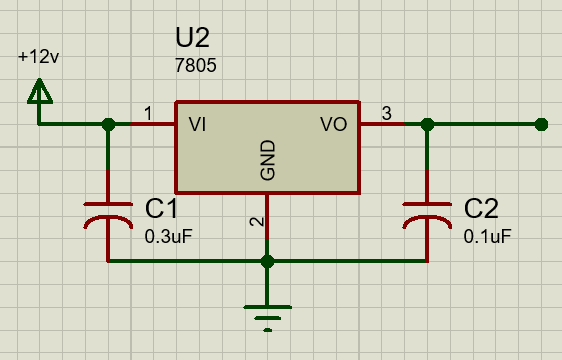
Model động cơ không chổi than được dùng có mã sản phẩm BH55FT15-07 với 11 đầu vào.



Hình 2.2 Hình 2. 2 Sơ đồ chân chiếu theo thứ tự như ảnh 2.1

Thông số chi tiết các chân:

* Chân 12V nối dương nguồn 12 - 24V cung cấp nguồn cho động cơ.
* Chân GND nối âm nguồn (GND).
* Chân S/S nối âm để khởi động động cơ, ngắt khỏi âm nguồn motor sẽ dừng.
* Chân CLK là chân cấp xung 5V, tần số 500 - 3kHz
* Chân 5V là chân cấp nguồn 5V cung cấp nguồn cho IC của động cơ.

Dựa theo các thông số trên, ta sẽ dùng nguồn 12-24v để cung cấp nguồn cho toàn mạch hoạt động. Do ESP và động cơ yêu cầu nguồn 5v nên ta phải thiết kế một mạch ổn áp để chuyển điện áp 12-24v cấp vào thành 5v đầu ra.

Hình 2.3 Sơ đồ mạch ổn áp 5V

Theo sơ đồ trên, các tụ C1 và C2 để lọc điện áp đầu vào cấp cho chân Vi và điện áp cấp cho tải tiêu thụ lấy từ chân Vo của IC 7805, cả 2 tụ này có tác dụng cung cấp điện áp tạm thời cho lần lượt các chân Vi và Vo khi nguồn đột ngột bị sụt áp.

Lưu ý thêm IC ổn áp 7805 này hoạt động có tỏa nhiệt ra bên ngoài nên cần lắp thêm tản nhiệt.

Các chân nguồn và đất cung cấp cho các linh kiện và động cơ đã hoàn tất. Nhiệm vụ xử lý các thuật toán logic và cung cấp xung cho động cơ sẽ do WeMos D1 Mini ESP8266 đảm nhiệm.

Hiện nay trên thi trường có rất nhiều loại vi điều khiển với chức năng và mục đích sử dụng khác nhau nhưng xét về tính nhỏ gọn và các chức năng liên quan tới IoT thì ESP8266 là lựa chọn tốt nhất.

A picture containing electronics, circuit

Description automatically generatedESP8266 cung cấp cho ta một số thư viện mặc định để có thể tùy chỉnh xung và độ rộng xung một cách dễ dàng hơn. Mặc định của tần số PWM là 1kHz nhưng có thể thay đổi tùy ý trong khoảng 100Hz tới 40000Hz.

