# **LỜI NÓI ĐẦU**

**Internet Of Things (IoT)** – **Internet vạn vật** dường như đang đứng trước một bước ngoặt để đi đến giai đoạn tiếp theo cho một thế giới hiện đại, văn minh. Đó là viễn cảnh mà mọi vật đều có thể kết nối với nhau thông qua Internet không dây. Các doanh nghiệp đang có xu hướng ứng dụng sản phẩm công nghệ IoT vào sản xuất ngày càng nhiều bởi thị trường sáng tạo tiềm năng và chi phí sản xuất ngày càng thấp.

Áp dụng Internet vạn vật vào các ngành, lĩnh vực trong cuộc sống trở nên rất cần thiết, đặc biệt là trong lĩnh vực Smart Home. Nó giúp cho con người tự do điều khiển thiết bị từ xa một cách chính xác và nhanh chóng.

LG là một công ty phát triển nhiều mảng trong linh vực điện điện tử ở Việt Nam trong đó có sản xuất điều hòa không khí (Air-Condition)

Như em tìm hiểu hiện nay có hai cách điều khiển điều hòa đó là sử dụng điều khiển cảu hãng hay với các model mới có tích hợp điều khiển qua app của hãng. Nhằm khắc phục nhược điểm khi mất điều khiển đối với model không tích hợp cùng với đó là giúp người dung điều khiển được điều hòa khi không ở trong nhà mình vì vậy em chọn đề tài: “Phát triển bộ điều khiển hồng ngoại điều khiển điều hòa LG qua môi trường Internet” nhằm giúp người dùng có một sự trải nghiệm tốt nhất không gây tốn kém chi phí.

Để hoàn thành đề tài này, ngoài sự nỗ lực của bản thân em xin chân thành gửi lời cảm ơn sâu sắc tới các thầy cô Khoa Công Nghệ Tự Động Hóa trường Đại học Công nghệ thông tin và Truyền thông Thái Nguyên, đặc biệt là thầy **TS Phạm Đức Long** đã tận tình hướng dẫn, gips đỡ và tạo điều kiện tốt nhất cho em kể từ khi nhận đề tài tới khi hoàn thành đề tài này.

Em xin chân thành cảm ơn!

# **MỤC LỤC**

[LỜI NÓI ĐẦU i](#_Toc54818171)

[MỤC LỤC ii](#_Toc54818172)

[DANH MỤC HÌNH ẢNH iii](#_Toc54818173)

[**CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI** 1](#_Toc54818174)

[**1.1** **Giới thiệu chung** 1](#_Toc54818175)

[**1.2** **Tổng quan các công nghệ sử dụng** 1](#_Toc54818178)

[*1.2.1* *Công nghệ WiFi* 1](#_Toc54818179)

[*1.2.2* *Tia hồng ngoại* 2](#_Toc54818180)

[*1.2.3* *Giao thức MQTT* 3](#_Toc54818181)

[**CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG** 7](#_Toc54818182)

[**2.1** **Đặc tả hệ thống** 7](#_Toc54818185)

[**2.2** **Phân tích và thiết kế** 7](#_Toc54818186)

[*2.2.1* *Sơ đồ khối của hệ thống và chức năng từng khối* 7](#_Toc54818187)

[*2.2.2* *Sơ đồ nguyên lý* 8](#_Toc54818188)

[*2.2.3* *Sơ đồ mạch in PCB* 9](#_Toc54818189)

[*2.2.4* *Lưu đồ thuật toán* 9](#_Toc54818190)

[**2.3** **Giới thiệu một số công cụ, thiết bị sử dụng** 10](#_Toc54818191)

[*2.3.1* *Esp8266* 10](#_Toc54818192)

[*2.3.2* *Visual Studio Code và Platform IO* 12](#_Toc54818193)

[*2.3.3* *Altium Design* 14](#_Toc54818194)

[*2.3.4* *Esp8266 MQTT client* 16](#_Toc54818195)

[*2.3.5* *MQTT dashboard* 17](#_Toc54818196)

[*2.3.6* *Led phát tia hồng ngoại* 17](#_Toc54818197)

[**CHƯƠNG 3: KẾT QUẢ ĐÁNH GIÁ** 20](#_Toc54818198)

[**3.1** **Các bước sử dụng hệ thống** 20](#_Toc54818200)

[**3.2** **Đánh giá hệ thống** 22](#_Toc54818201)

[*3.2.1* *Ưu điểm* 22](#_Toc54818202)

[*3.2.2* *Nhược điểm* 22](#_Toc54818203)

[**3.3** **Kết luận** 22](#_Toc54818204)

[**Tài liệu tham khảo** 23](#_Toc54818205)

[**Phụ lục** 24](#_Toc54818206)

# **DANH MỤC HÌNH ẢNH**

[Hình 1. 1 Hoạt động của WiFi 2](#_Toc54818547)

[Hình 1. 2 Cơ chế tổng quan 4](#_Toc54818548)

[Hình 1. 3 Kiến trúc thành phần 5](#_Toc54818549)

[Hình 2. 1 Sơ đồ khối hệ thống………………………………………..……………….7](#_Toc54818532)

[Hình 2. 2 Sơ đồ nguyên lý hệ thống 8](#_Toc54818533)

[Hình 2. 3 Mạch in PCB 9](#_Toc54818534)

[Hình 2. 4 Sơ đồ thuật toán hệ thống 10](#_Toc54818535)

[Hình 2. 5 ESP8266 11](#_Toc54818536)

[Hình 2. 6 Sơ đồ chân Esp8266 11](#_Toc54818537)

[Hình 2. 7 Visual Studio Code 12](#_Toc54818538)

[Hình 2. 8 Platform IO 13](#_Toc54818539)

[Hình 2. 9 Tạo Project 13](#_Toc54818540)

[Hình 2. 10 Điền thông tin project 14](#_Toc54818541)

[Hình 2. 11 Cấu trúc project sau khi tạo xong 14](#_Toc54818542)

[Hình 2. 12 Altium Design 15](#_Toc54818543)

[Hình 2. 13 Giao diện MQTT dashboard 17](#_Toc54818544)

[Hình 2. 14 Nguyên lý led phát hồng ngoại 18](#_Toc54818545)

[Hình 2. 15 Ứng dụng trong điều khiển tivi 19](#_Toc54818546)

[Hình 3. 1 Cố định thiết bị 20](#_Toc54818517)

[Hình 3. 2 Mở app điều khiển 20](#_Toc54818518)

[Hình 3. 3 Giao diện điều khiển của app 21](#_Toc54818519)

[Hình 3. 4 Điều hòa điều khiển qua App 21](#_Toc54818520)

# **CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI**

## **Giới thiệu chung**

**Internet Of Things (IoT)** – **Internet vạn vật** dường như đang đứng trước một bước ngoặt để đi đến giai đoạn tiếp theo cho một thế giới hiện đại, văn minh. Đó là viễn cảnh mà mọi vật đều có thể kết nối với nhau thông qua Internet không dây. Các doanh nghiệp đang có xu hướng ứng dụng sản phẩm công nghệ IoT vào sản xuất ngày càng nhiều bởi thị trường sáng tạo tiềm năng và chi phí sản xuất ngày càng thấp.