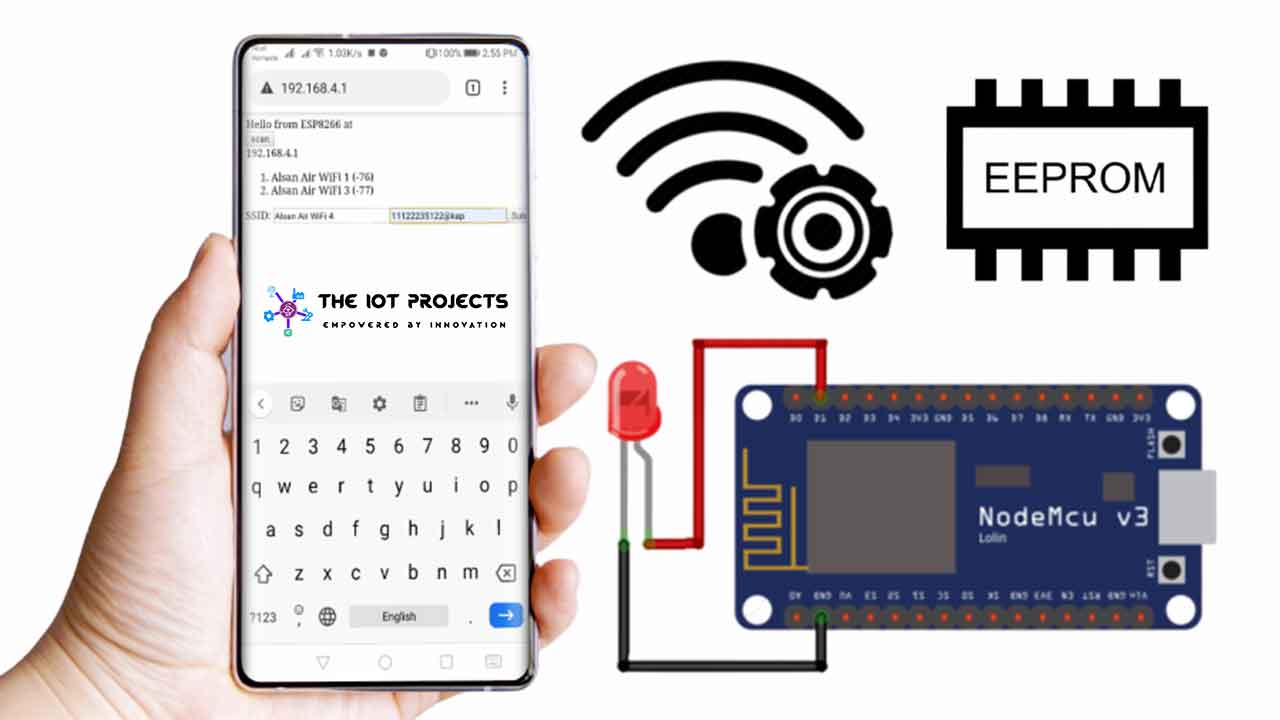
Hình 2.4 WeMos D1 Mini ESP8266

ESP8266 không có phần cứng liên quan tới PWM nên việc điều chỉnh thường bằng phần mềm. Với đầu ra xung PWM 40000Hz thì CPU sẽ tốn khá nhiều tài nguyên nên việc lựa chọn tần số thích hợp là vô cùng quan trọng. Vì động cơ BLDC ta dùng hoạt động trong dãy tần số từ 500 đến 3000Hz nên chúng ta có thể cài mặc định cho xung PWM đầu ra này.

Với 3 chức năng cơ bản: chế độ gió cố định, chế độ gió tự nhiên và chế độ hẹn giờ. Ở chế độ gió cố định ta có thể điều chỉnh tốc độ gió. Chế độ gió tự nhiên sẽ tự động tăng giảm tốc độ gió theo quy luật có sẵn và chế độ hẹn giờ sẽ cài giờ tắt cho thiết bị. Dựa vào 3 đặc điểm này ta sẽ xây dựng nên mạch điều khiển quạt thông minh.

Với khả năng thu và phát sóng wifi của mình, ta đồng thời có thể dùng ESP để vừa làm máy chủ phát sóng trong trường hợp không có kết nối và kết nối vào những mạng wifi có sẵn phục vụ cho quá trình truyền và nhận dữ liệu. Điểm đặc biệt của sóng Wifi là ta có thể truyền và nhận dữ liệu giữa các mạng cục bộ mà không yêu cầu kết nối internet và phạm vi tín hiệu cũng khá lớn.

Đồng thời nó cũng hỗ trợ nhiều thư viện để giao tiếp, ta sẽ sử dụng 2 giao thức phổ biến là HTTP và UDP để biến ESP thành một máy chủ xử lý dữ liệu. Với giao thức HTTP, mục tiêu là để các máy khách có thể dễ dàng phân biệt được thiết bị kết có có phải là chiếc quạt thông minh hay không đồng thời cũng là nơi trao đổi thông tin với vi điều khiển bằng những chuỗi JSON máy chủ webserver trả về. Với giao thức UDP là một giao thức truyền dữ liệu với tốc độ khá nhanh, vì vậy ESP sẽ đóng vai trò là một máy chủ UDP nhận dữ liệu từ máy khách. Các dữ liệu truyền từ máy khách sẽ được lưu dưới dạng JSON và truyền về máy chủ UDP. Mỗi ESP sẽ được định danh bằng một địa chỉ IP riêng và trên cùng một địa chỉ này ta sẽ triển khai 2 giao thức với 2 port khác nhau. Mặc định của webserver sẽ có port là 80 và trên máy chủ UDP ta sẽ chọn port 17499.

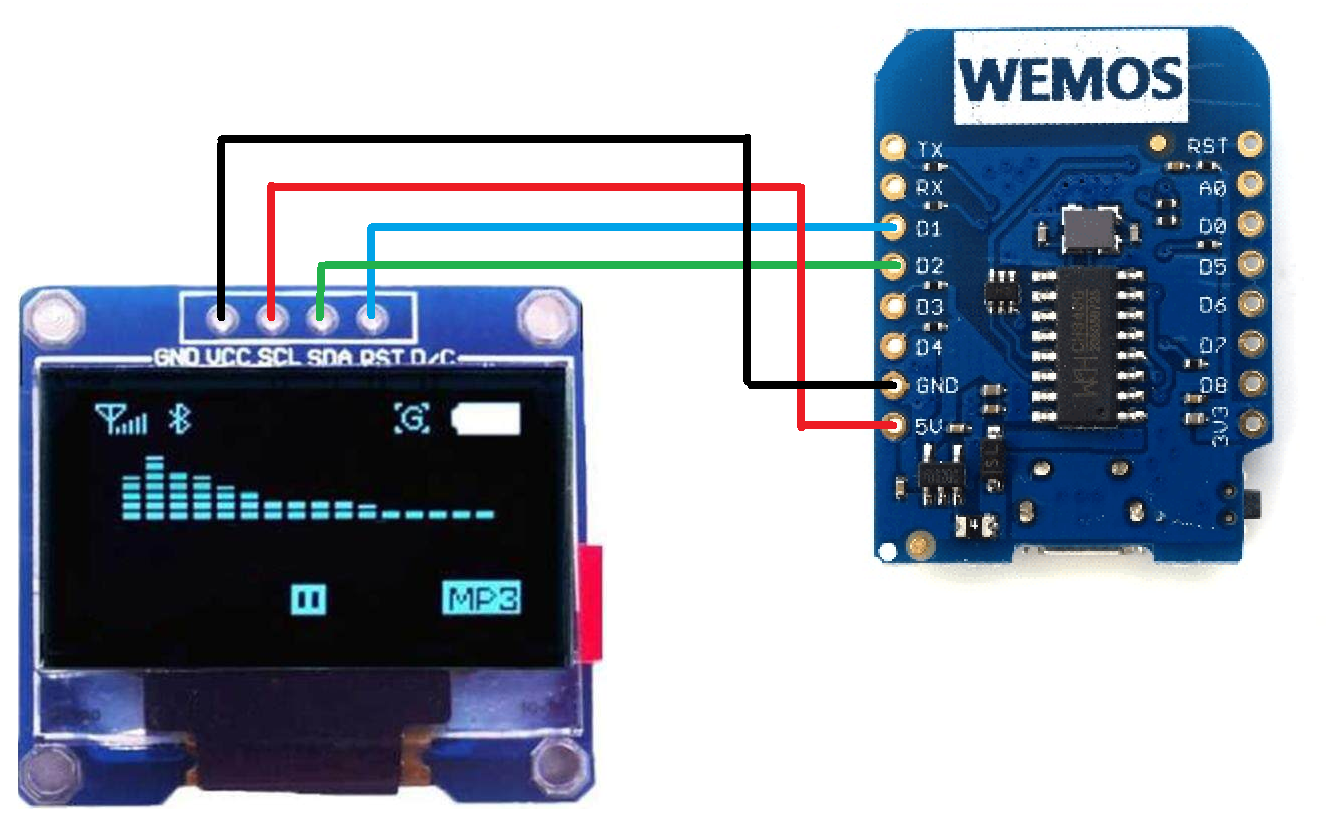


Hình 2.5 Lưu cấu hình wifi bằng EEPROM

Một vấn đề là mỗi khi ngắt nguồn, ESP sẽ reset lại toàn bộ dữ liệu nên việc lưu dữ liệu là một vấn đề quan trọng. ESP có hỗ trợ một khu vực lưu giữ liệu. Đó là EEPROM. EEPROM là bộ nhớ trong của vi điều khiển ESP cho phép lưu dữ liệu trong bộ nhớ sau khi khởi động lại bo mạch. Điều đặc biệt dữ liệu này sẽ không bị mất đi khi vô tình sập nguồn vi điều khiển. Điều lưu ý của bộ nhớ này là kích thước lưu trữ không được lớn nên ta sẽ sử dụng nó để lưu thông tin của kết nối wifi.

Việc hiển thị thông tin sẽ do màn hình OLED đảm nhiệm. Do nó giao tiếp bằng I2C nên việc kết nối với vi điểu khiển khá đơn giản. Để giao tiếp giữa ESP với OLED ta đấu mạch theo bảng sau:

|  |  |
| --- | --- |
| ESP8266 | OLED SSD1306 |
| 5V | VCC |
| GND | GND |
| SDA (D2) | SDA |
| SCL (D1) | SCL |



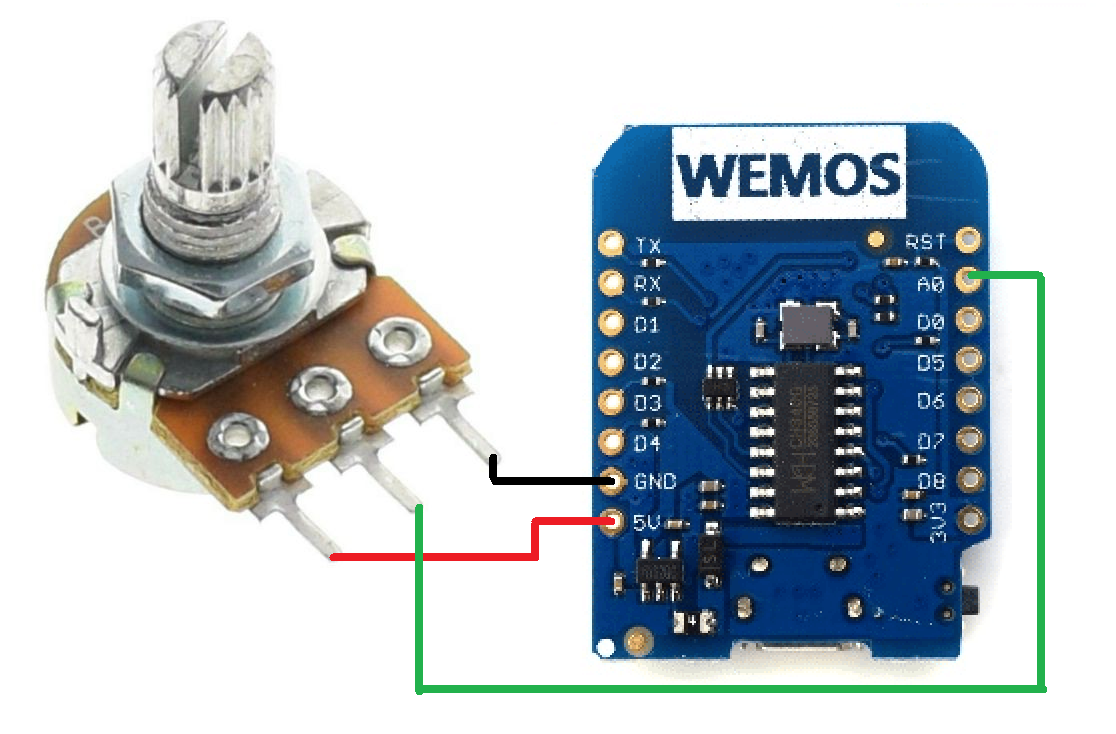
Hình 2.6 Sơ đồ mạch kết nối OLED với ESP