Áp dụng Internet vạn vật vào các ngành, lĩnh vực trong cuộc sống trở nên rất cần thiết, đặc biệt là trong lĩnh vực Smart Home. Nó giúp cho con người tự do điều khiển thiết bị từ xa một cách chính xác và nhanh chóng.

LG là một công ty phát triển nhiều mảng trong linh vực điện điện tử ở Việt Nam trong đó có sản xuất điều hòa không khí (Air-Condition)

Như em tìm hiểu hiện nay có hai cách điều khiển điều hòa đó là sử dụng điều khiển cảu hãng hay với các model mới có tích hợp điều khiển qua app của hãng. Nhằm khắc phục nhược điểm khi mất điều khiển đối với model không tích hợp cùng với đó là giúp người dung điều khiển được điều hòa khi không ở trong nhà mình vì vậy em chọn đề tài: “Phát triển bộ điều khiển hồng ngoại điều khiển điều hòa LG qua môi trường Internet” nhằm giúp người dùng có một sự trải nghiệm tốt nhất không gây tốn kém chi phí.



## **Tổng quan các công nghệ sử dụng**

### *Công nghệ WiFi*

*a, Khái niệm*

Wifi là viết tắt của **Wireless Fidelity**. Nhiệm vụ của wifi này là sử dụng sóng vô tuyến để truyền tín hiệu đến các thiết bị kết nối mà không kết đến đây. Loại sóng vô tuyến này giống như sóng điện thoại, radio,.. nhưng đường truyền sóng ngắn hơn. Đặc biệt là tất cả các thiết bị thông minh như máy tính bảng, smartphone, laptop đều có khả năng kết nối wifi, nên rất thuận tiện trong quá trình sử dụng.

*b, Nguyên tắc hoạt động của WiFi*

Wifi sử dụng sóng radio (sóng vô tuyến) để truyền thông tin qua hệ thống mạng. Máy tính của bạn bao gồm một card mạng không dây sẽ truyền dữ liệu gửi vào tín hiệu radio. Tương tự tín hiệu này sẽ được truyền đi thông qua một ăng-ten, một bộ giải mã gọi là router. Sau khi giải mã xong, dữ liệu sẽ được gửi đến Internet thông qua một kết nối Ethernet có dây. Khi mạng không dây hoạt động như đường 2 chiều, các dữ liệu nhận được từ internet cũng sẽ đi qua router và được mã hoá thành tín hiệu radio để card mạng không dây trên máy tính nhận.



Hình 1. 1 Hoạt động của WiFi

### *Tia hồng ngoại*

*a, Khái niệm*

Tia hồng ngoại hay còn được viết tắt là tia IR là một loại năng lượng bức xạ không thể nhìn thấy bằng mắt người nhưng chúng ta có thể cảm thấy vì nhiệt độ tia IR cao. Tất cả các vật thể trong vũ trụ đều phát ra một số mức độ bức xạ hồng ngoại, nhưng hai trong số các nguồn rõ ràng nhất là mặt trời và lửa.

Tia hồng ngoại là một loại bức xạ điện từ, một tần số liên tục được tạo ra khi các nguyên tử hấp thụ và sau đó giải phóng năng lượng. Từ tần số cao nhất đến tần số thấp nhất, bức xạ điện từ bao gồm tia gamma, tia X, tia cực tím, ánh sáng nhìn thấy, bức xạ hồng ngoại, sóng vi ba và sóng vô tuyến . Cùng với nhau, các loại bức xạ tạo nên phổ điện từ.

*b, Ứng dụng của tia hồng ngoại*

Tia hồng ngoại được ứng dụng rất nhiều trong đời sống, sản xuất các thiết bị điện tử, camera, máy quang phổ… Những ứng dụng cụ thể trong từng lĩnh vực gồm:

* Các thiết bị trong gia đình như đèn nhiệt và bếp điện từ, lò vi sóng sử dụng bức xạ hồng ngoại để truyền nhiệt. Điều khiển từ xa của TV, điều hòa dựa vào bức xạ hồng ngoại bắn ra các xung năng lượng hồng ngoại từ một đi-ốt phát sáng (LED) đến một bộ thu hồng ngoại trong TV.
* Ứng dụng để quay Camera vào ban đêm: Tất cả các vật thể trên Trái đất đều phát ra tia hồng ngoại dưới dạng nhiệt. Điều này có thể được phát hiện bởi các cảm biến điện tử, chẳng hạn như những cảm biến được sử dụng trong kính nhìn ban đêm và camera hồng ngoại.
* Trong ngành thiên văn học: Ứng dụng tia hồng ngoại cho phép các nhà thiên văn học quan sát chi tiết sự phân bố nguồn IR trong không gian. Cho thấy các cấu trúc phức tạp trong tinh vân, thiên hà và cấu trúc quy mô lớn của vũ trụ.

### *Giao thức MQTT*

*a, Định nghĩa*

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

*b, Tính năng, đặc điểm nổi bật*

Dạng truyền thông điệp theo mô hình Pub/Sub cung cấp việc truyền tin phân tán một chiều, tách biệt với phần ứng dụng.

Việc truyền thông điệp là ngay lập tức, không quan tâm đến nội dung được truyền.

Sử dụng TCP/IP là giao thức nền.