Sau khi kết nối ta tiến hành cài đặt, với giao tiếp i2c việc quan trọng nhất là tìm kiếm địa chỉ của i2c để có thể khởi chạy OLED một cách bình thường. Hiện nay trên mạng có khá nhiều thư viện hỗ trợ kiểm tra địa chỉ này. Ta sẽ dùng 2 thư viện Adafruit\_GFX.h và Adafruit\_SSD1306.h để điều khiển màn hình OLED.

Việc nhận tính hiệu để điều khiển vận tốc quay sẽ do biến trở, mạch hồng ngoại và máy chủ UDP trong ESP đảm nhiệm.

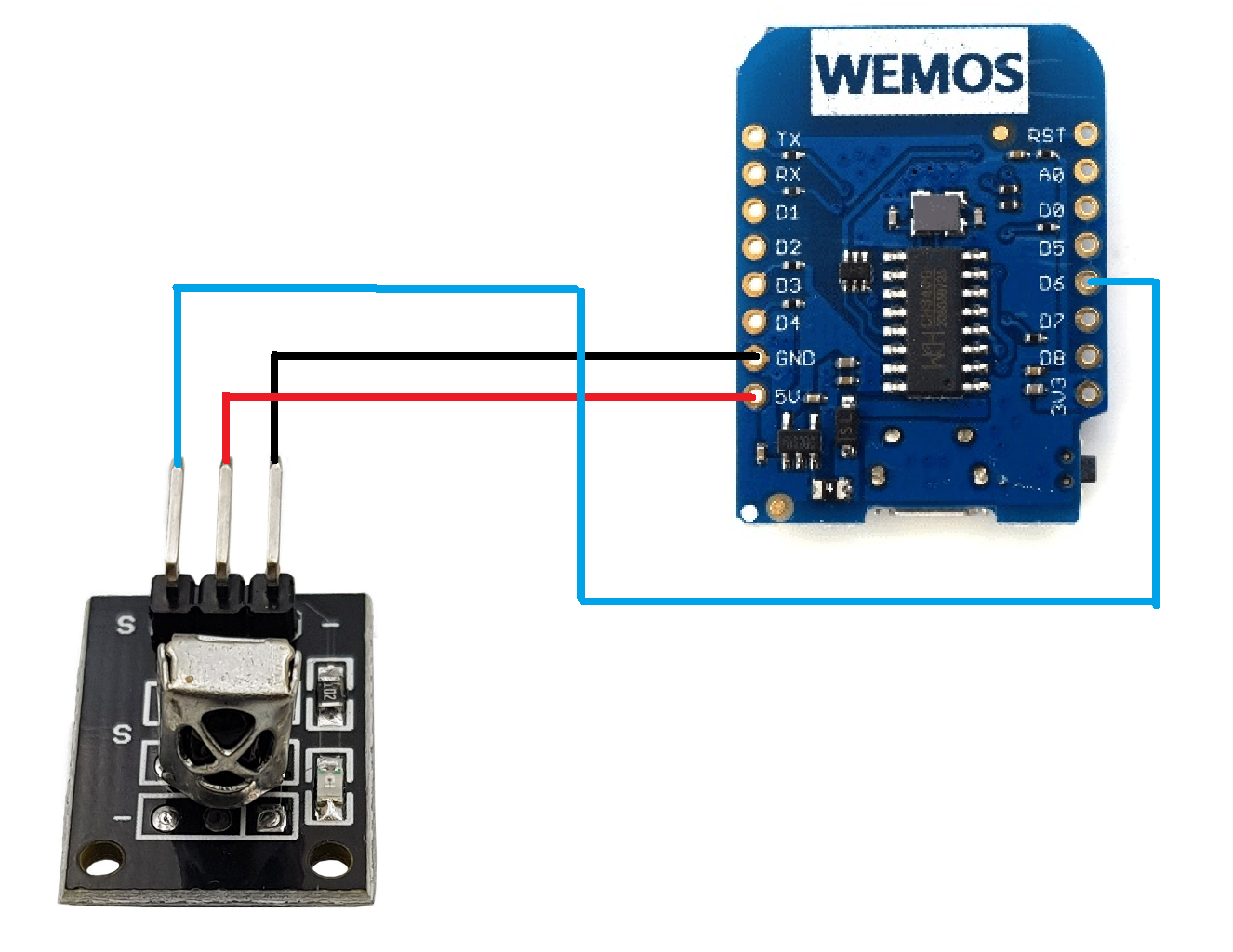
Các loại Arduino thường có 6 đến 12 chân analog tuy nhiên điểm hạn chế nhất của dòng chip ESP8266 này là chỉ có 1 chân ADC. Trên mạch, chân analog này được đặt tên là A0. và có thể đo được điện thế từ 0 đến 3.3v. Chúng ta có thể dùng để đọc điện áp bên ngoài, hay đọc điện áp VCC từ đó ta có thể tự do xử lý tính hiệu và chuyển về giá trị tương ứng với các độ rộng xung khác nhau.

|  |  |
| --- | --- |
| ESP8266 | Biến trở 10k |
| VCC | 5V |
| A0 | OUTPUT |
| GND | GND |



Hình 2.7 Sơ đồ mạch kết nối biến trở với ESP

Độ phân giải của chức năng chuyển đổi tín hiệu analog sang digital này của hầu hết các board ESP là 10 bits, tương đương với 210 = 1024, tức là dải giá trị từ 0 (0V) đến 1023 (3.3V).