Tồn tại ba mức độ tin cậy cho việc truyền dữ liệu (QoS: Quality of service)

* QoS 0: Broker/client sẽ gửi dữ liệu đúng một lần, quá trình gửi được xác nhận bởi chỉ giao thức TCP/IP.
* QoS 1: Broker/client sẽ gửi dữ liệu với ít nhất một lần xác nhận từ đầu kia, nghĩa là có thể có nhiều hơn 1 lần xác nhận đã nhận được dữ liệu.
* QoS 2: Broker/client đảm bảo khi gửi dữ liệu thì phía nhận chỉ nhận được đúng một lần, quá trình này phải trải qua 4 bước bắt tay.

Phần bao bọc dữ liệu truyền nhỏ và được giảm đến mức tối thiểu để giảm tải cho đường truyền.

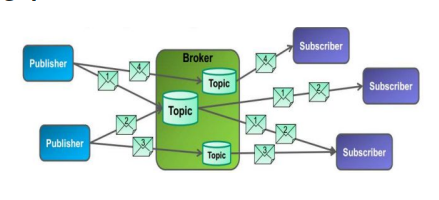
*c, Ưu điểm của MQTT*

Với những tính năng, đặc điểm nổi bật trên, MQTT mang lại nhiều lợi ích nhất là trong hệ thống SCADA (Supervisory Control And Data Acquisition) khi truy cập dữ liệu IoT.

* Truyền thông tin hiệu quả hơn.
* Tăng khả năng mở rộng.
* Giảm đáng kể tiêu thụ băng thông mạng.
* Rất phù hợp cho điều khiển và do thám.

………..

*d, Cơ chế hoạt động của MQTT theo mô hình Pub/Sub*

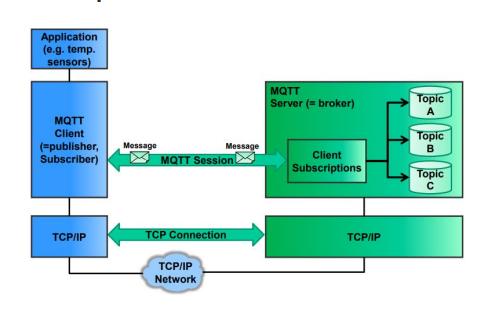


Hình 1. 2 Cơ chế tổng quan

MQTT hoạt động theo cơ chế client/server, nơi mà mỗi cảm biến là một khách hàng (client) và kết nối đến một máy chủ, có thể hiểu như một Máy chủ môi giới (broker), thông qua giao thức TCP (Transmission Control Protocol). Broker chịu trách nhiệm điều phối tất cả các thông điệp giữa phía gửi đến đúng phía nhận.

MQTT là giao thức định hướng bản tin. Mỗi bản tin là một đoạn rời rạc của tín hiệu và broker không thể nhìn thấy. Mỗi bản tin được publish một địa chỉ, có thể hiểu như một kênh (Topic). Client đăng kí vào một vài kênh để nhận/gửi dữ liệu, gọi là subscribe. Client có thể subscribe vào nhiều kênh. Mỗi client sẽ nhận được dữ liệu khi bất kỳ trạm nào khác gửi dữ liệu vào kênh đã đăng ký. Khi một client gửi một bản tin đến một kênh nào đó gọi là publish.

*e, Kiến trúc thành phần*



Hình 1. 3 Kiến trúc thành phần

Thành phần chính của MQTT là Client (Publisher/Subscriber), Server (Broker), Sessions (tạm dịch là Phiên làm việc), Subscriptions và Topics.

MQTT Client (Publisher/Subscriber): Clients sẽ subscribe một hoặc nhiều topics để gửi và nhận thông điệp từ những topic tương ứng.

MQTT Server (Broker): Broker nhận những thông tin subscribe (Subscriptions) từ client, nhận thông điệp, chuyển những thông điệp đến các Subscriber tương ứng dựa trên Subscriptions từ client.

Topic: Có thể coi Topic là một hàng đợi các thông điệp, và có sẵn khuôn mẫu dành cho Subscriber hoặc Publisher. Một cách logic thì các topic cho phép Client trao đổi thông tin với những ngữ nghĩa đã được định nghĩa sẵn. Ví dụ: Dữ liệu cảm biến nhiệt độ của một tòa nhà.

Session: Một session được định nghĩa là kết nối từ client đến server. Tất cả các giao tiếp giữa client và server đều là 1 phần của session.

Subscription: Không giống như session, subscription về mặt logic là kết nối từ client đến topic. Khi đã subscribe một topic, Client có thể nhận/gửi thông điệp (message) với topic đó.

# **CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG**



## **Đặc tả hệ thống**

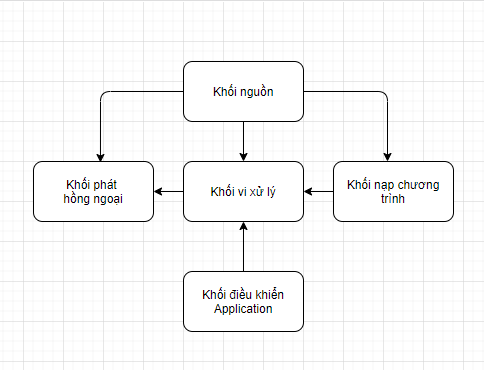
Hệ thống sau khi xây dựng xong phải đảm bảo các yêu cầu thiết kế sau:

* Hệ thống có khả năng kết nối WiFi
* Hệ thống có khả năng phát ra tia hồng ngoại
* Có thể điều khiển các loại điều hòa LG từ xa
* Thay thế được điều khiển điều hòa truyền thống
* Tiết kiệm năng lượng cho người dùng
* Tiết kiệm chi phí

## **Phân tích và thiết kế**

### *Sơ đồ khối của hệ thống và chức năng từng khối*

*a, Sơ đồ khối của hệ thống*



Hình 2. 1 Sơ đồ khối hệ thống

*b, Chức năng chi tiết từng khối*

*Khối nguồn:* Cung cấp điện áp ổn định tới các khối trong hệ thống, khối nguồn được sử dụng trong hệ thống có điện áp 5 VDC và 3.3 VDC.