Tiếp theo là điều khiển thông qua hồng ngoại, ta sẽ xử dụng module điều khiển hồng ngoại IR1838. Module này gồm 2 linh kiện là mạch thu tín hiệu và một remote phát tín hiệu hồng ngoại.

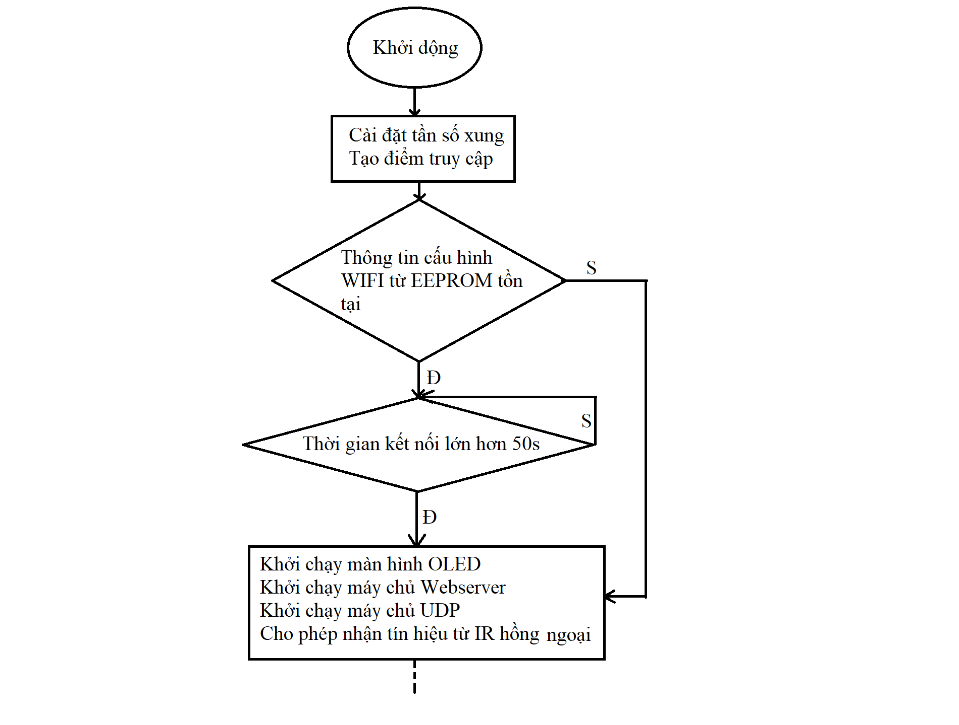
Hình 2.8 Sơ đồ mạch kết nối mạch thu hồng ngoại với ESP

|  |  |
| --- | --- |
| ESP8266 | Mạch thu hồng ngoại IR1838 |
| 5V | + (VCC) |
| GND | - (GND) |
| D6 | S (DAT) |

Mạch thu hồng ngoại này sẽ lấy các tính hiệu thu nhận được và trả về vi điều khiển dưới dạng mã HEX. Mỗi tín hiệu có một loại mã HEX đặc trưng.

Ta sẽ sử dụng remote này để điều khiển, vì vậy ta cần lấy từng giá trị HEX của từng nút để định danh dựa trên thư viện IRremoteESP8266.h ta sử dụng để điều khiển module này.

Hình 2.9 Remote IR1838

****Sau khi kết nối giữa ESP và các linh kiện, ta cần cấu hình hàm khởi động để ESP hoạt động đúng như mong muốn.

Hình 2.10 Sơ đồ thuật toán khởi động

Sau khi hoàn tất các bước kết nối và thuật toán khởi động của vi điều khiển, ta tiến hành cài đặt các thuật toán liên quan.

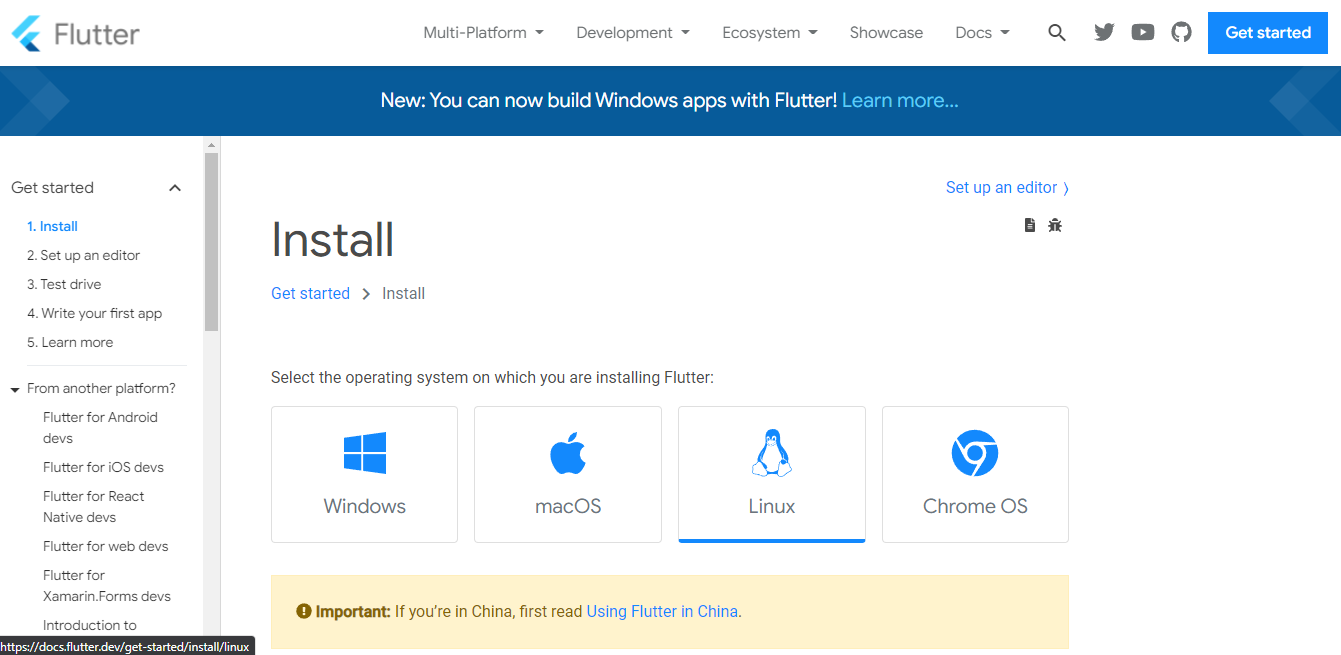
Dữ liệu vi điều khiển nhận được sẽ liên tục được đọc từ ba nguồn đó là chân Analog từ biến trở, chân dữ liệu của IR hồng ngoại và máy chủ webserver cũng như UDP server. Vì nhận dữ liệu cùng lúc khá nhiều nên việc sử dụng các hàm nghỉ là bắt buộc, trách trường hợp xung đột làm nhiễu các dữ liệu ta thu được.

Việc xử lý dữ liệu và quyết định trạng thái của động cơ sẽ do một hàm quyết định. Các dữ liệu nhận được sẽ được quy về một khuôn mẫu chung để thuật toán trở nên mượt mà hơn.

Màn hình OLED hiển thị dữ liệu cũng sẽ được làm mới mỗi khi dữ liệu thay đổi làm tối ưu khả năng tương tác với người dùng.

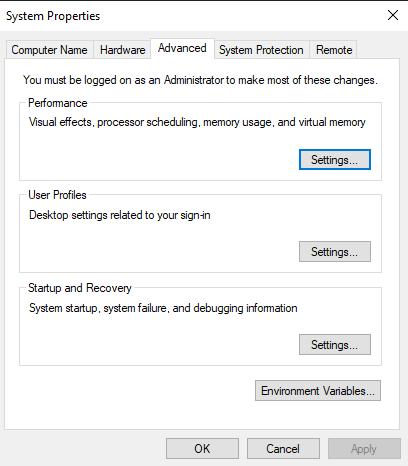
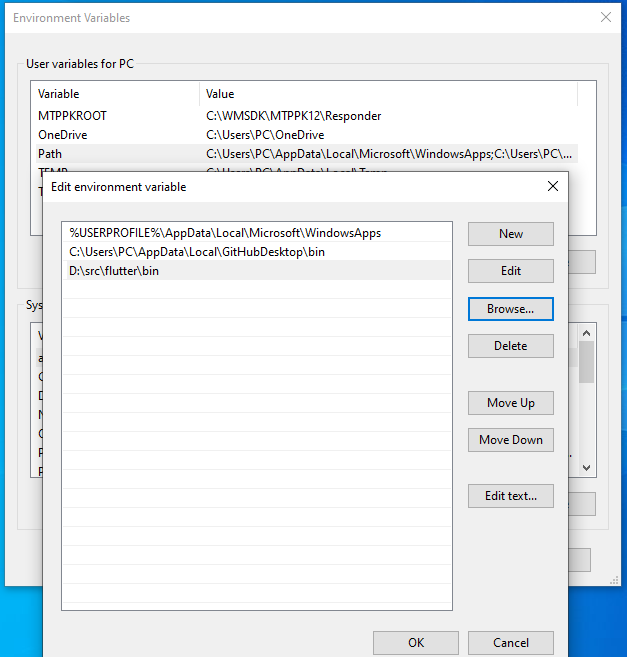
II- PHẦN MỀM

Mục tiêu của phần mềm này là gửi dữ liệu lên ESP.

Bước đầu tiên trong việc tạo một phần mềm là khởi tạo môi trường làm việc. Ở đây ta sẽ dùng Flutter làm công cụ hỗ trợ việc tạo phần mềm. Flutter hiện tại hỗ trợ trên 4 hệ điều hành khác nhau là Windows, macOS, Linux, Chrome OS, tiến hành tải xuống phiên bản phù hợp.