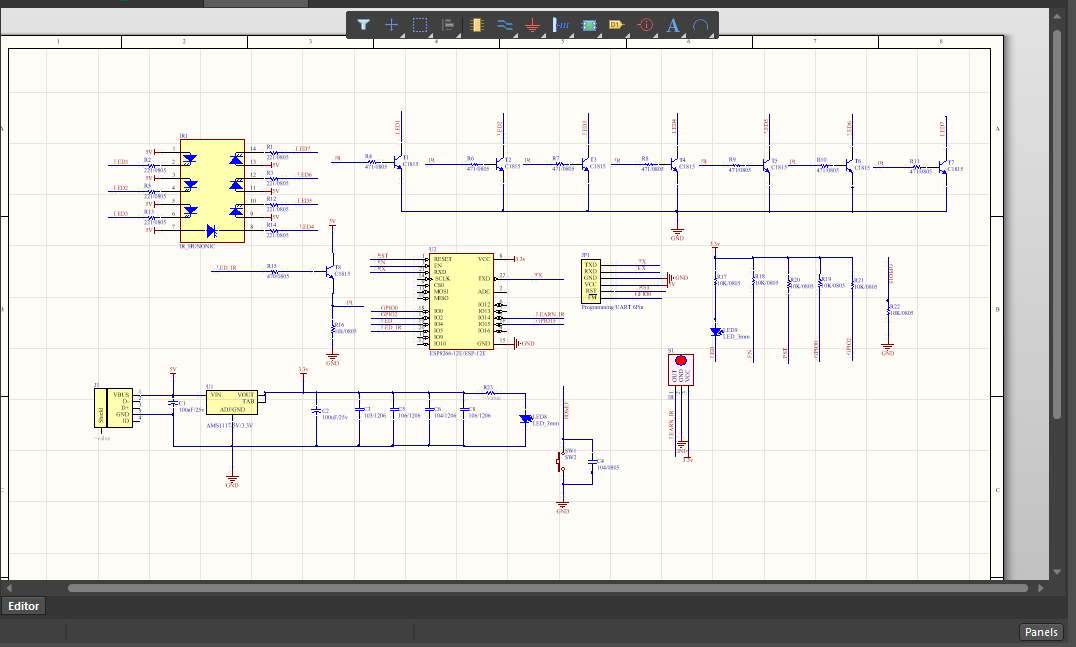
*Khối nạp chương trình:* Có chức năng giúp người lập trình đua code của mình vào trong vi điều khiển và hỗ trợ gỡ lỗi trong quá trình lập trình.

*Khối vi điều khiển:* Có chức năng kết nối internet, MQTT server, nhận dữ liệu từ khối điều khiển Application, tính toán dữ liệu và gửi về khối phát hồng ngoại. Vi điều khiển được sử dụng trong hệ thống là ESP8266

*Khối điều khiển Application:* Là một ứng dụng trên điện thoại di độngcó kết nối internet và MQTT client giúp gửi những thiết lập cần thiết để điều khiển điều hòa về cho vi điều khiển xử lý.

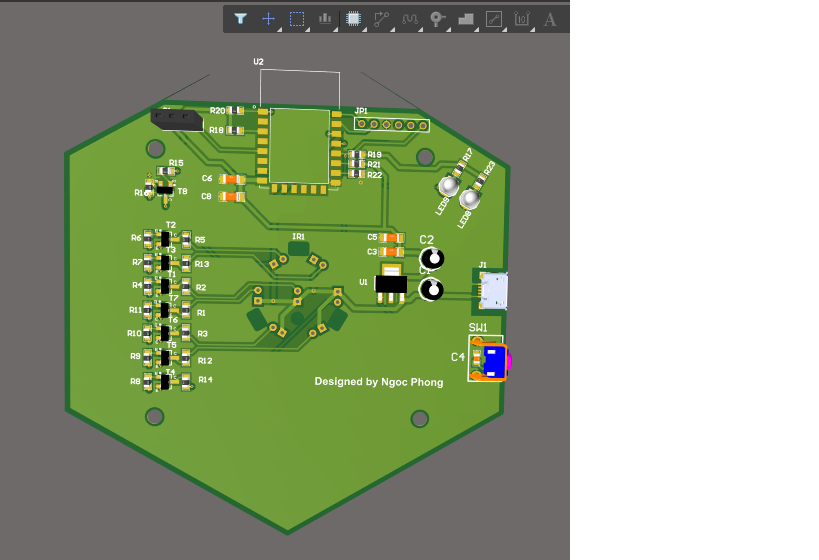
*Khối phát tia hồng ngoại:* Khi nhận được tín hiệu từ vi điều khiển thì khối này phát ra một tia hồng ngoại ở tần số 38Khz theo các tham số mà người dùng gửi về.

### *Sơ đồ nguyên lý*



Hình 2. 2 Sơ đồ nguyên lý hệ thống

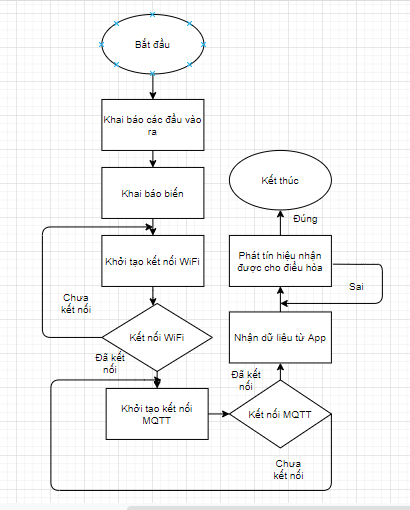
### *Sơ đồ mạch in PCB*



Hình 2. 3 Mạch in PCB

### *Lưu đồ thuật toán*

*a, Lưu đồ toàn hệ thống*



Hình 2. 4 Sơ đồ thuật toán hệ thống

## **Giới thiệu một số công cụ, thiết bị sử dụng**

### *Esp8266*

*a, Giới thiệu*

[**Mạch thu phát WiFi ESP8266 Uart ESP-01**](https://nshopvn.com/product/mach-thu-phat-wifi-esp8266-uart-esp-01/) sử dụng IC Wifi SoC ESP8266 của hãng Espressif, được sử dụng để kết nối với vi điều khiển thực hiện chức năng truyền nhận dữ liệu qua Wifi, mạch có thiết kế nhỏ gọn, sử dụng giao tiếp UART với bộ thư viện và code mẫu rất nhiều từ cộng đồng (search google esp-01), **mạch thu phát wifi ESP-01** được sử dụng trong các ứng dụng IoT và điều khiển thiết bị qua Wifi,…