Hình 2.11 Giao diện trang hướng dẫn và cài đặt

Sau khi giải nén tệp đã tải về vào một vị trí phù hợp, ta bắt đầu khai báo biến môi trường để sử dụng các tập lệnh của Flutte một cách dễ dàng hơn.

Tại giao diện System Properties ta chọn Environment Variables. Sau đó tại chọn vào mục Path trong User variables for PC, ta thêm thư mục bin trong đường dẫn chứa tệp vừa giải nén và tiến hành lưu lại.

Hình 2.12 Giao diện System Properties

Hình 2.13 Thêm PATH vào biến người dùng

Sau đó mở cửa sổ cmd dùng lệnh flutter doctor để kiểm tra bạn đã cài đặt hoàn tất các thành phần chưa.

Text

Description automatically generatedCài đặt Android Studio và cài plugin Flutter và tiến hành khởi tạo chương trình để điều khiển thiết bị của bạn.

Hình 2.14 Sơ đồ thư mục của một dự án Flutter

Thành phần của một dự án Flutter chia làm nhiều thư mục đảm nhận các nhiệm vụ khác nhau. Thư mục đảm bảo giao diện và thuật toán cho chương trình sẽ chứa trong thư mục lib và tùy vào các hệ điều hành mà có các thư mục ios, android và web, riêng các thư việc sẽ do pubspec.yaml đảm nhiệm khai báo.

Flutter SDK hỗ trợ rất nhiều thư viện, trong đó có các thư viện về IoT.

Graphical user interface, text, application

Description automatically generated

Hình 2.15 Thư viện ping\_discover\_network\_forked

Graphical user interface, text, application

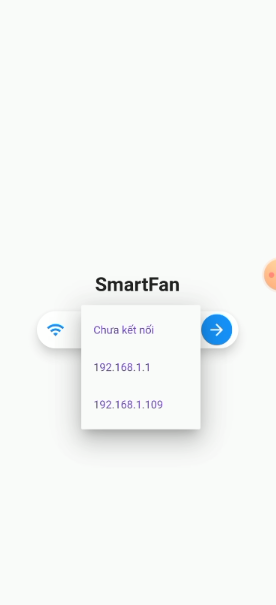
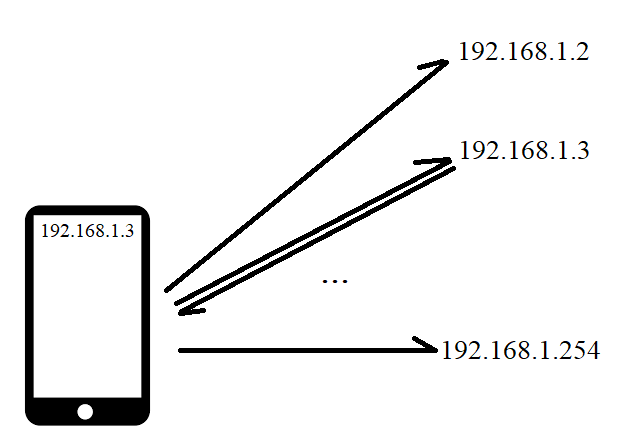
Description automatically generated

Hình 2.16 Thư viện wifi\_iot

Ta sẽ tập chung sử dụng hai thư viện ping\_discover\_network\_porked và wifi\_iot để thao tác hầu hết các chức năng với mạng wifi. Hai thư viện này đồng thời hỗ trợ trên android và ios nên sẽ không gặp bất kỳ lỗi nào khi chạy trên hai nền tảng khác nhau.

Ứng dụng điều khiển sẽ chia làm 3 giao diện đảm nhiệm 3 chức năng khác nhau:

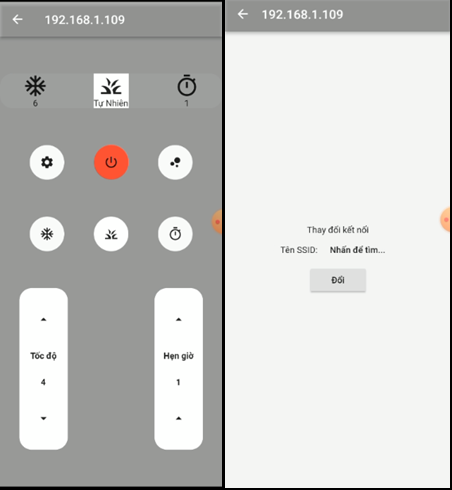
* Giao diện kết nối thiết bị: Đảm nhiệm việc quét và kết nối các thiết bị trong mạng cục bộ.
* Giao diện điều khiển: Truyển và nhận dữ liệu từ vi điều khiển.
* Giao diện cấu hình kết nối: Cấu hình kết nối wifi cho vi điều khiển

Ở giao diện kết nối thiết bị chỉ những địa chỉ ip hợp lệ mới có thể kết nối. Nguyên lý làm việc của quá trình dò tìm thiết bị của giao diện này là ứng dụng sẽ ping tới tất cả các ip trong mạng cục bộ, nếu bất kỳ ip nào phản hồi và trả về một giá trị hợp lệ sẽ xác nhận thiết bị.