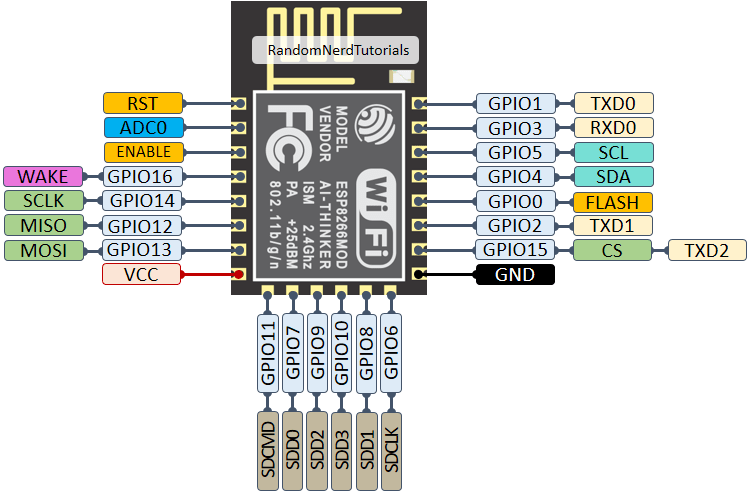


Hình 2. ESP8266

*b, Thông số kỹ thuật*

* Điện áp sử dụng: 3.3VDC
* Điện áp giao tiếp: 3.3VDC
* Dòng tiêu thụ: Max 320mA (nên sử dụng module cấp nguồn riêng cho mạch).
* Hỗ trợ chuẩn 802.11 b/g/n.
* Wi-Fi 2.4 GHz, hỗ trợ các chuẩn bảo mật như: OPEN, WEP, WPA\_PSK, WPA2\_PSK, WPA\_WPA2\_PSK.
* Hỗ trợ cả 2 giao tiếp TCP và UDP.
* Chuẩn giao tiếp UART với Firmware hỗ trợ bộ tập lệnh AT Command, tốc độ Baudrate mặc định 9600 hoặc 115200.
* Có 3 chế độ hoạt động: Client, Access Point, Both Client and Access Point.
* Kích thước: 24.8 x 14.3mm

*c, Chức năng các chân*



Hình 2. 6 Sơ đồ chân Esp8266

Tất cả các GPIO đều có trở kéo lên nguồn bên trong (ngoại trừ GPIO16 có trở kéo xuống GND). Người dùng có thể cấu hình kích hoạt hoặc không kích hoạt trở kéo này.

GPIO1 và GPIO3: hai GPIO này được nối với TX và RX của bộ UART0, NodeMCU nạp code thông qua bộ UART này nên tránh sử dụng 2 chân GPIO này.

GPIO0, GPIO2, GPIO15: đây là các chân có nhiệm vụ cấu hình mode cho ESP8266 điều khiển quá trình nạp code nên bên trong NodeMCU (có tên gọi là strapping pins) có các trở kéo để định sẵn mức logic cho chúng như sau: GPIO0: HIGH, GPIO2: HIGH, GPIO15: LOW. Vì vậy khi muốn sử dụng các chân này ở vai trò GPIO cần phải thiết kế một nguyên lý riêng để tránh xung đột đến quá trình nạp code.

GPIO9, GPIO10: hai chân này được dùng để giao tiếp với External Flash của ESP8266 vì vậy cũng không thể dùng được.

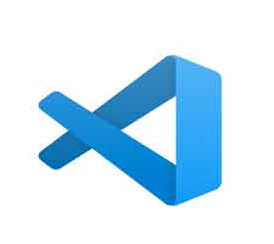
Như vậy, các GPIO còn lại: GPIO 4, 5, 12, 13, 14, 16 có thể sử dụng bình thường.

### *Visual Studio Code và Platform IO*

*a, Visual Studio Code*

Là một trình biên tập lập trình code miễn phí dành cho Windows, Linux và macOS, Visual Studio Code được phát triển bởi Microsoft. Nó được xem là một sự kết hợp hoàn hảo giữa IDE và Code Editor.

Visual Studio Code hỗ trợ chức năng debug, đi kèm với Git, có syntax highlighting, tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn. Nhờ tính năng tùy chỉnh, Visual Studio Code cũng cho phép người dùng thay đổi theme, phím tắt, và các tùy chọn khác.



Hình 2. 7 Visual Studio Code

Một số tính năng của Visual Studio Code:

* Hỗ trợ nhiều ngôn ngữ lập trình
* Hỗ trợ đa nền tảng
* Cung cấp kho tiện ích mở rộng
* Kho lưu trữ an toàn
* Hỗ trợ Web, lưu trữ dữ liệu đa dạng phân cấp

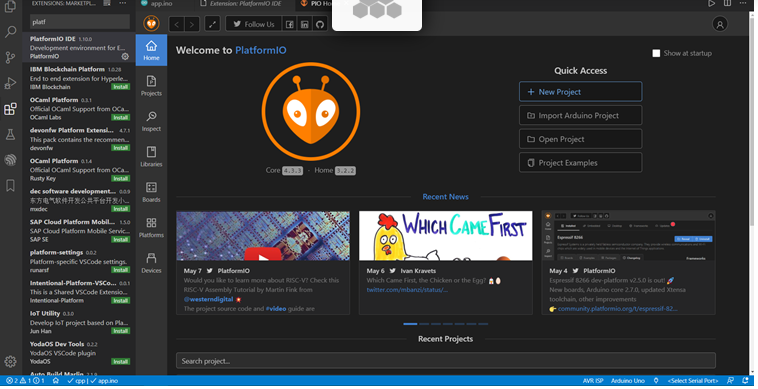
*b, Platform IO*

[PlatformIO](http://platformio.org/) là mội plugin có khả năng lập trình Arduino/ ARM mbed nó hỗ trợ tới hơn 250 board khác nhau.



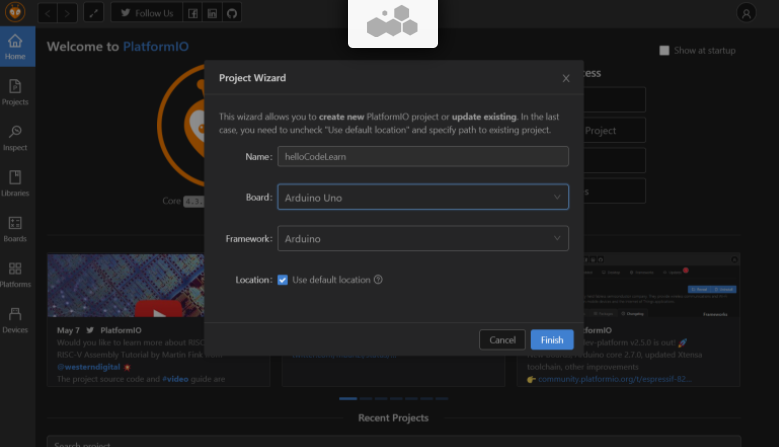
Hình 2. 8 Platform IO

Để bắt đầu với project mới, các bạn chọn New Project:



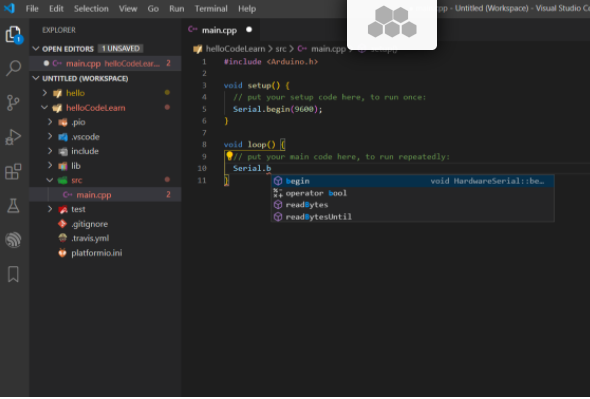
Hình 2. 9 Tạo Project

Điền thông tin vào, nhớ rằng platformio sẽ cấu hình luôn board mạch bạn sử dụng ngay khi khởi tạo project:



Hình 2. 10 Điền thông tin project

Kết quả sau khi tạo project:



Hình 2. 11 Cấu trúc project sau khi tạo xong

### *Altium Design*

*a, Giới thiệu*

Altium Designer trước kia có tên gọi quen thuộc là Protel DXP, là một trong những công cụ vẽ mạch điện tử mạnh nhất hiện nay. Được phát triển bởi hãng Altium Limited. Altium designer là một phần mềm chuyên nghành được sử dụng trong thiết kế mạch điện tử. Nó là một phần mềm mạnh với nhiều tính năng thú vị, tuy nhiên phần mềm này còn được ít người biết đến so với các phần mềm thiết kế mạch khác như orcad hay proteus.



Hình 2. 12 Altium Design

*b, Một số đặc trưng*

Giao diện thiết kế, quản lý và chỉnh sửa thân thiện, dễ dàng biên dịch, quản lý file, quản lý phiên bản cho các tài liệu thiết kế.

Hỗ trợ mạnh mẽ cho việc thiết kế tự động, đi dây tự động theo thuật toán tối ưu, phân tích lắp ráp linh kiện. Hỗ trợ việc tìm các giải pháp thiết kế hoặc chỉnh sửa mạch, linh kiện, netlist có sẵn từ trước theo các tham số mới.

Mở, xem và in các file thiết kế mạch dễ dàng với đầy đủ các thông tin linh kiện, netlist, dữ liệu bản vẽ, kích thước, số lượng…

Hệ thống các thư viện linh kiện phong phú, chi tiết và hoàn chỉnh bao gồm tất cả các linh kiện nhúng, số, tương tự…

 Đặt và sửa đối tượng trên các lớp cơ khí, định nghĩa các luật thiết kế, tùy chỉnh các lớp mạch in, chuyển từ schematic sang PCB, đặt vị trí linh kiện trên PCB.

Mô phỏng mạch PCB 3D, đem lại hình ảnh mạch điện trung thực trong không gian 3 chiều, hỗ trợ MCAD-ECAD, liên kết trực tiếp với mô hình STEP, kiểm tra khoảng cách cách điện, cấu hình cho cả 2D và 3D

Hỗ trợ thiết kế PCB sang FPGA và ngược lại.

*c, Tóm tắt các bước thiết kế mạch trên Altium*

- Đặt ra các yêu cầu bài toán.

-  Lựa chọn linh kiện.

- Thiết kế mạch nguyên lý.

- Lựa chọn các chân linh kiện để chuyển sang mạch in Update mạch nguyên lý sang mạch in.

-  Lựa chọn kích thước mạch in Sắp sếp các vị trí các loại linh kiện  như điện trở , tụ điện, IC...

-  Đặt kích thước các loại dây nối.

- Đi dây trên mạch.

- Kiểm tra toàn mạch.

### *Esp8266 MQTT client*

Có khá nhiều thư viện MQTT cho ESP8266 trên Arudino, ở đây mình dùng thư viện phổ biến là [PubSubClient](https://github.com/knolleary/pubsubclient/), bạn có thể tải thư viện này và bỏ vào trong thư mục library của Arduino

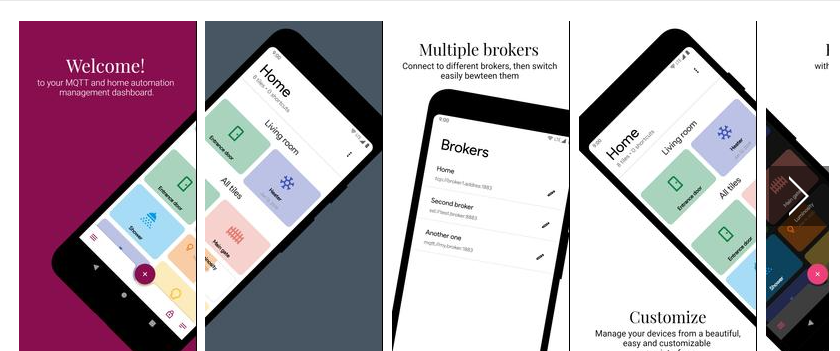
Chúng ta sẽ tạo một biến espClient thuộc lớp WiFiClient, biến này được khai báo là MQTT Client và sử dụng các thuộc tính của thư viện PubSubClient. Tại hàm setup() sẽ thiết lập ESP8266 ở chế độ station, kết nối đến mạng wifi. Bên cạnh đó hàm setup() cũng sẽ thực hiện chức năng tự động kết nối lại với MQTT Broker khi xảy ra mất kết nối đồng thời thực hiện các chức năng publish, subscribe của 1 MQTT Client thông qua hàm reconnect(). Hàm callback() có nhiệm vụ lấy dữ liệu của các puslisher trên topic đã subscribe và kiểm tra nội dung của message để điều khiển led ở GPIO2. Hàm loop() có chức năng kết nối Client là ESP8266 với Broker, thực hiện chức năng publish 1 message và subscribe topic. client.loop() sẽ kiểm tra thời gian kết nối của Client với gói KEEP\_ALIVE để đưa ra các thông tin về trạng thái kết nối của ESP8266 đồng thời lấy dữ liệu của message từ buffer để gửi đến các Client đã subcribe topic.