Hình 2.17 Nguyên lý dò thiết bị

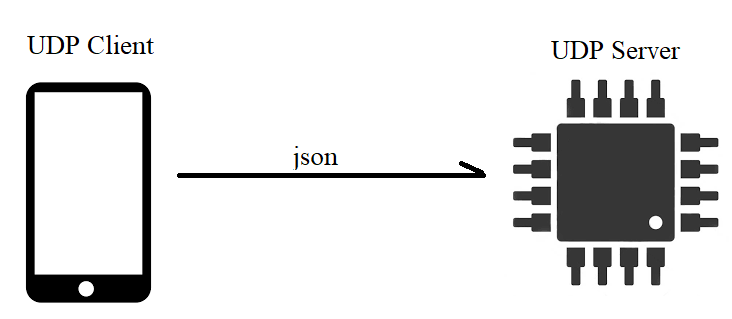
Hình 2.18 Giao diện kết nối thiết bị

Có thể hiểu việc ping này là quá trình gửi một yêu cầu http tới các ip trong mạng cục bộ. Và tại thiết bị kết nối, ESP đóng vai trò như một webserver.

Khi kết nối thành công, ở giao diện điều khiển ứng dụng sẽ tự động cập nhật trạng thái thiết bị nhận được từ webserver. Cơ bản ứng dụng điều khiển có chức năng chính như cấu hình kết nối wifi cho máy chủ, thay đổi qua lại giữa ba chế độ và hiển thị trạng thái quạt.

Hình 2.19 Giao diện điều khiển và cấu kết nối

Khi các nút được nhấn dữ liệu gồm chuỗi json thông tin sẽ được gửi lên máy chủ bằng giao thức UDP.



Hình 2.20 Quá trình gửi dữ liệu

Chuỗi json này chỉ chứa ba thông tin là tốc độ gió, thời gian và chế độ và sẽ có thêm hai trường tên và mật khẩu kết nối nếu thay đổi cấu hình kết nối.

# Chương 3: TỔNG KẾT

I- KẾT LUẬN

Thông qua đề tài này, em đã có thể hiểu rõ hơn về quá trình thiết kế, lập

trình vi điều khiển cho thiết bị đồng thời có các kiến thức cơ bản về IoT.

Ngoài ra, trong quá trình thực hiện đề tài được giao, nhóm em cũng đã gặp

nhiều vấn đề thực tế có khác biệt so với lí thuyết. Do đó, để bổ sung những phần còn

thiếu trong quá trình học tập, em đã phải tìm hiểu thêm về những kiến

thức, phương pháp mới để vượt qua những rào cản cũng như là vấn đề khó khăn trong quá trình thực hiện đề tài.

Bên cạnh đó, ngoài những kiến thức về thiết kế phần cứng, linh kiện điện tử,

khả năng đi dây, hàn mạch thì kĩ năng mềm về phân chia công việc hợp lý để đảm bảo tiến trình, cũng như khả năng tư duy, logic, giải thuật của em cũng được cải thiện đáng kể.

II- ƯU ĐIỂM

Sau khoảng thời gian tìm hiểu, học hỏi về cách linh kiện, các kiến thức về giải thuật, các thuật toán, kiến thức học thuật mới em đã hoàn thành đề tài “Mạch điều khiển quạt thông minh” với những ưu điểm sau:

* Điều chỉnh được tốc độ của động cơ BLDC
* Tương tác với vi điều khiển bằng các thiết bị ngoại vi có sẵn
* Tương tác bằng ứng dụng trên điện thoại
* Xây dựng một ứng dụng riêng để điều khiển thiết bị
* Cấu hình wifi cho vi điều khiển không cần phần mềm thứ ba

III- KHUYẾT ĐIỂM

Bên cạnh các ưu điểm, còn một số nhược điểm chưa khắc phục được.

Giao diện ứng dụng chưa được tối ưu cho người dùng. Cách bố trí và cách thức điều khiển khá phức tạp.

Màn hình hiển thị khá nhỏ, chưa đáp ứng được khả năng hiển thị với người dùng ở khoảng cách xa.

IV- HƯỚNG PHÁT TRIỂN

Trong tương lai quạt thông minh này có thể phát triển theo các hướng sau:

* Tích hợp thêm các nguồn điện dự trữ trong trường mất điện.
* Thêm các tính năng mới liên quan tới điều khiển.
* Bảo mật các kết nối tới thiết bị để chủ nhân của thiết bị mới có thể điều khiển

TÀI LIỆU THAM KHẢO

<https://randomnerdtutorials.com/esp8266-pinout-reference-gpios/>

<https://arduino-esp8266.readthedocs.io/en/latest/index.html>

<https://viblo.asia/p/tim-hieu-giao-thuc-tcp-va-udp-jvEla11xlkw>

<https://vncoder.vn/bai-hoc/gioi-thieu-flutter-204>

<https://www.guru99.com/tcp-vs-udp-understanding-the-difference.html>

1. Chức danh của CBHD: ThS. (Thạc sĩ); TS. (Tiến sĩ); PGS. TS. (Phó giáo sư Tiến sĩ); Thầy (Cữ nhân) [↑](#footnote-ref-1)
2. Chức danh của CBHD: ThS. (Thạc sĩ); TS. (Tiến sĩ); PGS. TS. (Phó giáo sư Tiến sĩ); Thầy (Cữ nhân) [↑](#footnote-ref-2)