### *MQTT dashboard*

Mqtt Dashboard là một công cụ đơn giản và đẹp mắt để điều khiển các thiết bị hỗ trợ MQTT và quản lý hệ thống tự động hóa trong nhà của bạn. Nó tương thích với Node-RED, Tasmota Sonoff, tất cả bảng Arduino hỗ trợ internet và nhiều hơn nữa…

Nếu nó có MQTT, nó sẽ hoạt động với ứng dụng này.

Nó giúp người sử dụng có thể tạo giao diện tùy chỉnh như ý muốn



Hình 2. 13 Giao diện MQTT dashboard

### *Led phát tia hồng ngoại*

*a, Giới thiệu*

Đèn LED phát hồng ngoại (IR LED) hoạt động giống như đèn LED thông thường, nhưng có thể sử dụng các vật liệu khác nhau để tạo ra ánh sáng hồng ngoại.

*b, Nguyên lý hoạt động*

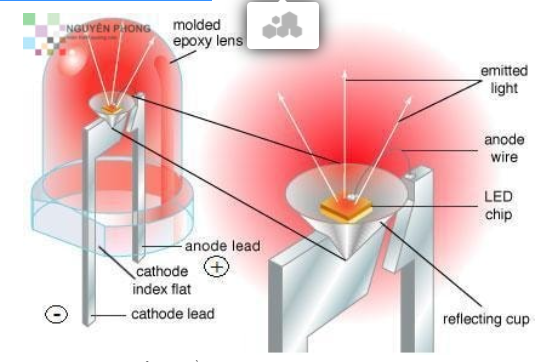
Một đèn LED phát hồng ngoại, giống như tất cả các đèn LED, một loại diode, hoặc chất bán dẫn đơn giản.

Điốt được thiết kế sao cho dòng điện chỉ có thể chạy theo một hướng. Khi dòng điện chạy, các electron rơi từ một phần của diode vào lỗ trên một phần khác. Để rơi vào các lỗ này, các electron phải làm năng lượng dưới dạng photon tạo ra ánh sáng.

Bước sóng và màu sắc của ánh sáng được tạo ra phụ thuộc vào vật liệu được sử dụng trong diode.

Đèn LED phát hồng ngoại sử dụng vật liệu tạo ra ánh sáng trong phần hồng ngoại của quang phổ, tức là, ngay dưới những gì mắt người có thể nhìn thấy.

Đèn LED hồng ngoại khác nhau có thể tạo ra ánh sáng hồng ngoại của các bước sóng khác nhau, giống như các đèn LED khác nhau tạo ra ánh sáng có màu sắc khác nhau.

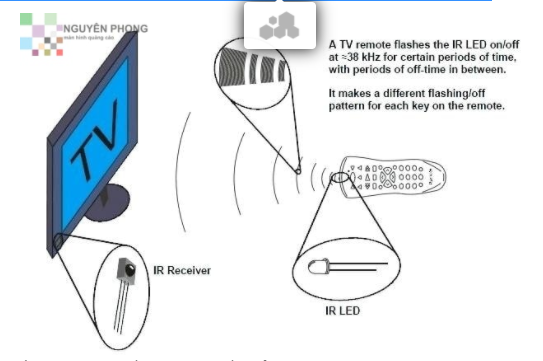


Hình 2. 14 Nguyên lý led phát hồng ngoại

*c, Ứng dụng của led hồng ngoại*

 Một nơi rất phổ biến để tìm đèn LED hồng ngoại là điều khiển từ xa cho TV hoặc thiết bị khác như [**màn hình quảng cáo**](https://manhinhquangcaodanang.com/man-hinh-quang-cao/)**,** quạt điện, điều hòa….

Một hoặc nhiều đèn LED bên trong điều khiển từ xa truyền các xung ánh sáng hồng ngoại nhanh đến một máy thu trên TV. Người nhận sau đó giải mã và diễn giải các xung này thành một lệnh và thực hiện thao tác mong muốn.



Hình 2. 15 Ứng dụng trong điều khiển tivi

Ánh sáng hồng ngoại cũng có thể được sử dụng để truyền dữ liệu giữa các thiết bị điện tử.

Điện thoại di động, trợ lý kỹ thuật số cá nhân (PDA) và một số máy tính xách tay có thể có đèn LED và bộ thu hồng ngoại được thiết kế để truyền dữ liệu tầm ngắn.

Một số bàn phím không dây và chuột máy tính cũng sử dụng đèn LED phát hồng ngoại và bộ thu để thay thế cáp.

Robot có thể sử dụng đèn LED thu phát hồng ngoại để phát hiện vật thể và một số đồng hồ tiện ích thậm chí còn có đèn LED phát hồng ngoại để truyền dữ liệu đến một công cụ để đọc đồng hồ dễ dàng…..

# **CHƯƠNG 3: KẾT QUẢ ĐÁNH GIÁ**



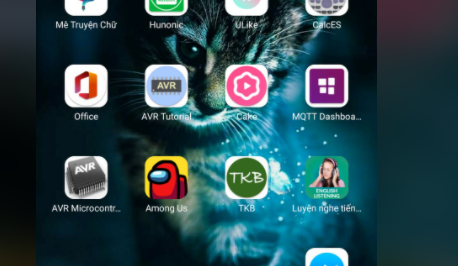
## **Các bước sử dụng hệ thống**

*Bước 1:* Đặt thiết bị lại gần điều hòa và cục kết nối WiFi sao cho hướng phát của bóng hồng ngoại trùng với mắt thu của thiết bị



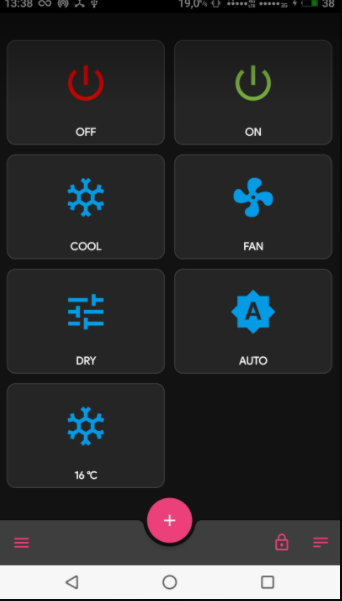
Hình 3. 1 Cố định thiết bị

*Bước 2:* Sau khi cố định thiết bị ta mở điện thoại và mở apps MQTT Dashboard



Hình 3. 2 Mở app điều khiển

*Bước 3:*Giao diện điều khiển của app cho người dùng thao tác theo ý muốn



Hình 3. 3 Giao diện điều khiển của app

*Bước 4*: Chờ đợi kết quả điều hòa hoạt động đúng theo app điều khiển



Hình 3. 4 Điều hòa điều khiển qua App

## **Đánh giá hệ thống**

### *Ưu điểm*

Hệ thống đáp ứng được đúng những yêu cầu đã được được đưa ra.

Giúp người sử dụng có thể điều khiển được thiết bị ngay cả khi không có mặt trực tiếp trong phòng

Giúp người dùng tiết kiệm điện năng

Tận dụng tối đa khả năng điều chỉnh nhiệt độ của điều hòa

Thay thế được điều khiển truyền thống

### *Nhược điểm*

Thiết bị chỉ sử dụng được khi có kết nối internet.

Không có khả năng kết nối WiFi tự động mà phải nhờ người lập trình nạp code có ssid và password của WiFi.

Khoảng cách điều khiển hạn chế (3m - 5m).

Sử dụng app và server bên thứ 3 nên nếu họ ngừng cung cấp thì ta phải thiết lập lại thiết bị từ đầu.

## **Kết luận**

Trải qua thời gian nghiên cứu và thực hiện đề tài đã giúp em hiểu ra hơn về vai trò, các ứng dụng của vi điều khiển trong thực tế, cũng như cách lập trình cho vi điều khiển Esp8266, cách thiết kế mạch trên phần mềm Altium.

Ngoài ra qua đề tài này em cũng hiểu rõ hơn về các cách truyền dữ liệu qua internet và các ứng dụng của lĩnh vực IOT trong đời sống từ đó có những hướng nghiên cứu và phát triển những thiết bị thông minh.

# **Tài liệu tham khảo**

[1] <https://github.com/crankyoldgit/IRremoteESP8266>

[2] <https://www.instructables.com/Web-IR-Remote-With-Esp8266-NodeMCU>

[3] <https://www.espressif.com/en/products/socs/esp8266>

# **Phụ lục**

#include <Arduino.h>

#include <ESP8266WiFi.h>

#include "IRremoteESP8266.h"

#include "IRutils.h"

#include "IRtext.h"

#include "IRrecv.h"

#include "IRsend.h"

#include "PubSubClient.h"

#include <ArduinoJson.h>

#include "ir\_LG.h"

#include <string.h>

/\*

\*define info mqtt

\*/

const char\* ssid = "Hunonic T2\_2Ghz";

const char\* password = "66668888";

const char\* mqtt\_client\_id = "ir\_controller";

const char\* mqtt\_server\_ip = "mqtt2.hunonicpro.com";

const char\* mqtt\_user = "bestbug";

const char\* mqtt\_password = "bigbugdmm";

const char\* mqtt\_topic\_sub = "recv";

const char\* mqtt\_topic\_pub = "gui";

int mqtt\_port = 1883;

const int kIrled = D1;

const int led\_state\_mqtt = D2;

const int button = 0;

WiFiClient espClient;

PubSubClient client(espClient);

IRsend irsend(kIrled);

IRLgAc lgAc(kIrled);

#define jsonName "name"

#define jsonPower "power"

#define jsonTemp "temp"

#define jsonMode "mode"

#define jsonFan "fan"

#define LG 1

void on\_air\_condition();

void off\_air\_condition();

void callback(char \*p\_toppic, uint8\_t \*p\_data, unsigned int length)

{

StaticJsonBuffer<1024> JSONBuffer;

JsonObject &root = JSONBuffer.parseObject((char \*)p\_data);

Serial.println((char\*)p\_data);

int name = (char)root[jsonName];

if(name == LG){

if (root[jsonPower] == 1)

{

lgAc.on();

lgAc.setTemp(root[jsonTemp]);

lgAc.setMode(root[jsonMode]);

lgAc.setFan(root[jsonFan]);

lgAc.send();

Serial.println(lgAc.toString());

}

else if (root[jsonPower] == 0)

{

lgAc.off();

lgAc.send();

Serial.println(lgAc.toString());

}

}

JSONBuffer.clear();

}

void reconnect()

{

while (!client.connected())

{

Serial.println("Dang ket noi MQTT...");

// Connect MQTT

if (client.connect(mqtt\_client\_id, mqtt\_user, mqtt\_password))

{

Serial.println("Da ket noi xong MQTT"); // Ket noi xong , hien thi

client.subscribe(mqtt\_topic\_sub);

}

else

{

Serial.print("No connect: ");

Serial.print(client.state());

Serial.println("Doi 5 giay");

delay(5000);

}

}

}

void setup() {

// put your setup code here, to run once:

Serial.begin(115200);

irsend.begin();

lgAc.begin();

WiFi.begin(ssid,password);

//WiFi.mode(WIFI\_STA);

Serial.println();

Serial.print("Dang ket noi wifi ");

Serial.println(ssid);

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

{

delay(10);

Serial.print("..........");

}

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

Serial.println(WiFi.localIP());

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

client.setCallback(callback);

pinMode(led\_state\_mqtt,OUTPUT);

pinMode(button,INPUT);

}

int dem =0;

void loop() {

// put your main code here, to run repeatedly:

if (!client.connected())

{

reconnect();

digitalWrite(led\_state\_mqtt,LOW);

}

client.loop();

//set button controller air condition

static int btnState, lastState;

btnState = digitalRead(button);

if(btnState !=lastState && !btnState ){

dem++;

if(dem ==1) on\_air\_condition();

else if(dem == 2) off\_air\_condition();

}

lastState = btnState;

dem =0;

}

//function on condition

void on\_air\_condition(){

lgAc.on();

lgAc.send();

Serial.println(lgAc.toString());

}

//function off condition

void off\_air\_condition(){

lgAc.off();

lgAc.send();

Serial.println(lgAc.toString());

}